排除Catalyst 9000交换机上的EtherChannel故障

目录

简介

本文档介绍如何了解Catalyst 9000系列交换机上的EtherChannel并对其进行故障排除。

先决条件

要求

Cisco 建议您了解以下主题：

- Catalyst 9000系列交换机架构
- Cisco IOS® XE软件架构
- 链路聚合控制协议 (LACP) 和端口聚合协议 (PAgP)

使用的组件

本文档中的信息基于以下硬件版本：

- Catalyst 9200
- Catalyst 9300
- Catalyst 9400
- Catalyst 9500
- Catalyst 9600

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

有关限制、限制、配置选项和注意事项的最新信息，以及与此功能相关的任何其他详细信息，请参阅Cisco正式发行说明和配置指南。

EtherChannel在交换机、路由器和服务器之间提供容错高速链路。 使用EtherChannel增加设备之间的带宽，并将其部署在网络中可能出现瓶颈的任何位置。EtherChannel为链路丢失提供自动恢复，它会在剩余链路之间重分配负载。如果链路发生故障，EtherChannel会将流量从故障链路重定向到信道中的其余链路，无需干预。

EtherChannel可以配置为不协商，也可以使用链路聚合协议（PAgP或LACP）支持动态协商。

当您启用PAgP或LACP时，交换机将了解合作伙伴的身份和每个接口的功能。然后，交换机将具有类似配置的接口动态分组到单个逻辑链路（通道或汇聚端口）中；交换机根据硬件、管理和端口参数限制对这些接口组进行分组。

LACP标志

LACP标志用于在端口通道启动时协商端口通道参数。看看每面旗帜的含义：

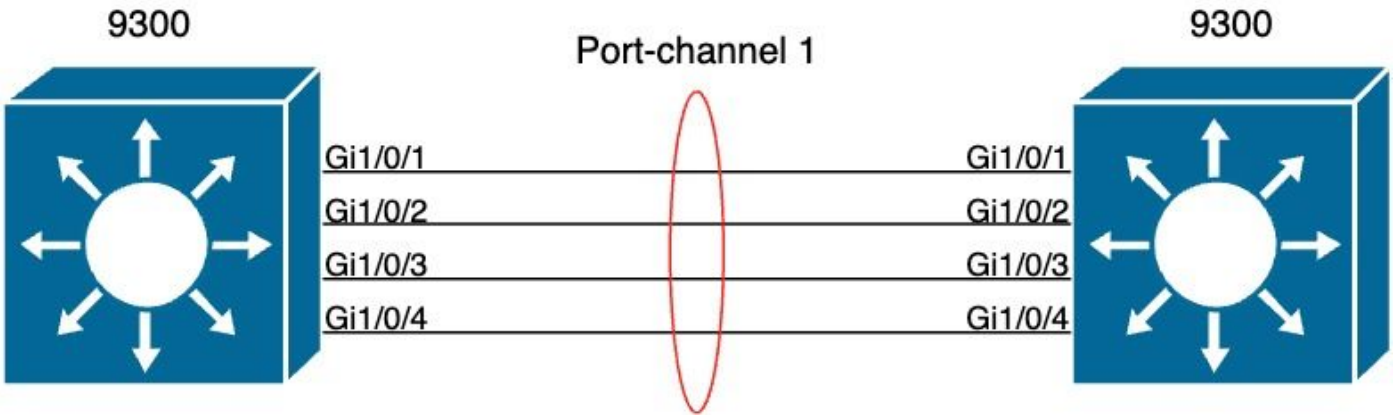
标志	状态
LACP活动（不太重要的位）	0 =被动模式 1 =主动模式
LACP Timeout：表示LACP发送/接收超时	0 =长超时。3 x 30秒（默认） 1 =短超时。3 x 1秒（LACP速率快速）
汇聚	0 =单个链路（不考虑聚合）

	1 =可聚合（潜在的聚合候选对象）
同步	0 =链路不同步（非良好状态） 1 =链路处于同步状态（正常状态）
收集	0 =未准备好接收/处理帧 1 =准备接收/处理帧
分发	0 =未准备好发送/传输帧 1 =准备发送/传输帧
已违约	0 =将接收的PDU中的信息用于合作伙伴 1 =它使用合作伙伴的默认信息
已过期（最高位）	0 = PDU已过期， 1 = PDU有效

LACP标志的预期值为0x3D（十六进制）或0111101（二进制），以达到P（捆绑在端口通道中）状态。

```
.... ...1 = LACP Activity (less significant bit)
.... ..0. = LACP Timeout
.... .1.. = Aggregation
.... 1... = Synchronization

...1 .... = Collecting
..1. .... = Distributing
.0.. .... = Defaulted
0... .... = Expired (most significant bit)
```

网络图



验证LACP操作

本节介绍如何验证LACP协议的正确状态和操作。

基本检查

使用以下命令检查LACP输出：

<#root>

show lacp sys-id

show lacp <*channel-group number*> neighbor

show lacp <*channel-group number*> counters

show interfaces <*interface ID*> accounting

debug lacp [event|packet|fsm|misc]

debug condition <*condition*>

第一个命令输出显示了交换机系统ID及其优先级（对于LACP）。

<#root>

switch#

show lacp sys-id

32768,

f04a.0206.1900 <-- Your system MAC address

检查LACP邻居的详细信息，例如运行模式、邻居系统Dev ID及其优先级。

<#root>

switch#

show lacp 1 neighbor

```
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 1 neighbors

                    LACP port                      Admin  Oper   Port    Port
Port          Flags  Priority
```

Dev ID

```
          Age  key    Key    Number  State
Gi1/0/1       SA     32768
```

f04a.0205.d600

```
  12s  0x0    0x1    0x102   0x3D
```

<-- Dev ID: Neighbor MAC Address

```
Gi1/0/2       SA     32768
```

f04a.0205.d600

```
  24s  0x0    0x1    0x103   0x3D
```

<-- Dev ID: Neighbor MAC Address

```
Gi1/0/3       SA     32768
```

f04a.0205.d600

```
  16s  0x0    0x1    0x104   0x3D
```

<-- Dev ID: Neighbor MAC Address

```
Gi1/0/4       SA     32768
```

f04a.0205.d600

```
  24s  0x0    0x1    0x105   0x3D
```

<-- Dev ID: Neighbor MAC Address

验证每个接口发送和接收的LACP数据包。如果检测到损坏的LACP数据包，则Pkts Err计数器将会增加。

<#root>

switch#

show lacp 1 counters

```
                LACPDUs         Marker          Marker Response  LACPDUs
Port            Sent   Recv     Sent   Recv     Sent   Recv      Pkts Err
--------------------------------------------------------------------------
Channel group: 1
Gi1/0/1
```

