

Catalyst 9000スイッチでのEtherChannelのトラブルシューティング

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[LACPフラグ](#)

[ネットワーク図](#)

[LACPの動作の確認](#)

[基本チェック](#)

[デバッグ](#)

[PAgPの動作確認](#)

[基本チェック](#)

[デバッグ](#)

[Etherchannelプログラミングの確認](#)

[ソフトウェアの確認](#)

[ハードウェアの確認](#)

[プラットフォームツール](#)

[組み込みパケットキャプチャ \(EPC\)](#)

[プラットフォーム転送](#)

[パケット状態ベクトル\(PSV\)](#)

[コントロールプレーンポリサー\(CoPP\)](#)

[FED CPUパケットキャプチャ](#)

[関連情報](#)

はじめに

このドキュメントでは、Catalyst 9000シリーズスイッチでのEtherChannelの理解とトラブルシューティングの方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Catalyst 9000シリーズスイッチのアーキテクチャ

- Cisco IOS® XEソフトウェアアーキテクチャ
- Link Aggregation Control Protocol (LACP) と Port Aggregation Protocol (PAgP)

使用するコンポーネント

このドキュメントの情報は、次のハードウェアのバージョンに基づくものです。

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

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

この機能に関する制限事項、制限事項、設定オプション、警告、およびその他の関連情報の最新の情報については、シスコの公式リリースノートと設定ガイドを参照してください。

EtherChannelは、スイッチ、ルータ、およびサーバ間に耐障害性のある高速リンクを提供します。EtherChannelを使用してデバイス間の帯域幅を増やし、ボトルネックが発生する可能性が高いネットワーク内の任意の場所に配置します。EtherChannelは、リンクの損失に対する自動リカバリを提供し、残りのリンクに負荷を再分配します。リンクに障害が発生すると、EtherChannelは介入することなく、障害が発生したリンクからチャンネル内の残りのリンクにトラフィックをリダイレクトします。

EtherChannelは、ネゴシエーションなしで設定することも、PAgPまたはLACPのいずれかのLink Aggregation Protocol (LACP ; リンク集約プロトコル) をサポートして動的にネゴシエートすることもできます。

PAgPまたはLACPをイネーブルにすると、スイッチはパートナーのアイデンティティと各インターフェイスの機能を学習します。次に、スイッチは、同様の設定を持つインターフェイスを単一の論理リンク (チャンネルまたは集約ポート) に動的にグループ化します。スイッチは、ハードウェア、管理、およびポートパラメータの制約に基づいてこれらのインターフェイスグループを作成します。

LACPフラグ

LACPフラグは、起動時にポートチャンネルパラメータをネゴシエートするために使用されます。すべてのフラグの意味を確認します。

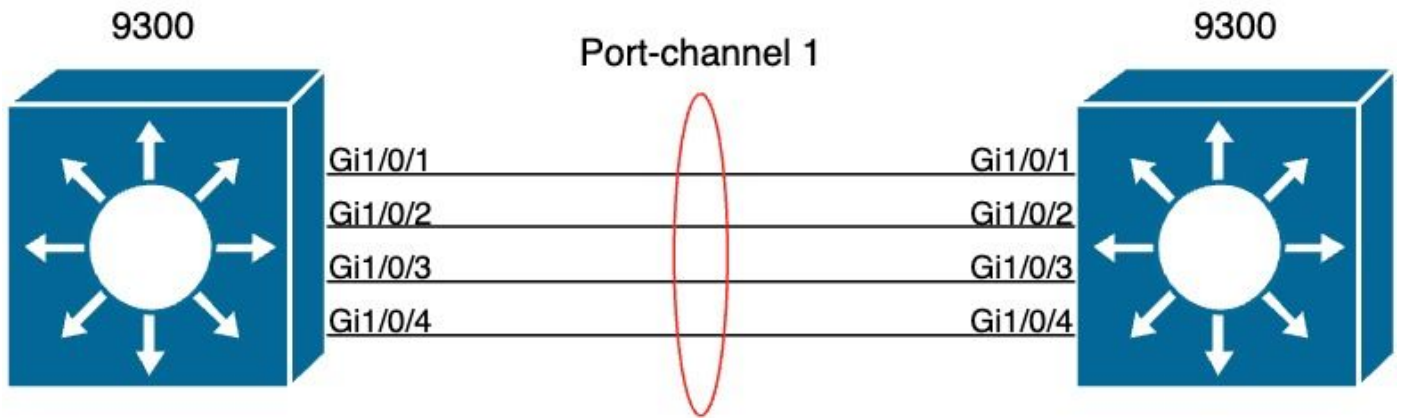
フラグ	ステータス
LACPアクティビティ (下位ビット)	0 =パッシブモード 1 =アクティブモード
LACPタイムアウト : LACPの送受信タイムアウトを示します。	0 =長時間タイムアウト。3 x 30秒 (デフォルト) 1 =ショートタイムアウト。3 x 1秒 (LACPレート高速)
集約	0 =個々のリンク (集約では考慮されない) 1 =集約可能 (集約の候補)
同期	0 =リンクが同期していない (状態が良好ではない) 1 =リンクは同期しています (良好な状態)
収集中	0 =フレームを受信/処理する準備ができていない 1 =フレームを受信/処理する準備ができています
配布	0 =フレームを送信/送信する準備ができていない 1 =フレームの送信/送信準備
デフォルト	0 =受信したPDUの情報をパートナー用に使用する 1 =パートナーのデフォルト情報を使用
期限切れ (最上位ビット)	0 = PDUが期限切れ、 1 = PDUは有効

