

Resolución de problemas de enlaces troncales de conexión de voz.

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Problema](#)

[Solución](#)

[Problemas comunes para troncales de conexión](#)

[Comenzar a solucionar problemas](#)

[Determinar qué llamadas están activas](#)

[Resolución de problemas de DTMF](#)

[Información Relacionada](#)

[Introducción](#)

Los enlaces de conexión de voz establecen de forma permanente llamadas de voz, ya sea voz sobre IP (VoIP), voz sobre Frame Relay (VoFR) o voz sobre ATM (VoATM). Las llamadas se establecen tan pronto como se activa el router y se completa la configuración. Tan pronto como se encienden los puertos de voz, los puertos de voz marcan automáticamente el número de teléfono falso especificado en el puerto de voz y realizan una llamada a la ubicación. Los puertos de voz completan la llamada al otro extremo a través de los pares de marcado correspondientes. Una vez establecida esta conexión, en lo que respecta al router, la llamada de voz está en sesión y conectada.

[Prerequisites](#)

[Requirements](#)

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. All of the devices used in this document started with a cleared

(default) configuration. Si su red está activa, asegúrese de comprender el impacto potencial de cualquier comando antes de utilizarlo.

[Convenciones](#)

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

[Problema](#)

Los problemas comunes que pertenecen a los troncales son transparentes para el router y muy difíciles de resolver. Los problemas comunes observados con los troncales de voz se manifiestan cuando se coloca una llamada sobre los troncales y no se oye nada. Este es uno de los problemas conocidos con los troncales de conexión y está causado por muchos problemas diferentes. Otro problema son los tonos de multifrecuencia de tono dual (DTMF) que no se pasan correctamente y la señalización de la centralita privada (PBX) a la PBX no se transporta correctamente. Este documento sirve para solucionar estos problemas.

Cuando los troncales de voz están activos, las señales se comportan de manera diferente en los troncales de conexión. Cualquier comando que ejecute normalmente en el puerto de voz para las características de señalización no es relevante ni útil. El troncal de voz se convierte en un conducto de señalización y retransmite la señal a través del link VoIP. Cuando utiliza los troncales de voz, la señalización PBX debe coincidir de extremo a extremo. En lo que respecta a las dos máquinas PBX, el objetivo es hacer que la conexión troncal de voz se vea idéntica a una línea T1 arrendada al PBX, con routers completamente transparentes mientras se establece un link claro entre los dos PBX en todo el proceso.

Cuando se activa el tronco, éste se convierte en un cable de software y el tipo de señal se considera un tipo de conector. Al tronco no le importa el tipo de señal que se utiliza. El tronco sigue apareciendo incluso si la señal no coincide en ambos extremos. Mientras los PBX en ambos extremos hagan la misma señalización, los troncales funcionan correctamente.

[Solución](#)

El enfoque que se debe seguir cuando se resuelven los problemas de conexión troncal es diferente del que se utiliza para las llamadas conmutadas. Para ver lo que realmente sucede después de que se verifican los troncales, debe buscar la señalización PBX. Antes de proceder a examinar la señalización, verifique que los troncales estén activos y que los Procesadores de señal digital (DSP) procesen los paquetes de voz.

Nota: Probablemente desee desactivar la detección de actividad de voz (VAD) para resolver problemas. Una vez verificado que los troncales funcionan correctamente, debe observar la señalización de telefonía para resolver problemas adicionales.

