음성 연결 트렁크 문제 해결

목차

소개

음성 연결 트렁크는 음성 통화(VoIP(Voice over IP), VoFR(Voice over Frame Relay) 또는 VoATM(Voice over ATM))를 영구적으로 설정합니다. 라우터가 켜지고 컨피그레이션이 완료되는 즉시 통화가 설정됩니다.음성 포트가 켜지는 즉시 음성 포트는 음성 포트 아래에 지정된 더미 전화 번호로 자동으로 전화를 걸어 해당 위치에 전화를 겁니다.음성 포트는 해당 다이얼 피어를 통해 다른 쪽 끝으로의 통화를 완료합니다.이 연결이 설정되면 라우터에 관한 한 음성 통화가 세션에 있고 연결됩니다.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다.이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다.l 네트워크가 작동 중인 경우, 명령을 사용하기 전에 모든 명령의 잠재적인 영향을 이해해야 합니다.

표기 규칙

문서 규칙에 대한 자세한 내용은 Cisco 기술 팁 표기 규칙을 참조하십시오.

문제

트렁크와 관련된 일반적인 문제는 라우터에 투명하며 트러블슈팅이 매우 어렵습니다.음성 트렁크에서 나타나는 일반적인 문제는 트렁크에 통화가 연결되고 아무것도 들리지 않을 때 나타납니다.이는 연결 트렁크에 대해 알려진 문제 중 하나이며 여러 가지 문제로 인해 발생합니다.또 다른 문제는 올바르게 전달되지 않은 DTMF(Dual Tone Multifrequency) 신호음과 PBX(Private Branch Exchange)에서 PBX로의 신호 처리가 제대로 전송되지 않는다는 것입니다.이 문서에서는 이러한 문제를 해결합니다.

음성 트럭이 작동 중이고 활성화되면 연결 트럭에서 신호가 다르게 작동합니다.신호 특성을 위해 음성 포트에서 일반적으로 실행하는 명령은 관련이 없으며 도움이 되지 않습니다.음성 트렁크는 시그널링 통로가 되며 VoIP 링크를 통해 신호를 릴레이합니다.음성 트렁크를 사용할 때 PBX 신호 처리가 엔드 투 엔드 일치해야 합니다.두 개의 PBX 시스템에 관한한, 음성 트렁크 연결이 라우터를 완전히 투명하게 사용하는 PBX에 임대 T1 회선과 동일해 보이게 하고 전체 프로세스에서 두 PBX 간에 명확한 링크가 설정되도록 하는 것이 목표입니다.

트렁크가 나타나면 트렁크는 소프트웨어 케이블이 되고 신호 유형은 커넥터 유형으로 간주됩니다.트렁크는 사용되는 신호 유형에 대해 중요하지 않습니다.신호가 양쪽 끝에서 일치하지 않아도 트렁크가 계속 나타납니다.양쪽 끝에 있는 PBX가 동일한 신호 처리를 하는 한 트렁크는 제대로 작동합니다.

솔루션

연결 트렁크 문제를 해결할 때 사용하는 접근 방식은 스위치드 통화에 사용되는 접근 방식과 다릅니다.트렁크가 확인된 후 실제로 어떤 일이 발생하는지 확인하려면 PBX 신호 처리를 확인해야 합니다.신호를 확인하기 전에 트렁크가 작동 중인지, DSP(Digital Signal Processor)가 음성 패킷을 처리하는지 확인하십시오.

참고: 문제 해결을 위해 VAD(Voice Activity Detection)를 비활성화해야 할 수도 있습니다.트렁크가 올바르게 작동하는지 확인한 후 더 많은 문제를 해결하려면 텔레포니 신호 처리를 확인해야 합니다.

