

# Resolución de problemas de enlaces troncales de conexión de voz.

## Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Problema](#)

[Solución](#)

[Problemas comunes para troncales de conexión](#)

[Comience a resolver problemas](#)

[Determine qué llamadas están para arriba](#)

[Troubleshooting DTMF](#)

[Información Relacionada](#)

## [Introducción](#)

Los links troncales de conexión de voz establecen permanentemente las llamadas de voz, la voz sobre IP (VoIP), voz sobre Frame Relay (VoFR), o Voz por ATM (VoATM). Las llamadas se establecen tan pronto como se activa el router y se completa la configuración. Tan pronto como los puertos de voz se den vuelta para arriba, los puertos de voz marcan automáticamente el número de teléfono simulado especificado bajo el puerto de voz y ponen una llamada a la ubicación. Los puertos de voz completan la llamada al otro extremo a través de los dial peer correspondientes. Una vez que se establece esta conexión, por lo que al router, la llamada de voz está en la sesión y está conectada.

## [prerrequisitos](#)

### [Requisitos](#)

No hay requisitos previos específicos para este documento.

### [Componentes Utilizados](#)

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

La información que se presenta en este documento se originó a partir de dispositivos dentro de un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se

pusieron en funcionamiento con una configuración verificada (predeterminada). Si su red está vivo, asegúrese que usted entiende el impacto potencial del comando any antes de que usted lo utilice.

## [Convenciones](#)

Para obtener más información sobre las convenciones del documento, consulte las [Convenciones de Consejos Técnicos de Cisco](#).

## [Problema](#)

Los problemas comunes que pertenecen a los trunks son transparentes al router y muy difíciles de resolver problemas. Los problemas frecuentes considerados con los trunks de la Voz se manifiestan cuando una llamada se pone sobre los trunks y no se oye nada. Éste es uno de los problemas conocidos con los troncos de conexión y es causado por muchos diversos problemas. El Otro problema es los tonos de múltiples frecuencias del tono dual (DTMF) que no se pasan correctamente, y la señalización de la Central telefónica privada (PBX) al PBX no se transporta correctamente. Este documento sirve para solucionar estos problemas.

Cuando los camiones de la Voz son ascendentes y activos, las señales se comportan diferentemente en los camiones de la conexión. Los comandos any que usted publica normalmente bajo el puerto de voz para las características de señalización no son relevantes y útiles. El trunk de la Voz se convierte en un conducto de la señalización y retransmite la señal a través del link VoIP. Cuando usted utiliza los trunks de la Voz, señalización PBX debe hacer juego de punta a punta. Por lo que las dos máquinas PBX, la meta es hacer la mirada de la conexión de tronco de la Voz idéntica a una línea arrendada T1 al PBX, con el Routers totalmente transparente mientras que un vínculo claro se establece entre los dos PBX en el proceso completo.

Cuando sube el trunk, el trunk se convierte en un cable del software y consideran al tipo de señal un Tipo de conector. El trunk no cuida sobre el tipo de señal se utiliza que. El trunk todavía sube incluso si la señal no hace juego en los ambos extremos. Mientras los PBX en los ambos extremos hagan la misma señalización, los trunks funcionan correctamente.

## [Solución](#)

El acercamiento a tomar cuando usted resuelve problemas los problemas del tronco de conexión es diferente que eso se utiliza para las llamadas conmutadas. Para ver qué sucede realmente después de que se verifiquen los trunks, usted necesita mirar al señalización PBX. Antes de que usted proceda a mirar la señalización, verifique que los trunks sean ascendentes y que los procesadores de señales digitales (DSPs) procesan los paquetes de voz.

**Nota:** Usted quiere probablemente inhabilitar la detección de actividad de la Voz (VAD) para resolver problemas. Una vez que se verifica que funcionan los trunks correctamente, usted necesita mirar la Señalización de telefonía para resolver problemas más lejos.

