

POS 인터페이스에서 "Line Protocol is Down" 문제 해결

목차

[소개](#)

[show interface pos 명령 해석](#)

[POS 프로토콜 스택 개요](#)

[디버그 명령 사용](#)

[회선 프로토콜이 HDLC로 다운됨](#)

[회선 프로토콜이 PPP로 다운됨](#)

[링크 구성](#)

[링크 유지 관리\(keepalive 사용\)](#)

[링크 종료](#)

[샘플 문제 해결 순서](#)

[문제 해결 참고 사항](#)

[루프백 테스트](#)

[APS가 있는 회선 프로토콜 상태](#)

[알려진 문제](#)

[관련 정보](#)

소개

이 문서에서는 회선 프로토콜 상태가 "down"인 POS(SONET) 라우터 인터페이스를 통해 패킷을 해결하는 방법에 대해 설명합니다.

또한 회선 프로토콜이 다운되었음을 식별할 수 있을 뿐만 아니라 PPP(Point-to-Point Protocol) 및 HDLC(High-Level Data Link Control) 캡슐화의 문제를 해결하는 데 사용할 **show** 및 **debug** 명령을 설명합니다. 또한 문서화된 랩 설정을 기반으로 하는 일반적인 문제 해결 시나리오를 소개합니다.

[show interface pos 명령 해석](#)

문서의 경우 **show interface pos**의 출력은 이 출력에 나와 있습니다. 디스플레이의 강조 표시된 부분과 설명을 확인합니다.

```
RTR12410-2#show interface pos 6/0
  POS6/0 is up, line protocol is down
!--- The line protocol is down . Hardware is Packet over SONET MTU 4470 bytes, BW 2488000 Kbit,
DLY 100 usec, rely 255/255, load 1/255          Encapsulation HDLC, crc 32, loopback not set
!--- The loopback has not been set. Keepalive set (10 sec)
!--- The keepalive is set as every ten seconds. Scramble disabled Last input never, output
00:00:05, output hang never Last clearing of "show interface" counters never Queueing strategy:
fifo Output queue 0/40, 0 drops; input queue 0/75, 0 drops 5 minute input rate 0 bits/sec, 0
```

```
packets/sec 5 minute output rate 0 bits/sec, 0 packets/sec 0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles 0 parity 0 input errors, 0 CRC, 0 frame, 0
overrun, 0 ignored, 0 abort 3 packets output, 1074 bytes, 0 underruns 0 output errors, 0
applique, 1 interface resets 0 output buffer failures, 0 output buffers swapped out 2 carrier
transitions
```

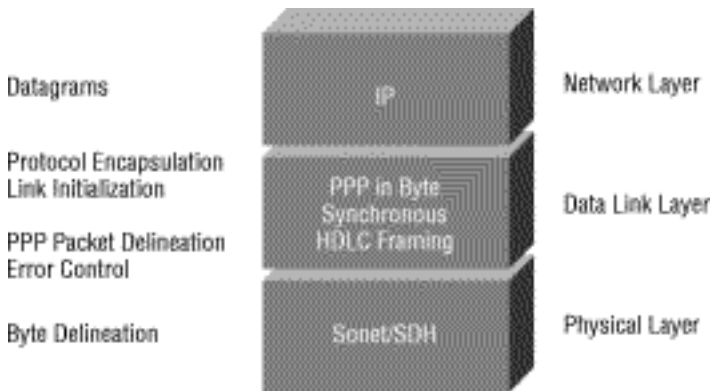
Cisco IOS® Command Reference는 라인 프로토콜 필드 상태가 "라인 프로토콜을 처리하는 소프트웨어 프로세스가 사용 가능한 라인(즉, keepalive가 성공함)으로 간주하는지 또는 관리자가 중단했는지 여부를 나타냅니다."

show interface pos 출력의 기타 중요한 필드는 다음과 같습니다.

- 캡슐화 - 인터페이스에 할당된 캡슐화 방법입니다.
- loopback - 루프백이 설정되었는지 여부를 나타냅니다.
- keepalive - keepalive가 설정되었는지 여부를 나타냅니다.

POS 프로토콜 스택 개요

이 다이어그램은 POS 인터페이스에서 사용되는 프로토콜 스택을 보여줍니다.



POS 인터페이스는 HDLC, PPP 및 Frame Relay 등 여러 캡슐화를 지원합니다. 따라서 Packet over SONET은 SONET을 통한 PPP 또는 HDLC를 통한 SONET에 보다 정확하게 PPP를 생성합니다. 이 문서에서는 프레임 릴레이 캡슐화에 대해 다루지 않습니다.

PPP 및 HDLC는 밀접한 관련이 있으며 다음과 같은 특징을 공유합니다.

- 헤더와 트레일러가 있는 프레임 구조를 제공합니다. 트레일러는 오류 검사를 제공합니다.
- 패킷과 프레임이 시작하고 끝나는 정확한 위치를 수신자에 대해 정의하는 프레임 설명을 제공합니다. HDLC 및 PPP에서 프레임 설명은 특수 인터프레임 채우기 패턴 또는 유휴 패턴을 통해 제공됩니다. 패턴은 0x7E 또는 0111 110입니다.
- 최소 및 최대 패킷 길이를 정의합니다.
- IP 패킷을 전송하고 수신자가 도착하는 프레임 내에서 정확한 패킷 유형을 확인할 수 있는 방법을 제공합니다.

그러나 PPP와 HDLC는 서로 관련이 있지만 서로 다른 debug 명령을 사용하여 회선 프로토콜 문제를 해결합니다.

디버그 명령 사용

다양한 debug privileged EXEC 명령의 출력은 많은 인터넷워킹 이벤트의 프로토콜 상태 및 네트워크 활동과 관련된 진단 정보를 제공합니다.

주의: 디버깅 출력은 CPU 프로세스에 높은 우선 순위가 할당되므로 시스템을 사용할 수 없게 만들 수 있습니다. 따라서 debug 명령만 사용하여 특정 문제를 해결하거나 Cisco 기술 지원 담당자와의 문제 해결 세션 중에만 사용합니다. 또한 네트워크 트래픽이 낮고 사용자가 적은 기간 동안 debug 명령을 사용하는 것이 가장 좋습니다. 이러한 기간 동안 디버깅을 수행하면 디버그 명령 처리 오버헤드가 증가하여 시스템 사용에 영향을 미칠 가능성이 줄어듭니다. debug 명령을 모두 사용하는 경우 특정 no debug 명령 또는 no debug all 명령으로 비활성화해야 합니다.