3111   3085

```
     0      0        0      0
```

0

```
Gi1/0/2
```

3075   3057

```
     0      0        0      0
```

0

```
Gi1/0/3
```

3081   3060

```
     0      0        0      0
```

0

```
Gi1/0/4
```

3076   3046

```
     0      0        0      0
```

0

还有一个选项可用于检查LACP的接口记帐。

<#root>

switch#

show interface gigabitEthernet1/0/1 accounting

```
GigabitEthernet1/0/1
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   Other          0          0      10677     640620
                    PAgP        879      78231        891      79299
           Spanning Tree        240      12720         85       5100
                     CDP       2179     936495       2180     937020
                     DTP       3545     170160       3545     212700
```

```
LACP       3102     384648       3127     387748
```

调试

当没有LACP同步或远程对等体不运行LACP时，生成系统日志消息。

```
%ETC-5-L3DONTBNDL2: Gig1/0/1 suspended: LACP currently not enabled on the remote port.
%ETC-5-L3DONTBNDL2: Gig/1/0/1 suspended: LACP currently not enabled on the remote port.
```

使用以下命令启用LACP调试：

<#root>

debug lacp [event|packet|fsm|misc]

debug condition <*condition*>

如果发现LACP协商问题，请启用LACP调试以分析原因。

<#root>

switch#

debug lacp event

Link Aggregation Control Protocol events debugging is on
switch#

debug lacp packet

Link Aggregation Control Protocol packet debugging is on
switch#

debug lacp fsm

Link Aggregation Control Protocol fsm debugging is on
switch#

debug lacp misc

Link Aggregation Control Protocol miscellaneous debugging is on

如果需要，还应启用特定接口的调试条件并过滤输出。

<#root>

switch#

debug condition interface gigabitEthernet 1/0/1

注意：LACP调试与平台无关。

验证调试和过滤器已设置。

<#root>

switch#

show debugging

Packet Infra debugs:

Ip Address Port
---|----------

LACP:
  Link Aggregation Control Protocol

miscellaneous

 debugging is

on

  Link Aggregation Control Protocol

packet

 debugging is

on

  Link Aggregation Control Protocol

fsm

 debugging is

on

  Link Aggregation Control Protocol

events

 debugging is

on

```
Condition 1: interface Gi1/0/1 (1 flags triggered)

        Flags: Gi1/0/1
```

分析LACP调试，并使用show logging命令显示它们。调试输出显示了端口通道接口启动之前的最后一个LACP帧：

<#root>

```
switch#

show logging

<omitted output>

LACP :lacp_bugpak: Send LACP-PDU packet via Gi1/0/1
LACP : packet size: 124

LACP: pdu: subtype: 1, version: 1

LACP: Act: tlv:1, tlv-len:20, key:0x1, p-pri:0x8000, p:0x102, p-state:0x3D, s-pri:0x8000, s-mac:f04a.020
LACP: Part: tlv:2, tlv-len:20, key:0x1, p-pri:0x8000, p:0x102, p-state:0xF, s-pri:0x8000, s-mac:f04a.020

LACP: col-tlv:3, col-tlv-len:16, col-max-d:0x8000
LACP: term-tlv:0 termr-tlv-len:0
LACP: HA: Attempt to sync events -- no action  (event type 0x1)


LACP :lacp_bugpak: Receive LACP-PDU packet via Gi1/0/1
LACP : packet size: 124

LACP: pdu: subtype: 1, version: 1

LACP: Act: tlv:1, tlv-len:20, key:0x1, p-pri:0x8000, p:0x102, p-state:0x3D, s-pri:0x8000, s-mac:f04a.020
LACP: Part: tlv:2, tlv-len:20, key:0x1, p-pri:0x8000, p:0x102, p-state:0x3D, s-pri:0x8000, s-mac:f04a.02

LACP: col-tlv:3, col-tlv-len:16, col-max-d:0x8000
LACP: term-tlv:0 termr-tlv-len:0

LACP: Gi1/0/1 LACP packet received, processing    <-- beginning to process LACP PDU

    lacp_rx Gi1/0/1 - rx: during state CURRENT, got event 5(recv_lacpdu)
@@@ lacp_rx Gi1/0/1 - rx: CURRENT -> CURRENT
LACP: Gi1/0/1 lacp_action_rx_current entered

LACP: recordPDU Gi1/0/1 LACP PDU Rcvd. Partners oper state is hex F    <-- operational state

LACP: Gi1/0/1 partner timeout mode changed to 0
    lacp_ptx Gi1/0/1 - ptx: during state FAST_PERIODIC, got event 2(long_timeout)
@@@ lacp_ptx Gi1/0/1 - ptx: FAST_PERIODIC -> SLOW_PERIODIC
LACP: Gi1/0/1 lacp_action_ptx_fast_periodic_exit entered
LACP: lacp_p(Gi1/0/1) timer stopped
LACP: Gi1/0/1 lacp_action_ptx_slow_periodic entered
LACP: timer lacp_p_s(Gi1/0/1) started with interval 30000.

LACP: recordPDU Gi1/0/1 Partner in sync and aggregating    <-- peer is in sync
LACP: Gi1/0/1 Partners oper state is hex 3D    <-- operational state update

LACP: timer lacp_c_l(Gi1/0/1) started with interval 90000.
LACP: Gi1/0/1 LAG_PARTNER_UP.
LACP: Gi1/0/1 LAG unchanged
    lacp_mux Gi1/0/1 - mux: during state COLLECTING_DISTRIBUTING, got event 5(in_sync) (ignored)
```

```
lacp_handle_standby_port_internal called, depth = 1
LACP: lacp_handle_standby_port_internal: No Standby port found for LAG 1
lacp_handle_standby_port_internal called, depth = 1
LACP: lacp_handle_standby_port_internal: No Standby port found for LAG 1
lacp_handle_standby_port_internal called, depth = 1
LACP: lacp_handle_standby_port_internal: No Standby port found for LAG 1
LACP: lacp_t(Gi1/0/1) timer stopped
LACP: lacp_t(Gi1/0/1) expired

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/2, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/3, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/4, changed state to up

%LINK-3-UPDOWN: Interface Port-channel1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel1, changed state to up
```

如果重点关注LACP调试的两条最重要的线路，则可以定义一些LACP PDU概念。

```
<#root>

LACP:

 Act

: tlv:1, tlv-len:20,

key:0x1

, p-pri:0x8000, p:0x102,

p-state:0x3D

, s-pri:0x8000,

s-mac:f04a.0205.d600


LACP:

Part

: tlv:2, tlv-len:20,

key:0x1

, p-pri:0x8000, p:0x102,

p-state:0x3D

, s-pri:0x8000,

s-mac:f04a.0206.1900
```

概念	描述
行动	代表演员（你）

部件	代表合作伙伴（您的邻居/同行）
密钥	它是配置的端口通道号。
p状态	表示端口状态，这是最重要的概念。它使用8位（LACP标志）构建。有关详细信息，请查看背景信息部分。
s-mac	它是LACP使用的系统MAC地址。

