

Fehlerbehebung bei Sprachverbindungen-Trunks

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Problem](#)

[Lösung](#)

[Häufige Probleme bei Verbindungs-Trunks](#)

[Fehlerbehebung beginnen](#)

[Festlegen der eingehenden Anrufe](#)

[DTMF-Fehlerbehebung](#)

[Zugehörige Informationen](#)

Einführung

Die Sprachverbindungen-Trunks stellen permanent Sprachanrufe her, entweder Voice over IP (VoIP), Voice over Frame Relay (VoFR) oder Voice over ATM (VoATM). Die Anrufe werden eingerichtet, sobald der Router eingeschaltet und die Konfiguration abgeschlossen ist. Sobald die Sprach-Ports aktiviert sind, wählen die Sprach-Ports automatisch die Dummy-Telefonnummer, die unter dem Sprach-Port angegeben ist, und tätigen einen Anruf am Standort. Die Sprach-Ports führen den Anruf am anderen Ende über die entsprechenden DFÜ-Peers aus. Sobald diese Verbindung hergestellt ist, ist der Sprachanruf für den Router in einer Sitzung und verbunden.

Voraussetzungen

Anforderungen

Für dieses Dokument bestehen keine besonderen Voraussetzungen.

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardwareversionen beschränkt.

Die in diesem Dokument enthaltenen Informationen wurden aus Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen, bevor Sie ihn verwenden.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie in den [Cisco Technical Tips Conventions](#).

Problem

Die allgemeinen Probleme, die sich auf die Trunks beziehen, sind für den Router transparent und schwer zu beheben. Die allgemeinen Probleme, die bei den Sprach-Trunks auftreten, treten auf, wenn ein Anruf über die Trunks gestellt wird und nichts gehört wird. Dies ist eines der bekannten Probleme mit den Verbindungstüren und wird durch viele verschiedene Probleme verursacht. Ein weiteres Problem sind die DTMF-Töne (Dual Tone Multifrequency), die nicht ordnungsgemäß übertragen werden, und die Signalisierung von PBX (Private Branch Exchange) zu PBX werden nicht ordnungsgemäß übertragen. Dieses Dokument behebt diese Probleme.

Wenn die Sprach-LKWs eingeschaltet und aktiv sind, verhalten sich die Signale in den Verbindungs-LKWs anders. Alle Befehle, die Sie normalerweise unter dem Sprach-Port für Signalisierungseigenschaften ausgeben, sind nicht relevant und hilfreich. Der Sprach-Trunk wird zu einem Signalkanal und leitet das Signal über die VoIP-Verbindung weiter. Wenn Sie die Sprach-Trunks verwenden, muss die PBX-Signalisierung mit der End-to-End-Signalisierung übereinstimmen. Was die beiden PBX-Systeme betrifft, so soll die Sprach-Trunk-Verbindung mit einer gemieteten T1-Leitung identisch sein mit der PBX-Anlage, wobei die Router vollständig transparent sind, während zwischen den beiden PBX-Systemen im gesamten Prozess eine klare Verbindung besteht.

Wenn der Trunk hochgefahren wird, wird der Trunk zu einem Softwarekabel, und der Signaltyp wird als Verbindungstyp betrachtet. Der verwendete Signaltyp ist für den Trunk nicht von Bedeutung. Der Trunk wird immer noch angezeigt, auch wenn das Signal an beiden Enden nicht übereinstimmt. Solange die PBX-Systeme an beiden Enden dieselbe Signalisierung ausführen, funktionieren die Trunks ordnungsgemäß.

Lösung

Der Ansatz, der bei der Behebung von Verbindungsproblemen zu wählen ist, unterscheidet sich von dem bei Switched Calls. Um zu sehen, was wirklich geschieht, nachdem die Trunks verifiziert wurden, müssen Sie die PBX-Signalisierung verwenden. Bevor Sie mit der Signalisierung fortfahren, überprüfen Sie, ob die Trunks betriebsbereit sind und ob die Digital Signal Processors (DSPs) die Sprachpakete verarbeiten.