이러한 debug 명령은 POS 인터페이스 문제를 해결할 때 유용합니다. 이러한 각 명령의 기능 및 출력에 대한 자세한 내용은 Cisco Debug Command Reference 게시물을 참조하십시오.

- **debug serial interface** - HDLC keepalive 패킷이 증가하는지 확인합니다. 그렇지 않으면 인터페이스 카드 또는 네트워크에 가능한 타이밍 문제가 있습니다.
- **debug ppp negotiation** - PPP 시작 중에 전송된 PPP 패킷을 표시합니다. 여기서 PPP 옵션은 협상됩니다.
- **debug ppp packet** - 보내고 받는 PPP 패킷을 표시합니다. 이 명령은 낮은 수준의 패킷 덤프를 표시합니다.
- **debug ppp errors** - PPP 연결 협상 및 작업과 연결된 PPP 오류(예: 불법 또는 잘못된 프레임)를 표시합니다.

자세한 내용은 [직렬 회선 문제 해결](#)을 참조하십시오.

회선 프로토콜이 HDLC로 다운됨

HDLC는 POS 라우터 인터페이스의 기본 캡슐화 유형입니다. HDLC는 국제 표준이지만 공급업체 구현은 하나 이상의 필드 또는 크기와 형식의 헤더 또는 트레일러에 따라 다릅니다. SONET를 정의하는 Telecordia GR-253 사양은 HDLC-over-SONET 매핑에 대해 설명합니다(Issue 3, Section 3.4.2.3, pp.3-59 참조). HDLC 프레임이 SONET 프레임과 바이트 정렬되도록 지정하고 자체 동기화 스크램블러, CRC(순환 이중화 검사) 및 HDLC 플래그 패턴을 인터프레임 채우기로 사용하여 도착하는 HDLC 프레임의 변수 특성을 고려하도록 지정합니다.

show interface pos 명령에 회선 및 프로토콜이 HDLC 캡슐화를 통해 다운된 것으로 표시되면 **debug serial interface** 명령을 사용하여 연결 오류의 원인으로 회선 문제를 격리할 수 있습니다. HDLC는 keepalive를 사용하고 디버그 출력에 있는 3개의 카운터 값을 보고합니다.

- **myseq** - 라우터가 keepalive 패킷을 원격 라우터로 전송할 때마다 하나씩 증가합니다.
- **mineseen** - mineseeq 카운터의 값은 원격 라우터가 라우터에서 수신한 마지막 myseq 시퀀스 번호를 반영합니다. 원격 라우터는 이 값을 표시된 카운터에 저장하고 keepalive 패킷의 해당 값을 라우터로 전송합니다.
- **yourseen** - 원격 라우터에서 keepalive 패킷에서 라우터가 받은 myseq 시퀀스 번호의 값을 반영합니다.

mineseq, yoursen 및 myseen 필드의 keepalive 값이 후속 출력 라인에서 증가하지 않으면 연결의 한 쪽에서 문제가 발생합니다. myseq 및 mineseen 필드의 값 차이가 3을 초과하면 라인이 중단되고 인터페이스가 재설정됩니다.

keepalive가 양쪽 끝에서 올바르게 수신될 때 HDLC 연결에 대한 **debug serial interface** 명령의 샘플 출력입니다.

```
hswan-12008-2a#debug serial interface
Serial network interface debugging is on
hswan-12008-2a#
Oct 31 11:47:16: POS4/0: HDLC myseq 180, mineseen 0*, yourseen 1, line up
```

```
Oct 31 11:47:17: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS4/0,
changed state to up
!--- Local router sees a remote keepalive with a sequence number of 1. Oct 31 11:47:26: POS4/0:
HDLC myseq 181, mineseen 181*, yourseen 2, line up Oct 31 11:47:36: POS4/0: HDLC myseq 182,
mineseen 182*, yourseen 3, line up Oct 31 11:47:46: POS4/0: HDLC myseq 183, mineseen 183*,
yourseen 4, line up Oct 31 11:47:56: POS4/0: HDLC myseq 184, mineseen 184*, yourseen 5, line up
Oct 31 11:48:06: POS4/0: HDLC myseq 185, mineseen 185*, yourseen 6, line up !--- Keepalives are
sent every 10 seconds by default. !--- Both sides report incrementing sequence numbers.
```

원격 인터페이스가 종료되고 로컬 인터페이스에서 keepalive가 3개 이상 누락될 때 HDLC 연결에 대한 HDLC 연결에 대한 debug serial interface 명령의 샘플 출력입니다.

```
hswan-12008-2a#
Oct 31 11:49:46: POS4/0: HDLC myseq 195, mineseen 192, yourseen 13, line down
Oct 31 11:49:47: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS4/0,
changed state to down
!--- The local router has failed to receive three keepalives and !--- brings down the line
protocol. Note the difference between !--- "myseq 195" and "mineseen 192". Oct 31 11:49:56:
POS4/0: HDLC myseq 196, mineseen 192, yourseen 13, line down Oct 31 11:50:06: POS4/0: HDLC myseq
197, mineseen 192, yourseen 13, line down Oct 31 11:50:16: POS4/0: HDLC myseq 198, mineseen 192,
yourseen 13, line down Oct 31 11:50:26: POS4/0: HDLC myseq 199, mineseen 192, yourseen 13, line
down Oct 31 11:50:36: POS4/0: HDLC myseq 200, mineseen 0*, yourseen 1, line up Oct 31 11:50:37:
%LINEPROTO-5-UPDOWN: Line protocol on Interface POS4/0, changed state to up !--- After you
execute the no shut command on the remote router, !--- the local router receives a keepalive
again and brings up !--- the line protocol.
```

```
Oct 31 11:50:46: POS4/0: HDLC myseq 201, mineseen 201*, yourseen 2, line up
Oct 31 11:50:56: POS4/0: HDLC myseq 202, mineseen 202*, yourseen 3, line up
Oct 31 11:51:06: POS4/0: HDLC myseq 203, mineseen 203*, yourseen 4, line up
Oct 31 11:51:16: POS4/0: HDLC myseq 204, mineseen 204*, yourseen 5, line up
Oct 31 11:51:26: POS4/0: HDLC myseq 205, mineseen 205*, yourseen 6, line up
Oct 31 11:51:36: POS4/0: HDLC myseq 206, mineseen 206*, yourseen 7, line up
!--- After the shut/no shut, the remote router re-initialized its !--- sequence number.
```

회선 프로토콜이 PPP로 다운됨

[RFC 1661](#) 은 PPP를 프로토콜로 정의합니다. POS 인터페이스는 [RFC 1662](#) 에 지정된 대로 HDLC(High-Level Data Link Control) 유사 프레임에서 PPP를 지원하며, 레이어 2에서 데이터 캡슐화를 지원합니다. HDLC 유사 프레임의 PPP에 대한 프레임 형식은 이 그림에 나와 있습니다.