注意：在调试中看到的值是十六进制。要正确读取这些值，必须将其转换为十进制或二进制系统。

验证PAgP操作

本节介绍如何验证PAgP协议的正确状态和操作。

基本检查

使用以下命令检查PAgP输出：

<#root>

show pagp <*channel-group number*> neighbor

show pagp <*channel-group number*> counters

show interfaces <*interface ID*> accounting

检查PAgP邻居的详细信息，例如操作模式、伙伴系统ID、主机名和优先级。

<#root>

```
switch#
```

show pagp 1 neighbor

```
Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.        P - Device learns on physical port.
```

```
Channel group 1 neighbors
            Partner
```

Partner

```
         Partner                    Partner Group
Port        Name
```

Device ID

```
      Port        Age  Flags    Cap.
Gi1/0/1     switch
```

f04a.0205.d600

```
   Gi1/0/1     16s  SC       10001
```

<-- Dev ID: Neighbor MAC Address

```
Gi1/0/2     switch
```

f04a.0205.d600

```
   Gi1/0/2     19s  SC       10001
```

<-- Dev ID: Neighbor MAC Address

```
Gi1/0/3     switch
```

f04a.0205.d600

```
   Gi1/0/3     17s  SC       10001
```

<-- Dev ID: Neighbor MAC Address

```
Gi1/0/4     switch
```

f04a.0205.d600

```
   Gi1/0/4     15s  SC       10001
```

<-- Dev ID: Neighbor MAC Address

验证每个接口发送和接收的PAgP数据包的输出详细信息。如果检测到损坏的PAgP数据包，则Pkts Err计数器将会增加。

<#root>

```
switch#
```

show pagp 1 counters

```
            Information         Flush        PAgP
Port        Sent    Recv     Sent    Recv    Err Pkts
--------------------------------------------------------
Channel group: 1
Gi1/0/1
```

29   17

```
       0       0
```

0

```
Gi1/0/2
 28      17
      0       0
 0

Gi1/0/3
  28      16
      0       0
 0

Gi1/0/4
 29      16
       0       0
0
```

还有一个选项可用于检查PAgP的接口记帐。

```
<#root>
switch#
show int gi1/0/1 accounting

GigabitEthernet1/0/1
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                   Other          0          0      10677     640620
                    PAgP        879      78231        891      79299
           Spanning Tree        240      12720         85       5100
                     CDP       2179     936495       2180     937020
                     DTP       3545     170160       3545     212700
                    LACP       3102     384648       3127     387748
```

调试

如果您注意到PAgP协商问题，请启用PAgP调试以分析原因。

```
<#root>
switch#
debug pagp event
```

```
Port Aggregation Protocol events debugging is on
switch#
```

debug pagp packet

```
Port Aggregation Protocol packet debugging is on
switch#
```

debug pagp fsm

```
Port Aggregation Protocol fsm debugging is on
switch#
```

debug pagp misc

```
Port Aggregation Protocol miscellaneous debugging is on
```

如果需要，请为特定接口启用调试条件并过滤输出。

<#root>

switch#

debug condition interface gigabitEthernet 1/0/1

注意：PAgP调试与平台无关。

验证调试和过滤器已设置。

<#root>

switch#

show debugging

```
Packet Infra debugs:

Ip Address                                              Port
-------------------------------------------------------|----------

PAGP:
  Port Aggregation Protocol
```

miscellaneous

debugging is

on

Port Aggregation Protocol

```
packet
 debugging is
on
  Port Aggregation Protocol
fsm
 debugging is
on
  Port Aggregation Protocol
events
 debugging is
on

Condition 1: interface Gi1/0/1 (1 flags triggered)
        Flags: Gi1/0/1
```

分析PAgP调试。调试输出显示端口通道接口启动之前的最后一个PAgP帧：

```
<#root>

PAgP: Receive information packet via Gi1/0/1, packet size: 89
flags: 5, my device ID: f04a.0205.d600, learn-cap: 2, port-priority: 128, sent-port-ifindex: 9, group-ca
your device ID: f04a.0206.1900, learn-cap: 2, port-priority: 128, sent-port-ifindex: 9, group-cap: 10001


partner count: 1, num-tlvs: 2
device name TLV: switch
port name TLV: Gi1/0/1

PAgP: Gi1/0/1 PAgP packet received, processing    <-- Processing ingress PAgP frame
PAgP: Gi1/0/1 proved to be bidirectional    <--

PAgP: Gi1/0/1 action_b0 is entered
PAgP: Gi1/0/1 Input = Transmission State, V12 Old State = U5  New State = U5
PAgP: Gi1/0/1 action_a6 is entered
PAgP: Gi1/0/1 action_b9 is entered

PAgP: set hello interval from 1000 to 30000 for port Gi1/0/1    <--

PAgP: Gi1/0/1 Input = Transmission State, V10 Old State = U5  New State = U6
PAgP: set partner 0 interval from 3500 to 105000 for port Gi1/0/1
PAgP: Gi1/0/1 Setting hello flag
PAgP: timer pagp_p(Gi1/0/1) started with interval 105000.
PAgP: pagp_i(Gi1/0/1) timer stopped
PAgP: Gi1/0/1 Input = Port State, E5 Old State = S7  New State = S7
PAgP: pagp_h(Gi1/0/1) expired


PAgP: Send information packet via Gi1/0/1, packet size: 89
flags: 5, my device ID: f04a.0206.1900, learn-cap: 2, port-priority: 128, sent-port-ifindex: 9, group-ca
your device ID: f04a.0205.d600, learn-cap: 2, port-priority: 128, sent-port-ifindex: 9, group-cap: 10001
```

```
partner count: 1, num-tlvs: 2
device name TLV: switch
port name TLV: Gi1/0/1
PAgP: 89 bytes out Gi1/0/1
```

PAgP: Gi1/0/1 Transmitting information packet

PAgP: timer pagp_h(Gi1/0/1) started with interval 30000   <--
%LINK-3-UPDOWN: Interface Port-channel1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel1, changed state to up

验证Etherchannel编程

本节介绍如何验证EtherChannel的软件和硬件设置。

验证软件

验证软件条目。

<#root>

show run interface <*interface ID*>

show etherchannel <*channel-group number*> summary

检查EtherChannel配置。

<#root>

```
switch#
```

show run interface gigabitEthernet 1/0/1

```
<output omitted>
interface GigabitEthernet1/0/1
 channel-group 1 mode active
end

switch#
```

show run interface gigabitEthernet 1/0/2

<output omitted> interface GigabitEthernet1/0/2 channel-group 1 mode active end switch#

show run interface gigabitEthernet 1/0/3

<output omitted> interface GigabitEthernet1/0/3 channel-group 1 mode active end switch#

show run interface gigabitEthernet 1/0/4

<output omitted> interface GigabitEthernet1/0/4 channel-group 1 mode active end switch#

show run interface port-channel 1

```
 <output omitted> interface Port-channel1 end
```

验证端口信道中捆绑了所有端口成员。

<#root>

```
switch#
```

show etherchannel 1 summary

```
<output omitted>
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)         LACP        Gi1/0/1(P)      Gi1/0/2(P)
                                   Gi1/0/3(P)      Gi1/0/4(P)
```

