

Catalyst 9000 스위치의 EtherChannel 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[LACP 플러그](#)

[네트워크 다이어그램](#)

[LACP 작업 확인](#)

[기본 검사](#)

[디버그](#)

[PAgP 작업 확인](#)

[기본 검사](#)

[디버그](#)

[Etherchannel 프로그래밍 확인](#)

[소프트웨어 확인](#)

[하드웨어 확인](#)

[플랫폼 톨](#)

[EPC\(Embedded Packet Capture\)](#)

[플랫폼 전달](#)

[패킷 상태 벡터\(PSV\)](#)

[CoPP\(컨트롤 플레인 폴리서\)](#)

[FED CPU 패킷 캡처](#)

[관련 정보](#)

소개

이 문서에서는 Catalyst 9000 Series 스위치의 EtherChannel을 이해하고 문제를 해결하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Catalyst 9000 Series 스위치 아키텍처
- Cisco IOS® XE 소프트웨어 아키텍처
- LACP(Link Aggregation Control Protocol) 및 PAgP(Port Aggregation Protocol)

사용되는 구성 요소

이 문서의 내용은 다음 하드웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

제한 사항, 제한 사항, 컨피그레이션 옵션, 주의 사항 및 이 기능에 대한 기타 관련 세부 사항에 대한 최신 정보는 Cisco 공식 릴리스 노트 및 컨피그레이션 가이드를 참조하십시오.

EtherChannel은 스위치, 라우터 및 서버 간에 내결함성 고속 링크를 제공합니다. EtherChannel을 사용하여 디바이스 간 대역폭을 늘리고, 병목 현상이 발생할 가능성이 높은 네트워크 어디에나 구축합니다. EtherChannel은 링크 손실에 대한 자동 복구를 제공하며 나머지 링크에 로드를 재분배합니다. 링크에 장애가 발생하면 EtherChannel은 장애가 발생한 링크에서 채널의 나머지 링크로 트래픽을 아무런 작업 없이 리디렉션합니다.

EtherChannel은 협상 없이 구성하거나 PAgP 또는 LACP와 같은 Link Aggregation Protocol의 지원을 통해 동적으로 협상할 수 있습니다.

PAgP 또는 LACP를 활성화하면 스위치는 파트너의 ID와 각 인터페이스의 기능을 학습합니다. 그런 다음 유사한 컨피그레이션의 인터페이스를 단일 논리적 링크(채널 또는 종합 포트)로 동적으로 그룹화합니다. 스위치는 하드웨어, 관리 및 포트 매개변수 제약을 기반으로 인터페이스 그룹을 설정합니다.

LACP 플래그

LACP 플래그는 시작할 때 포트 채널 매개변수를 협상하는 데 사용됩니다. 모든 깃발의 의미를 살펴 보십시오.

플래그	상태
LACP 활동(하위 비트)	0 = 패시브 모드 1 = 활성 모드

LACP Timeout(LACP 시간 초과): 전송/수신된 LACP 시간 초과를 나타냅니다.	0 = 긴 시간 초과. 3 x 30초(기본값) 1 = 짧은 시간 초과. 3 x 1초(빠른 LACP 속도)
Aggregation(어그리게이션)	0 = 개별 링크(어그리게이션에 고려되지 않음) 1 = 집계 가능(잠재적 집계 후보)
Synchronization(동기화)	0 = 링크가 동기화되지 않음(비정상 상태) 1 = 링크가 동기화 중(정상 상태)
Collecting(수집 중)	0 = 프레임을 수신/처리할 준비가 되지 않았습니다. 1 = 프레임을 수신/처리할 준비가 되었습니다.
배포	0 = 프레임을 전송/전송할 준비가 되지 않았습니다. 1 = 프레임을 전송/전송할 준비 완료
Defaulted(기본값 지정됨)	0 = 파트너에 대해 수신된 PDU의 정보를 사용합니다. 1 = 파트너에 대한 기본 정보 사용
만료됨(최상위 비트)	0 = PDU가 만료되었습니다. 1 = PDU가 유효함

LACP 플래그의 예상 값은 0x3D(16진수) 또는 0111101(이진)입니다.