LACPフラグの期待値は、P (ポートチャネルにバンドル) ステータスに達するためには0x3D (16進数) または0111101 (バイナリ) です。

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

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

ネットワーク図



LACPの動作の確認

このセクションでは、LACPプロトコルの正しい状態と動作を確認する方法について説明します。

基本チェック

次のコマンドを使用して、LACP出力を確認します。

```
<#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>
```

最初のコマンド出力は、スイッチシステムID(SID)と優先度 (LACPの場合) を表示します。

```
<#root>
switch#
show lacp sys-id

32768,
f04a.0206.1900 <-- Your system MAC address
```

LACPネイバーの詳細 (動作モード、ネイバーシステムのデバイスID、優先順位など) を確認します。

```
<#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

Port	Flags	LACP port Priority	Admin	Oper	Port	Port
------	-------	-----------------------	-------	------	------	------

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
```

各インターフェイスで送受信されるLACPパケットを検証します。破損したLACPパケットが検出されると、Pkts Errカウンタが増加します。

```
<#root>
```

```
switch#
```

```
show lacp 1 counters
```

Port	LACPDUs		Marker		Marker Response		LACPDUs	
	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								

LACPのインターフェイスアカウンティングを確認するオプションもあります。

```
<#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
```

デバッグ

LACP同期がない場合、またはリモートピアでLACPが実行されていない場合、Syslogメッセージが生成されます。

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

次のコマンドを使用して、LACPデバッグを有効にします。

```
<#root>
```

```
debug lacp [event|packet|fsm|misc]
```

```
debug condition <condition>
```

LACPネゴシエーションの問題が発生した場合は、LACPデバッグを有効にして、その原因を分析します。

```
<#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
```

必要に応じて、特定のインターフェイスに対してデバッグ条件を有効にし、出力をフィルタ処理します。

```
<#root>
```

```
switch#
```

```
debug condition interface gigabitEthernet 1/0/1
```

 注:LACPのデバッグはプラットフォームに依存しません。

デバッグとフィルタの設定を検証します。

```
<#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
```

LACPデバッグを分析し、show loggingコマンドを使用して表示します。デバッグ出力には、ポートチャネルインターフェイスがアップする前の最後のLACPフレームが示されています。

```
<#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.02
LACP: Part: tlv:2, tlv-len:20, key:0x1, p-pri:0x8000, p:0x102, p-state:0xF, 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: 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.02
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

```

LACPデバッグの2つの最も重要な行に焦点を当てると、いくつかの概念を定義する価値があります。

```
<#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	説明
ACT	アクター (あなた) を表します
部品	パートナー (ネイバーピア) を表す
キー	これは、設定されているポートチャネルの番号です。
Pステート	ポート状態を表し、最も重要な概念です。8ビット (LACPフラグ) で構成される詳細は「背景説明」セクションを参照してください。
S-MAC#S-MAC#	LACPで使用されるシステムMACアドレスである。

 注：デバッグに表示される値は16進数です。値を正しく読み取るには、値を10進法または2進法に変換する必要があります。

PAgPの動作確認

このセクションでは、PAgPプロトコルの正しい状態と動作を確認する方法について説明します。

基本チェック

次のコマンドを使用して、PAgPの出力を確認します。