트렁크가 설정되고 통화를 시도하지 않는 경우 트렁크 keepalive 메시지가 원격 박스 간에 송수신됩니다.이러한 keepalives는 트렁크 연결을 확인하고 엔드 투 엔드 신호 정보를 전달합니다.이러한 keepalive를 확인하려면 debug vpm signal 명령을 실행합니다.트렁크가 많은 경우 debug vpm 명령의 출력은 debug vpm **port *x*** 명령 옵션을 실행하면 출력을 단일 포트로 제한할 수 있습니다. 여기서 "x"는 문제의 음성 포트입니다.다음은 모든 포트를 볼 때 실행된 **debug vpm signal** 명령의 출력입니다.

```
21:18:12: [3/0:10(11)] send to dsp sig DCBA state 0x0
21:18:12: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:12(13)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:20(21)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:12(13)] send to dsp SIG DCBA state 0x0
21:18:12: [3/0:20(21)] send to dsp SIG DCBA state 0x0
21:18:12: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:18:12: [3/0:3(4)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:9(10)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:3(4)] send to dsp SIG DCBA state 0x0
21:18:13: [3/0:9(10)] send to dsp SIG DCBA state 0x0
21:18:13: [3/0:19(20)] rcv from dsp SIG DCBA state 0x0
```

debug vpm port *x* 명령을 사용하여 이를 제한하면 다음 예와 같이 디버그가 해석하기가 훨씬 쉬워

집니다.

```
21:21:08: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:12: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:13: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:17: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:18: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:22: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:23: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:27: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:28: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:32: [3/0:0(1)] send to dsp SIG DCBA state 0x0
```

keepalive는 5초마다 전송되고 수신됩니다."sent to dsp" 및 "received from dsp"라는 용어는 Cisco IOS® 관점에서 제공됩니다.DSP를 PBX로 대체하여 더 이해할 수 있도록 합니다.트렁크에 활동이 없는 동안 표시되는 메시지입니다.keepalive 메시지를 통해 회로의 각 끝에 있는 라우터가 트렁크가 아직 작동 중임을 알 수 있습니다.이 메시지 중 5개가 한 줄로 누락되면 트렁크가 다운됩니다.원인 중 하나는 트렁크가 네트워크에서 지속적으로 플랩하는 경우입니다.음성 트렁크 keepalive가 전송 및 수신되는지 확인하려면 debug vpm trunk**-sc** 명령**을** 실행합니다.이 디버그는 트렁크 keepalive가 누락될 때까지 출력을 생성하지 않습니다.keepalive가 누락되었을 때 **debug vpm trunk-sc** 명령 출력의 예입니다.

```
22:22:38: 3/0:22(23): lost Keepalive
22:22:38: 3/0:22(23): TRUNK_SC state : TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE
22:22:38: 3/0:22(23): trunk_rtc_set_AIS on
22:22:38: 3/0:22(23): trunk_rtc_gen_pattern : SIG pattern 0x0
22:22:38: 3/0:22(23): TRUNK_SC, TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
22:22:39: 3/0:13(14): lost Keepalive
22:22:39: 3/0:13(14): TRUNK_SC state : TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE
22:22:39: 3/0:13(14): trunk_rtc_set_AIS on
22:22:39: 3/0:13(14): trunk_rtc_gen_pattern : SIG pattern 0x0
22:22:39: 3/0:13(14): TRUNK_SC, TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
```

debug vpm trunk-sc 명령을 **실행할** 때 출력이 표시되지 않으면 keepalive가 누락되지 않습니다 .keepalive가 누락되어도 5개의 순차적 메시지가 누락될 때까지 트렁크가 켜져 있습니다.즉, 트렁크가 중단되기 전에 25초 동안 연결이 중단되어야 합니다.