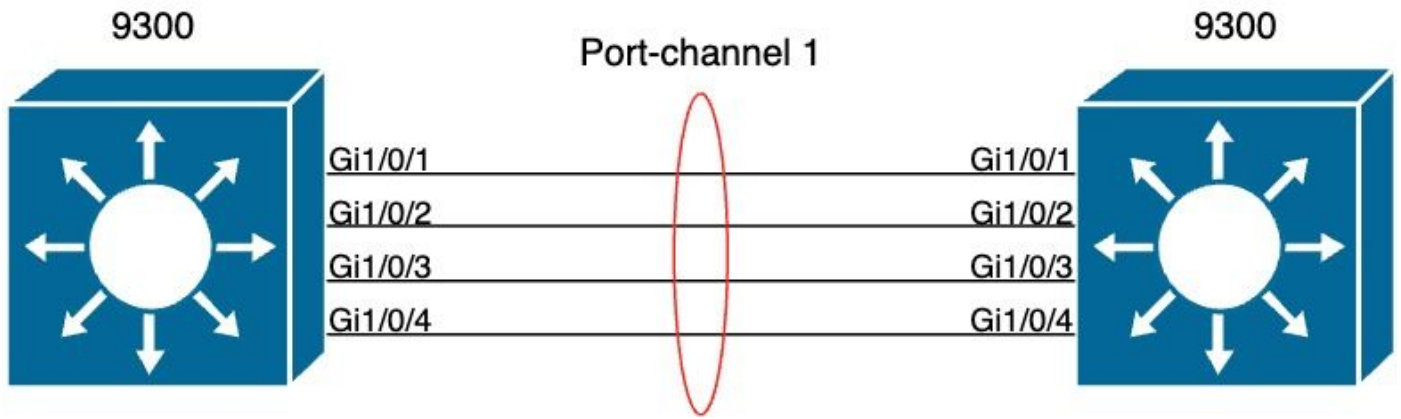
```

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

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

```

네트워크 다이어그램



LACP 작업 확인

이 섹션에서는 LACP 프로토콜의 올바른 상태 및 작동을 확인하는 방법에 대해 설명합니다.

기본 검사

다음 명령을 사용하여 LACP 출력을 확인합니다.

```
<#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>
```

첫 번째 명령 출력은 스위치 시스템 ID 및 그 우선순위(LACP의 경우)를 표시합니다.

```
<#root>
```

```
switch#
```

```
show lacp sys-id
```

```
32768,
```

f04a.0206.1900 <-- Your system MAC address

운영 모드, 네이버 시스템 Dev ID, 우선 순위 등 LACP 네이버의 세부 정보를 확인합니다.

<#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
--	-----	-----	-----	--------	-------

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

각 인터페이스에서 보내고 받은 LACP 패킷을 확인합니다. 손상된 LACP 패킷이 감지되면 Pkts Err 카운터가 증가합니다.

<#root>

switch#

show lacp 1 counters

Port	LACPDU		Marker		Marker Response		LACPDU	
	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								

LACP에 대한 인터페이스 어카운팅을 확인하는 옵션도 있습니다.

<#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

디버그

LACP 동기화가 없거나 원격 피어에서 LACP를 실행하지 않는 경우 Syslog 메시지가 생성됩니다.

```
%ETC-5-L3DONTBNL2: Gig1/0/1 suspended: LACP currently not enabled on the remote port.  
%ETC-5-L3DONTBNL2: Gig/1/0/1 suspended: LACP currently not enabled on the remote port.
```

다음 명령을 사용하여 LACP 디버그를 활성화합니다.

```
<#root>
```

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

```
debug condition <condition>
```

LACP 협상 문제가 발견되면 LACP 디버그를 활성화하여 그 이유를 분석합니다.

```
<#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
```

필요한 경우 특정 인터페이스에 대한 디버그 조건도 활성화하고 출력을 필터링합니다.

```
<#root>
```

```
switch#
```

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

 참고: LACP 디버깅은 플랫폼에 구애받지 않습니다.

디버그 및 필터가 설정되었는지 확인합니다.

```
<#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 디버그를 분석하고 show logging 명령을 사용하여 표시합니다. 디버그 출력에는 포트 채널 인터페이스가 시작되기 전 마지막 LACP 프레임이 표시됩니다.

<#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.020

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 디버그의 가장 중요한 두 줄에 초점을 맞추면 일부 LACP PDU 개념을 정의하는 데 가치가 있는 몇 가지 개념이 있습니다.

<#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

개념	설명
행위	배우(you)를 의미합니다.

부품	파트너(neighbor/peer)를 의미합니다.
키	구성된 포트 채널의 번호입니다.
P-상태	포트 스테이트를 의미하며 가장 중요한 개념입니다. 8비트(LACP 플래그)로 구축됩니다. 자세한 내용은 Background Information 섹션을 참조하십시오.
S-맥	LACP에서 사용하는 시스템 mac 주소입니다.

 참고: 디버깅에 표시되는 값은 16진수입니다. 값을 제대로 읽으려면 십진수 또는 이진 시스템으로 변환해야 합니다.

PAgP 작업 확인

이 섹션에서는 PAgP 프로토콜의 올바른 상태 및 작동을 확인하는 방법에 대해 설명합니다.

기본 검사

다음 명령으로 PAgP 출력을 확인합니다.

