ケース スタディ：Cisco IP Phone と Cisco IOS Gateway 間のコールのトラブルシューティング

付録 B「ケース スタディ：Cisco IP Phone コールのトラブルシューティング」のケース スタディでは、クラスタ内コールのコール フローについて説明しました。この付録のケース スタディでは、ローカル PBX または Public Switched Telephone Network（PSTN; 公衆電話交換網）に接続された電話機に Cisco IOS Gateway を介してコールを発信する Cisco IP Phone について説明します。概念的には、コールが Cisco IOS Gateway に到達すると、ゲートウェイはそのコールを FXS ポートまたは PBX に接続された電話機のどちらかに転送します。コールが PBX に転送された場合、そのコールはローカル PBX に接続された電話機で終端するか、PBX によって PSTN に転送されて PSTN 上のどこかで終端します。

この章では、次のトピックについて取り上げます。

- 「コール フロー トレース」
- 「Cisco IOS Gatekeeper のデバッグ メッセージと表示コマンド」
- 「Cisco IOS Gateway のデバッグ メッセージと表示コマンド」
- 「T1/PRI インターフェイスを使用する Cisco IOS Gateway」
- 「T1/CAS インターフェイスを使用する Cisco IOS Gateway」

コール フロー トレース

この項では、Cisco CallManager トレース ファイル CCM000000000 の例を使用して、コール フローについて説明します。付録 B「ケース スタディ：Cisco IP Phone コールのトラブルシューティング」で詳細なトレース情報（初期化、登録、KeepAlive のメカニズムなど）についてはすでに説明したので、このケース スタディのトレースでは、コール フロー自体に焦点を絞っています。

このコール フローでは、Cluster 2 に配置された Cisco IP Phone（電話番号 1001）が、PSTN に配置された電話機（電話番号 3333）にコールを発信しています。TCP ハンドル値、タイム スタンプ、またはデバイスの名前を調べることで、デバイスをトレース上で追跡できます。デバイスをリブートするかオフラインにするまで、デバイスの TCP ハンドル値は変わりません。

次のトレースでは、Cisco IP Phone（1001）はオフフックになっています。このトレースは、一意のメッセージ、TCP ハンドル、および発信側の番号を示しています。これらは Cisco IP Phone に表示されます。この時点では、まだユーザが番号をダイヤルしていないので、着信側の番号は表示されていません。

```
16:05:46.37515:20:18.390 CCM|StationInit - InboundStim - OffHookMessageID
tcpHandle=0x5138d98

15:20:18.390 CCM|StationD - stationOutputDisplayText tcpHandle=0x5138d98, Display=1001
```

次のトレースでは、ユーザが DN 3333 をダイヤルしています（数字を 1 つずつダイヤルしています）。3333 という番号は電話機の宛先番号であり、この電話機は PSTN ネットワークに配置されています。Cisco CallManager の番号分析プロセスは現在アクティブになっていて、コールのルーティング先を検出するために番号を分析しています。番号分析については、付録 B「ケース スタディ：Cisco IP Phone コールのトラブルシューティング」で詳細に説明しています。

```
15:20:18.390 CCM|Digit analysis: match(fqcn="", cn="1001", pss="", dd="")
15:20:19.703 CCM|Digit analysis: match(fqcn="", cn="1001", pss="", dd="3")
15:20:20.078 CCM|Digit analysis: match(fqcn="", cn="1001", pss="", dd="33")
15:20:20.718 CCM|Digit analysis: match(fqcn="", cn="1001", pss="", dd="333")
15:20:21.421 CCM|Digit analysis: match(fqcn="", cn="1001", pss="", dd="3333")
15:20:21.421 CCM|Digit analysis: analysis results
```

次のトレースでは、番号分析が完了して発信側と着信側が一致し、情報の解析が完了しています。

```
|CallingPartyNumber=1001
|DialingPattern=3333
|DialingRoutePatternRegularExpression=(3333)
|PretransformDigitString=3333
|PretransformPositionalMatchList=3333
|CollectedDigits=3333
|PositionalMatchList=3333
```

次のトレースでは、番号 0 は発信元のロケーションを示し、番号 1 は宛先のロケーションを示しています。BW = -1 によって発信元のロケーションの帯域幅が決定されています。値 -1 は、帯域幅が無限であることを意味します。帯域幅が無限であるのは、LAN 環境に配置された Cisco IP Phone からコールが発信されたためです。BW = 64 によって宛先のロケーションの帯域幅が決定されています。コールの宛先には PSTN に配置された電話機が指定されていて、使用されるコーデック タイプは G.711（64 Kbps）です。

```
15:20:21.421 CCM|Locations:Orig=0 BW=-1 Dest=1 BW=64 (-1 implies infinite bw
available)
```