연결 트렁크의 일반적인 문제

음성 트렁크 연결과 관련된 몇 가지 버그가 있습니다.특이한 것이 발견되면 이 버그를 확인하세요 .Cisco IOS Software 12.2가 출시되었을 때 이러한 문제의 대부분은 해결되고 통합되었습니다.버그를 통해 이러한 원인이 이전 코드 문제의 원인임을 인식할 수 있습니다.가장 일반적인 문제 중 하나는 PBX가 트렁크 연결을 통해 올바르게 신호를 보내는 것입니다.트렁크를 내려놓고 라우터가 각 끝에서 작동하도록 구성하는 것이 좋은 생각인 것 같습니다. 그러나 이제 변경된 모든 항목이 트렁크가 설정되면 이동하므로 이 방식은 역효과가 없습니다.가장 좋은 트러블슈팅 방법은 트렁크를 켜고 작동하는 것입니다.

문제 해결 시작

이러한 기능이 올바르게 작동하려면 기본 사항을 확인해야 합니다.

- 트렁크가 설정되었습니까?show **voice call summary** 명령을 실행하고 트렁크가 S_CONNECTED 상태에 있는지 확인합니다.
- DSP가 패킷을 처리합니까?이를 확인하려면 **show voice dsp** 명령을 실행합니다.DSP에서 처리되는 패킷이 표시되지 않는 경우 VAD가 활성화되고 패킷이 억제되기 때문입니다.VAD를 끄

고 트렁크를 다시 설정하고 다시 확인합니다.또한 **show call active voice brief** 명령이 실행될 때 패킷 카운터가 증가하는지 확인합니다.이 명령은 또한 문제의 통화 로그에 대해 VAD가 활성화되었는지 여부를 표시합니다.

트렁크가 어느 사이트에서나 아날로그 포트에 연결되는 경우 트렁크 모드가 아닌 모드에서 PBX의 작동을 확인하는 것이 가장 좋습니다.아날로그 E&M 연결 문제를 해결하려면 아날로그 E&M 인터페이스 유형 및 배선 배열 이해 및 문제 해결을 참조하십시오.모든 것이 확인되고 제대로 작동하면 트렁크를 들고 PBX 간에 전달되는 신호를 살펴봅니다.

음성 트렁크 연결 문제를 해결하는 가장 좋은 방법은 PBX 간에 전달되는 신호를 조사하는 것입니다.한 쪽에서 다른 쪽으로 신호가 전달될 때 신호를 관찰할 수 있도록 문제가 있는 각 라우터에 대한 텔넷 세션을 갖는 것이 좋습니다.이 문서에서는 E&M 윙크 시그널링을 사용합니다. 이는 매우 인기 있고 윙크 타이밍을 고려해야 하기 때문입니다.