RFC 2615는 SONET 또는 SDH 링크를 통해 PPP 캡슐화를 사용하도록 지정합니다. PPP는 포인트-투-포인트 링크에 사용하도록 설계되었으며 링 토폴로지에서 포인트투포인트 회로로 프로비저닝 되는 SONET 또는 SDH 링크에 적합합니다.

포인트 투 포인트 링크를 표시할 때 PPP는 상태 다이어그램에서 그릴 수 있는 여러 가지 단계를 거칩니다. 캐리어 감지 또는 네트워크 관리자 컨피그레이션과 같은 외부 이벤트가 물리적 레이어를 사용할 준비가 되었음을 나타내는 경우 PPP는 링크 설정 단계로 진행합니다. 이 단계로 전환하면 LCP(Link Control Protocol)에 대한 UP 이벤트가 생성되며, 이 이벤트는 여러 기능을 제공합니다. 한 가지 기능은 링크가 제대로 작동하는 시기와 장애가 발생한 시기를 결정하는 것입니다. 포인트 투 포인트 링크를 통해 통신을 설정하려면 PPP 링크의 각 끝은 먼저 LCP 패킷을 전송하여 데이터 링크를 구성하고 테스트해야 합니다.

그런 다음 PPP는 하나 이상의 네트워크 레이어 프로토콜을 선택하고 구성하려면 NCP(Network Control Protocol) 패킷을 전송해야 합니다. 선택한 각 네트워크 레이어 프로토콜이 구성되면 각 네트워크 레이어 프로토콜의 데이터그램을 링크를 통해 전송할 수 있습니다.

이 표에는 LCP 패킷의 세 가지 클래스가 나열되어 있습니다.

LCP 패킷 클래스	LCP 패킷 유형	목적
링크 구성	Configure-Request, Configure-Ack, Configure-Nak 및 Configure-Reject	링크를 설정하고 구성하는 데 사용됩니다.
링크 종료	Terminate-Request 및 Terminate-Ack	링크를 종료하는 데 사용됩니다.
링크 유지 관리	코드 거부, 프로토콜 거부, 에코 요청, 에코 응답 및 폐기 요청	링크를 관리하고 디버깅하는 데 사용됩니다.

링크 구성

LCP는 Configure 패킷의 교환을 통해 연결을 설정하는 데 사용됩니다. Configure-Ack 패킷이 전송 및 수신되면 이 교환이 완료되고 LCP Opened 상태가 입력됩니다.

이 샘플 출력은 POS 인터페이스의 LCP 링크 컨피그레이션 단계를 캡처합니다.

```
4d01h: PO3/1 LCP: State is Open
4d01h: PO3/1 PPP: I pkt type 0x8021, datagramsize 14 LCP_UP
(0x639FCAD8) id 0 (0s.) queued 1/1/2
4d01h: PO3/1 PPP: Phase is UP
4d01h: PO3/1 IPCP: O CONFREQ [Closed] id 152 len 10
4d01h: PO3/1 IPCP: Address 172.16.1.1 (0x0306AC100101)
4d01h: PO3/1 PPP: I pkt type 0x8021, datagramsize 14
4d01h: PO3/1 IPCP: I CONFREQ [REQsent] id 1 len 10
4d01h: PO3/1 IPCP: Address 172.16.1.2 (0x0306AC100102)
4d01h: PO3/1 IPCP: O CONFACK [REQsent] id 1 len 10
4d01h: PO3/1 IPCP: Address 172.16.1.2 (0x0306AC100102)
4d01h: PO3/1 IPCP: I CONFACK [ACKsent] id 152 len 10
4d01h: PO3/1 IPCP: Address 172.16.1.1 (0x0306AC100101)
4d01h: PO3/1 IPCP: State is Open
4d01h: PO3/1 IPCP: Install route to 172.16.1.2
4d01h: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS3/1,
changed state to up
```

참고: PPP 캡슐화로 구성된 POS 인터페이스는 PPP 세션 설정을 계속 시도합니다. 따라서, 파이버가 제거되더라도 지속적인 문제가 발생할 경우 정기적으로 라인 프로토콜이 잠깐 나타납니다.

링크 유지 관리(keepalive 사용)

LCP Echo-Request 및 Echo-Reply 패킷은 링크의 양방향에 대한 레이어 2 루프백 메커니즘을 제공합니다. LCP Opened 상태에서 Echo-Request를 수신할 경우 Echo-Reply를 전송해야 합니다.

RFC 1661의 이 다이어그램은 PPP keepalive 패킷의 형식을 보여줍니다.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Code	Identifier	Length
Magic-Number		
Data...		

이러한 LCP 패킷에는 다음 키 필드가 포함됩니다.

- **코드** - 에코 요청에는 9이고 에코 응답에는 10입니다.
- **Identifier(식별자)** - 전송 시 Data(데이터) 필드의 내용이 변경될 때마다 그리고 이전 요청에 대해 유효한 회신이 수신될 때마다 Identifier(식별자) 필드를 변경해야 합니다. 재전송의 경우 식별자는 변경되지 않을 수 있습니다. 수신에서 Echo-Request의 Identifier 필드가 Echo-Reply 패킷의 Identifier 필드에 복사됩니다.
- **Magic-Number**—Magic-Number 필드는 4개의 8진수로 루프백(loopback) 상태의 링크 탐지를 지원합니다. Magic-Number Configuration Option이 성공적으로 협상될 때까지 Magic-Number를 0으로 전송해야 합니다. 자세한 내용은 RFC 1661의 Magic-Number 컨피그레이션 옵션을 참조하십시오.
- **Data**—Data 필드는 0초 이상이며 발신자가 사용할 해석되지 않은 데이터를 포함합니다. 데이터는 이진 값으로 구성될 수 있습니다. 필드의 끝은 길이로 표시됩니다.

keepalive가 활성화된 경우 **디버그 ppp 협상**의 예를 소개합니다.

```
4d01h: PO3/1 LCP: O ECHOREQ [Open] id 1 len 12 magic 0x1A45933B
4d01h: PO3/1 PPP: I pkt type 0xC021, datagramsize 16
4d01h: PO3/1 LCP: I ECHOREP [Open] id 1 len 12 magic 0x00000002
4d01h: PO3/1 LCP: Received id 1, sent id 1, line up
```

링크 종료

PPP는 언제든지 링크를 종료할 수 있습니다. 캐리어 손실, 인증 실패, 링크 품질 오류, 유효 기간 타이머의 만료, 링크 관리 종료 등이 발생할 수 있습니다.

