



## CHAPTER 12

# ケーススタディ：Cisco Unified IP Phone と Cisco IOS ゲートウェイ間のコールのトラ ブルシューティング

付録 11 「ケーススタディ：Cisco Unified 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 を経由して接続している電話機、または公衆電話交換網（PSTN）上の電話機に、Cisco IOS ゲートウェイを経由してコールを発信する Cisco Unified IP Phone について説明します。概念的には、コールが Cisco IOS ゲートウェイに到達すると、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いずれかに転送します。PBX に転送されたコールは、ローカル PBX に接続された電話機で終端するか、または PBX によって PSTN 経由で転送され、PSTN 上で終端します。

この項では、次のトピックを扱います。

- コールフローのトレース
- Cisco IOS ゲートキーパーのデバッグメッセージと表示コマンド
- Cisco IOS ゲートウェイ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 ゲートウェイ
- T1/CAS インターフェイスを使用する Cisco IOS ゲートウェイ

## コールフローのトレース

ここでは、Cisco Communications Manager 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。初期化、登録、キープアライブメカニズムなど、トレース情報の詳細については付録 11 「ケーススタディ：Cisco Unified IP Phone コールのトラブルシューティング」で説明しているため、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 にある Cisco Unified IP Phone（電話番号 1001）が PSTN 上の電話機（電話番号 3333）にコールを発信します。トレースでデバイスを追跡するには、デバイスの TCP ハンドル値、タイムスタンプ、または名前を調べます。デバイスの TCP ハンドル値は、デバイスがリブートされるかオフラインになるまで変わりません。

次のトレースでは、Cisco Unified IP Phone（1001）がオフフックになっています。トレースは、Cisco Unified IP Phone に表示される固有のメッセージ、TCP ハンドル、発信番号を示しています。ユーザはまだ番号をダイヤルしていないため、この時点では着信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 つずつダイヤルします)。番号 3333 は、PSTN ネットワーク上の電話機の宛先番号です。Cisco Unified Communications Manager の番号分析プロセスが現在アクティブになっており、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 11 「ケース スタディ : Cisco Unified IP Phone コールの問題シューティング」で詳しく説明しています。

```
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、情報が解析されています。

```
|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 が発信元のロケーションを示し、番号 1 が宛先のロケーションを示しています。BW = -1 は、発信元のロケーションの帯域幅を示します。値 -1 は、帯域幅が無限であることを暗黙的に示します。コールは LAN 環境にある Cisco Unified IP Phone から発信されたため、帯域幅は無限であると見なされます。BW = 64 は、宛先のロケーションの帯域幅を示します。コールの宛先は PSTN 上にある電話機で、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、発信側および着信側の情報を示しています。この例では、管理者が表示名 (John Smith など) を設定していないため、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、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、16 進表記の着信側の 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、発信側および着信側の情報と H.225 アラートメッセージを示しています。また、Cisco Unified IP Phone の 16 進数値と IP アドレスのマッピングも示されています。Cisco Unified 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、このコールで使用されている圧縮タイプ (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.323 は H.225 アラートメッセージが送信されたあとで、H.245 を初期化します。次のトレースは、発信側および着信側の情報と 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同じで、同じコールが続い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 接続メッセージと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、コールが接続します。

```
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, forwarding,
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 Unified IP Phone (1001) からのオンフックメッセージを受信していることを示しています。オンフックメッセージを受信するとすぐに、H.225 と Skinny Station デバイスの接続解除メッセージが送信され、H.225 メッセージの全文が表示されます。この最後のメッセージには、コールが終了したことが示されています。

```
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): TCPPIID = [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): TCPPIID = [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): TCPPIID = [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): TCPPIID = [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 ゲートキーパーのデバッグメッセージと表示コマンド

「コールフローのトレース」(P.12-1) で、Communications Manager SDI トレースについて詳しく説明しています。このケーススタディのトポロジでは、Cisco IOS ゲートキーパーで `debug ras` コマンドがオンになっています。