次のトレースは、発信側と着信側の情報を示しています。この例では、管理者が John Smith などの表示名を設定していないので、発信側の名前と番号は同じです。

```
15:20:21.421 CCM|StationD - stationOutputCallInfo CallingPartyName=1001,
CallingParty=1001, CalledPartyName=, CalledParty=3333, tcpHandle=0x5138d98
```

次のトレースは、H.323 コードが初期化されて H.225 セットアップ メッセージを送信していることを示しています。従来の HDLC SAPI メッセージ、着信側の 16 進表記の IP アドレス、およびポート番号も確認できます。

```
15:20:21.421 CCM|Out Message -- H225SetupMsg -- Protocol= H225Protocol
15:20:21.421 CCM|MMan_Id= 1. (iep=  0 dsl=  0 sapi=  0 ces=  0 IpAddr=e24610ac
IpPort=47110)
```

次のトレースは、発信側と着信側の情報および H.225 アラート メッセージを示しています。また、Cisco IP Phone の 16 進数値と IP アドレスのマッピングも示しています。Cisco IP Phone（1001）の IP アドレスは 172.16.70.231 です。

```
15:20:21.437 CCM|StationD - stationOutputCallInfo CallingPartyName=1001,
CallingParty=1001, CalledPartyName=, CalledParty=3333, tcpHandle=0x5138d98
15:20:21.453 CCM|In  Message -- H225AlertMsg -- Protocol= H225Protocol
15:20:21.953 CCM|StationD - stationOutputOpenReceiveChannel tcpHandle=0x5138d98 myIP:
e74610ac (172.16.70.231)
```

次のトレースは、このコールに使用される圧縮タイプ（G.711 mu-law）を示しています。

```
15:20:21.953 CCM|StationD - ConferenceID: 0 msecPacketSize: 20
compressionType:(4)Media_Payload_G711Ulaw64k
```

H.225 アラート メッセージが送信された後、H.323 は H.245 を初期化します。次のトレースは、発信側と着信側の情報および H.245 メッセージを示しています。TCP ハンドル値はこれまでと変わらず、同一コールが継続していることを示しています。

```
ONE FOR EACH Channel- 16:53:36.855 CCM|H245Interface(3) paths established ip =
e98e6b80, port = 1304|<CT::1,100,105,1.1682><IP::128.107.142.233>
ONE FOR EACH Channel- 16:53:37.199 CCM|H245Interface(3) OLC outgoing confirm ip =
b870701, port = 49252|<CT::1,100,128,3.9><IP::1.7.135.11>

H323 EP has answered the call and H245 channel setup in progress:
16:53:13.479 CCM|In  Message -- H225ConnectMsg -- Protocol= H225Protocol|

16:03:25.359 CCM|StationD(1):  TCPPid = [1.100.117.1] CallInfo callingPartyName=''
callingParty=13001 cgpnVoiceMailbox=    calledPartyName='' calledParty=11002
cdpnVoiceMailbox=    originalCalledPartyName='' originalCalledParty=11002
originalCdpnVoiceMailbox= originalCdpnRedirectReason=0    lastRedirectingPartyName=''
lastRedirectingParty=11002 lastRedirectingVoiceMailbox= lastRedirectingReason=0
callType=2(OutBound) lineInstance=1 callReference=16777217. version:
0|<CT::1,100,11,2.1><IP::><DEV::>

16:03:25.328 CCM|StationD(1):   TCPPid = [1.100.117.1] OpenReceiveChannel
conferenceID=16777217 passThruPartyID=16777233 millisecondPacketSize=20
compressionType=4(Media_Payload_G711Ulaw64k) qualifierIn=?. myIP: e98e6b80
(128.107.142.233)|<CT::1,100,11,1.1><IP::><DEV::>
16:03:25.359 CCM|StationD(2):   TCPPid = [1.100.117.2] StartMediaTransmission
conferenceID=16777218 passThruPartyID=16777249 remoteIpAddress=e98e6b80(64.255.0.0)
remotePortNumber=65344 milliSecondPacketSize=20
compressType=4(Media_Payload_G711Ulaw64k) qualifierOut=?. myIP: e98e6b80
(128.107.142.233)|<CT::1,100,105,1.213><IP::128.107.142.233>
16:03:25.375 CCM|StationD(2):   TCPPid = [1.100.117.2]
star_StationOutputStartMultiMediaTransmission conferenceID=16777218
passThruPartyID=16777250 remoteIpAddress=e98e6b80(66.255.0.0)
remotePortNumber=65346 compressType=101(Media_Payload_H263) qualifierOut=?. myIP:
e98e6b80 (128.107.142.233)|<CT::1,100,105,1.215><IP::128.107.142.233>
16:03:25.328 CCM|StationD(1):   TCPPid=[1.100.117.1] OpenMultiReceiveChannel
conferenceID=16777217 passThruPartyID=1000011 compressionType=101(Media_Payload_H263)
qualifierIn=?.   myIP: e98e6b80 (128.107.142.233)|<CT::1,100,11,1.1><IP::><DEV::>
```