LCP는 Terminate packets를 사용하여 링크를 닫습니다. Terminate-Request의 발신자는 Terminate-Ack을 받은 후 또는 Restart 카운터가 만료된 후에 연결을 끊어야 합니다. Terminate-Request의 수신자는 피어의 연결이 끊길 때까지 기다려야 하며, Terminate-Ack을 보낸 후 하나 이상의 Restart 시간이 경과할 때까지 연결을 끊지 않아야 합니다.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Code	Identifier	Length
Magic-Number		
Data...		

Terminate LCP packets에는 다음 키 필드가 포함됩니다.

- **코드** - Terminate-Request의 경우 5이고 Terminate-Ack의 경우 6입니다.
- **Identifier(식별자)** - 전송 시 Data(데이터) 필드의 내용이 변경될 때마다, 그리고 이전 요청에 대해 유효한 회신이 수신될 때마다 Identifier(식별자) 필드를 변경해야 합니다. 재전송의 경우 식별자는 변경되지 않을 수 있습니다. 수신 시 Terminate-Request의 Identifier 필드가 Terminate-Ack 패킷의 Identifier 필드에 복사됩니다.

Data 필드는 0개 이상의 8진이며 보낸 사람이 사용할 해석되지 않은 데이터를 포함합니다. 데이터는 모든 이진 값으로 구성될 수 있습니다. 필드의 끝은 길이를 표시합니다.

다음은 TERMREQ 패킷을 수신할 때 **debug ppp 협상** 출력의 예입니다.

```
4d01h: PO3/1 PPP: I pkt type 0xC021, datagramsize 8
4d01h: PO3/1 LCP: I TERMREQ [Open] id 4 len 4
4d01h: PO3/1 LCP: O TERMACK [Open] id 4 len 4
4d01h: PO3/1 PPP: I pkt type 0xC021, datagramsize 18
4d01h: PO3/1 IPCP: State is Closed
4d01h: PO3/1 PPP: Phase is TERMINATING
4d01h: PO3/1 LCP: I CONFREQ [TERMsent] id 1 len 14
4d01h: PO3/1 LCP: MRU 1500 (0x010405DC)
4d01h: PO3/1 LCP: MagicNumber 0x00000002 (0x050600000002)
4d01h: PO3/1 LCP: Dropping packet, state is TERMsent
!--- While in the TERMsent state, PPP should drop all other packets.
4d01h: PO3/1 IPCP: Remove route to 172.16.1.2
4d01h: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS3/1, changed state to down
```

샘플 문제 해결 순서

이 섹션에서는 PPP 캡슐화를 사용하는 POS 링크의 샘플 문제 해결 시나리오에 대해 설명합니다. 다음 컨피그레이션을 사용합니다.

라우터 A 컨피그레이션
<pre>interface POS1/0 ip address 1.1.1.6 255.255.255.0 no ip directed-broadcast encapsulation ppp crc 16 clock source internal</pre>
라우터 B 컨피그레이션
<pre>interface POS2/0 ip address 1.1.1.5 255.255.255.0 no ip directed-broadcast encapsulation ppp crc 16</pre>

참고: 이러한 디버그는 백투백 랩 설정의 두 라우터에서 캡처되었습니다. 따라서 clocking은 한 쪽의 internal로 설정되고 다른 쪽 끝의 line으로 기본값으로 설정됩니다.

디버그 ppp 협상

이 출력은 LCP의 링크 설정 단계에서 디버그 ppp 협상으로 캡처된 패킷 교환을 보여줍니다.

라우터 A 디버그 출력

Router A Debug Output

(1)

!--- The router sends an outgoing confreq. hswan-12008-2a# *Nov 7 08:27:00: %LINK-3-UPDOWN: Interface POS1/0, changed state to up *Nov 7 08:27:00: PO1/0 PPP: Treating connection as a dedicated line *Nov 7 08:27:00: PO1/0 PPP: Phase is ESTABLISHING, Active Open *Nov 7 08:27:00: PO1/0 LCP: O CONFREQ [Closed] id 7 len 14 *Nov 7 08:27:00: PO1/0 LCP: MRU 4470 (0x01041176) *Nov 7 08:27:00: PO1/0 LCP: MagicNumber 0x4F46AF4D (0x05064F46AF4D)

(4)

!--- Router A receives an incoming confreq from router B. *Nov 7 08:27:00: PO1/0 LCP: I CONFREQ [REQsent] id 45 len 14 *Nov 7 08:27:00: PO1/0 LCP: MRU 4470 (0x01041176) *Nov 7 08:27:00: PO1/0 LCP: MagicNumber 0x2631E6D2 (0x05062631E6D2)

(5)

!--- Router A responds with a confack and receives a !-- - confack from Router B. The LCP state is open. *Nov 7 08:27:00: PO1/0 LCP: O CONFACK [REQsent] id 45 len 14 *Nov 7 08:27:00: PO1/0 LCP: MRU 4470 (0x01041176) *Nov 7 08:27:00: PO1/0 LCP: MagicNumber 0x2631E6D2 (0x05062631E6D2) *Nov 7 08:27:00: PO1/0 LCP: I CONFACK [ACKsent] id 7 len 14 Nov 7 08:27:00: PO1/0 LCP: MRU 4470 (0x01041176) *Nov 7 08:27:00: PO1/0 LCP: MagicNumber 0x4F46AF4D (0x05064F46AF4D) *Nov 7 08:27:00: PO1/0 LCP: State is Open *Nov 7 08:27:00: PO1/0 PPP: Phase is UP

(7)

