

音声接続トランクのトラブルシューティング

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[問題](#)

[解決方法](#)

[接続トランクの一般的な問題](#)

[トラブルシューティングの開始](#)

[アップしているコールの判別](#)

[DTMF のトラブルシューティング](#)

[関連情報](#)

概要

音声接続トランクとは常に確立されている音声コールのことで、Voice over IP (VoIP)、Voice over Frame Relay (VoFR)、Voice over ATM (VoATM) のいずれかです。コールは、ルータがオンになって設定が完了すると同時に確立されます。音声ポートがアップにされると、音声ポートは自動的に音声ポートで指定されたダミーの電話番号に電話をかけ、その場所へコールを発信します。音声ポートは、対応するダイヤル ピアを通じて相手側へのコールを確立します。この接続が確立された後は、このルータに関する限り、音声コールはセッション中で接続されています。

前提条件

要件

このドキュメントに関しては個別の前提条件はありません。

使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このマニュアルの情報は、特定のラボ環境に置かれたデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。ネットワークが実稼働中である場合は、コマンドを使用する前に、コマンドによる潜在的な影響について理解しておく必要があります。

表記法

ドキュメント表記の詳細は、「[シスコテクニカルティップスの表記法](#)」を参照してください。

問題

トランクに関係する一般的な問題はルータで認識されず、トラブルシューティングが非常に困難です。音声トランクに関する一般的な問題は、トランクに対してコールを発信しても何も聞こえないという形で現れます。これは接続トランクによくある問題の1つで、多くの異なる問題が原因になっています。その他に、デュアルトーン多重周波数 (DTMF) トーンが正しく渡されない、構内交換機 (PBX) 間でシグナリングが正しく転送されないなどの問題があります。このドキュメントでは、このような問題のトラブルシューティングについて説明します。

音声トランクがアップして稼働状態になると、接続トランクでの信号の動作が変わります。そのため、音声ポートでシグナリングの特性を取得するために一般に発行されるコマンドは使用に適さず、役に立ちません。音声トランクはシグナリングのコンジットとなり、VoIP リンク全体に信号をリレーします。音声トランクを使用している場合、PBX のシグナリングはエンドツーエンドで一致する必要があります。それは、2 台の PBX マシンが接続されている限り、音声トランク接続を PBX に接続されている T1 専用回線と同一に見えるようにするためです。その際、ルータはプロセス全体で 2 台の PBX 間にクリアなリンクを確立しながら、完全に透過的になっています。

トランクはアップするとソフトウェア ケーブルになり、信号タイプがコネクタのタイプと見なされます。トランクは使用されている信号タイプについて関知せず、両端の信号が一致していなくてもアップしたままになります。両端の PBX が同じシグナリングを実行している限り、トランクは正常に動作します。

解決方法

接続トランク問題のトラブルシューティング方法は、スイッチド コールで使用される方法とは異なります。トランクを確認した後、実際に何が起きているかを突き止めるために PBX のシグナリングを調べる必要があります。シグナリングを調べる前に、トランクがアップしていて、デジタルシグナルプロセッサ (DSP) が音声パケットを処理していることを確認します。

注：トラブルシューティングを行うには、Voice Activity Detection (VAD；音声アクティビティ検出) を無効にする必要があります。トランクが正常に機能していることを確認したら、トラブルシューティングをさらに進めるためにテレフォニーのシグナリングを調べます。

トランクが確立されていて、だれもコールを発信していない場合、トランクのキープアライブメッセージがリモートのデバイス間で送受信されます。これらのキープアライブメッセージはトランクの接続を確認し、シグナリング情報をエンドツーエンドで伝送します。これらのキープアライブメッセージを確認するには、[debug vpm signal コマンドを発行します](#)。トランクが多くある場合、debug vpm コマンドの出力は、debug vpm port x コマンドオプションを発行すると、1つのポートに出力を制限できません。ここで、「x」は問題の音声ポートです。これは、すべてのポートを対象として debug vpm signal コマンドを発行した場合の出力です。

```
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 コマンドでポートを限定すると、この例に示すように、デバッグの解釈が容易になります。](#)

```
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
```

キープアライブ メッセージは 5 秒間隔で送受信されます。「sent to dsp」と「received from dsp」という表現は、Cisco IOS® の観点から見たものです。DSP を PBX に置き換えると、わかりやすくなります。これらは、トランクでアクティビティがないときに表示されるメッセージです。回線の両端のルータは、キープアライブ メッセージによってトランクがまだアップしていることを認識します。このメッセージが 5 回連続して失われた場合、トランクがダウンします。ネットワーク上のトランクが常にフラップする場合、これが原因の 1 つです。音声トランクのキープアライブが送受信されているかどうかを確認するには、**debug vpm trunk-sc** コマンドを発行します。この debug コマンドは、トランクのキープアライブが失われるまで出力を生成しません。これは、キープアライブが失われたときの **debug vpm trunk-sc** コマンドの出力例です。

```
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_KEEPAALIVE
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_KEEPAALIVE
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 コマンドを発行したときに出力が生成されなければ、キープアライブは失われていません。](#)キープアライブが失われても、5 回連続して失われない限りトランクはアップし続けます。つまり、25 秒間接続がダウンすると、トランクがダウンします。

[接続トランクの一般的な問題](#)

音声トランク接続に関連するバグがいくつかあります。異常が起こった場合は、これらのバグをチェックしてください。Cisco IOS ソフトウェア 12.2 がリリースされるまでに、これらの問題はほとんど解決し、統合されています。バグに目を通し、古いコードを使用している場合はこれらのバグが問題の原因であることを認識してください。最も一般的な問題の 1 つに、PBX がトランク接続経路で信号を正しく送信するにはどうすればよいかという問題があります。トランクをダウンさせて、両端のルータが動作するように設定すればよいように思われますが、トランクの接続が確立されると変更内容は無意味になるため、この方法では実際に望ましい結果は得られません。最善のトラブルシューティング方法は、トランクをアップさせたまま行うことです。