```
<#root>
```

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

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

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

PAgP 인접 디바이스의 세부 정보(예: 운영 모드, 파트너 시스템 ID, 호스트 이름, 우선순위)를 확인합니다.

```
<#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
```

```
Port          Partner Name          Partner Group
```

```
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
```

각 인터페이스에서 보내고 받은 PAgP 패킷의 출력 세부사항을 검증합니다. 손상된 PAgP 패킷이 감지되면 Pkts Err 카운터가 증가합니다.

```
<#root>
```

```
switch#
```

```
show pagp 1 counters
```

```
Port          Information Sent  Recv      Flush Sent  Recv      PAgP 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
```

PAgP에 대한 인터페이스 어카운팅도 확인할 수 있는 옵션이 있다.

<#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

디버그

PAGP 협상 문제가 발견되면 PAGP 디버그를 활성화하여 이유를 분석합니다.

<#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
```

필요한 경우 특정 인터페이스에 대한 디버그 조건을 활성화하고 출력을 필터링합니다.

```
<#root>
```

```
switch#
```

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

 참고: PAgP 디버깅은 플랫폼에 구애받지 않습니다.

디버그 및 필터가 설정되었는지 확인합니다.

```
<#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 디버그를 분석합니다. 디버그 출력은 포트 채널 인터페이스가 시작되기 전 마지막 PAGP 프레임에 표시합니다.

<#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: 10000, your device ID: f04a.0206.1900, learn-cap: 2, port-priority: 128, sent-port-ifindex: 9, group-cap: 10000

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: 1000
your device ID: f04a.0205.d600, learn-cap: 2, port-priority: 128, sent-port-ifindex: 9, group-cap: 1000
```

```
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 프로그래밍 확인

이 섹션에서는 EtherChannel에 대한 소프트웨어 및 하드웨어 설정을 확인하는 방법에 대해 설명합니다.

소프트웨어 확인

소프트웨어 항목을 확인합니다.

```
<#root>
```

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

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

EtherChannel 컨피그레이션을 확인합니다.

```
<#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
```

모든 포트 멤버가 포트 채널에 번들되는지 확인합니다.

```
<#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)

하드웨어 확인

하드웨어 레벨에서 소프트웨어 항목을 검증합니다.

```
<#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>
```

포트 채널 및 번들 인터페이스의 ID를 확인합니다.

```
<#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 섹션에 초점을 맞추고 값(16진수)이 이전 명령에서 관찰된 ID(10진수)와 같은지 확인합니다.

<#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
```

다음 명령으로 포트 채널의 IF ID를 가져옵니다. 이 값은 이전 명령의 값과 일치해야 합니다.

```
<#root>
Switch#
show platform software fed switch active ifm mappings etherchannel

Mappings Table

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

다음 명령에는 IF ID를 사용합니다. 표시된 정보는 이전에 수집된 출력과 일치해야 합니다.

```
<#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

```

플랫폼 툴

이 표에서는 사용 시기를 쉽게 이해할 수 있도록 어떤 도구와 기능을 사용할 수 있는지 보여줍니다.

툴	수준	사용 시기
EPC	하드웨어 및 소프트웨어	물리적 인터페이스에 연결된 LACP 프레임을 검증하거나 CPU에 도달했는지 검증하는 데 사용합니다.
플랫폼 전달	하드웨어	LACP 프레임이 스위치에 연결된 것을 확인한 경우 이 툴을 사용하여 스위치의 내부 포워딩 결정을 확인합니다.
PSV	하드웨어	LACP 프레임이 스위치에 연결된 것을 확인한 경우 이 툴을 사용하여 스위치의 내부 포워딩 결정을 확인합니다.
CoPP	하드웨어	그러나 하드웨어 관점에서 패킷이 CPU에 전달된 경우 소프트웨어(CPU) 레벨에서 해당 패킷이 표시되지 않았습니다. 이 기능은 하드웨어와 CPU 간의 경로를 따라 LACP 프레임을 삭제했을 가능성이 높습니다.
FED CPU 패킷 캡처	소프트웨어	LACP 프레임이 올바른 큐를 통해 CPU에 펀트되었는지 검증하고, CPU가 LACP 프레임을 다시 하드웨어로 전송하는지 검증하는 데 사용합니다.



참고: 이러한 툴을 사용하여 LACP 프로토콜만 분석되지만, PAgP 프레임을 분석하는 데에도 사용할 수 있습니다.

EPC(Embedded Packet Capture)

EPC(Wireshark)를 설정하고 인그레스/이그레스 LACP PDU를 캡처하는 명령입니다.