```
<#root>
```

```
show pagp <channel-group number> neighbor
```

```
show pagp <channel-group number> counters
```

```
show interfaces <interface ID> accounting
```

動作モード、パートナーシステムID、ホスト名、優先順位など、PAgPネイバーの詳細を確認します。

```
<#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 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
```

各インターフェイスで送受信されたPAgPパケットの出力の詳細を検証します。破損したPAgPパケットが検出されると、Pkts Errカウンタが増加します。

```
<#root>
```

```
switch#
```

```
show pagp 1 counters
```

Port	Information		Flush		PAgP	
	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
```

PAgPのインターフェイスのアカウントिंगを確認するオプションもあります。

```
<#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

デバッグ

PAgPネゴシエーションの問題が見つかった場合は、その原因を分析するためにPAgPデバッグを有効にします。

```
<#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
```

必要に応じて、特定のインターフェイスに対してデバッグ条件を有効にし、出力をフィルタリングします。

```
<#root>
```

```
switch#
```

```
debug condition interface gigabitEthernet 1/0/1
```

 注:PAGPのデバッグはプラットフォームに依存しません。

デバッグとフィルタの設定を検証します。

```
<#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
```

PAGPデバッグを分析します。デバッグ出力には、ポートチャネルインターフェイスが起動する前の最後のPAGPフレームが示されています。

```
<#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-cap: 10001
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-cap: 10001
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
```

Etherchannelプログラミングの確認

このセクションでは、EtherChannelのソフトウェアとハードウェアの設定を確認する方法について説明します。

ソフトウェアの確認

ソフトウェアエントリを検証します。

```
<#root>
```

```
show run interface <interface ID>
```



```
show etherchannel <channel-group number> summary
```

EtherChannel設定を確認します。

```
<#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
```

すべてのポートメンバーがポートチャンネルにバンドルされていることを検証します。

```
<#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)
```

ハードウェアの確認

ハードウェアレベルでソフトウェアエントリを検証します。

```
<#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>
```

ポートチャンネルとバンドルインターフェイスのIDを確認します。

```
<#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
```

IF IDセクションに注目し、値 (16進数) が前のコマンドで表示されたID (10進数) と等しいことを確認します。

```
<#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
```

```
0000000000000000c
```

```
    true    7777777777777777
```

```
<-- IfId Hex 0xc = 12 decimal
```

```
GigabitEthernet1/0/3
```

```
0000000000000000b
```

```
    true    bbbbbbbbbbbbbbbb
```

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

```
GigabitEthernet1/0/2
```

```
0000000000000000a
```

```
    true    dddddddddddddddd
```

```
<-- IfId Hex 0xa = 10 decimal
```

```
GigabitEthernet1/0/1
```

```
00000000000000009
```

```
    true    eeeeeeeeeeeeeeee
```

```
<-- IfId Hex 0x9 = 10 decimal
```

次のコマンドを使用して、ポートチャネルのIF IDを取得します。この値は、前のコマンドの値と一致している必要があります。

```
<#root>
```

```
Switch#
```

```
show platform software fed switch active ifm mappings etherchannel
```

Mappings Table

Chan Interface IF_ID

1 Port-channel1

0x0000004c

次のコマンドにIF IDを使用します。表示される情報は、前述の手順で収集した出力と一致している必要があります。

<#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

```

プラットフォームツール

次の表に、これらのツールと機能を使用するタイミングを理解するために使用できるツールと機能を示します。

ツール	レベル	いつ使用するか
EPC	ハードウェア	このコマンドを使用して、物理インターフェイスに到着したLACPフレー

	およびソフトウェア	ムを検証したり、CPUに到達していることを検証できます。
プラットフォーム転送	ハードウェア	LACPフレームがスイッチに到着したことを確認した場合は、このツールを使用してスイッチの内部転送の判断を確認します。
PSV	ハードウェア	LACPフレームがスイッチに到着したことを確認した場合は、このツールを使用してスイッチの内部転送の判断を確認します。
CoPP	ハードウェア	ただし、ハードウェアの観点からパケットがCPUに転送された場合は、ソフトウェア(CPU)レベルでは見られません。この機能により、ハードウェアとCPU間のパスでLACPフレームが廃棄された可能性が高くなります。
FED CPUパケットキャプチャ	[ソフトウェア (Software)]	これを使用して、LACPフレームが正しいキューを通じてCPUにパントされたことを検証します。また、CPUがLACPフレームをハードウェアに送り返しているかどうかを検証します。