次のトレースは、H.225 接続メッセージおよびその他の情報を示しています。H.225 接続メッセージが受信されると、コールが接続されます。

```
15:20:22.968 CCM|In  Message -- H225ConnectMsg -- Protocol= H225Protocol
15:20:22.968 CCM|StationD - stationOutputCallInfo CallingPartyName=1001,
CallingParty=1001, CalledPartyName=, CalledParty=3333, tcpHandle=0x5138d98
15:20:22.062 CCM|MediaCoordinator - wait_AuConnectInfoInd
15:20:22.062 CCM|StationD - stationOutputStartMediaTransmission tcpHandle=0x5138d98
myIP: e74610ac (172.16.70.231)
15:20:22.062 CCM|StationD - RemoteIpAddr: e24610ac (172.16.70.226)
RemoteRtpPortNumber: 16758 msecPacketSize: 20
compressionType:(4)Media_Payload_G711Ulaw64k
15:20:22.062 CCM|Locations:Orig=0 BW=-1Dest=1 BW=6(-1 implies infinite bw available)
16:03:25.359 CCM|MediaManager(1) - wait_AuConnectInfo - recieved response, fowarding,
CI(16777217,16777218)|<CT::1,100,105,1.213><IP::128.107.142.233>
16:03:25.359 CCM|MediaCoordinator -
wait_AuConnectInfoInd|<CT::1,100,105,1.213><IP::128.107.142.233>
16:03:25.359 CCM|ConnectionManager - wait_AuConnectInfoInd,
CI(16777217,16777218)|<CT::1,100,105,1.213><IP::128.107.142.233>
```

次のメッセージは、Cisco IP Phone（1001）からのオンフック メッセージが受信されていることを示しています。オンフック メッセージが受信されるとすぐに、H.225 メッセージと Skinny Station デバイス接続解除メッセージが送信され、H.225 メッセージ全体が表示されます。最後のメッセージは、コールが終了したことを示しています。

```
15:20:27.296 CCM|StationInit - InboundStim - OnHookMessageID tcpHandle=0x5138d98
15:20:27.296 CCM|ConnectionManager -wait_AuDisconnectRequest (16777247,16777248): STOP
SESSION
15:20:27.296 CCM|MediaManager - wait_AuDisconnectRequest - StopSession sending
disconnect to (64,5) and remove connection from list
15:20:27.296 CCM| Device SEP003094C26105 , UnRegisters with SDL Link to monitor
NodeID= 1
15:20:27.296 CCM|StationD - stationOutputCloseReceiveChannel tcpHandle=0x5138d98 myIP:
e74610ac (172.16.70.231)
15:20:27.296 CCM|StationD - stationOutputStopMediaTransmission tcpHandle=0x5138d98
myIP: e74610ac (172.16.70.231)
15:20:28.328 CCM|In  Message -- H225ReleaseCompleteMsg -- Protocol= H225Protocol
16:03:33.344 CCM|StationInit - InboundStim - StationOnHookMessageID: Msg
Size(received, defined) = 4, 12|<CT::1,100,105,1.219><IP::128.107.142.233>
16:03:33.359 CCM|ConnectionManager - wait_AuDisconnectRequest(16777217,16777218): STOP
SESSION|<CT::1,100,105,1.219><IP::128.107.142.233>
16:03:33.359 CCM|StationD(2):   TCPPid = [1.100.117.2] CloseReceiveChannel
conferenceID=16777218 passThruPartyID=16777249.  myIP: e98e6b80
(128.107.142.233)|<CT::1,100,105,1.219><IP::128.107.142.233>
16:03:33.359 CCM|StationD(2):   TCPPid = [1.100.117.2] StopMediaTransmission
conferenceID=16777218 passThruPartyID=16777249.  myIP: e98e6b80
(128.107.142.233)|<CT::1,100,105,1.219><IP::128.107.142.233>
16:03:33.359 CCM|StationD(2):   TCPPid = [1.100.117.2]
star_StationOutputCloseMultiMediaReceiveChannel conferenceID=16777218
passThruPartyID=16777249.  myIP: e98e6b80
(128.107.142.233)|<CT::1,100,105,1.219><IP::128.107.142.233>
16:03:33.359 CCM|StationD(2):   TCPPid = [1.100.117.2]
star_StationOutputStopMultiMediaTransmission conferenceID=16777218
passThruPartyID=16777250.  myIP: e98e6b80
(128.107.142.233)|<CT::1,100,105,1.219><IP::128.107.142.233>
```