!--- Router A begins the IPCP stage and negotiates an IP address. !-- In this setup, the peer router already has an address and !-- sends it in a confreq. If the peer router accepts the address, !-- it responds with a confack. *Nov 7 08:27:00: PO1/0 IPCP: O CONFREQ [Closed] id 7 len 10 *Nov 7 08:27:00: PO1/0 IPCP: Address 1.1.1.6 (0x030601010106) *Nov 7 08:27:00: PO1/0 CDPCP: O CONFREQ [Closed] id 7 len 4 *Nov 7 08:27:00: PO1/0 IPCP: I CONFREQ [REQsent] id 9 len 10 *Nov 7 08:27:00: PO1/0 IPCP: Address 1.1.1.5 (0x030601010105) *Nov 7 08:27:00: PO1/0 IPCP: O CONFACK [REQsent] id 9 len 10 *Nov 7 08:27:00: PO1/0 IPCP: Address 1.1.1.5 (0x030601010105) *Nov 7 08:27:00: PO1/0 CDPCP: I CONFREQ [REQsent] id 9 len 4 *Nov 7 08:27:00: PO1/0 CDPCP: O CONFACK [REQsent] id 9 len 4 *Nov 7 08:27:00: PO1/0 IPCP: I CONFACK [ACKsent] id 7 len 10 *Nov 7 08:27:00: PO1/0 IPCP: Address 1.1.1.6 (0x030601010106) *Nov 7 08:27:00: PO1/0 IPCP: State is Open *Nov 7 08:27:00: PO1/0 CDPCP: I CONFACK [ACKsent] id 7 len 4 *Nov 7 08:27:00: PO1/0 CDPCP: State is Open *Nov 7 08:27:00: PO1/0 IPCP: Install route to 1.1.1.5 *Nov 7 08:27:01: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS1/0, changed state to up

라우터 B 디버그 출력

(2)

```
!--- Router B receives an incoming confrg from Router A.
hswan-12008-2b# Nov 7 10:29:19.043: PO2/0 LCP: I CONFREQ
[Open] id 7 len 14 Nov 7 10:29:19.043: PO2/0 LCP: MRU
4470 (0x01041176) Nov 7 10:29:19.043: PO2/0 LCP:
MagicNumber 0x4F46AF4D (0x05064F46AF4D) Nov 7
10:29:19.043: PO2/0 IPCP: State is Closed Nov 7
10:29:19.043: PO2/0 CDPCP: State is Closed Nov 7
10:29:19.043: PO2/0 PPP: Phase is TERMINATING Nov 7
10:29:19.043: PO2/0 PPP: Phase is ESTABLISHING
```

(3)

```
!--- Router B sends its own LCP confreq. Nov 7
10:29:19.043: PO2/0 LCP: O CONFREQ [Open] id 45 len 14
Nov 7 10:29:19.043: PO2/0 LCP: MRU 4470 (0x01041176) Nov
7 10:29:19.043: PO2/0 LCP: MagicNumber 0x2631E6D2
(0x05062631E6D2)
```

(6)

```
!--- Router B responds with a confack and receives a
confack from Router A. The LCP state is open. Nov 7
10:29:19.043: PO2/0 LCP: O CONFACK [Open] id 7 len 14
Nov 7 10:29:19.043: PO2/0 LCP: MRU 4470 (0x01041176) Nov
7 10:29:19.043: PO2/0 LCP: MagicNumber 0x4F46AF4D
(0x05064F46AF4D) Nov 7 10:29:19.043: PO2/0 IPCP: Remove
route to 1.1.1.6 Nov 7 10:29:19.047: PO2/0 LCP: I
CONFACK [ACKsent] id 45 len 14 Nov 7 10:29:19.047: PO2/0
LCP: MRU 4470 (0x01041176) Nov 7 10:29:19.047: PO2/0
LCP: MagicNumber 0x2631E6D2 (0x05062631E6D2) Nov 7
10:29:19.047: PO2/0 LCP: State is Open Nov 7
10:29:19.047: PO2/0 PPP: Phase is UP
```

(8)

```
!--- Router B also begins the IPCP stage and negotiates
an IP address. Nov 7 10:29:19.047: PO2/0 IPCP: O CONFREQ
[Closed] id 9 len 10 Nov 7 10:29:19.047: PO2/0 IPCP:
Address 1.1.1.5 (0x030601010105) Nov 7 10:29:19.047:
PO2/0 CDPCP: O CONFREQ [Closed] id 9 len 4 Nov 7
10:29:19.047: PO2/0 IPCP: I CONFREQ [REQsent] id 7 len
10 Nov 7 10:29:19.047: PO2/0 IPCP: Address 1.1.1.6
(0x030601010106) Nov 7 10:29:19.047: PO2/0 IPCP: O
CONFACK [REQsent] id 7 len 10 Nov 7 10:29:19.047: PO2/0
IPCP: Address 1.1.1.6 (0x030601010106) Nov 7
10:29:19.047: PO2/0 CDPCP: I CONFREQ [REQsent] id 7 len
4 Nov 7 10:29:19.047: PO2/0 CDPCP: O CONFACK [REQsent]
id 7 len 4 Nov 7 10:29:19.047: PO2/0 IPCP: I CONFACK
[ACKsent] id 9 len 10 Nov 7 10:29:19.047: PO2/0 IPCP:
Address 1.1.1.5 (0x030601010105) Nov 7 10:29:19.047:
PO2/0 IPCP: State is Open Nov 7 10:29:19.047: PO2/0
CDPCP: I CONFACK [ACKsent] id 9 len 4 Nov 7
10:29:19.047: PO2/0 CDPCP: State is Open Nov 7
10:29:19.047: PO2/0 IPCP: Install route to 1.1.1.6 *Nov
7 10:29:19.048: %LINEPROTO-5-UPDOWN: Line protocol on
Interface POS2/0, changed state to up
```

디버그 ppp 패킷

이 출력은 링크가 설정되는 동안 디버그 ppp 패킷으로 캡처된 패킷 교환을 보여줍니다. 이 디버그는

PPP 패킷에서 프로토콜 필드의 값을 캡처합니다. RFC 1661은 프로토콜 필드를 1 또는 2개의 8진수로 정의합니다. 이 필드의 값은 패킷의 Information(정보) 필드에 캡슐화된 데이터그램을 식별합니다.