注：これらのツールを使用して分析できるのはLACPプロトコルだけですが、PAgPフレームの分析にも使用できます。

組み込みパケット キャプチャ (EPC)

Wireshark(EPC)を設定し、入力/出力LACP PDUをキャプチャするコマンド。

```
<#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 <destination MAC address>|<destination MAC address>]
```

```
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：コマンドは特権モードで入力します。

Wiresharkキャプチャを設定します。

 ヒント：特定のバンドルインターフェイスや特定の送信元MACアドレスに注目する場合は、インターフェイスを調整し、MACキーワードを照合します。

```
<#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：キャプチャで定義された宛先MACアドレス0180.c200.0002は、LACPフレームのフィルタリングに役立ちます。

Wiresharkが正しく設定されていることを確認します。

```
<#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)
```

キャプチャを開始します。

```
<#root>
```

```
switch#
```

```
monitor capture CAP start
```

```
Started capture point : CAP
```

LACP rate fast timerを使用しない場合は、少なくとも30秒後に停止します。

```
<#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

キャプチャされたフレーム :

<#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

特定のフレームのLACPフィールドを確認する必要がある場合は、detailedキーワードを使用します。

<#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: 000000000000000000000000
TLV Type: Terminator (0x00)
TLV Length: 0x00
Pad: 0000000000000000000000000000000000000000000000000000000000000000...
```

```
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:Wiresharkの出力形式は、9200デバイスでは異なる場合があります、スイッチから読み取ることはできません。キャプチャをエクスポートし、その場合はPCから読み取ります。

プラットフォーム転送

転送情報をデバッグして、ハードウェア転送プレーンのパケットパスをトレースするには、`show platform hardware fed switch <switch number or role> forward interface`コマンドを使用します。このコマンドは、ユーザ定義のパケットをシミュレートし、ハードウェアフォワーディングプレーンから転送情報を取得します。このコマンドで指定したパケットパラメータに基づいて、入力ポートでパケットが生成されます。また、PCAPファイルに保存されているキャプチャされたパケットから完全なパケットを提供することもできます。

この項では、インターフェイス転送固有のオプション、つまり`show platform hardware fed switch {switch_num|active|standby} forward interface`コマンドで使用できるオプションについてのみ詳しく説明します。

<#root>

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

プラットフォーム転送キャプチャを定義します。この場合、フレールCAP.pcap ム1が分析されます。

<#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.
```

プラットフォーム転送キャプチャが完了すると、次のsyslogメッセージが表示されます。

```
<#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
```

プラットフォーム転送キャプチャを分析します。「出力」の項では、内部転送の決定が何であったかを説明しています。LACPフレームとPAgPフレームはCPUにバントされることが想定されています。

```
<#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 (L2
```

```
###[ 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 .
```

```
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 *****
```

パケット状態ベクトル(PSV)

PSVはプラットフォーム転送キャプチャに似ていますが、PSVはトリガー基準に一致するネットワークからのライブ入力フレームをキャプチャする点が異なります。

 注:PSVがサポートされているのは、C9500-32C、C9500-32QC、C9500-24Y4C、C9500-48Y4C、およびC9606Rの各プラッ

 トフォームだけです。

```
<#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> <destination MAC address>
```

```
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
```

相互に接続された2つのC9500-48Y4Cは、次のポートチャネルとPSVキャプチャに使用されます。

```
<#root>
```

```
switch#
```

```
show etherchannel 1 summary
```

```
<output omitted>
```

```
Group Port-channel Protocol Ports
```

```
-----+-----+-----+-----
```

```
1 Po1(SU) LACP
```

```
Twel/0/1(P)
```

```
Twel/0/2(P)
```

トリガー基準を設定します。layer2キーワードを使用して、特定の送信元MACアドレスおよび宛先としてのLACP MACアドレスと照合します。

```
<#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:PSVキャプチャで定義されているMACアドレス0000.0000.0000は、すべてに一致することを意味します。

検証トリガー基準が設定されました。

```
<#root>
```

```
switch#
```

```
show platform hardware fed active capture trigger
```

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

PSTがトリガーされると、ステータスはCompletedと表示されます。

```
<#root>
```

```
switch#
```

```
show platform hardware fed active capture status
```

```
Asic: 0
```

```
Status: Completed
```

次のコマンドを使用して、PSVキャプチャの出力を分析します。LACPフレームとPAgPフレームがCPUにパントされることが想定されています。

