

語音連線中繼故障排除

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[慣例](#)

[問題](#)

[解決方案](#)

[連線中繼的常見問題](#)

[開始排除故障](#)

[確定哪些呼叫已啟動](#)

[DTMF故障排除](#)

[相關資訊](#)

簡介

語音連線中繼會永久建立語音呼叫，包括IP語音(VoIP)、幀中繼語音(VoFR)或ATM語音(VoATM)。一旦路由器啟動且配置完成，呼叫就會建立。一旦語音埠開啟，語音埠將自動撥打語音埠下指定的虛擬電話號碼，並向該位置發出呼叫。語音埠通過相應的撥號對等體完成到另一端的呼叫。建立此連線後，就路由器而言，語音呼叫處於會話中且已連線。

必要條件

需求

本文件沒有特定先決條件。

採用元件

本文件所述內容不限於特定軟體和硬體版本。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的I網路正在作用，請確保您在使用之前瞭解任何指令可能造成的影響。

慣例

如需文件慣例的詳細資訊，請參閱[思科技術提示慣例](#)。

問題

與中繼相關的常見問題對路由器是透明的，並且很難進行故障排除。當呼叫置於中繼上且未聽到任何消息時，會顯現出語音中繼上出現的常見問題。這是連線中繼的已知問題之一，由許多不同問題引起。另一個問題是未正確傳遞的雙音多頻(DTMF)音，並且從專用交換機(PBX)到PBX的信令傳輸不正確。本檔案將疑難排解。

當語音卡處於開啟狀態並處於活動狀態時，連線卡中的訊號行為會有所不同。通常在語音連線埠下發出有關訊號特性的任何指令都無關且實用。語音中繼線成為一條信令管道，通過VoIP鏈路中繼訊號。使用語音中繼時，PBX信令必須端到端匹配。就這兩台PBX電腦而言，其目標是使語音中繼連線看起來與租用T1線路與PBX完全相同，使路由器完全透明，同時在整個過程中兩個PBX之間建立明確的鏈路。

當中繼啟動時，中繼變為軟體電纜，並且訊號型別被視為聯結器型別。中繼並不關心所使用的訊號型別。即使兩端訊號不匹配，中繼仍然會啟動。只要兩端的PBX執行相同的信令，中繼就能正常工作。

解決方案

排除連線中繼故障時採用的方法與用於交換呼叫的方法不同。要檢視驗證中繼後實際發生的情況，您需要檢視PBX信令。繼續檢視信令之前，請確認中繼已啟動，並且數位訊號處理器(DSP)處理語音資料包。

注意：您可能要禁用語音活動檢測(VAD)以進行故障排除。驗證中繼是否正常工作後，您需要檢視電話信令以進一步排除故障。

如果中繼已建立，並且沒有人嘗試呼叫，則中繼保持連線消息會在遠端盒子之間來回傳送。這些keepalive會驗證中繼連線並從端對端傳送訊號資訊。若要驗證這些keepalive，請發出[debug vpm signal](#) 指令。如果有多個中繼，則debug vpm命令的輸出可限制為單個埠(如果您發出debug vpm port x 命令選項，其中「x」是相關語音埠)。以下是在檢視所有連線埠時發出的debug vpm signal命令的輸出：

```
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
```

如果加以限制，則使用[debug vpm port x](#) 指令時，偵錯會更容易解讀，如以下範例所示：

```
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
```

keepalive每五秒傳送和接收一次。術語「傳送到dsp」和「從dsp接收」是從Cisco IOS®的角度的。用PBX代替DSP，使其更易於理解。這些是中繼上沒有活動時看到的消息。keepalive訊息可讓電路兩端的路由器知道主幹仍處於接通狀態。當連續錯過其中五條消息時，中繼將關閉。其中一個原因是在網路中中繼不斷抖動。若要驗證是否已傳送和接收語音中繼keepalive，請發出**debug vpm trunk-sc**命令。在錯過了中繼keepalive之前，此調試不會產生任何輸出。以下是在keepalive遺失時**debug vpm trunk-sc**指令輸出的範例：

```
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
```

如果在發出**debug vpm trunk-sc**命令時未看到輸出，則不會遺漏keepalive。即使keepalive被丟失，中繼也會保持運作，直到五個連續消息被丟失。這表示連線必須中斷25秒才能中斷中繼。

連線中繼的常見問題

有多個錯誤與語音中繼連線相關。如果您看到任何異常情況，請檢查這些錯誤。在Cisco IOS軟體12.2發佈時，大多數問題都已得到解決和整合。您可以檢視這些錯誤，讓自己意識到這些都是舊代碼出現問題的原因。最常見的問題之一就是使PBX通過中繼連線正確發出訊號。關閉trunk並配置路由器以使其在每一端都工作，這似乎是一個好主意，但這種方法實際上會適得其反，因為您更改的任何內容在建立trunk後都變得無關緊要。進行故障排除的最佳方法是使中繼啟動並正常工作。