验证硬件

验证硬件级别的软件条目：

<#root>

show platform software interface switch <*switch number or role*> r0 br

show platform software fed switch <*switch number or role*> etherchannel <*channel-group number*> group-mask

show platform software fed switch <*switch number or role*> ifm mappings etherchannel

show platform software fed switch <*switch number or role*> ifm if-id <*if ID*>

检查端口通道和捆绑接口的ID。

<#root>

```
switch#
```

show platform software interface switch active r0 br

```
Forwarding Manager Interfaces Information

Name
```

ID

```
               QFP ID
------------------------------------------------------------
<output omitted>
GigabitEthernet1/0/1

9

                  0
GigabitEthernet1/0/2

10

                 0
GigabitEthernet1/0/3

11

                 0
GigabitEthernet1/0/4

12

                 0
<output omitted> Port-channel1

76

 0
```

关注IF ID部分并确保值（十六进制数）与上一个命令中观察到的ID（十进制数）相等。

<#root>

```
switch#

show platform software fed switch active etherchannel 1 group-mask


Group Mask Info

Aggport IIF Id: 000000000000004c    <-- IfId Hex 0x4c = 76 decimal

Active Port: : 4

Member Ports
If Name

If Id

              local  Group Mask
--------------------------------------------------------------------------
GigabitEthernet1/0/4

000000000000000c

  true   7777777777777777

<-- IfId Hex 0xc = 12 decimal


GigabitEthernet1/0/3

000000000000000b
```

```
  true   bbbbbbbbbbbbbbbb
<-- IfId Hex 0xb = 11 decimal

GigabitEthernet1/0/2
000000000000000a
  true   dddddddddddddddd
<-- IfId Hex 0xa = 10 decimal

GigabitEthernet1/0/1
0000000000000009
  true   eeeeeeeeeeeeeeee
<-- IfId Hex 0x9 = 10 decimal
```

使用下一命令获取端口通道的IF ID。该值必须与之前命令中的值匹配。

```
<#root>

Switch#
show platform software fed switch active ifm mappings etherchannel

Mappings Table

Chan Interface IF_ID
---------------------------------------------------
1 Port-channel1
0x0000004c
```

使用下一个命令的IF ID。显示的信息必须与之前收集的输出相符。

```
<#root>

switch#
show platform software fed switch active ifm if-id 0x0000004c

Interface IF_ID         : 0x000000000000004c
Interface Name          : Port-channel1

Interface Block Pointer : 0x7f0178ca1a28
Interface Block State   : READY
Interface State         : Enabled
Interface Status        : ADD, UPD
Interface Ref-Cnt       : 8

Interface Type          : ETHERCHANNEL
```

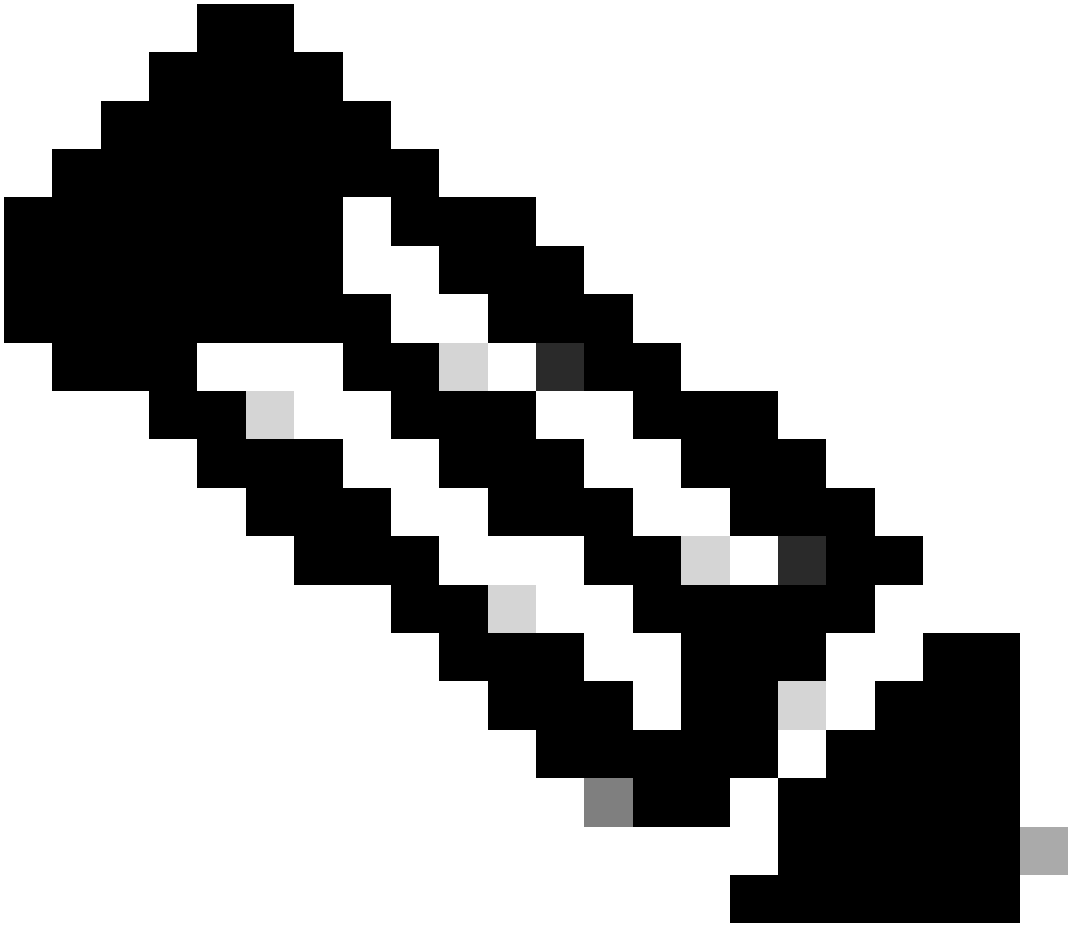
```
Port Type         : SWITCH PORT
Channel Number    : 1

SNMP IF Index     : 78
Port Handle       : 0xdd000068
# Of Active Ports : 4
Base GPN          : 1536

        Index[2]  : 000000000000000c
        Index[3]  : 000000000000000b
        Index[4]  : 000000000000000a
        Index[5]  : 0000000000000009


Port Information
Handle ............ [0xdd000068]

Type ............... [L2-Ethchannel]

Identifier ........ [0x4c]
Unit ............... [1]

DI ................. [0x7f0178c058a8]
Port Logical Subblock
        L3IF_LE handle .... [0x0]
        Num physical port . [4]
        GPN Base .......... [1536]
        Physical Port[2] .. [0x7b000027]
        Physical Port[3] .. [0x1f000026]
        Physical Port[4] .. [0xc000025]
        Physical Port[5] .. [0xb7000024]
        Num physical port on asic [0] is  [0]
        DiBcam handle on asic [0].... [0x0]
        Num physical port on asic [1] is  [4]
        DiBcam handle on asic [1].... [0x7f0178c850a8]
        SubIf count ....... [0]
Port L2 Subblock
        Enabled ............. [No]
        Allow dot1q ......... [No]
        Allow native ........ [No]
        Default VLAN ........ [0]
        Allow priority tag ... [No]
        Allow unknown unicast  [No]
        Allow unknown multicast[No]
        Allow unknown broadcast[No]
        Allow unknown multicast[Enabled]
        Allow unknown unicast  [Enabled]
        Protected ............ [No]
        IPv4 ARP snoop ....... [No]
        IPv6 ARP snoop ....... [No]
        Jumbo MTU ............ [0]
        Learning Mode ........ [0]
        Vepa ................. [Disabled]
        App Hosting........... [Disabled]
Port QoS Subblock
        Trust Type ................... [0x7]
        Default Value ................ [0]
        Ingress Table Map ............ [0x0]
        Egress Table Map ............. [0x0]
        Queue Map .................... [0x0]
Port Netflow Subblock
Port Policy Subblock
List of Ingress Policies attached to an interface
List of Egress Policies attached to an interface
```

```
        Port CTS Subblock
                Disable SGACL ..................... [0x0]
                Trust ............................. [0x0]
                Propagate ......................... [0x0]
                Port SGT .......................... [0xffff]
Ref Count : 8 (feature Ref Counts + 1)
IFM Feature Ref Counts
        FID : 97 (AAL_FEATURE_L2_MULTICAST_IGMP), Ref Count : 1
        FID : 119 ((null)), Ref Count : 1
        FID : 84 (AAL_FEATURE_L2_MATM), Ref Count : 1
No Sub Blocks Present
```