次のデバッグメッセージは、Cisco IOS ゲートキーパーが Cisco Unified Communications Manager (172.16.70.228) に対するアドミッション要求 (ARQ) を受信し、その他の正常なリモートアクセスサーバ (RAS) メッセージが後続していることを示しています。最後に、Cisco IOS ゲートキーパーがアドミッション確認 (ACF) メッセージを Cisco Unified Communications Manager に送信しています。

```

*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 ゲートキーパーが Cisco Unified Communications Manager (172.16.70.228) から解除要求 (DRQ) を受信し、Cisco IOS ゲートキーパーが解除確認 (DCF) を Cisco Unified Communications Manager 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 ゲートキーパーに対する `show gatekeeper endpoints` コマンドは、4 つの Cisco Unified Communications Manager がすべて Cisco IOS ゲートキーパー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、4 つの Cisco Unified Communications Manager があります。この Cisco IOS ゲートキーパーには 2 つのゾーンがあり、各ゾーンに 2 つの Cisco Unified Communications Manager 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 ゲートウェイのデバッグメッセージと表示コマンド

「Cisco IOS ゲートキーパーのデバッグメッセージと表示コマンド」(P.12-4) では、Cisco IOS ゲートキーパーの表示コマンドとデバッグ出力について詳しく説明しました。ここでは、Cisco IOS ゲートウェイのデバッグ出力と表示コマンドについて説明します。このケーススタディのトポロジでは、コールが Cisco IOS ゲートウェイを通過します。Cisco IOS ゲートウェイは T1/CAS または T1/PRI のいずれかのインターフェイスを使用して PSTN または PBX に接続します。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です。

次のデバッグ出力では、Cisco IOS ゲートウェイが Cisco Unified Communications Manager (172.16.70.228) からの TCP 接続要求を H.225 のポート 2328 で受け入れています。

```
*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 TPDKT03000065080000100
```

次のデバッグ出力は、この TCP セッションで、H.225 データが Cisco Unified Communications Manager から着信していることを示しています。このデバッグ出力には、使用されている H.323 バージョンを表す `protocolIdentifier` が示されています。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。また、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 は着信コールを示します。次の出力では、着信側および発信側の情報も確認できません。CCAPi は、デフォルトのダイヤル ピアであるダイヤル ピア 0 に一致します。CCAPi がダイヤル ピア 0 に一致するのは、発信番号に対する他のダイヤル ピアが見つからず、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), cllcd(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（着信番号 3333）を照合します。peer\_tag はダイヤル ピアを意味します。要求パケット内の発信側と着信側の番号が示されています。

```
*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 Unified Communications Manager 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)csz(0)in(0)fDest(0)-cid2(17)st(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)csz(0)in(0)fDest(0)-cid2(17)st(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 Unified Communications Manager に送信しています。Cisco IOS ゲートウェイは到達不能なアドレスを送信する場合がありますため、無音声または片通話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 セッションが開始され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、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、両方の側が適切にネゴシエートし、160 バイトデータの 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後続します。

```

*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)csz(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 04:03:59.373:         ip 'AC1046E2'H,
*Mar 12 04:03:59.373:         port 011008
*Mar 12 04:03:59.373:       },
*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 ゲートウェイ

上記のように、2つのタイプのコールが Cisco IOS ゲートウェイを通過します。Cisco IOS ゲートウェイは T1/CAS または T1/PRI のいずれかのインターフェイスを使用して PSTN または PBX に接続します。次の例は、Cisco IOS ゲートウェイが T1/PRI インターフェイスを使用する場合のデバッグ出力です。

Cisco IOS ゲートウェイの `debug isdn q931` コマンドがオンになり、それにより 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 がイネーブルになります。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、`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` のデータ コールです。これが `0x8890218F` だった場合は、`56 Kbps` のデータ コールになり、音声コールの場合は `0x8090A3` になります。次のデバッグ出力では、ベアラ機能は `0x8090A3` で、音声に適用されます。例には、着信側と発信側の番号が示されています。

`callref` では、最初の数字に異なる値を使用し (TX と RX を識別するため)、2 番目の値は同じです (SETUP の最後の数字は 0、CONNECT\_ACK も 0 です)。ルータはベアラ チャネル (B チャネル) を割り当てる際に PSTN または PBX に完全に依存します。PSTN または PBX がルータにチャネルを割り当てないと、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、交換機から受信される CONNECT メッセージに、ALERTING 用に受信されたものと同じ参照番号 (`0x800B`) が含まれています。最後に、コールが接続解除されるとき、DISCONNECT メッセージの交換後に RELEASE メッセージと RELEASE\_COMP メッセージが続くことを確認できます。RELEASE\_COMP メッセージのあとには、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 ゲートウェイ

2つのタイプのコールが Cisco IOS ゲートウェイを通過します。Cisco IOS ゲートウェイは T1/CAS または T1/PRI のいずれかのインターフェイスを使用して PSTN または PBX に接続します。次のデバッグ出力は、Cisco IOS ゲートウェイが T1/CAS インターフェイスを使用する場合に発生します。Cisco IOS ゲートウェイでは `debug cas` がオンになっています。

次のデバッグメッセージは、Cisco IOS ゲートウェイ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 ゲートウェイから閉ループ信号を受信後、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 ゲートウェイ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、コールが進行中の Cisco IOS ゲートウェイでの `show call active voice brief` を示しています。出力には、着信側と発信側の番号およびその他の役立つ情報も示されています。

```

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/o:-36/-42 dBm

```

■ T1/CAS インターフェイスを使用する Cisco IOS ゲートウェイ