Cisco IOS Gatekeeper のデバッグ メッセージと表示コマンド

「コール フロー トレース」では、Cisco CallManager SDI トレースについて詳細に説明しました。このケース スタディのトポロジでは、debug ras コマンドが Cisco IOS Gatekeeper でオンになっています。

次のデバッグ メッセージは、Cisco IOS Gatekeeper が Cisco CallManager（172.16.70.228）に対する admission request（ARQ; アドミッション要求）を受信し、その他の正常な Remote Access Server（RAS）メッセージがその後に続いていることを示しています。最後に、Cisco IOS Gatekeeper が admission confirmed（ACF; アドミッション確認）メッセージを Cisco CallManager に送信します。

```
*Mar 12 04:03:57.181: RASLibRASRecvData ARQ (seq# 3365) rcvd from [172.16.70.228883]
on sock [0x60AF038C]
*Mar 12 04:03:57.181: RASLibRAS_WK_TInit ipsock [0x60A7A68C] setup successful
*Mar 12 04:03:57.181: RASlibras_sendto msg length 16 from 172.16.70.2251719 to
172.16.70.228883
*Mar 12 04:03:57.181: RASLibRASSendACF ACF (seq# 3365) sent to 172.16.70.228
```

次のデバッグ メッセージは、コールが進行中であることを示しています。

```
*Mar 12 04:03:57.181: RASLibRASRecvData successfully rcvd message of length 55 from
172.16.70.228883
```

次のデバッグ メッセージは、Cisco IOS Gatekeeper が Cisco CallManager（172.16.70.228）から disengaged request (DRQ; 解除要求）を受信し、Cisco IOS Gatekeeper が disengage confirmed（DCF; 解除確認）を Cisco CallManager に送信したことを示しています。

```
*Mar 12 04:03:57.181: RASLibRASRecvData DRQ (seq# 3366) rcvd from [172.16.70.228883]
on sock [0x60AF038C]
*Mar 12 04:03:57.181: RASlibras_sendto msg length 3 from 172.16.70.2251719 to
172.16.70.228883
*Mar 12 04:03:57.181: RASLibRASSendDCF DCF (seq# 3366) sent to 172.16.70.228
*Mar 12 04:03:57.181: RASLibRASRecvData successfully rcvd message of length 124 from
172.16.70.228883
```

Cisco IOS Gatekeeper に対するコマンド show gatekeeper endpoints は、4 つの Cisco CallManager がすべて Cisco IOS Gatekeeper に登録されていることを表示します。このケース スタディのトポロジでは、各クラスタに 2 つずつ、計 4 つの Cisco CallManager が存在することに注意してください。この Cisco IOS Gatekeeper には 2 つのゾーンがあり、各ゾーンには 2 つの Cisco CallManager があります。

```
R2514-1#show gatekeeper endpoints
                    GATEKEEPER ENDPOINT REGISTRATION
                    ================================
CallSignalAddr  Port  RASSignalAddr   Port  Zone Name         Type
--------------- ----- --------------- ----- ---------         ----
172.16.70.228       2     172.16.70.228       1493  gka.cisco.com     VOIP-GW
    H323-ID: ac1046e4->ac1046f5
172.16.70.229       2     172.16.70.229        3923  gka.cisco.com     VOIP-GW
    H323-ID: ac1046e5->ac1046f5
172.16.70.245       1     172.16.70.245        1041  gkb.cisco.com     VOIP-GW
    H323-ID: ac1046f5->ac1046e4
172.16.70.243        1     172.16.70.243       2043  gkb.cisco.com     VOIP-GW
    H323-ID: ac1046f5->ac1046e4
Total number of active registrations = 4
```

Cisco IOS Gateway のデバッグ メッセージと表示コマンド

「Cisco IOS Gatekeeper のデバッグ メッセージと表示コマンド」の項では、Cisco IOS Gatekeeper の表示コマンドとデバッグ出力について詳細に説明しました。この項では、Cisco IOS Gateway のデバッグ出力と表示コマンドについて取り上げます。このケース スタディのトポロジでは、コールは Cisco IOS Gateway を経由します。Cisco IOS Gateway は、T1/CAS または T1/PRI のいずれかのインターフェイスで PSTN または PBX に接続しています。次の例は、debug voip ccapi inout、debug H225 events、debug H225 asn1 などのコマンドのデバッグ出力を示しています。