Hinweis: Zur Fehlerbehebung sollten Sie möglicherweise die Sprachaktivitätserkennung (VAD) deaktivieren. Wenn sichergestellt ist, dass die Trunks korrekt funktionieren, müssen Sie die Telefonie-Signalisierung prüfen, um eine weitere Fehlerbehebung durchführen zu können.

Wenn die Trunks eingerichtet sind und niemand versucht, einen Anruf zu tätigen, werden die Trunk-Keepalive-Nachrichten zwischen den Remote-Feldern hin und her gesendet. Diese Keepalives überprüfen die Trunk-Verbindung und übertragen die Signalisierungsinformationen von Ende zu Ende. Um diese Keepalives zu überprüfen, führen Sie den [Befehl `debug vpm signal aus`](#). Wenn es viele Trunks gibt, die Ausgabe von `debug vpm`-Befehlen, können Sie die Ausgabe auf einen einzelnen Port begrenzen, wenn Sie die Befehloption `debug vpm port x` ausgeben, wobei "x" der betreffende Sprach-Port ist. Dies ist die Ausgabe des Befehls `debug vpm`, der beim Betrachten aller Ports ausgegeben wird:

```
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
```

Wenn Sie dies begrenzen, kann das Debuggen mit dem Befehl [debug vpm port x](#) einfacher interpretiert werden, wie in diesem Beispiel gezeigt:

```
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
```

Die Keepalives werden alle fünf Sekunden gesendet und empfangen. Die Begriffe "an den DSP gesendet" und "vom DSP empfangen" stammen aus der Cisco IOS[®]-Sicht. Ersetzen Sie PBX durch DSP, um es verständlicher zu machen. Dies sind die Meldungen, die angezeigt werden, während keine Aktivität auf den Trunks verzeichnet wird. Die Keepalive-Nachrichten informieren die Router an jedem Ende des Schaltkreises darüber, dass die Trunks immer noch aktiv sind. Wenn fünf dieser Nachrichten nacheinander verpasst werden, wird der Trunk unterbrochen. Eine der Ursachen ist, dass die Trunks ständig in einem Netzwerk flapping. Führen Sie den Befehl **debug vpm trunk-sc** aus, um zu überprüfen, ob die Sprachtrunk-Keepalives gesendet und empfangen werden. Dieses Debuggen generiert erst dann eine Ausgabe, wenn die Trunk-Keepalives verpasst wurden. Dies ist ein Beispiel für die Ausgabe des Befehls **debug vpm trunk-sc**, wenn Keepalives verpasst werden:

```
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
```

Wenn beim [Befehl debug vpm trunk-sc](#) keine Ausgabe angezeigt wird, werden keine Keepalives verpasst. Auch wenn Keepalives verpasst werden, bleibt der Trunk so lange erhalten, bis fünf sequenzielle Nachrichten verpasst werden. Das bedeutet, dass eine Verbindung 25 Sekunden lang unterbrochen werden muss, bevor die Trunks ausfallen.

[Häufige Probleme bei Verbindungs-Trunks](#)

Es gibt mehrere Fehler, die mit den Sprach-Trunk-Verbindungen verbunden sind. Überprüfen Sie

diese Fehler, wenn Sie etwas Ungewöhnliches sehen. Zum Zeitpunkt der Veröffentlichung der Cisco IOS Software 12.2 waren die meisten dieser Probleme behoben und integriert. Sie können die Fehler durchsehen, um sich bewusst zu machen, dass dies Ursachen für Probleme mit älterem Code sind. Eines der häufigsten Probleme besteht darin, die PBX-Systeme dazu zu bringen, über die Trunk-Verbindung ordnungsgemäß zu signalisieren. Es scheint eine gute Idee zu sein, die Trunks herunterzufahren und die Router so zu konfigurieren, dass sie an jedem Ende funktionieren. Der Ansatz ist jedoch wirklich kontraproduktiv, da alles, was Sie geändert haben, nach der Einrichtung der Trunks unbedeutend wird. Die beste Methode zur Fehlerbehebung besteht darin, die Trunks betriebsbereit zu machen.