平台工具

下表显示了可用的工具和功能，以帮助了解何时使用它们：

工具	级别	何时使用
EPC	硬件与软件	使用它来验证位于物理接口的LACP帧，或验证它们与CPU的连通性。
平台前进	Hardware	如果确认了LACP帧已降级到交换机上，请使用此工具了解交换机的内部转发决策。
PSV	Hardware	如果确认了LACP帧已降级到交换机上，请使用此工具了解交换机的内部转发决策。
CoPP	Hardware	但是，如果从硬件角度将数据包转发到CPU，则在软件(CPU)级别上看不到该数据包。此功能很可能丢弃了硬件和CPU之间路径上的LACP帧。
FED CPU数据包捕获	软件	使用它来验证LACP帧是否通过正确的队列传送到CPU，还验证CPU是否将LACP帧发送回硬件。



注意：使用这些工具只能分析LACP协议，但是，它们也可用于分析PAgP帧。

嵌入式数据包捕获(EPC)

用于设置Wireshark (EPC)和捕获入口/出口LACP PDU的命令。

<#root>

monitor capture <*capture name*> [control-plane|interface <*interface ID*>] BOTH

monitor capture <*capture name*> match mac [any|host <*source MAC address*>|<*source MAC address*>][any|host <

monitor capture <*capture name*> file location flash:<*name*>.pcap

show monitor capture <*capture name*> parameter

```
show monitor capture <capture name>

monitor capture <capture name> start

monitor capture <capture name> stop

show monitor capture file flash:<name>.pcap [detailed]
```

注意：在特权模式下输入命令。

设置Wireshark捕获。

提示：如果您想专注于特定捆绑接口和/或特定源MAC地址，请调整接口并匹配mac关键字。

<#root>

```
monitor capture CAP interface GigabitEthernet1/0/1 BOTH

monitor capture CAP interface GigabitEthernet1/0/2 BOTH

monitor capture CAP interface GigabitEthernet1/0/3 BOTH

monitor capture CAP interface GigabitEthernet1/0/4 BOTH

monitor capture CAP match mac any host 0180.c200.0002

show monitor capture CAP file location flash:CAP.pcap
```

注意：在捕获上定义的目标MAC地址0180.c200.0002可帮助您过滤LACP帧。

验证是否已正确配置Wireshark：

<#root>

```
switch#
show monitor capture CAP parameter
```

```
monitor capture CAP interface GigabitEthernet1/0/1 BOTH
monitor capture CAP interface GigabitEthernet1/0/2 BOTH
monitor capture CAP interface GigabitEthernet1/0/3 BOTH
monitor capture CAP interface GigabitEthernet1/0/4 BOTH
monitor capture CAP match mac any host 0180.c200.0002
monitor capture CAP file location flash:LACP.pcap

switch#
```

show monitor capture CAP

```
Status Information for Capture CAP
  Target Type:
 Interface: GigabitEthernet1/0/1, Direction: BOTH
 Interface: GigabitEthernet1/0/2, Direction: BOTH
 Interface: GigabitEthernet1/0/3, Direction: BOTH
 Interface: GigabitEthernet1/0/4, Direction: BOTH
   Status : Inactive
  Filter Details:
   MAC
     Source MAC: 0000.0000.0000 mask:ffff.ffff.ffff
     Destination MAC: 0180.c200.0002 mask:0000.0000.0000
  Buffer Details:
   Buffer Type: LINEAR (default)
  File Details:
   Associated file name: flash:CAP.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)
```

开始捕获：

<#root>

```
switch#
```

monitor capture CAP start

```
Started capture point : CAP
```

如果您不使用LACP速率快速计时器，请在30秒后（至少）停止它：

<#root>

```
switch#
```

monitor capture CAP stop

```
Capture statistics collected at software:
        Capture duration - 58 seconds
```

```
        Packets received - 16
        Packets dropped - 0
        Packets oversized - 0

Bytes dropped in asic - 0

Stopped capture point : CAP
```

捕获的帧：

```
<#root>

switch#

show monitor capture file flash:CAP.pcap


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

    1   0.000000 f0:4a:02:06:19:04 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:06:19:00 P: 261 K
    2   2.563406 f0:4a:02:05:d6:01 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:05:d6:00 P: 258 K
    3   3.325148 f0:4a:02:05:d6:04 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:05:d6:00 P: 261 K
    4   5.105978 f0:4a:02:06:19:01 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:06:19:00 P: 258 K
    5   6.621438 f0:4a:02:06:19:02 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:06:19:00 P: 259 K
    6   8.797498 f0:4a:02:05:d6:03 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:05:d6:00 P: 260 K
    7  13.438561 f0:4a:02:05:d6:02 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:05:d6:00 P: 259 K
    8  16.658497 f0:4a:02:06:19:03 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:06:19:00 P: 260 K
    9  28.862344 f0:4a:02:06:19:04 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:06:19:00 P: 261 K
   10  29.013031 f0:4a:02:05:d6:01 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:05:d6:00 P: 258 K
   11  30.756138 f0:4a:02:05:d6:04 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:05:d6:00 P: 261 K
   12  33.290542 f0:4a:02:06:19:01 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:06:19:00 P: 258 K
   13  36.387119 f0:4a:02:06:19:02 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:06:19:00 P: 259 K
   14  37.598788 f0:4a:02:05:d6:03 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:05:d6:00 P: 260 K
   15  40.659931 f0:4a:02:05:d6:02 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:05:d6:00 P: 259 K
   16  45.242014 f0:4a:02:06:19:03 b^F^R 01:80:c2:00:00:02 LACP 124 v1 ACTOR f0:4a:02:06:19:00 P: 260 K
```