```
<#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
```

コントロールプレーンポリサー(CoPP)

CoPPは基本的に、高CPUの問題を回避するために、データプレーン(ハードウェア)とコントロールプレーン(CPU)間のパイプに適用されるQoSポリサーです。CoPPは、LACPフレームとPAgPフレームが機能によって確立されたしきい値を超えた場合、これら

のフレームをフィルタリングできます。

CoPPがLACPパケットをドロップするかどうかを検証します。

<#root>

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

次のコマンドL2制御キューの出力には、ドロップがありません。

<#root>

switch#

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

```
CPU Queue Statistics =====
```

```
(set)
```

```
Queue Queue
```

```
QId PlcIdx
```

```
Queue Name
```

```
Enabled Rate
```

```
Rate
```

```
Drop(Bytes) Drop(Frames) -----
```

```
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 Polic
```

```
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 -----
```

```
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 =====
```

```
1 1 20 L2 Control Yes
```


2 14 21 Forus traffic Yes
<output omitted>

L2制御キューに過大な負荷がかかることは想定されていません。コントロールプレーンパケットキャプチャは、その逆が見られる場合に必要です。

FED CPUパケットキャプチャ

LACPパケットがインターフェイスレベルで受信され、ELAM/PSVでLACPフレームがCoPPレベルで観察されるドロップなしでCPUにパントされたことを確認した場合は、FED CPUパケットキャプチャツールを使用します。

FED CPUパケットキャプチャは、パケットがハードウェアからCPUにパントされた理由を示し、パケットが送信されたCPUキューも示します。FED CPUパケットキャプチャでは、ハードウェアに注入されたCPUによって生成されたパケットもキャプチャできません。

<#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
```

パント

LACPパケットのみをフィルタリングするパケットキャプチャを定義します。

```
<#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
```

キャプチャの開始.

```
<#root>
```

```
switch#
```

```
debug platform software fed sw active punt packet-capture start
```

```
Punt packet capturing started.
```

LACP rate fast timerを使用しない場合は、少なくとも30秒後に停止します。

```
<#root>
```

```
switch#
```

```
debug platform software fed switch active punt packet-capture stop
```

```
Punt packet capturing stopped.
```

Captured 11 packet(s)

FED CPUパケットキャプチャステータスを確認します。

<#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"

FED CPUパケットキャプチャの出力を分析します。

<#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

挿入

LACPパケットのみをフィルタリングするパケットキャプチャを定義します。

<#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

キャプチャの開始.

```
<#root>
```

```
switch#
```

```
debug platform software fed sw active inject packet-capture start
```

Punt packet capturing started.

LACP rate fast timerを使用しない場合は、少なくとも30秒後に停止します。

```
<#root>
```

```
switch#
```

```
debug platform software fed switch active inject packet-capture stop
```

Inject packet capturing stopped.

Captured 12 packet(s)

FED CPUパケットキャプチャステータスを確認します。

<#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"

FED CPUパケットキャプチャの出力を分析します。

<#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

関連情報

- [IEEE 802番号](#)
- [IEEE:Link Aggregation Control Protocol \(リンク集約制御プロトコル\)](#)
- [『Cisco IOS XE Amsterdam 17.3.x Layer 2 Configuration Guide』 \(Catalyst 9200スイッチ \) – 章 : EtherChannelの設定](#)
- [『Cisco IOS XE Cupertino 17.7.x Layer 2 Configuration Guide』 \(Catalyst 9300スイッチ \) – 章 : EtherChannelの設定](#)
- [『Cisco IOS XE Amsterdam 17.3.x Layer 2 Configuration Guide』 \(Catalyst 9400スイッチ \) – 章 : EtherChannelの設定](#)
- [『Cisco IOS XE Cupertino 17.9.x Layer 2 Configuration Guide』 \(Catalyst 9500スイッチ \) – 章 : EtherChannelの設定](#)
- [『Cisco IOS XE Cupertino 17.9.x Layer 2 Configuration Guide』 \(Catalyst 9600スイッチ \) – 章 : EtherChannelの設定](#)
- [章 : インターフェイスおよびハードウェアコマンド – show platform hardware fed switch forward interface](#)

- [Catalyst 9000スイッチでのFED CPUパケットキャプチャの設定](#)
- [テクニカル サポートとドキュメント - Cisco Systems](#)

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。