"0****" - "3****" 범위의 프로토콜 필드 값은 특정 패킷의 네트워크 레이어 프로토콜을 식별하고, "8****" - "b****" 범위의 값은 연결된 NCP(Network Control Protocols)에 속하는 패킷을 식별합니다(있는 경우). "c****" - "f****" 범위의 프로토콜 필드 값은 패킷을 LCP와 같은 링크 레이어 제어 프로토콜로 식별합니다. 공급업체별 다양한 값도 있습니다. PPP [프로토콜 필드 값의 전체 목록을 보려면 여기를 클릭하십시오](#).

라우터 A 디버그 출력

(1)

```
*Nov 7 10:19:58: PO1/0 PPP: I pkt type 0xC021,
datagramsize 18
!--- 0xC021 identifies LCP. *Nov 7 10:19:58: PO1/0 LCP:
I CONFREQ [Closed] id 7 len 14 *Nov 7 10:19:58: PO1/0
LCP: MRU 4470 (0x01041176) *Nov 7 10:19:58: PO1/0 LCP:
MagicNumber 0x269933F4 (0x0506269933F4) *Nov 7 10:19:58:
PO1/0 LCP: O CONFREQ [Closed] id 57 len 14^Z *Nov 7
10:19:58: PO1/0 LCP: MRU 4470 (0x01041176) *Nov 7
10:19:58: PO1/0 LCP: MagicNumber 0x4FAE1B0C
(0x05064FAE1B0C) *Nov 7 10:19:58: PO1/0 LCP: O CONFACK
[REQsent] id 7 len 14 *Nov 7 10:19:58: PO1/0 LCP: MRU
4470 (0x01041176) *Nov 7 10:19:58: PO1/0 LCP:
MagicNumber 0x269933F4 (0x0506269933F4) *Nov 7 10:19:58:
%LINK-3-UPDOWN: Interface POS1/0, changed state to up
*Nov 7 10:19:58: PO1/0 PPP: I pkt type 0xC021,
datagramsize 18 *Nov 7 10:19:58: PO1/0 LCP: I CONFACK
[ACKsent] id 57 len 14ppp *Nov 7 10:19:58: PO1/0 PPP: I
pkt type 0x8021, datagramsize 14
!--- 0x8021 identifies IPCP, PPP internet protocol
control protocol. *Nov 7 10:19:58: PO1/0 LCP: MRU 4470
(0x01041176) *Nov 7 10:19:58: PO1/0 PPP: I pkt type
0x8207, datagramsize 8
!--- 0x8207 identifies Cisco discovery protocol control.
*Nov 7 10:19:58: PO1/0 LCP: MagicNumber 0x4FAE1B0C
(0x05064FAE1B0C) *Nov 7 10:19:58: PO1/0 IPCP: O CONFREQ
[Closed] id 15 len 10 *Nov 7 10:19:58: PO1/0 IPCP:
Address 1.1.1.6 (0x030601010106) *Nov 7 10:19:58: PO1/0
CDPCP: O CONFREQ [Closed] id 13 len 4 *Nov 7 10:19:58:
PO1/0 IPCP: I CONFREQ [REQsent] id 14 len 10packet *Nov
7 10:19:58: PO1/0 IPCP: Address 1.1.1.5 (0x030601010105)
*Nov 7 10:19:58: PO1/0 IPCP: O CONFACK [REQsent] id 14
len 10 *Nov 7 10:19:58: PO1/0 IPCP: Address 1.1.1.5
(0x030601010105) *Nov 7 10:19:58: PO1/0 PPP: I pkt type
0x8021, datagramsize 14 *Nov 7 10:19:58: PO1/0 CDPCP: I
CONFREQ [REQsent] id 15 len 4 *Nov 7 10:19:58: PO1/0
CDPCP: O CONFACK [REQsent] id 15 len 4 *Nov 7 10:19:58:
PO1/0 IPCP: I CONFACK [ACKsent] id 15 len 10 *Nov 7
10:19:58: PO1/0 PPP: I pkt type 0x8207, datagramsize 8
*Nov 7 10:19:58: PO1/0 IPCP: Address 1.1.1.6
(0x030601010106) *Nov 7 10:19:58: PO1/0 CDPCP: I CONFACK
[ACKsent] id 13 len 4 *Nov 7 10:19:59: PO1/0 PPP: I pkt
type 0x0207, datagramsize 376
!--- 0x0207 identifies Cisco Discovery Protocol (CDP).
*Nov 7 10:19:59: PO1/0 PPP: I pkt type 0x0207,
datagramsize 376 *Nov 7 10:19:59: PO1/0 PPP: I pkt type
0x0207, datagramsize 376 *Nov 7 10:19:59: %LINEPROTO-5-
UPDOWN: Line protocol on Interface POS1/0, changed state
```

to up

(3)

!--- ECHOREQ and ECHOREP packets for PPP keepalives use packet type values !--- of 0xC021. *Nov 7 10:20:05: PO1/0 PPP: I pkt type 0xC021, datagramsize 16 *Nov 7 10:20:05: PO1/0 LCP: I ECHOREQ [Open] id 1 len 12 magic 0x269933F4 *Nov 7 10:20:05: PO1/0 LCP: O ECHOREP [Open] id 1 len 12 magic 0x4FAE1B0C *Nov 7 10:20:07: PO1/0 LCP: O ECHOREQ [Open] id 1 len 12 magic 0x4FAE1B0C *Nov 7 10:20:07: PO1/0 PPP: I pkt type 0xC021, datagramsize 16 *Nov 7 10:20:07: PO1/0 PPP: O pkt type 0x0207, datagramsize 376 *Nov 7 10:20:07: PO1/0 LCP: I ECHOREP [Open] id 1 len 12 magic 0x269933F4 *Nov 7 10:20:07: PO1/0 LCP: Received id 1, sent id 1, line up

라우터 B 디버그 출력

(2)