開始排除故障

必須檢視基本資訊才能正確確定這些功能：

- 是否已建立中繼？發出**show voice call summary**命令，並確保中繼處於S_CONNECTED狀態。
- DSP是否正在處理資料包？發出**show voice dsp**命令以驗證這點。如果您沒有看到封包是由DSP處理的，這是因為VAD已啟用且會抑制封包。關閉VAD，重新建立中繼並重新檢視。此外，請檢查在發出**show call active voice brief**命令時資料包計數器是否增加。此命令還顯示是否為有問題的呼叫日誌啟用VAD。

如果中繼連線到任何站點上的模擬埠，最好檢驗PBX在非中繼模式下的運行。要排除模擬E&M連線故障，請參閱[瞭解和排除模擬E&M介面型別和佈線安排](#)。驗證所有內容並正常工作後，開啟中繼並檢視在PBX之間傳遞的信令。

解決語音中繼連線問題的理想方法是檢查PBX之間傳遞的信令。最好對每台有問題的路由器都建立Telnet作業階段，以便在從一端傳送到另一端時觀察訊號。本檔案使用E&M wink訊號傳送，因為此訊號非常流行，且必須考慮訊號傳送時間。

這是連線到發起呼叫的PBX的路由器的輸出：

```
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 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

此輸出顯示路由器終止呼叫。網路時間協定(NTP)已同步。

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

注意：此輸出顯示使用E&M wink訊號的語音中繼兩邊發生的訊號。可以看到使用這些相同調試的其他型別的信令。如果您看到呼叫建立正確（如此處所示），則必須存在雙向音訊。如果您檢視**show voice dsp**或**show call active voice brief**命令輸出，可以驗證這一點。如果一切正常，並且模擬連線出現音訊問題（無音訊或單向），請再次檢查這些連線。

確定哪些呼叫已啟動

由於檢視中繼呼叫的 `show call active voice` 或 `show voice call summary` 命令輸出幾乎不會產生任何效果，因此您需要一種簡單的方法來確定哪些語音中繼支援活動呼叫。最簡單的方法之一就是結合使用 `include` 引數發出 `show voice trunk-conditioning signaling` 命令，然後使用 `ABCD` 作為包含的字串，如下所示：

```
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=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
```

注意：此輸出顯示一個呼叫在時間段10上處於活動狀態，另一個呼叫在時間段8上啟動。如果您經常使用這個較長的命令，則希望為其建立別名。

DTMF故障排除

除了摘機和掛機信令外，路由器在PBX之間傳輸的唯一其他功能（除語音外）是DTMF音。還有音訊路徑，因此這通常不是問題，但確實有問題。您如何通過該路徑執行音訊時出現問題。有時最好使用低位元率編解碼器以節省頻寬。問題是這些低位元率編解碼器是通過為人類語音編寫的演算法設計的。除非客戶使用g711編解碼器，否則DTMF音調與這些演算法不太相符，需要一些其他方法來傳送。答案在 `dtmf-relay` 命令中。此功能允許DSP在最後、啟動音訊以識別DTMF音訊並將其與常規音訊流分離。然後，DSP基於其配置方式，將該音調編碼為不同型別的即時協定(RTP)資料包，或編碼為要在鏈路上與音訊流分開傳送的h245消息。這是 `fax-relay` 和 `modem-relay` 命令背後的相同進程。

此功能為中繼故障排除帶來了另一個調試問題。如果沒有呼叫設定，並且必須從路由器之間的資料包流提取該資訊，如何驗證傳遞了哪些數字？如何執行此操作取決於使用的 `dtmf-relay` 命令型別。

如本示例所示，`dtmf-relay cisco-rtp` 命令使用專有的Cisco負載型別，因此您必須向下檢視DSP才能看到此情況。您可以將 `debug vpm signal` 命令與 `debug vpm port x/x:y.z` 命令一起發出（將輸出限制到相關埠），以檢視傳遞給始發端DSP的數字。此輸出顯示在起始端，而不是終止端。

```
*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.

```

您可以使用[dtmf-relay h245-alphanumeric](#)命令驗證從始發端傳送哪些數字。[dtmf-relay h245-alphanumeric](#)命令使用h.245的字母數字部分來傳達色調。如本例所示，啟用[debug h245 asn1](#)命令後，可以在中繼的始發端和終止端很容易看到數字：

始發端：

```

*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

```

終端端：

```

*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 h245-signal](#)命令非常相似，當使用與[dtmf-relay h245-alphanumeric](#)命令相同的調試時可以看到。總之，如果不提到調試，使用[dtmf-relay](#)命令對連線中繼進行故障排除相當困難。

相關資訊

- [配置和故障排除透明CCS](#)
- [語音技術支援](#)
- [語音和IP通訊產品支援](#)
- [Cisco IP電話故障排除](#)
- [技術支援 - Cisco Systems](#)