トラブルシューティングの開始

基本的な事柄を調べて、次のことが正しく動作していることを確認する必要があります。

- トランクが確立されているか `show voice call summary` コマンドを使用して、トランクが `S_CONNECTED`
- DSP がパケットを処理しているか `show voice dsp` コマンドを発行してこれを確認します。DSP によってパケットが処理されていない場合、VAD が有効になっていてパケットを抑制しているためです。VAD を無効にしてトランクを再確立し、もう一度確認してください。また、`show call active voice brief` コマンドを発行したとき、パケット カウンタが増えることを確認します。このコマンドでは、問題のあるコール ログで VAD が有効になっているかどうかも確認できます。

いずれかのサイトでトランクがアナログ ポートに接続している場合は、非トランク モードで PBX の動作を確認するのが最もよい方法です。アナログ E&M の接続問題をトラブルシューティングするには、『[アナログ E&M インターフェイスのタイプおよび配線の説明とトラブルシューティング](#)』を参照してください。すべてが正しく動作することを確認したら、トランクをアップして、PBX 間で渡されるシグナリングを調べます。

音声トランク接続問題に対する最もよいトラブルシューティング方法は、PBX 間で渡されるシグナリングを調べることです。一方のルータから他方のルータにシグナリングが渡される状況を監視するには、問題のある各ルータに対して Telnet で接続するのが最もよい方法です。このドキュメントでは、広く普及していることと、ウインクのタイミングを考慮する必要があるという理由から、E&M ウィンクシグナリングを使用しています。

これは、コールの発信側の PBX に接続されているルータからの出力です。

```
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
```


この出力は、ルータがコールを終端することを示します。Network Time Protocol (NTP; ネットワーク タイム プロトコル) が同期しています。

```
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:
```



```
!--- 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：この出力は、タイムスロット10でアクティブなコールと、タイムスロット8で開始された別のコールを示しています。このコマンドを頻繁に使用する場合は、この長いコマンドにエイリアスを割り当てると便利です。

DTMF のトラブルシューティング

オフフック信号やオンフック信号の他に、ルータが PBX との間で受け渡すのは（音声を除いて）、DTMF トーンだけです。音声パスもあるためこれは通常問題にはなりません、実際は問題が起きます。問題は、そのパスで音声をどのように送信するかに関係しています。帯域幅を節約するために、低いビットレートコーデックを使用した方がよい場合があります。しかしこれらの低いビットレートコーデックは、人間の会話向けに開発されたアルゴリズムを用いて設計されているため、問題が生じます。DTMF トーンはこれらのアルゴリズムに十分に適応できないため、お客様が G711 コーデックを使用しない場合は、他の送信方法が必要となります。これを解決するには、**dtmf-relay** コマンドを使用します。この機能によって、トーン発信側の DSP は DTMF トーンを認識し、通常の音声ストリームと区別できるようになります。設定方法に基づき、DSP はこのトーンを Real-Time Transport Protocol (RTP) パケットの異なるタイプか、H.245 メッセージのどちらかとして符号化し、リンク上を音声ストリームとは別に送信します。これは、fax-relay コマンドおよび modem-relay コマンドで行われるのと同じ処理です。

この機能を使用すると、トランクのトラブルシューティング時に他のデバッグ問題が生じます。コール設定がなく、ルータ間のパケットストリームから情報を取得しなければならない場合、どのディジットが渡されたかをどのように確認すればよいでしょうか。その確認方法は、使用されている dtmf-relay コマンドのタイプによって異なります。

この例に示すように、**dtmf-relay cisco-rtp コマンドはシスコ独自のペイロードタイプを使用するため、DSP のレベルでディジットを調べる必要があります。** debug vpm signal コマンドを debug vpm port x/x.y.z コマンドと組み合わせて発行すると（問題のポートへの出力を制限するため）、発信側の DSP に渡されたディジットを確認できます。この出力は、終端側ではなく、発信側に表示されます。

```
*Mar 1 00:22:39.592: htsp_digit_ready: digit = 31  
*Mar 1 00:22:39.592: [1/0:1(2), S_TRUNCED, 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_TRUNCED, 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_TRUNCED, 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_TRUNCED, 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_TRUNCED, 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_TRUNCED, 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_TRUNCED, 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_TRUNCED, 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_TRUNCED, 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.
```

発信側から送信されるディジットを確認するには、[dtmf-relay h245-alphanumeric](#)コマンドを使用します。[dtmf-relay h245-alphanumeric](#) コマンドは、H.245 の英数字部分を使用してトーンを送信します。この例に示すように、[debug h245 asn1](#) コマンドを有効にすると、送信されたディジットはトランクの発信側と終端側の両方で簡単に参照できます。

発信側：

```
*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
```

終端側：

```
*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](#)コマンドは非常に類似しており、[dtmf-relay h245-alphanumeric](#)コマンドと同じデバッグが使用されている場合に表示されます。一般に、[dtmf-relay](#) コマンドによる接続トランクのトラブルシューティングは、これまでに説明したデバッグ方法を使用しなければ、かなり難しくなります。

関連情報

- [透過型 CCSの設定とトラブルシューティング](#)
- [音声に関する技術サポート](#)

- [音声と IP 通信製品サポート](#)
- [Cisco IP Telephony のトラブルシューティング](#)
- [テクニカルサポート - Cisco Systems](#)