Nov 7 12:22:16.947: PO2/0 PPP: I pkt type 0xC021, datagramsize 18
Nov 7 12:22:16.947: PO2/0 LCP: I CONFREQ [REQsent] id 57 len 14
Nov 7 12:22:16.947: PO2/0 LCP: MRU 4470 (0x01041176)
Nov 7 12:22:16.947: PO2/0 PPP: I pkt type 0xC021, datagramsize 18
Nov 7 12:22:16.947: PO2/0 LCP: MagicNumber 0x4FAE1B0C (0x05064FAE1B0C)
Nov 7 12:22:16.947: PO2/0 LCP: O CONFACK [REQsent] id 57 len 14
Nov 7 12:22:16.947: PO2/0 LCP: MRU 4470 (0x01041176)
Nov 7 12:22:16.947: PO2/0 LCP: MagicNumber 0x4FAE1B0C (0x05064FAE1B0C)
Nov 7 12:22:16.947: PO2/0 LCP: I CONFACK [ACKsent] id 7 len 14
Nov 7 12:22:16.947: PO2/0 LCP: MRU 4470 (0x01041176)
Nov 7 12:22:16.947: PO2/0 LCP: MagicNumber 0x269933F4 (0x0506269933F4)
Nov 7 12:22:16.947: PO2/0 IPCP: O CONFREQ [Closed] id 14 len 10
Nov 7 12:22:16.947: PO2/0 IPCP: Address 1.1.1.5 (0x030601010105)
Nov 7 12:22:16.947: PO2/0 CDPCP: O CONFREQ [Closed] id 15 len 4
Nov 7 12:22:16.947: PO2/0 PPP: I pkt type 0x8021, datagramsize 14
Nov 7 12:22:16.951: PO2/0 PPP: I pkt type 0x8207, datagramsize 8
Nov 7 12:22:16.951: PO2/0 IPCP: I CONFREQ [REQsent] id 15 len 10
Nov 7 12:22:16.951: PO2/0 IPCP: Address 1.1.1.6 (0x030601010106)
Nov 7 12:22:16.951: PO2/0 IPCP: O CONFACK [REQsent] id 15 len 10
Nov 7 12:22:16.951: PO2/0 IPCP: Address 1.1.1.6 (0x030601010106)
Nov 7 12:22:16.951: PO2/0 PPP: I pkt type 0x8021, datagramsize 14
Nov 7 12:22:16.951: PO2/0 CDPCP: I CONFREQ [REQsent] id 13 len 4
Nov 7 12:22:16.951: PO2/0 CDPCP: O CONFACK [REQsent] id

```

13 len 4
Nov 7 12:22:16.951: PO2/0 PPP: I pkt type 0x8207,
datagramsize 8
Nov 7 12:22:16.951: PO2/0 IPCP: I CONFACK [ACKsent] id
14 len 10
Nov 7 12:22:16.951: PO2/0 IPCP: Address 1.1.1.5
(0x030601010105)
Nov 7 12:22:16.951: PO2/0 CDPCP: I CONFACK [ACKsent] id
15 len 4
Nov 7 12:22:17.947: %LINEPROTO-5-UPDOWN: Line protocol
on Interface POS2/0,
changed state to up

```

(4)

```

!--- ECHOREQ and ECHOREP packets for PPP keepalives use
packet type !--- values of 0xC021. Nov 7 12:22:17.947:
PO2/0 PPP: O pkt type 0x0207, datagramsize 376 Nov 7
12:22:17.947: PO2/0 PPP: O pkt type 0x0207, datagramsize
376 Nov 7 12:22:17.947: PO2/0 PPP: O pkt type 0x0207,
datagramsize 376 Nov 7 12:22:23.403: PO2/0 LCP: O
ECHOREQ [Open] id 1 len 12 magic 0x269933F4 Nov 7
12:22:23.403: PO2/0 PPP: I pkt type 0xC021, datagramsize
16 Nov 7 12:22:23.403: PO2/0 LCP: I ECHOREP [Open] id 1
len 12 magic 0x4FAE1B0C Nov 7 12:22:23.403: PO2/0 LCP:
Received id 1, sent id 1, line up Nov 7 12:22:25.595:
PO2/0 PPP: I pkt type 0xC021, datagramsize 16

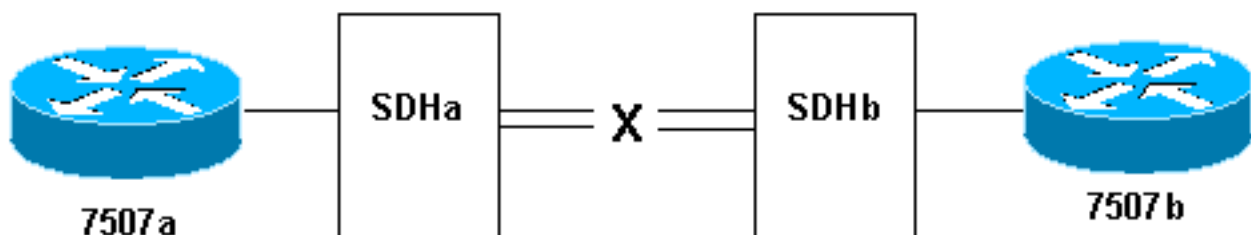
```

문제 해결 참고 사항

PPP 또는 HDLC 캡슐화가 포함된 POS 인터페이스는 링크 장애를 알리는 두 가지 메커니즘을 지원합니다. 레이어 2 keepalive 및 SONET 레이어 경보 keepalive는 고유한 SONET 경보 구조보다 문제를 보고하는 데 더 오랜 시간이 걸립니다. 그러나 레이어 2 keepalives는 라인 카드 CPU에서 라인 카드 CPU로의 경로를 SONET 레벨 알람과 같이 프레임에서 프레임으로 확인하므로 유용합니다. LCP가 즉시 중단되기 때문에 PPP는 상태 변경 링크 작업에 더 신속하게 응답합니다. 반면 HDLC는 keepalive를 시간 초과해야 합니다.

두 라우터 간의 백 투 백 설정에서 파이버 선 중 하나를 가져오면 레이어 1 연결이 끊어지고 두 POS 인터페이스 모두 작동 중지/중지 상태로 변경됩니다. 그러나 두 라우터 POS 인터페이스가 SONET/SDH 장비를 사용하여 Telco 클라우드를 통해 연결되면 레이어 1 손실 정보가 원격 엔드로 전파되지 않습니다. 이 컨피그레이션에서 keepalive는 링크를 다운시키는 메커니즘입니다.

이 설정을 고려하십시오.



다음은 SDHb에서 SDHa로 링크의 전송 파이버 선을 끌어올 때 발생하는 작업입니다.

- 라우터 7507a는 keepalive를 받지 않습니다.

- 라우터 7507b는 수신 파이버가 계속 작동하므로 7507a에서 keepalive를 확인합니다. 디버그 직렬 인터페이스를 사용하여 확인합니다.