PBX에 연결된 라우터의 출력으로 통화를 시작합니다.

```
May 22 19:39:03.582: [3/0:0(1)] rcv from dsp sig DCBA state 0x0
!--- It is in idle state. May 22 19:39:07.774: [3/0:0(1)] send to dsp SIG DCBA state 0x0 !---
ABCD bits=0000. May 22 19:39:08.586: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
19:39:12.778: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:13.586: [3/0:0(1)] rcv from
dsp SIG DCBA state 0x0 May 22 19:39:17.777: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22
19:39:18.593: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:22.781: [3/0:0(1)] send to
dsp SIG DCBA state 0x0 May 22 19:39:23.593: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
19:39:27.781: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:28.597: [3/0:0(1)] rcv from
dsp SIG DCBA state 0x0 May 22 19:39:32.785: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22
19:39:33.597: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:37.789: [3/0:0(1)] send to
dsp SIG DCBA state 0x0 May 22 19:39:38.601: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
19:39:39.777: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:39.797: [3/0:0(1)] rcv
from dsp SIG DCBA state 0x0 May 22 19:39:39.817: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF !---
Receives off-hook from PBX, and passes to remote end. May 22 19:39:39.837: [3/0:0(1)] rcv from
dsp SIG DCBA state 0xF May 22 19:39:39.857: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22
19:39:39.877: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:39.897: [3/0:0(1)] rcv
from dsp SIG DCBA state 0xF May 22 19:39:39.917: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May
22 19:39:39.937: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:39.957: [3/0:0(1)] rcv
from dsp SIG DCBA state 0xF May 22 19:39:39.977: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May
22 19:39:39.997: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.017: [3/0:0(1)] rcv
from dsp SIG DCBA state 0xF May 22 19:39:40.037: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May
22 19:39:40.057: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.077: [3/0:0(1)] rcv
from dsp SIG DCBA state 0xF May 22 19:39:40.089: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May
22 19:39:40.097: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.109: [3/0:0(1)] send
to dsp SIG DCBA state 0x0 May 22 19:39:40.117: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF !---
Receiving wink from remote side, and passes to PBX. May 22 19:39:40.129: [3/0:0(1)] send to dsp
SIG DCBA state 0xF May 22 19:39:40.137: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22
19:39:40.149: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.157: [3/0:0(1)] rcv from
dsp SIG DCBA state 0xF May 22 19:39:40.169: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22
19:39:40.177: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.189: [3/0:0(1)] send to
dsp SIG DCBA state 0xF May 22 19:39:40.197: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22
19:39:40.213: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.217: [3/0:0(1)] rcv from
dsp SIG DCBA state 0xF May 22 19:39:40.229: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22
19:39:40.237: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.249: [3/0:0(1)] send to
dsp SIG DCBA state 0xF May 22 19:39:40.257: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22
19:39:40.269: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.289: [3/0:0(1)] send to
dsp SIG DCBA state 0xF May 22 19:39:40.309: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22
19:39:40.329: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.349: [3/0:0(1)] send to
dsp SIG DCBA state 0x0 !--- Wink ended from remote side, and passes to PBX. May 22 19:39:40.369:
[3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.389: [3/0:0(1)] send to dsp SIG DCBA
state 0x0 May 22 19:39:40.409: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.429:
[3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.449: [3/0:0(1)] send to dsp SIG DCBA
state 0x0 May 22 19:39:40.469: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.493:
[3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.509: [3/0:0(1)] send to dsp SIG DCBA
```

state 0x0 May 22 19:39:40.529: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.549: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.569: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.589: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.613: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.629: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.649: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.669: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.689: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.709: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.729: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.749: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.769: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:45.773: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:50.081: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:50.101: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:50.121: [3/0:0(1)] send to dsp SIG DCBA state 0xF *!--- Wink ends, the remote end is now off-hook, the conversation happens.* May 22 19:39:50.141: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.161: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.181: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.197: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.221: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.241: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.261: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.261: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.281: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.301: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.321: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.341: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.361: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.381: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.401: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.421: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.441: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.461: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.481: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.501: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.521: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.541: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.561: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:55.265: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:55.561: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:00.269: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:00.565: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:05.268: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:05.564: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:10.272: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:10.568: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:15.276: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:15.572: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.676: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.696: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.716: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.736: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.756: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.776: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.796: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.796: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:19.816: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.816: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:19.836: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.836: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 *!--- Both side hung up, back to idle state.* May 22 19:40:19.856: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.856: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.876: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.876: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.896: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.896: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.916: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.916: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.936: [3/0:0(1)] send to dsp SIG DCBA state 0x0

이 출력은 라우터가 통화를 종료하는 것을 보여줍니다.NTP(Network Time Protocol)가 동기화됩니다.

May 22 19:39:03.582: [3/0:0(1)] send to dsp SIG DCBA state 0x0
May 22 19:39:07.774: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
!--- Idle state, both side on-hook. May 22 19:39:08.586: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:12.774: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:13.586: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:15.383: [1/0:0(1)] Signaling RTP packet has no particle *!--- You will see this message if you are running Cisco IOS !--- Software Release 12.2(1a) or later. It is not an error !--- message, it is a normal functioning state.* May 22 19:39:17.774: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:18.590: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:22.778: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:23.594: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:27.782: [3/0:0(1)]