Fehlerbehebung beginnen

Es ist notwendig, die Grundlagen zu prüfen, um sicherzustellen, dass diese Funktion richtig:

- Sind die Trunks eingerichtet? Geben Sie den Befehl **show voice call summary** ein, und stellen Sie sicher, dass sich die Trunks im `S_CONNECTED`-Status befinden.
- Verarbeiten die DSPs Pakete? Geben Sie den Befehl **show voice dsp** ein, um dies zu überprüfen. Wenn Pakete nicht von den DSPs verarbeitet werden, liegt dies daran, dass VAD aktiviert ist und die Pakete unterdrückt. Deaktivieren Sie VAD, stellen Sie Trunks wieder her, und sehen Sie erneut nach. Überprüfen Sie außerdem, ob die Paketzähler inkrementiert werden, wenn der Befehl **show call active voice brief** ausgegeben wird. Dieser Befehl zeigt auch an, ob VAD für das betreffende Anrufprotokoll aktiviert ist.

Wenn die Trunks an einem Standort mit analogen Ports verbunden sind, empfiehlt es sich, den Betrieb des PBX-Systems im Nicht-Trunk-Modus zu überprüfen. Informationen zur Behebung analoger E&M-Verbindungsprobleme finden Sie unter [Verstehen und Beheben von analogen E&M-Schnittstellentypen und -Verkabelungseinstellungen](#). Wenn alle Elemente überprüft wurden und ordnungsgemäß funktionieren, stellen Sie die Trunks wieder her, und überprüfen Sie die Signalisierung, die zwischen den PBX-Systemen weitergeleitet wird.

Die ideale Lösung zur Behebung von Verbindungsproblemen im Sprach-Trunk ist die Überprüfung der Signalisierung, die zwischen den PBX-Systemen weitergeleitet wird. Es ist am besten, eine Telnet-Sitzung mit jedem betroffenen Router einzurichten, damit die Signalisierung beobachtet werden kann, wenn sie von einem Ende an das andere übertragen wird. In diesem Dokument wird E&M-Wink-Signalisierung verwendet, da diese sehr beliebt ist und das Wink-Timing berücksichtigt werden muss.

Dies ist die Ausgabe des Routers, der mit dem PBX-System verbunden ist und den Anruf auslöst:

```
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
```


19:40:19.696: [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 19:40:19.816: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.836: [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 0xF May 22 19:40:19.876: [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 0xF May 22 19:40:19.916: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.936: [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 19:40:19.876: [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 19:40:19.916: [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

Diese Ausgabe zeigt an, dass der Router den Anruf beendet. Network Time Protocol (NTP) wird synchronisiert.

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

```

Hinweis: Diese Ausgabe zeigt die Signalisierung an, die auf beiden Seiten eines Sprach-Trunks erfolgt, der E&M-Wink-Signalisierung verwendet. Es sind andere Signalisierungstypen sichtbar, die dieselben Debuggen verwenden. Wenn Anrufe richtig eingerichtet werden (wie hier gezeigt), muss Zweiwege-Audio vorhanden sein. Dies kann überprüft werden, wenn entweder der Befehl **show voice dsp** oder der Befehl **show call active voice brief** angezeigt wird. Wenn dort alles in Ordnung zu sein scheint und Sie Audioprobleme (ohne Audio oder unidirektionale Verbindung) mit analogen Verbindungen bekommen, überprüfen Sie diese Verbindungen erneut.

[Festlegen der eingehenden Anrufe](#)

Da es kaum oder nicht gut ist, die Befehlsausgabe **show call active voice** oder **voice call summary** für Trunked Calls zu betrachten, benötigen Sie eine einfache Methode, um festzustellen, welche

Sprach-Trunks aktive Anrufe unterstützen. Eine der einfachsten Möglichkeiten hierfür besteht darin, den Befehl **show voice trunk Conditioning signaling** in Verbindung mit dem **include-Parameter** auszugeben und *ABCD als enthaltene Zeichenfolge zu verwenden, wie hier gezeigt:*