또는 이 테스트를 수행할 때 SONET 경보를 표시하는 **show controller pos** 명령을 실행합니다. 라우터 7507a의 경로 경보 신호(P-AIS)와 7507b의 경로 원격 결함 표시(P-RDI)가 표시되어야 합니다.

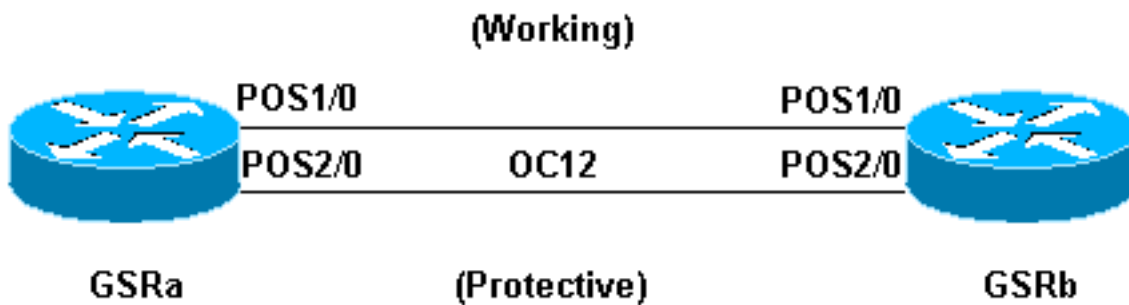
루프백 테스트

show interfaces pos 명령의 출력에서 직렬 회선이 작동하지만 회선 프로토콜이 다운되었음을 나타내는 경우 루프백 테스트를 사용하여 문제의 원인을 확인합니다. 먼저 로컬 루프 테스트를 수행한 다음 원격 테스트를 수행합니다. 지침은 [Cisco 라우터의 루프백 모드 이해](#)를 참조하십시오.

참고: 루프백을 사용할 때 캡슐화를 PPP에서 HDLC로 변경합니다. PPP로 구성된 인터페이스의 회선 프로토콜은 모든 LCP 및 NCP 세션이 성공적으로 협상될 때만 나타납니다.

APS가 있는 회선 프로토콜 상태

APS(Automatic Protection Switching)용으로 구성된 POS 인터페이스는 인터페이스가 작동 채널이 아닌 보호 채널인 경우 회선 프로토콜을 중단시킵니다. 다음의 샘플 토폴로지를 고려하십시오.



이 샘플 로그 출력은 GSRb의 POS 1/0 인터페이스의 파이버 케이블을 제거한 후 캡처되었습니다. APS 전환이 발생할 때 두 인터페이스에서 라인 프로토콜 상태가 변경됩니다. 또한 OSPF(Open Shortest Path First) 인접성 상태의 변경 사항을 확인합니다. (자세한 내용은 [APS 기술 지원 페이지](#)를 참조하십시오.)

```

*Sep 5 17:41:46: %SONET-4-ALARM: POS1/0: SLOS
*Sep 5 17:41:46: %SONET-4-ALARM: POS2/0: APS enabling channel
*Sep 5 17:41:46: %SONET-6-APSREMSWI: POS2/0: Remote APS status now Protect
*Sep 5 17:41:46: %SONET-4-ALARM: POS1/0: APS disabling channel
*Sep 5 17:41:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS2/0,
changed state to up
*Sep 5 17:41:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS1/0,
changed state to down
*Sep 5 17:41:48: %LINK-3-UPDOWN: Interface POS1/0, changed state to down
*Sep 5 17:41:48: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.100.100 on POS1/0
from FULL to DOWN, Neighbor Down: Interface down or detached
*Sep 5 17:41:56: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.100.100 on POS2/0
from LOADING to FULL, Loading Done
  
```

PPP 캡슐화를 사용하여 POS 인터페이스에서 APS를 구성하지 마십시오. PPP는 APS를 인식하지 못합니다. APS 선택 취소로 인해 인터페이스가 작동/중지된 경우 PPP는 인터페이스 재설정을 시도하고 PPP 협상 패킷을 지속적으로 전송합니다.

또한 keepalive를 비활성화하여 불필요한 라인 프로토콜 플랩을 방지합니다. keepalive는 대부분의 POS 라우터 하드웨어에서 자동으로 비활성화됩니다.

APS가 비활성화된 경우 APS 작동 또는 보호 모드의 Cisco 12000 Series POS 인터페이스는 가동/중단 상태(루프백이 있는 경우에도)로 고정될 수 있습니다. 동일한 슬롯에 삽입된 다른 카드에 이 문제가 발생합니다. 적절한 회선 프로토콜 상태를 복원하려면 카드를 새 슬롯으로 이동합니다. 이 문제는 Cisco IOS Software 릴리스 12.0(19)S의 Cisco 버그 ID [CSCdt43759](#)([등록된](#) 고객만 해당)에서 해결되었습니다.

다음 단계를 해결 방법으로 사용합니다.

1. `aps protect` 명령을 구성합니다.
2. `aps force 1` 명령을 실행합니다.
3. `no aps protect` 명령을 구성합니다.

[알려진 문제](#)

POS 인터페이스로 회선 프로토콜 문제를 해결할 때 다음 주의 사항을 참고하십시오.

- 캡슐화가 PPP에서 HDLC로 변경된 후 PA-POS 인터페이스가 지속적으로 재설정될 수 있습니다. 이 문제는 Cisco 버그 ID CSCdk [30893](#)의 PA-POS에 대해 보고되며([등록된](#) 고객만 해당) Cisco 버그 ID [CSCdk1877](#)([등록된](#) 고객만 해당) 및 Cisco 버그 ID [CSC11187757](#)의 [다양한 인터페이스만 지원 PPP 및 HDLC 캡슐화](#). 캡슐화가 변경될 때 PPP가 완전히 종료되지 않은 경우 문제가 발생합니다.
- HDLC 캡슐화 및 keepalive로 구성된 POS 인터페이스는 원격 엔드로부터 keepalive를 받지 못한 경우 회선 프로토콜을 종료하지 않고 반복 인터페이스 플랩을 거칩니다. 이 문제는 Cisco 버그 ID [CSCdp86387](#)에서 [해결됩니다](#)([등록된](#) 고객만 해당).

[관련 정보](#)

- [옵티컬 기술 지원](#)
- [기술 지원 및 문서 - Cisco Systems](#)