次のデバッグ出力では、Cisco IOS Gateway が Cisco CallManager（172.16.70.228）からの TCP 接続要求を H.225 用のポート 2328 で受け入れます。

```
*Mar 12 04:03:57.169: H225Lib::h225TAccept: TCP connection accepted from
172.16.70.228:2328 on socket [1]
*Mar 12 04:03:57.169: H225Lib::h225TAccept: Q.931 Call State is initialized to be
[Null].
*Mar 12 04:03:57.177: Hex representation of the received TPKT03000065080000100
```

次のデバッグ出力は、この TCP セッションで Cisco CallManager から H.225 データが到達していることを示しています。このデバッグ出力では、使用されている H.323 バージョンを指定する protocolIdentifier に注意してください。次のデバッグは、H.323 バージョン 2 が使用されていることを示しています。この例は、着信側と発信側の番号も示しています。

```
- Source Address H323-ID
- Destination Address e164
*Mar 12 04:03:57.177:       H225Lib::h225RecvData: Q.931 SETUP received from socket
[1]value H323-UserInformation ::=
*Mar 12 04:03:57.181: {
*Mar 12 04:03:57.181:   h323-uu-pdu
*Mar 12 04:03:57.181:   {
*Mar 12 04:03:57.181:     h323-message-body setup :
*Mar 12 04:03:57.181:       {
*Mar 12 04:03:57.181:         protocolIdentifier { 0 0 8 2250 0 2 },
*Mar 12 04:03:57.181:         sourceAddress
*Mar 12 04:03:57.181:         {
*Mar 12 04:03:57.181:           h323-ID : "1001"
*Mar 12 04:03:57.181:         },
*Mar 12 04:03:57.185:         destinationAddress
*Mar 12 04:03:57.185:         {
*Mar 12 04:03:57.185:           e164 : "3333"
*Mar 12 04:03:57.185:         },
*Mar 12 04:03:57.189:       H225Lib::h225RecvData: State changed to [Call Present].
```

次のデバッグ出力は、Call Control Application Programming Interface（CCAPi）を示しています。Call Control APi は着信コールを指定します。次の出力では、着信側と発信側の情報も確認できます。CCAPi はダイヤルピア 0 と一致します。0 はデフォルトのダイヤルピアです。CCAPi がダイヤルピア 0 と一致するのは、発信側の番号について他のダイヤルピアが見つからなかったので、デフォルトのダイヤルピアを使用しているためです。

```
*Mar 12 04:03:57.189: cc_api_call_setup_ind (vdbPtr=0x616C9F54, callInfo={called=3333,
calling=1001, fdest=1 peer_tag=0}, callID=0x616C4838)
*Mar 12 04:03:57.193: cc_process_call_setup_ind (event=0x617A2B18) handed call to app
"SESSION"
*Mar 12 04:03:57.193: sess_appl: ev(19=CC_EV_CALL_SETUP_IND), cid(17), disp(0)
*Mar 12 04:03:57.193: ccCallSetContext (callID=0x11, context=0x61782BBC)
Mar 12 04:03:57.193: ssaCallSetupInd finalDest cllng(1001), clled(3333)
*Mar 12 04:03:57.193: ssaSetupPeer cid(17) peer list:  tag(1)
*Mar 12 04:03:57.193: ssaSetupPeer cid(17), destPat(3333), matched(4), prefix(),
peer(6179E63C)
*Mar 12 04:03:57.193: ccCallSetupRequest (peer=0x6179E63C, dest=, params=0x61782BD0
mode=0, *callID=0x617A87C0)
*Mar 12 04:03:57.193: callingNumber=1001, calledNumber=3333, redirectNumber=
*Mar 12 04:03:57.193: accountNumber=,finalDestFlag=1,
guid=0098.89c8.9233.511d.0300.cddd.ac10.46e6
```

CCAPi は、ダイヤルピア 1 と宛先パターン（着信側の番号 3333）を一致させます。peer_tag はダイヤルピアを意味することに留意してください。要求パケット内の発信側と着信側の番号に注目してください。

```
*Mar 12 04:03:57.193: peer_tag=1
*Mar 12 04:03:57.197: ccIFCallSetupRequest: (vdbPtr=0x617BE064, dest=,
callParams={called=3333, calling=1001, fdest=1, voice_peer_tag=1}, mode=0x0)
```