```
<#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]
```

 참고: 명령은 권한 모드에서 입력됩니다.

Wireshark 캡처 설정

 팁: 특정 번들 인터페이스 및/또는 특정 소스 MAC 주소에 중점을 두려는 경우 인터페이스를 조정하고 mac 키워드를 일치시킵니다.

```
<#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
```

 참고: 캡처에 정의된 대상 MAC 주소 0180.c200.0002를 사용하면 LACP 프레임을 필터링할 수 있습니다.

Wireshark가 올바르게 구성되었는지 확인합니다.

```
<#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)
```

캡처를 시작합니다.

```
<#root>
```

```
switch#
```

```
monitor capture CAP start
```

```
Started capture point : CAP
```

LACP 속도 고속 타이머를 사용하지 않는 경우 최소 30초 후에 중지합니다.

```
<#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

캡처된 프레임:

<#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

특정 프레임에서 LACP 필드를 확인해야 하는 경우 detailed 키워드를 사용합니다.

<#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]
```

 참고: Wireshark 출력 형식은 9200 디바이스에서 다를 수 있으며 스위치에서 읽을 수 없습니다. 캡처를 내보내고 PC에서 읽습니다(해당하는 경우).

플랫폼 전달

포워딩 정보를 디버깅하고 하드웨어 포워딩 플레인에서 패킷 경로를 추적하려면 이 명령을 `show platform hardware fed switch <switch number or role> forward interface` 사용합니다. 이 명령은 사용자 정의 패킷을 시뮬레이션하고 하드웨어 포워딩 플레인에서 포워딩 정보를 검색합니다. 이 명령에서 지정한 패킷 매개 변수를 기반으로 인그레스 포트에서 패킷이 생성됩니다. PCAP 파일에 저장된 캡처된 패킷에서 전체 패킷을 제공할 수도 있습니다.

이 항목에서는 인터페이스 전달 관련 옵션, 즉 명령에서 사용할 수 있는 옵션에 대해서만 `show platform hardware fed switch {switch_num|active|standby} forward interface` 설명합니다.

<#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>
```

Platform Forward 캡처를 정의합니다. 이 경우, CAP.pcap 프레임 1이 분석됩니다.

```
<#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.

Platform Forward 캡처가 완료되면 다음 Syslog 메시지가 표시됩니다.

```
<#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
```

Platform Forward 캡처 분석 Egress(이그레스) 섹션에서는 내부 포워딩 결정이 무엇이었는지 알려줍니다. LACP 및 PAgP 프레임은 CPU에 펀팅될 것으로 예상됩니다.

```
<#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 (L

###[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

패킷 상태 벡터(PKV)

PKV는 트리거 기준과 일치하는 네트워크에서 실시간 인그레스 프레임 캡처한다는 점을 제외하면 Platform Forward 캡처와 유사합니다.



참고: PKV는 C9500-32C, C9500-32QC, C9500-24Y4C, C9500-48Y4C 및 C9606R 플랫폼에서만 지원됩니다.

<#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
```

서로 연결된 2개의 C9500-48Y4C가 다음 포트 채널 및 PSV 캡처에 사용됩니다.

<#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)

트리거 기준을 설정합니다. destination으로 특정 소스 MAC 주소 및 LACP MAC 주소와 일치시키려면 layer2 키워드를 사용합니다.

<#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.



참고: PSV 캡처에 정의된 MAC 주소 0000.0000.0000은 match any를 의미합니다.

트리거 기준이 설정되었는지 확인합니다.

<#root>

switch#

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

```
Trigger Set:
Ingress Interface: TwentyFiveGigE1/0/1
```


Dest Mac: 0180.c200.0002

PST가 트리거되면 상태가 Completed(완료됨)로 표시됩니다.

```
<#root>
```

```
switch#
```

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

```
Asic: 0
```

```
Status: Completed
```

다음 명령으로 PSV 캡처 출력을 분석합니다. LACP 및 PAgP 프레임이 CPU에 편팅되는 것을 볼 수 있을 것으로 예상됩니다.