Si se establecen los trunks, y nadie intenta hacer una llamada, los mensajes del keepalive del tronco se envían hacia adelante y hacia atrás entre los cuadros remotos. Este Keepalives verifica la conectividad del tronco y lleva la información de señalización de punta a punta. Para verificar este Keepalives, publique el [comando debug vpm signal](#). Si hay muchos trunks, la salida de los

comandos debug vpm, usted puede limitar la salida a un puerto único si usted aplicación la opción del comando **debug vpm port x**, donde está el puerto de voz “x” en la pregunta. Ésta es la salida del comando **debug vpm signal** publicado cuando usted mira todos los puertos:

```
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
```

Si usted limita esto, con el [comando debug vpm port x](#), los debugs mucho más fáciles interpretar, tal y como se muestra en de este ejemplo:

```
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
```

El Keepalives se envía y recibió cada cinco segundos. Los términos “enviaron al dsp” y “recibido del dsp” sea desde el punto de vista del <sup>®</sup> del Cisco IOS. Substituya el PBX para que el DSP lo haga más comprensible. Éstos son los mensajes se consideran que mientras que no hay actividad en los trunks. Los mensajes de la señal de mantenimiento dejan que los routers en cada extremo del circuito sepan que los troncos aún están activos. Cuando cinco de estos mensajes se faltan en fila, el trunk va abajo. Una de las causas es si los trunks agitan constantemente en una red. Para verificar si los keepalives del tronco de la Voz estén enviados y recibidos, publique el **comando debug vpm trunk-sc**. Este debug no genera ninguna salida hasta que se falten los keepalives del tronco. Éste es un ejemplo del **comando debug vpm trunk-sc** hecho salir cuando se falta el Keepalives:

```
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
```

Si no se considera ninguna salida cuando publican el [comando debug vpm trunk-sc](#), después no se falta ningún Keepalives. Incluso si se falta el Keepalives, el trunk permanece para arriba hasta que se falten cinco mensajes secuenciales. Esto significa que una conexión debe estar abajo para 25 segundos antes de que van los trunks abajo.

## [Problemas comunes para troncales de conexión](#)

Hay varios bug asociados a las conexiones de tronco de la Voz. Marque estos bug si usted ve

cualquier cosa inusual. Para el momento en que el Cisco IOS Software 12.2 fuera liberado, la mayor parte de estos problemas habían sido abordados e integrados. Usted puede mirar a través de los bug para hacerse enterado que éstas son causas de los problemas con un más viejo código. Uno de la mayoría de los problemas frecuentes es conseguir los PBX para señalar correctamente sobre la conexión de tronco. Parece como una buena idea de derribar los trunks y de configurar al Routers de modo que él trabaje en cada extremo, pero el acercamiento es realmente contraproducente puesto que cualquier cosa que usted ahora cambió llega a ser discutible los trunks se establece una vez. La mejor manera de resolver problemas está con los trunks ascendentes y funcionales.

## Comience a resolver problemas

Es necesario mirar los fundamentos para establecer que funcionan éstos correctamente:

- ¿Están establecidas las troncales? Publique el **comando show voice call summary**, y asegurese que los trunks están en `S_CONNECTED` el estado.
- ¿Los DSP están procesando paquetes? Publique el **comando show voice dsp** de verificar esto. Si usted no ve que los paquetes son procesados por el DSPs, es porque se habilita el VAD y es suprime los paquetes. Apague el VAD, vuelva a establecer los troncos y mire otra vez. También, control que los contadores de paquetes incrementan cuando el publican **comando show call active voice brief**. Este comando también muestra si el VAD está habilitado para la pregunta del login de la llamada.

Si los trunks conectan con los puertos analógicos en cualquier sitio, es el mejor verificar la operación del PBX en el modo NON-trunked. Para resolver problemas los problemas de conectividad analógicos E&M, refiera a [entender y a resolver problemas los tipos de interfaz analógicos y las disposiciones de cableado E&M](#). Una vez que todo se verifica y funciona correctamente, traiga los trunks para arriba y mire la señalización que se pasa entre los PBX.

La forma ideal de resolver problemas los problemas de conexión de tronco de la Voz es examinar la señalización que se pasa entre los PBX. Es el mejor tener una sesión telnet a cada router en la pregunta para poder observar la señalización como ella esté pasada a partir de un extremo al otro. Este documento utiliza la señalización del guiño E&M puesto que es bastante popular y la sincronización de Wink tiene que ser tomada en la consideración.

Ésta es la salida del router conectado con el PBX que origina la llamada:

```
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
```





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 19:40:19.916: [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

Esta salida muestra que el router termina la llamada. El Protocolo de tiempo de red (NTP) está sincronizado.