次のデバッグ出力は、H.225 アラート メッセージが Cisco CallManager に返されていることを示しています。

```
*Mar 12 04:03:57.197: ccCallSetContext (callID=0x12, context=0x61466B30)
*Mar 12 04:03:57.197: ccCallProceeding (callID=0x11, prog_ind=0x0)
*Mar 12 04:03:57.197: cc_api_call_proceeding(vdbPtr=0x617BE064, callID=0x12,
prog_ind=0x0)
*Mar 12 04:03:57.197: cc_api_call_alert(vdbPtr=0x617BE064, callID=0x12, prog_ind=0x8,
sig_ind=0x1)
*Mar 12 04:03:57.201: sess_appl: ev(17=CC_EV_CALL_PROCEEDING), cid(18), disp(0)
*Mar 12 04:03:57.201: ssa:
cid(18)st(1)oldst(0)cfid(-1)csize(0)in(0)fDest(0)-cid2(17)st2(1)oldst2(0)
*Mar 12 04:03:57.201: ssaIgnore cid(18), st(1),oldst(1), ev(17)
*Mar 12 04:03:57.201: sess_appl: ev(7=CC_EV_CALL_ALERT), cid(18), disp(0)
*Mar 12 04:03:57.201: ssa:
cid(18)st(1)oldst(1)cfid(-1)csize(0)in(0)fDest(0)-cid2(17)st2(1)oldst2(0)
*Mar 12 04:03:57.201: ssaFlushPeerTagQueue cid(17) peer list: (empty)
*Mar 12 04:03:57.201: ccCallAlert (callID=0x11, prog_ind=0x8, sig_ind=0x1)
*Mar 12 04:03:57.201: ccConferenceCreate (confID=0x617A8808, callID1=0x11,
callID2=0x12, tag=0x0)
*Mar 12 04:03:57.201: cc_api_bridge_done (confID=0x7, srcIF=0x616C9F54,
srcCallID=0x11, dstCallID=0x12, disposition=0, tag=0x0)value H323-UserInformation
*Mar 12 04:03:57.201: {
*Mar 12 04:03:57.201:   h323-uu-pdu
*Mar 12 04:03:57.201:   {
*Mar 12 04:03:57.201:     h323-message-body alerting :
*Mar 12 04:03:57.201:       {
*Mar 12 04:03:57.201:         protocolIdentifier { 0 0 8 2250 0 2 },
*Mar 12 04:03:57.205:         destinationInfo
*Mar 12 04:03:57.205:         {
*Mar 12 04:03:57.205:           mc FALSE,
*Mar 12 04:03:57.205:           undefinedNode FALSE
*Mar 12 04:03:57.205:         },
```

このパケットでは、Cisco IOS が H.245 アドレスとポート番号も Cisco CallManager に送信していることに注意してください。Cisco IOS Gateway は到達不能なアドレスを送信する場合があるため、無音声または単方向音声になることがあります。

```
*Mar 12 04:03:57.205:          h245Address ipAddress :
*Mar 12 04:03:57.205:            {
*Mar 12 04:03:57.205:              ip 'AC1046E2'H,
*Mar 12 04:03:57.205:              port 011008
*Mar 12 04:03:57.205:            },
*Mar 12 04:03:57.213: Hex representation of the ALERTING TPKT to send.0300003D0100
*Mar 12 04:03:57.213:
*Mar 12 04:03:57.213:       H225Lib::h225AlertRequest: Q.931 ALERTING sent from socket
[1]. Call state changed to [Call Received].
*Mar 12 04:03:57.213: cc_api_bridge_done (confID=0x7, srcIF=0x617BE064,
srcCallID=0x12, dstCallID=0x11, disposition=0, tag=0x0)
```

次のデバッグ出力は、H.245 セッションが開始していることを示しています。コーデック ネゴシエーションの機能表示および各音声パケットに含まれるバイト数を確認できます。

```
*Mar 12 04:03:57.217: cc_api_caps_ind (dstVdbPtr=0x616C9F54, dstCallId=0x11,
srcCallId=0x12, caps={codec=0xEBFB, fax_rate=0x7F, vad=0x3, modem=0x617C5720
codec_bytes=0, signal_type=3})
*Mar 12 04:03:57.217: sess_appl: ev(23=CC_EV_CONF_CREATE_DONE), cid(17), disp(0)
*Mar 12 04:03:57.217: ssa:
cid(17)st(3)oldst(0)cfid(7)csize(0)in(1)fDest(1)-cid2(18)st2(3)oldst2(1)
*Mar 12 04:03:57.653: cc_api_caps_ind (dstVdbPtr=0x617BE064, dstCallId=0x12,
srcCallId=0x11, caps={codec=0x1, fax_rate=0x2, vad=0x2, modem=0x1, codec_bytes=160,
signal_type=0})
```