```
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
```

Hinweis: Diese Ausgabe zeigt einen aktiven Anruf auf Timeslot 10 und einen weiteren Anruf an, der auf Timeslot acht gestartet wurde. Sie möchten einen Alias für diesen ziemlich langen Befehl erstellen, wenn Sie ihn häufig verwenden.

DTMF-Fehlerbehebung

Neben der Signalisierung bei abgehobenem Hörer und aufgelegtem Hörer übergeben die Router zwischen den PBX-Systemen (außer Sprachübertragung) nur DTMF-Töne. Es gibt auch einen Audiopfad, sodass dies normalerweise kein Problem ist, aber ein Problem besteht. Das Problem besteht darin, wie Sie Audio über diesen Pfad ausführen. Manchmal ist es vorzuziehen, Codecs mit niedriger Bitrate zu verwenden, um Bandbreite zu sparen. Das Problem ist, dass diese Codecs mit niedriger Bitrate mithilfe von Algorithmen entwickelt wurden, die für die menschliche Sprache geschrieben wurden. DTMF-Töne entsprechen diesen Algorithmen nicht sehr gut und erfordern eine andere Methode, um sie zu übertragen, es sei denn, der Kunde verwendet den g711-Codec. Die Antwort liegt im Befehl **dtmf-relais**. Diese Funktion ermöglicht es den DSPs am Ende, startet den Ton, erkennt den DTMF-Ton und trennt ihn vom normalen Audio-Stream. Je nach Konfiguration kodiert der DSP diesen Ton dann entweder als ein anderes Real Time Protocol (RTP)-Paket oder als eine h245-Nachricht, die separat vom Audio-Stream über die Verbindung gesendet wird. Dies ist der gleiche Prozess hinter den **Fax-Relay-** und **Modem-Relay-**Befehlen.

Diese Funktion stellt ein weiteres Debugproblem bei der Trunk-Fehlerbehebung dar. Wie überprüfen Sie, welche Ziffern übergeben werden, wenn keine Anruferichtung vorhanden ist und Sie diese Informationen aus dem Paket-Stream zwischen den Routern extrahieren müssen? Wie das funktioniert, hängt davon ab, welche Art von **dtmf-relais-**Befehl verwendet wird.

Wie in diesem Beispiel gezeigt, verwendet der [Befehl dtmf-relais cisco-rtp](#) einen proprietären Payload-Typ von Cisco. Um dies zu sehen, müssen Sie sich die DSPs ansehen. Sie können den Befehl **debug vpm signal** zusammen mit dem Befehl **debug vpm port x/x.y.z** (um die Ausgabe auf den betreffenden Port zu begrenzen) ausgeben, um die an die DSPs übergebenen Ziffern an der ursprünglichen Seite anzuzeigen. Diese Ausgabe wird auf der Ausgangsseite und nicht auf der Abschlussseite angezeigt.

```
*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_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.

```

Mit dem Befehl [dtmf-relais h245-alphanumerisch](#) können Sie überprüfen, welche Ziffern von der ursprünglichen Seite gesendet werden. Der Befehl [dtmf-relais h245-alphanumerisch](#) verwendet den alphanumerischen Teil von h.245, um die Töne zu übertragen. Wie in diesem Beispiel gezeigt, können die Ziffern sowohl auf der Ausgangs- als auch auf der terminierenden Seite des Trunks leicht sichtbar sein, wenn der Befehl [debug h245 asn1](#) aktiviert ist:

Ursprungsseite:

```

*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

```

Abschlussseite:

```

*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"
```

Der [Befehl dtmf-relais h245-signal](#) ist sehr ähnlich und kann angezeigt werden, wenn die gleichen [Debug-Befehle wie der Befehl dtmf-relais h245-alphanumerisch](#) verwendet werden. Insgesamt ist die Fehlerbehebung für die Verbindungs-Trunks mit dem Befehl **dtmf-relais** ohne die genannten Debuggen recht schwierig.

[Zugehörige Informationen](#)

- [Konfiguration und Fehlerbehebung in transparentem CCS](#)
- [Unterstützung von Sprachtechnologie](#)
- [Produkt-Support für Sprach- und IP-Kommunikation](#)
- [Fehlerbehebung bei Cisco IP-Telefonie](#)
- [Technischer Support - Cisco Systems](#)