如果需要检查特定帧的LACP字段，请使用detailed关键字。

```
<#root>

switch#

show monitor capture file flash:CAP.pcap detailed

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


Frame 1: 124 bytes on wire (992 bits), 124 bytes captured (992 bits)

 on interface 0
    Interface id: 0 (/tmp/epc_ws/wif_to_ts_pipe)
        Interface name: /tmp/epc_ws/wif_to_ts_pipe
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 28, 2023 15:48:14.985430000 UTC
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1680018494.985430000 seconds
```

```
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 124 bytes (992 bits)
    Capture Length: 124 bytes (992 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:slow:lacp]
```

Ethernet II, Src: f0:4a:02:06:19:04 (f0:4a:02:06:19:04), Dst: 01:80:c2:00:00:02 (01:80:c2:00:00:02)

```
    Destination: 01:80:c2:00:00:02 (01:80:c2:00:00:02)
        Address: 01:80:c2:00:00:02 (01:80:c2:00:00:02)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: f0:4a:02:06:19:04 (f0:4a:02:06:19:04)
        Address: f0:4a:02:06:19:04 (f0:4a:02:06:19:04)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: Slow Protocols (0x8809)
```

Slow Protocols
**   Slow Protocols subtype: LACP (0x01)**
Link Aggregation Control Protocol

```
    LACP Version: 0x01
    TLV Type: Actor Information (0x01)
    TLV Length: 0x14
    Actor System Priority: 32768
    Actor System ID: f0:4a:02:06:19:00 (f0:4a:02:06:19:00)
    Actor Key: 1
    Actor Port Priority: 32768
    Actor Port: 261
    Actor State: 0x3d, LACP Activity, Aggregation, Synchronization, Collecting, Distributing
        .... ...1 = LACP Activity: Active
        .... ..0. = LACP Timeout: Long Timeout
        .... .1.. = Aggregation: Aggregatable
        .... 1... = Synchronization: In Sync
        ...1 .... = Collecting: Enabled
        ..1. .... = Distributing: Enabled
        .0.. .... = Defaulted: No
        0... .... = Expired: No
    [Actor State Flags: **DCSG*A]
    Reserved: 000000
    TLV Type: Partner Information (0x02)
    TLV Length: 0x14
    Partner System Priority: 32768
    Partner System: f0:4a:02:05:d6:00 (f0:4a:02:05:d6:00)
    Partner Key: 1
    Partner Port Priority: 32768
    Partner Port: 261
    Partner State: 0x3d, LACP Activity, Aggregation, Synchronization, Collecting, Distributing
        .... ...1 = LACP Activity: Active
        .... ..0. = LACP Timeout: Long Timeout
        .... .1.. = Aggregation: Aggregatable
        .... 1... = Synchronization: In Sync
        ...1 .... = Collecting: Enabled
        ..1. .... = Distributing: Enabled
        .0.. .... = Defaulted: No
        0... .... = Expired: No
    [Partner State Flags: **DCSG*A]
    Reserved: 000000
```

```
    TLV Type: Collector Information (0x03)
    TLV Length: 0x10
    Collector Max Delay: 32768
    Reserved: 00000000000000000000000
    TLV Type: Terminator (0x00)
    TLV Length: 0x00
    Pad: 000000000000000000000000000000000000000000000000...

Frame 2: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface 0
    Interface id: 0 (/tmp/epc_ws/wif_to_ts_pipe)
        Interface name: /tmp/epc_ws/wif_to_ts_pipe
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 28, 2023 15:48:17.548836000 UTC
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1680018497.548836000 seconds
    [Time delta from previous captured frame: 2.563406000 seconds]
    [Time delta from previous displayed frame: 2.563406000 seconds]
    [Time since reference or first frame: 2.563406000 seconds]
```

注意：9200台设备上的Wireshark输出格式可能不同，并且无法从交换机读取。导出捕获并从PC中读取（如果出现这种情况）。

平台前进

要调试转发信息并跟踪硬件转发平面中的数据包路径，请使用show platform hardware fed switch <switch number or role> forward interface命令。此命令模拟用户定义的数据包并从硬件转发平面检索转发信息。根据您在此命令中指定的数据包参数，在入口端口上生成数据包。您还可以提供来自PCAP文件中存储的捕获数据包的完整数据包。

本主题仅详细介绍接口转发特定的选项，即show platform hardware fed switch {switch_num|active|standby}forward interface命令提供的选项。

<#root>

```
show platform hardware fed switch <switch number or role> forward interface <interface ID> <source mac a
show platform hardware fed switch <switch number or role> forward interface <interface ID> pcap <pcap fi
show platform hardware fed switch <switch number or role> forward interface <interface ID> vlan <VLAN ID
```

定义平台转发捕获。在这种情况下，将分析CAP.pcap 帧1。

<#root>

switch#

show platform hardware fed switch active forward interface gigabitEthernet 1/0/1 pcap flash:CAP.pcap num

show forward is running in the background. After completion, syslog will be generated.

平台转发捕获完成后，会显示下一条Syslog消息。

<#root>

switch#

show logging

<output omitted>
*Mar 28 16:47:57.289: %SHFWD-6-PACKET_TRACE_DONE: Switch 1 R0/0: fed: Packet Trace Complete: Execute (s
*Mar 28 16:47:57.289: %SHFWD-6-PACKET_TRACE_FLOW_ID: Switch 1 R0/0: fed: Packet Trace Flow id is 100990

分析平台转发捕获。Egress部分告诉您内部转发决策是什么。LACP和PAgP帧需要传送到CPU。

<#root>

switch#

show platform hardware fed switch active forward last summary

Input Packet Details:

###[Ethernet]### dst = 01:80:c2:00:00:02 src. = f0:4a:02:06:19:04 type = 0x8809 <-- slow protocols (LA

###[Raw]###
load = '01 01 01 14 80 00 F0 4A 02 06 19 00 00 01 80 00 01 05 3D 00 00 00 02 14 80 00 F0 4A 0

```
Ingress:
   Port                      :
   Global Port Number        : 1536
   Local Port Number         : 0
   Asic Port Number          : 0
   Asic Instance             : 1
   Vlan                      : 1
   Mapped Vlan ID            : 4
   STP Instance              : 2
   BlockForward              : 0
   BlockLearn                : 0
   L3 Interface              : 37
       IPv4 Routing          : enabled
       IPv6 Routing          : enabled
       Vrf Id                : 0
   Adjacency:
       Station Index         : 107    [SI_CPUQ_L2_CONTROL]
       Destination Index     : 21106
       Rewrite Index         : 1
       Replication Bit Map   : 0x20   ['coreCpu']
   Decision:
       Destination Index     : 21106  [DI_CPUQ_L2_CONTROL]
       Rewrite Index         : 1      [RI_CPU]
       Dest Mod Index        : 0      [IGR_FIXED_DMI_NULL_VALUE]
       CPU Map Index         : 0      [CMI_NULL]
       Forwarding Mode       : 0      [Bridging]
       Replication Bit Map   :        ['coreCpu']
       Winner                :        L2DESTMACVLAN LOOKUP
       Qos Label             : 65
       SGT                   : 0
       DGTID                 : 0
```