次のデバッグ出力は、両方の側が正常にネゴシエートし、160 バイトのデータを持つ G.711 コーデックで合意したことを示しています。

```
*Mar 12 04:03:57.653: cc_api_caps_ack (dstVdbPtr=0x617BE064, dstCallId=0x12,
srcCallId=0x11, caps={codec=0x1, fax_rate=0x2, vad=0x2, modem=0x1, codec_bytes=160,
signal_type=0})
*Mar 12 04:03:57.653: cc_api_caps_ind (dstVdbPtr=0x617BE064, dstCallId=0x12,
srcCallId=0x11, caps={codec=0x1, fax_rate=0x2, vad=0x2, modem=0x, codec_bytes=160,
signal_type=0})
*Mar 12 04:03:57.653: cc_api_caps_ack (dstVdbPtr=0x617BE064, dstCallId=0x12,
srcCallId=0x11, caps={codec=0x1, fax_rate=0x2, vad=0x2, modem=0x1, codec_bytes=160,
signal_type=0})
*Mar 12 04:03:57.657: cc_api_caps_ack (dstVdbPtr=0x616C9F54, dstCallId=0x11,
srcCallId=0x12, caps={codec=0x1, fax_rate=0x2, vad=0x2, modem=0x1, codec_bytes=160,
signal_type=0})
*Mar 12 04:03:57.657: cc_api_caps_ack (dstVdbPtr=0x616C9F54, dstCallId=0x11,
srcCallId=0x12, caps={codec=0x1, fax_rate=0x2, vad=0x2, modem=0x1, codec_bytes=160,
signal_type=0})
```

H.323 接続および接続解除のメッセージがこの後に続きます。

```
*Mar 12 04:03:59.373: cc_api_call_connected(vdbPtr=0x617BE064, callID=0x12)
*Mar 12 04:03:59.373: sess_appl: ev(8=CC_EV_CALL_CONNECTED), cid(18), disp(0)
*Mar 12 04:03:59.373: ssa:
cid(18)st(4)oldst(1)cfid(7)csize(0)in(0)fDest(0)-cid2(17)st2(4)oldst2(3)
*Mar 12 04:03:59.373: ccCallConnect (callID=0x11)
*Mar 12 04:03:59.373: {
*Mar 12 04:03:59.373:   h323-uu-pdu
*Mar 12 04:03:59.373:   {
*Mar 12 04:03:59.373:     h323-message-body connect :
*Mar 12 04:03:59.373:       {
*Mar 12 04:03:59.373:         protocolIdentifier { 0 0 8 2250 0 2 },
*Mar 12 04:03:59.373:         h245Address ipAddress :
*Mar 12 04:03:59.373:           {
*Mar 12 04:03:59.377:             ip 'AC1046E2'H,
*Mar 12 04:03:59.377:             port 011008
*Mar 12 04:03:59.377:           },
*Mar 12 04:03:59.389: Hex representation of the CONNECT TPKT to send.03000052080
*Mar 12 04:03:59.393: H225Lib::h225SetupResponse: Q.931 CONNECT sent from socket [1]
*Mar 12 04:03:59.393: H225Lib::h225SetupResponse: Q.931 Call State changed to
[Active].
*Mar 12 04:04:08.769: cc_api_call_disconnected(vdbPtr=0x617BE064, callID=0x12,
cause=0x10)
*Mar 12 04:04:08.769: sess_appl: ev(12=CC_EV_CALL_DISCONNECTED), cid(18), disp(0)
```

T1/PRI インターフェイスを使用する Cisco IOS Gateway

前述したように、2 つのタイプのコールが Cisco IOS Gateway を経由し、Cisco IOS Gateway は、T1/CAS または T1/PRI のいずれかのインターフェイスで PSTN または PBX に接続しています。次の例は、Cisco IOS Gateway が T1/PRI インターフェイスを使用する場合のデバッグ出力を示しています。

Cisco IOS Gateway で debug isdn q931 コマンドがオンになっていて、ISDN 環境にある D チャネル用のレイヤ 3 シグナリング プロトコルである Q.931 が有効になっています。T1/PRI インターフェイスからコールが発信されるたびに、セットアップ パケットが送信される必要があります。セットアップ パケットには必ずプロトコル記述子 pd = 8 が含まれており、callref 用にランダムな 16 進数値が生成されます。callref はコールを追跡します。たとえば、2 つのコールが発信された場合、callref の値によって、RX（受信済み）メッセージの対象になっているコールを判別できます。ベアラ機能 0x8890 は 64 Kbps データ コールを意味します。これが 0x8890218F だった場合は、56 Kbps データ コールになり、音声コールでは 0x8090A3 になります。下記のデバッグ出力では、ベアラ機能は 0x8090A3（音声用）です。この例は、着信側と発信側の番号を示しています。