```
<#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

CoPP(컨트롤 플레인 폴리서)

CoPP는 기본적으로 높은 CPU 문제를 방지하기 위해 데이터 플레인(하드웨어)과 컨트롤 플레인(CPU) 사이의 파이프에 적용되는 QoS 폴리서입니다. CoPP는 LACP 및 PAgP 프레임이 기능에 의해 설정된 임계값을 초과할 경우 필터링할 수 있습니다.

CoPP가 LACP 패킷을 삭제하는지 확인합니다.

```
<#root>
```

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

이 명령의 출력인 L2 제어 대기열에는 삭제가 없습니다.

```
<#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)

QId	PlcIdx	Queue Name	Enabled	Rate	Drop(Bytes)	Drop(Frames)
0	11	DOT1X Auth	Yes	1000	1000	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
---	----	---------------	-----	------	------	---

<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)
PlcIdx	PlcIdx	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	level-2 Enabled
0	11	21	DOT1X Auth	Yes

1 1 20 L2 Control Yes

2 14 21 Forus traffic Yes

<output omitted>

L2 제어 대기열을 압도할 것으로 예상되지 않습니다. 반대의 상황이 관찰될 경우 컨트롤 플레인 패킷 캡처가 필요합니다.

FED CPU 패킷 캡처

LACP 패킷이 인터페이스 레벨에서 수신되었는지 확인한 경우, EPC 및 ELAM/PSV에서 확인한 LACP 프레임이 CoPP 레벨에서 관찰된 삭제 없이 CPU에 펀팅되었는지 확인한 다음 FED CPU 패킷 캡처 툴을 사용합니다.

FED CPU 패킷 캡처는 패킷이 하드웨어에서 CPU로 펀트된 이유를 알려주며 패킷이 전송된 CPU 대기열도 알려줍니다. FED CPU 패킷 캡처는 하드웨어에 삽입된 CPU에서 생성된 패킷도 캡처할 수 있습니다.

<#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
```

핀트

LACP 패킷만 필터링하도록 패킷 캡처를 정의합니다.

```
<#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

캡처를 시작합니다.

```
<#root>
```

```
switch#
```

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

Punt packet capturing started.

LACP rate fast timer를 사용하지 않는 경우 최소 30초 후에 중지합니다.

```
<#root>
```

switch#

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

Punt packet capturing stopped.

Captured 11 packet(s)

FED CPU 패킷 캡처 상태를 확인합니다.

<#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 패킷 캡처 출력을 분석합니다.

<#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

삽입

LACP 패킷만 필터링하도록 패킷 캡처를 정의합니다.

```
<#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

캡처를 시작합니다.

```
<#root>
```

```
switch#
```

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

Punt packet capturing started.

LACP rate fast timer를 사용하지 않는 경우 최소 30초 후에 중지합니다.

```
<#root>
```

```
switch#
```

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

Inject packet capturing stopped.

Captured 12 packet(s)

FED CPU 패킷 캡처 상태를 확인합니다.

<#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 패킷 캡처 출력을 분석합니다.

```
<#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

관련 정보

- [IEEE 802 번호](#)
- [IEEE - Link Aggregation Control Protocol](#)
- [레이어 2 컨피그레이션 가이드, Cisco IOS XE Amsterdam 17.3.x\(Catalyst 9200 스위치\) - 장: EtherChannel 구성](#)
- [레이어 2 컨피그레이션 가이드, Cisco IOS XE Cupertino 17.7.x\(Catalyst 9300 스위치\) - 장: EtherChannel 구성](#)
- [레이어 2 컨피그레이션 가이드, Cisco IOS XE Amsterdam 17.3.x\(Catalyst 9400 스위치\) - 장: EtherChannel 구성](#)

- [레이어 2 컨피그레이션 가이드, Cisco IOS XE Cupertino 17.9.x\(Catalyst 9500 스위치\) - 장: EtherChannel 구성](#)
- [레이어 2 컨피그레이션 가이드, Cisco IOS XE Cupertino 17.9.x\(Catalyst 9600 스위치\) - 장: EtherChannel 구성](#)
- [장: 인터페이스 및 하드웨어 명령 - show platform hardware fed switch forward interface](#)
- [Catalyst 9000 스위치에서 FED CPU 패킷 캡처 구성](#)
- [기술 지원 및 문서 - Cisco Systems](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.