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

**Nota:** Esta salida muestra la señalización que ocurre a ambos lados de un trunk de la Voz qué señalización del guiño de las aplicaciones E&M. Otros tipos de señalización pueden ser considerados que utilice estos mismos debugs. Si usted ve las llamadas establecidas correctamente (como se muestra aquí), el audio de dos vías debe estar presente. Esto puede ser verificada si usted mira el **DSP de voz de la demostración** o la salida del comando **show call active voice brief**. Si todo mira para ser fino allí, y usted está consiguiendo los problemas de audio (no audio o unidireccional) con las conexiones analógicas, marque estas conexiones otra vez.

## [Determine qué llamadas están para arriba](#)

Puesto que hace poco o nada de bueno mirar el comando **show call active voice** o **show voice call summary** hecho salir para las llamadas de tronco, usted necesita un método simple determinar que expresan las llamadas activas del soporte de los trunks. Una de las maneras más fáciles de hacer esto es publicar el comando **show voice trunk-conditioning signaling** conjuntamente con el parámetro del *incluido* y utilizar el *ABCD* como la cadena incluida, tal y como se muestra en de aquí:

```
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=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
```

**Nota:** Esta salida muestra un active de la llamada en el intervalo de tiempo 10 y otra llamada que es comenzado en el intervalo de tiempo ocho. Usted quiere hacer un alias para este comando bastante largo si usted lo utiliza mucho.

## Troubleshooting DTMF

Aparte del descolgado y la señalización activada, el únicos la otra cosa que el Routers pasa entre los PBX (además de la Voz) son tonos DTMF. Hay también un trayecto de audio así que esto no es generalmente un problema, pero hay un problema. El problema se presenta con cómo usted hace el audio sobre esa trayectoria. Es a veces preferible utilizar el codecs del Low Bit Rate para salvar el ancho de banda. El problema sube que este codecs del Low Bit Rate está diseñado mediante los algoritmos que fueron escritos para lo que dice una persona. Los tonos DTMF no se ajustan a estos algoritmos muy bien y necesitan un cierto otro método transportar a menos que el cliente utilice el codificador-decodificador g711. La respuesta miente en el **comando dtmf-relay**. Esta característica permite el DSPs en el extremo, comienza el tono, para reconocer el tono DTMF y para separarlo de la secuencia de audio regular. Basado sobre cómo se configura, el DSP entonces cifra este tono como un tipo diferente de paquete del Real-Time Protocol (RTP) o como mensaje h245 que se enviará a través del link por separado de la secuencia de audio. Este es el mismo proceso detrás de los comandos fax-relay y modem-relay.

Esta característica plantea otro problema del debug para el Troubleshooting troncal. ¿Cómo usted verifica lo que se pasan los dígitos si no hay configuración de la llamada y usted tiene que extraer esa información de la secuencia de paquetes entre el Routers? Cómo hacer esto depende emplean de qué tipo de **comando dtmf-relay**.

Tal y como se muestra en de este ejemplo, el [comando dtmf-relay cisco-rtp](#), utiliza un tipo de carga útil propietario de Cisco, así que usted debe mirar abajo el DSPs para ver esto. Usted puede publicar el **comando debug vpm signal** conjuntamente con el **debug vpm port x/x: comando y.z** (limitar la salida al puerto en la pregunta) de ver los dígitos pasajeros al DSPs en el lado de origen. Esta salida se visualiza en el lado de origen, no en el lado de finalización.

```
*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.

```

Usted puede verificar qué dígitos se envían del lado de origen con el [comando dtmf-relay h245-alphanumeric](#). El comando `dtmf-relay h245-alphanumeric` utiliza la porción alfanumérica de h.245 para transportar los tonos. Tal y como se muestra en de este ejemplo, los dígitos se pueden considerar fácilmente en originar y los lados de finalización del trunk cuando habilitan al `comando debug h245 asn1`:

### Lado de origen:

```

*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

```

### Lado de finalización:

```

*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"

```

[El comando dtmf-relay h245-signal](#) es muy similar y puede ser considerado cuando se utilizan lo mismo los debugs que el `comando dtmf-relay h245-alphanumeric`. Total, resolver problemas los troncos de conexión con el `comando dtmf-relay` es bastante difícil sin los debugs mencionados.

## [Información Relacionada](#)

- [Configuración y resolución de problemas de CCS transparente](#)

- [Soporte de tecnología de voz](#)
- [Soporte para productos de comunicaciones IP y por voz](#)
- [Troubleshooting de Cisco IP Telephony](#)
- [Soporte Técnico - Cisco Systems](#)