callref では、最初の数字に異なる値が使用され（TX と RX を区別するため）、2 番目の値は同じです（SETUP には最後の数字に 0 が設定され、CONNECT_ACK にも 0 が設定されています）。ルータは PSTN または PBX に完全に依存して Bearer チャネル（B チャネル）を割り当てます。PSTN または PBX がルータにチャネルを割り当てない場合、コールはルーティングされません。このケース スタディでは、ALERTING 用に受信されたものと同じ参照番号（0x800B）を使用して、CONNECT メッセージが交換機から受信されます。最後に、コールが接続解除されるとき、DISCONNECT メッセージの交換の後に、RELEASE メッセージおよび RELEASE _COMP メッセージが続きます。RELEASE_COMP メッセージの後には、コール拒否の理由 ID が続きます。理由 ID は 16 進数値です。理由の内容は、16 進数値のデコードとプロバイダーのフォローアップによって確認できます。

```
 *Mar  1 225209.694 ISDN Se115 TX ->  SETUP pd = 8  callref = 0x000B
 *Mar  1 225209.694         Bearer Capability i = 0x8090A3
 *Mar  1 225209.694         Channel ID i = 0xA98381
 *Mar  1 225209.694         Calling Party Number i = 0x2183, '1001'
 *Mar  1 225209.694         Called Party Number i = 0x80, '3333'
 *Mar  1 225209.982 ISDN Se115 RX <-  ALERTING pd = 8  callref = 0x800B
 *Mar  1 225209.982         Channel ID i = 0xA98381
 *Mar  1 225210.674 ISDN Se115 RX <-  CONNECT pd = 8  callref = 0x800B
 *Mar  1 225210.678 ISDN Se115 TX ->  CONNECT_ACK pd = 8  callref = 0x000B
 *Mar  1 225215.058 ISDN Se115 RX <-  DISCONNECT pd = 8  callref = 0x800B
 *Mar  1 225215.058         Cause i = 0x8090 - Normal call clearing  225217 %ISDN-6
DISCONNECT Int S10  disconnected from unknown , call lasted 4 sec
 *Mar  1 225215.058 ISDN Se115 TX ->  RELEASE pd = 8  callref = 0x000B
 *Mar  1 225215.082 ISDN Se115 RX <-  RELEASE_COMP pd = 8  callref = 0x800B
*Mar  1 225215.082  Cause i = 0x829F - Normal, unspecified or Special  intercept, call
blocked group restriction
```

T1/CAS インターフェイスを使用する Cisco IOS Gateway

2 つのタイプのコールが Cisco IOS Gateway を経由し、Cisco IOS Gateway は、T1/CAS または T1/PRI のいずれかのインターフェイスで　PSTN　または　PBX　に接続しています。次の例は、Cisco IOS Gateway が T1/CAS インターフェイスを使用する場合のデバッグ出力を示しています。Cisco IOS Gateway で debug cas はオンになっています。

次のデバッグ メッセージは、Cisco IOS Gateway がオフフック信号を交換機に送信していることを示しています。

```
Apr  5 17:58:21.727: from NEAT(0): (0/15): Tx LOOP_CLOSURE (ABCD=1111)
```

次のデバッグ メッセージは、交換機が Cisco IOS Gateway から閉ループ信号を受信した後にウィンクを送信していることを示しています。

```
Apr  5 17:58:21.859: from NEAT(0): (0/15): Rx LOOP_CLOSURE (ABCD=1111)
Apr  5 17:58:22.083: from NEAT(0): (0/15): Rx LOOP_OPEN (ABCD=0000)
```

次のデバッグ メッセージは、Cisco IOS Gateway がオフフックしようとしていることを示しています。

```
Apr  5 17:58:23.499: from NEAT(0): (0/15): Rx LOOP_CLOSURE (ABCD=1111)
```

次の出力は、コール進行中の Cisco IOS Gateway での show call active voice brief を示しています。この出力は、着信側と発信側の番号およびその他の有用な情報も示しています。

```
R5300-5#show call active voice brief
<ID>: <start>hs.<index> +<connect> pid:<peer_id> <dir> <addr> <state>
tx:<packets>/<bytes> rx:<packets>/<bytes> <state>
 IP <ip>:<udp> rtt:<time>ms pl:<play>/<gap>ms lost:<lost>/<early>/<late>
delay:<last>/<min>/<max>ms <codec>
 FR <protocol> [int dlci cid] vad:<y/n> dtmf:<y/n> seq:<y/n> sig:<on/off> <codec>
(payload size)
 Tele <int>: tx:<tot>/<v>/<fax>ms <codec> noise:<l> acom:<l> i/o:<l>/<l> dBm
511D : 156043737hs.1 +645 pid:0 Answer 1001 active
 tx:1752/280320 rx:988/158080
 IP172.16.70.228:18888 rtt:0ms pl:15750/80ms lost:0/0/0 delay:25/25/65ms g711ulaw
511D : 156043738hs.1 +644 pid:1 Originate 3333 active
 tx:988/136972 rx:1759/302548
 Tele 1/0/0 (30): tx:39090/35195/0ms g711ulaw noise:-43 acom:0  i/0:-36/-42 dBm
```