rcv from dsp SIG DCBA state 0x0 May 22 19:39:28.598: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:32.782: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:33.598: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:37.786: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:38.602: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:39.778: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:39.798: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:39.818: [3/0:0(1)] send to dsp SIG DCBA state 0xF *!--- Remote side off-hook, this is conveyed to the PBX.* May 22 19:39:39.838: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.858: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.878: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.898: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.918: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.938: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.958: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.978: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:39.998: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.018: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.038: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.058: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.078: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.090: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.098: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.110: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.118: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.130: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF *!--- Receive wink from PBX.* May 22 19:39:40.138: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.150: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.158: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.170: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.178: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.190: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.198: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.210: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.218: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.230: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.238: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.250: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.258: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.270: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.290: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.310: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.330: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.350: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 *!--- Wink ended, waiting for an answer.* May 22 19:39:40.370: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.390: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.410: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.430: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.450: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.470: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.490: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.510: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.530: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.550: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.570: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.590: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.610: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.630: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.650: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.670: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.690: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.710: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.730: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.750: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:40.770: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:45.262: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:45.770: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:50.077: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:50.097: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:50.117: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF *!--- Receive off-hook from PBX.* May 22 19:39:50.137: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.157: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.177: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.197: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.217: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.237: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.257: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.261: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.277: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.297: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.317: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.337: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.357: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.377: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.397: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.417: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.437: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.457: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.477: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.497: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.517: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.537: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.557: [3/0:0(1)] rcv from dsp SIG DCBA

```
state 0xF !--- Both sides off-hook, the conversation happens. May 22 19:39:55.265: [3/0:0(1)]
send to dsp SIG DCBA state 0xF May 22 19:39:55.557: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF
May 22 19:40:00.269: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:00.561: [3/0:0(1)]
rcv from dsp SIG DCBA state 0xF May 22 19:40:05.269: [3/0:0(1)] send to dsp SIG DCBA state 0xF
May 22 19:40:05.561: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:10.273: [3/0:0(1)]
send to dsp SIG DCBA state 0xF May 22 19:40:10.565: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF
May 22 19:40:15.273: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:15.569: [3/0:0(1)]
rcv from dsp SIG DCBA state 0xF May 22 19:40:19.673: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF
May 22 19:40:19.693: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:19.713: [3/0:0(1)]
rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.733: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
May 22 19:40:19.753: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.773: [3/0:0(1)]
rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.793: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
May 22 19:40:19.797: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.813: [3/0:0(1)]
rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.817: [3/0:0(1)] send to dsp SIG DCBA state 0xF
May 22 19:40:19.833: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.837: [3/0:0(1)]
send to dsp SIG DCBA state 0x0 !--- Both sides are back on-hook, back to idle. May 22
19:40:19.853: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:40:19.857: [3/0:0(1)] send to
dsp SIG DCBA state 0x0 May 22 19:40:19.873: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
19:40:19.877: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.893: [3/0:0(1)] rcv from
dsp SIG DCBA state 0x0 May 22 19:40:19.897: [3/0:0(1)] send to dsp SIG DCBA state 0x0
```

참고: 이 출력은 E&M 윙크 신호를 사용하는 음성 트렁크의 양쪽에서 발생하는 신호를 보여줍니다 .이와 같은 디버그를 사용하는 다른 유형의 신호 처리를 확인할 수 있습니다.통화가 올바르게 설정된 경우(여기에 표시됨) 양방향 오디오가 있어야 합니다.이는 **show voice dsp** 또는 **show call active voice brief** 명령 출력을 보면 확인할 수 있습니다.모든 것이 정상적으로 작동하는 것처럼 보이지만 아날로그 연결을 통해 오디오 문제(오디오 또는 단방향 없음)가 발생하는 경우 이러한 연결을 다시 확인하십시오.