Egress: Possible Replication : Port : CPU_Q_L2_CONTROL Output Port Data : Port : CPU

```
       Asic Instance         : 0
```

CPU Queue : 1 [CPU_Q_L2_CONTROL]

```
       Unique RI             : 0
       Rewrite Type          : 0      [NULL]
       Mapped Rewrite Type   : 15     [CPU_ENCAP]



 Vlan : 1



      Mapped Vlan ID        : 4
********************************************************************************
```

数据包状态矢量(PSV)

PSV与Platform Forward（平台转发）捕获类似，不同之处在于PSV从符合触发标准的网络中捕获实时入口帧。

注意：仅C9500-32C、C9500-32QC、C9500-24Y4C、C9500-48Y4C和C9606R平台支持PSV。

<#root>

debug platform hardware fed <*switch number or role*> capture trigger interface <*interface ID*> ingress

debug platform hardware fed <*switch number or role*> capture trigger layer2 <*source MAC address*> <*destina*

show platform hardware fed <*switch number or role*> capture trigger

show platform hardware fed <*switch number or role*> capture status

show platform hardware fed <*switch number or role*> capture summary

两个相互连接的C9500-48Y4C用于下一个端口通道和PSV捕获。

<#root>

switch#

show etherchannel 1 summary

```
<output omitted>
Group Port-channel Protocol Ports
------+-------------+-----------+-----------------------------------------------
1 Po1(SU) LACP
```

Twe1/0/1(P)

Twe1/0/2(P)

设置触发条件。使用layer2关键字匹配特定源MAC地址和LACP MAC地址作为目标。

<#root>

switch#debug platform hardware fed active capture trigger interface twentyFiveGigE1/0/1 ingress
switch#debug platform hardware fed active capture trigger layer2

0000.0000.0000 0180.c200.0002 <-- match source MAC: any, match destination MAC: LACP MAC address

Capture trigger set successful.

注意：在PSV捕获上定义的MAC地址0000.0000.0000表示match any。

验证触发条件已设置。

<#root>

switch#

show platform hardware fed active capture trigger

Trigger Set:
Ingress Interface: TwentyFiveGigE1/0/1
Dest Mac: 0180.c200.0002

触发PST后，状态将显示为“已完成”。

<#root>

switch#

show platform hardware fed active capture status

Asic: 0

Status: Completed

使用下一条命令**分析**PSV捕获输出。预期会看到LACP和PAgP帧被传送到CPU。

<#root>

switch#

show platform hardware fed active capture summary

Trigger: Ingress Interface:TwentyFiveGigE1/0/1 Dest Mac:0180.c200.0002

Input Output State Reason

Tw1/0/1 cpuQ 1 PUNT

Bridged

控制平面策略器(CoPP)

CoPP基本上是应用于数据平面（硬件）和控制平面(CPU)之间的管道的QoS监察器，以避免高CPU问题。CoPP可以过滤LACP和PAgP帧（如果这些帧超过该功能建立的阈值）。

验证CoPP是否丢弃LACP数据包。

<#root>

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

此命令**L2控制**队列的输出没有丢弃：

<#root>

switch#

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

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

(set)

```
     Queue         Queue
```

QId PlcIdx

Queue Name

Enabled Rate

Rate

Drop(Bytes) Drop(Frames)

---

0 11 DOT1X Auth Yes 1000 1000 0 0

1 1 L2 Control Yes 2000 2000 0 0 <-- L2 Control queue filters LACP packets, rate set to 2000 (packets pe

2 14 Forus traffic Yes 4000 4000 0 0

<output omitted>

* NOTE: CPU queue policer rates are configured to the closest hardware supported value

CPU Queue Policer Statistics

==========

Policer Index | Policer Accept Bytes | Policer Accept Frames | Policer Drop Bytes | Policer Drop Frames

---

0 0 0 0 0

1 13328202 79853 0 0 <-- QId = 1 matches policer index (level 1) = 1, no drops

```
2          0               0              0             0

<output omitted>

                  Second Level Policer Statistics
==========================================================================



20 34149506 389054 0 0 <-- Policer index (level 2) no drops



21         76896           596            0             0

Policer Index Mapping and Settings
--------------------------------------------------------------------
level-2   :   level-1                    (default)   (set)
PlcIndex  :   PlcIndex                     rate       rate
--------------------------------------------------------------------



20 : 1 2 8 13000 13000 <-- Policer index (level 1) = 1 matches policer index (level 2) = 20



21        :   0 4 7 9 10 11 12 13 14 15      6000       6000
==========================================================================

               Second Level Policer Config
==========================================================================
    level-1 level-2                            level-2
QId PlcIdx  PlcIdx  Queue Name                 Enabled
--------------------------------------------------------------------------
0    11      21      DOT1X Auth                  Yes



1 1 20 L2 Control Yes



2    14      21      Forus traffic               Yes

<output omitted>
```

它不会超过L2控制队列。当观察到相反情况时，需要捕获控制平面数据包。

FED CPU数据包捕获

如果您已确保在接口级别收到LACP数据包，EPC和ELAM/PSV确认已将LACP帧传送到CPU，并且在CoPP级别未观察到丢包现象

，则使用FED CPU数据包捕获工具。

FED CPU数据包捕获会告诉您数据包从硬件传送到CPU的原因，还会告诉您数据包被发送到哪个CPU队列。FED CPU数据包捕获还可以捕获注入硬件的CPU生成的数据包。

<#root>

debug platform software fed sw active punt packet-capture set-filter *<filter>*

debug platform software fed switch active punt packet-capture start

debug platform software fed switch active punt packet-capture stop

show platform software fed switch active punt packet-capture status

show platform software fed switch active punt packet-capture brief

debug platform software fed sw active inject packet-capture set-filter <*filter*>

debug platform software fed switch active inject packet-capture start

debug platform software fed switch active inject packet-capture stop

show platform software fed switch active inject packet-capture status

show platform software fed switch active inject packet-capture brief

传送

定义数据包捕获以仅过滤LACP数据包。

<#root>

switch#

debug platform software fed sw active punt packet-capture set-filter "eth.dst==0180.c200.0002"

Filter setup successful. Captured packets will be cleared

开始捕获。

<#root>

switch#

debug platform software fed sw active punt packet-capture start

Punt packet capturing started.