통화 진행 상황 확인

트렁크 통화에 대해 **show call active voice** 또는 **show voice call summary** 명령 출력을 보는 것이 별로 좋지 않거나 좋지 않기 때문에 활성 통화를 지원하는 음성 트렁크를 확인하는 간단한 방법이 필요합니다.이렇게 하는 가장 쉬운 방법 중 하나는 **show voice trunk-conditioning signaling** 명령을 **include** 매개 변수와 함께 **실행하고 *ABCD*를 포함된 문자열로 *사용하는* 것입니다.**

```
Phoenix#show voice trunk-conditioning signaling | include ABCD
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=1111, last-RX-ABCD=0000
!--- Timeslot 8. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=1111, last-RX-ABCD=1111 !---
Timeslot 10. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-
ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000
```

참고: 이 출력은 타임 슬롯 10에서 활성화된 통화와 타임 슬롯 8에서 시작된 다른 통화를 보여줍니다.이 명령을 많이 사용하면 이 긴 명령에 대한 별칭을 만들 수 있습니다.

DTMF 문제 해결

오프 후크 및 온후크 신호 외에도 라우터가 PBX(음성 외) 간에 전달하는 다른 유일한 것은 DTMF 신호음입니다.오디오 경로도 있으므로 일반적으로 문제가 아니지만 문제가 있습니다.문제는 해당 경로를 통해 오디오를 수행하는 방식에서 발생합니다.대역폭을 절약하기 위해 낮은 비트 속도 코덱을 사용하는 것이 좋습니다.문제는 이러한 낮은 비트 속도 코덱이 인간의 말을 위해 작성된 알고리

즘에 의해 설계되었다는 것입니다.DTMF 신호음은 이러한 알고리즘에 잘 맞지 않으며 고객이 g711 코덱을 사용하지 않는 한 전달할 다른 방법이 필요합니다.답은 dtmf-**relay** 명령에 있습니다.이 기능을 사용하면 DSP가 마지막에 신호음을 시작하고 DTMF 신호음을 인식하여 일반 오디오 스트림과 분리할 수 있습니다.그런 다음 DSP는 구성 방식에 따라 이 신호음을 다른 유형의 RTP(Real Time Protocol) 패킷 또는 오디오 스트림과 별도로 링크를 통해 전송되는 h245 메시지로 코딩합니다.이는 **fax-relay** 및 **modem-relay** 명령의 배후에 있는 동일한 프로세스입니다.

이 기능은 트렁크 문제 해결을 위해 또 다른 디버그 문제를 발생시킵니다.통화 설정이 없고 라우터 간 패킷 스트림에서 해당 정보를 추출해야 하는 경우 어떤 번호가 전달되는지 어떻게 확인합니까?이 작업을 수행하는 방법은 어떤 유형의 dtmf **릴레이** 명령이 사용되는지에 따라 달라집니다.

이 예에서와 같이 dtmf-relay cisco-rtp 명령은 전용 Cisco 페이로드 유형을 사용하므로 이를 보려면 DSP를 아래를 살펴보아야 합니다.debug vpm **signal** 명령**을 debug vpm port x/x:y.z** 명령과 함께 실행할 수 있습니다(문제의 포트로 출력을 제한하기 위해). 이 명령은 원래 쪽의 DSP에 전달된 숫자를 볼 수 있습니다.이 출력은 종료 쪽이 아니라 원래 쪽에 표시됩니다.

```
*Mar  1 00:22:39.592: htsp_digit_ready: digit = 31
*Mar  1 00:22:39.592: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:40.021: htsp_digit_ready: digit = 32
*Mar  1 00:22:40.021: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:40.562: htsp_digit_ready: digit = 33
*Mar  1 00:22:40.562: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:40.810: [1/0:1(2)] rcv from dsp SIG DCBA state 0xF
*Mar  1 00:22:41.131: htsp_digit_ready: digit = 34
*Mar  1 00:22:41.131: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:41.499:  [1/0:1(2)] Signaling RTP packet has no partical
*Mar  1 00:22:41.499: [1/0:1(2)] send to dsp SIG DCBA state 0xF
*Mar  1 00:22:41.672: htsp_digit_ready: digit = 35
*Mar  1 00:22:41.672: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:42.192: htsp_digit_ready: digit = 36
*Mar  1 00:22:42.192: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:42.789: htsp_digit_ready: digit = 37
*Mar  1 00:22:42.789: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:43.350: htsp_digit_ready: digit = 38
*Mar  1 00:22:43.350: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:44.079: htsp_digit_ready: digit = 39
*Mar  1 00:22:44.079: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:45.249: htsp_digit_ready: digit = 30
*Mar  1 00:22:45.249: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:45.810: [1/0:1(2)] rcv from dsp SIG DCBA state 0xF
*Mar  1 00:22:46.007: htsp_digit_ready: digit = 2A
*Mar  1 00:22:46.011: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:46.572: [1/0:1(2)] Signaling RTP packet has no partical
*Mar  1 00:22:46.572: [1/0:1(2)] send to dsp SIG DCBA state 0xF
*Mar  1 00:22:46.628: htsp_digit_ready: digit = 23
*Mar  1 00:22:46.628: [1/0:1(2), S_TRUNKED, E_VTSP_DIGIT]
*Mar  1 00:22:50.815: [1/0:1(2)] rcv from dsp SIG DCBA state 0xF
all digits 0-9 are represented by 30-39, * = 2A and # = 23.
```