如果不使用LACP速率快速计时器，请在30秒后（至少）停止它。

<#root>

switch#

debug platform software fed switch active punt packet-capture stop

```
Punt packet capturing stopped.
```

Captured 11 packet(s)

检查FED CPU数据包捕获状态。

```
<#root>

switch#
```

show platform software fed switch active punt packet-capture status

```
Punt packet capturing: disabled. Buffer wrapping: disabled
```

Total captured so far: 11 packets.

```
 Capture capacity : 4096 packets
```

Capture filter : "eth.dst==0180.c200.0002"

分析FED CPU数据包捕获输出。

<#root>

switch#

show platform software fed switch active punt packet-capture brief

Punt packet capturing: disabled. Buffer wrapping: disabled

Total captured so far: 11 packets

. Capture capacity : 4096 packets

Capture filter : "eth.dst==0180.c200.0002"

------ Punt Packet Number: 1, Timestamp: 2023/03/31 00:27:54.141 ------
 interface :

physical: GigabitEthernet1/0/2[if-id: 0x0000000a]

, pal: GigabitEthernet1/0/2 [if-id: 0x0000000a]

<-- interface that punted the frame

 metadata :

cause: 96 [Layer2 control protocols],

sub-cause: 0,

q-no: 1

, linktype: MCP_LINK_TYPE_LAYER2 [10]

<-- LACP frame was punted due to L2 ctrl protocol to queue 1 (L2 control)

ether hdr :

dest mac: 0180.c200.0002, src mac: f04a.0205.d602 <-- source and destination MAC addresses

ether hdr : ethertype: 0x8809

------ Punt Packet Number: 2, Timestamp: 2023/03/31 00:27:58.436 ------
interface :

physical: GigabitEthernet1/0/4[if-id: 0x0000000c]

, pal: GigabitEthernet1/0/4 [if-id: 0x0000000c]
metadata :

cause: 96 [Layer2 control protocols]

, sub-cause: 0,

q-no: 1

, linktype: MCP_LINK_TYPE_LAYER2 [10]
 ether hdr : dest mac: 0180.c200.0002,

src mac: f04a.0205.d604

 ether hdr : ethertype: 0x8809

------ Punt Packet Number: 3, Timestamp: 2023/03/31 00:28:00.758 ------
 interface :

physical: GigabitEthernet1/0/1[if-id: 0x00000009]

, pal: GigabitEthernet1/0/1 [if-id: 0x00000009]
 metadata :

cause: 96 [Layer2 control protocols]

, sub-cause: 0,

q-no: 1

, linktype: MCP_LINK_TYPE_LAYER2 [10]
 ether hdr : dest mac: 0180.c200.0002,

src mac: f04a.0205.d601

```
 ether hdr : ethertype: 0x8809

------ Punt Packet Number: 4, Timestamp: 2023/03/31 00:28:11.888 ------
 interface :
```

physical: GigabitEthernet1/0/3[if-id: 0x0000000b]

```
, pal: GigabitEthernet1/0/3 [if-id: 0x0000000b]
 metadata  :
```

cause: 96 [Layer2 control protocols]

```
, sub-cause: 0,
```

q-no: 1

```
, linktype: MCP_LINK_TYPE_LAYER2 [10]
 ether hdr : dest mac: 0180.c200.0002,
```

src mac: f04a.0205.d603

```
 ether hdr : ethertype: 0x8809
```

注入

定义数据包捕获以仅过滤LACP数据包。

```
    <#root>

switch#
```

debug platform software fed sw active inject packet-capture set-filter "eth.dst==0180.c200.0002"

```
Filter setup successful. Captured packets will be cleared
```

开始捕获。

```
<#root>

switch#
```

debug platform software fed sw active inject packet-capture start

```
Punt packet capturing started.
```

如果不使用LACP速率快速计时器，请在30秒后（至少）停止它。

```
<#root>

switch#
```

debug platform software fed switch active inject packet-capture stop

```
Inject packet capturing stopped.
```

Captured 12 packet(s)

检查FED CPU数据包捕获状态。

<#root>

switch#

show platform software fed sw active inject packet-capture status

Inject packet capturing: disabled. Buffer wrapping: disabled

Total captured so far: 12 packets.

Capture capacity : 4096 packets

Capture filter : "eth.dst==0180.c200.0002"

分析FED CPU数据包捕获输出。

<#root>

switch#

show platform software fed sw active inject packet-capture brief

Inject packet capturing: disabled. Buffer wrapping: disabled

Total captured so far: 12

packets. Capture capacity : 4096 packets

Capture filter : "eth.dst==0180.c200.0002"

------ Inject Packet Number: 1, Timestamp: 2023/03/31 19:59:26.507 ------
interface :

pal: GigabitEthernet1/0/2 [if-id: 0x0000000a] <-- interface that LACP frame is destined to

metadata :

cause: 1 [L2 control/legacy]

, sub-cause: 0,

q-no: 7

, linktype: MCP_LINK_TYPE_LAYER2 [10]

<-- cause L2 ctrl, queue=7 (high priority)

ether hdr :

dest mac: 0180.c200.0002, src mac: f04a.0206.1902 <-- source and destination MAC addresses

ether hdr : ethertype: 0x8809

------ Inject Packet Number: 2, Timestamp: 2023/03/31 19:59:28.538 ------
interface :

pal: GigabitEthernet1/0/3 [if-id: 0x0000000b]

metadata :

cause: 1 [L2 control/legacy]

, sub-cause: 0,

q-no: 7

, linktype: MCP_LINK_TYPE_LAYER2 [10]
ether hdr :

dest mac: 0180.c200.0002, src mac: f04a.0206.1903

 ether hdr : ethertype: 0x8809

------ Inject Packet Number: 3, Timestamp: 2023/03/31 19:59:30.050 ------
 interface :

pal: GigabitEthernet1/0/1 [if-id: 0x00000009]

 metadata :

cause: 1 [L2 control/legacy]

, sub-cause: 0,

q-no: 7

, linktype: MCP_LINK_TYPE_LAYER2 [10]
 ether hdr :

dest mac: 0180.c200.0002, src mac: f04a.0206.1901

 ether hdr : ethertype: 0x8809

------ Inject Packet Number: 4, Timestamp: 2023/03/31 19:59:33.467 ------
 interface : pal:

GigabitEthernet1/0/4 [if-id: 0x0000000c]

 metadata :

cause: 1 [L2 control/legacy]

, sub-cause: 0,

q-no: 7

, linktype: MCP_LINK_TYPE_LAYER2 [10]
 ether hdr :

 dest mac: 0180.c200.0002, src mac: f04a.0206.1904

 ether hdr : ethertype: 0x8809

相关信息

- IEEE 802编号
- IEEE -链路聚合控制协议
- 第2层配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9200交换机）-章节：配置EtherChannel
- 第2层配置指南，Cisco IOS XE Cupertino 17.7.x（Catalyst 9300交换机）-章节：配置EtherChannel
- 第2层配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9400交换机）-章节：配置EtherChannel
- 第2层配置指南，Cisco IOS XE Cupertino 17.9.x（Catalyst 9500交换机）-章节：配置EtherChannel
- 第2层配置指南，Cisco IOS XE Cupertino 17.9.x（Catalyst 9600交换机）-章节：配置EtherChannel
- 章节：接口和硬件命令- show platform hardware fed switch forward interface
- 在Catalyst 9000交换机上配置FED CPU数据包捕获
- 技术支持和文档 - Cisco Systems

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）。