dtmf-relay h245-영숫자 명령을 사용하여 원래 측에서 어떤 숫자가 전송되는지 확인할 수 있습니다.dtmf-**relay h245-영숫자** 명령은 h.245의 영숫자 부분을 사용하여 신호음을 전달합니다.이 예제에서 볼 수 있듯이 debug h245 asn1 명령이 활성화되면 트렁크의 시작 면과 종료 면 모두에서 숫자를 쉽게 볼 수 있습니다.

원래 측면:

```
*Mar  1 00:34:17.749: H245 MSC OUTGOING PDU ::=
```

```
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "1"

*Mar  1 00:34:17.749: H245 MSC OUTGOING ENCODE BUFFER::= 6D 400131
*Mar  1 00:34:17.753:
*Mar  1 00:34:18.350: H245 MSC OUTGOING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "2"

*Mar  1 00:34:18.350: H245 MSC OUTGOING ENCODE BUFFER::= 6D 400132
*Mar  1 00:34:18.350:
*Mar  1 00:34:18.838: H245 MSC OUTGOING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "3"

*Mar  1 00:34:18.838: H245 MSC OUTGOING ENCODE BUFFER::= 6D 400133
```

종료 면:

```
*Mar  1 17:45:16.424: H245 MSC INCOMING ENCODE BUFFER::= 6D 400131
*Mar  1 17:45:16.424:
*Mar  1 17:45:16.424: H245 MSC INCOMING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "1"

*Mar  1 17:45:17.025: H245 MSC INCOMING ENCODE BUFFER::= 6D 400132
*Mar  1 17:45:17.025:
*Mar  1 17:45:17.025: H245 MSC INCOMING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "2"

*Mar  1 17:45:17.514: H245 MSC INCOMING ENCODE BUFFER::= 6D 400133
*Mar  1 17:45:17.514:
*Mar  1 17:45:17.514: H245 MSC INCOMING PDU ::=
value MultimediaSystemControlMessage ::= indication : userInput : alphanumeric : "3"
```

dtmf-relay h h245-signal 명령은 매우 유사하며 dtmf-relay h h245-영숫자 명령과 동일한 디버그를 **사용할** 때 **볼 수 있습니다**.전체적으로 dtmf-relay 명령으로 연결 트렁크를 트러블슈팅하는 것은 디버그가 언급되지 않은 상태에서 다소 어렵습니다.

관련 정보

- 투명 CCS 구성 및 문제 해결
- 음성 기술 지원
- 음성 및 IP 커뮤니케이션 제품 지원
- Cisco IP 텔레포니 문제 해결
- Technical Support - Cisco Systems