Si se establecen los troncales y nadie intenta realizar una llamada, los mensajes de keepalive del tronco se envían de ida y vuelta entre las casillas remotas. Estas señales de mantenimiento verifican la conectividad del tronco y llevan la información de señalización de extremo a extremo. Para verificar estas señales de mantenimiento, ejecute el comando [debug vpm signal](#). Si hay muchos trunks, el resultado de los comandos `debug vpm`, puede limitar el resultado a un solo puerto si ejecuta la opción de comando `debug vpm port x`, donde "x" es el puerto de voz en cuestión. Este es el resultado del comando `debug vpm signal` emitido cuando se observa 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](#), las depuraciones son mucho más fáciles de interpretar, como se muestra en 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
```

Las señales de mantenimiento se envían y reciben cada cinco segundos. Los términos "enviado a dsp" y "recibido desde dsp" provienen del punto de vista de Cisco IOS®. Sustituya el PBX por el DSP para hacerlo más comprensible. Estos son los mensajes que se ven mientras no hay actividad en los troncales. Los mensajes de la señal de mantenimiento dejan que los routers en cada extremo del circuito sepan que los troncos aún están activos. Cuando se pierden cinco de estos mensajes en una fila, el tronco se desactiva. Una de las causas es si los troncales se inundan constantemente en una red. Para verificar si se envían y reciben las señales de mantenimiento del tronco de voz, ejecute el comando **debug vpm trunk-sc**. Esta depuración no genera ningún resultado hasta que se pierden las señales de mantenimiento del tronco. Este es un ejemplo de la salida del comando **debug vpm trunk-sc** cuando se omiten 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 ve ningún resultado cuando se ejecuta el comando [debug vpm trunk-sc, no se pierden señales de mantenimiento](#). Incluso si se pierden señales de mantenimiento, el tronco permanece activo hasta que se pierden cinco mensajes secuenciales. Esto significa que una conexión debe estar inactiva durante 25 segundos antes de que los troncales se desactiven.

[Problemas comunes para troncales de conexión](#)

Hay varios errores asociados con las conexiones del tronco de voz. Revise estos errores si ve algo inusual. Cuando se lanzó Cisco IOS Software 12.2, la mayoría de estos problemas se habían abordado e integrado. Puede examinar los errores de funcionamiento para darse cuenta de que son causas de problemas con el código antiguo. Uno de los problemas más comunes es hacer que los PBX señalen correctamente a través de la conexión troncal. Parece una buena idea derribar los troncales y configurar los routers para que funcionen en cada extremo, pero el enfoque es realmente contraproducente ya que cualquier cosa que haya cambiado ahora se vuelve discutible una vez que se establecen los troncales. La mejor manera de resolver problemas es con los troncales activos y funcionales.

Comenzar a solucionar problemas

Es necesario analizar los fundamentos para establecer que estas funciones funcionan correctamente:

- ¿Están establecidas las troncales? Ejecute el comando **show voice call summary** y asegúrese de que los troncales estén en el estado `S_CONNECTED`.
- ¿Los DSP están procesando paquetes? Ejecute el comando **show voice dsp** para verificar esto. Si no ve que los DSP procesan los paquetes, es porque VAD está habilitado y suprime los paquetes. Apague el VAD, vuelva a establecer los troncos y mire otra vez. Además, verifique que los contadores de paquetes aumenten cuando se ejecuta el comando **show call active voice brief**. Este comando también muestra si VAD está habilitado para el registro de llamadas en cuestión.

Si los troncales se conectan a puertos analógicos en cualquier sitio, es mejor verificar el funcionamiento del PBX en modo no troncal. Para resolver problemas de conectividad E/M analógica, consulte [Comprensión y Troubleshooting de Tipos de Interfaz E/M Analógicos y Arreglos de Cableado](#). Una vez que todo se verifica y funciona correctamente, active los troncales y observe la señalización que se pasa entre los PBX.

La manera ideal de resolver problemas de conexión troncal de voz es examinar la señalización que se pasa entre los PBX. Es mejor tener una sesión Telnet a cada router en cuestión para que la señalización pueda observarse a medida que se pasa de un extremo al otro. Este documento utiliza señalización de guiño E/M, ya que es bastante popular y se debe tener en cuenta el tiempo de guiño.

Este es el resultado del router conectado al 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
```


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

Este resultado muestra que el router finaliza 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 en ambos lados de un tronco de voz que utiliza señalización de guiño E/M. Se pueden ver otros tipos de señalización que utilizan estas mismas depuraciones. Si ve llamadas establecidas correctamente (como se muestra aquí), debe haber audio bidireccional. Esto se puede verificar si observa el resultado del comando **show voice dsp** o el comando **show call active voice brief**. Si todo parece estar bien allí y está teniendo problemas de audio (sin audio o unidireccional) con conexiones analógicas, compruebe estas conexiones de nuevo.

[Determinar qué llamadas están activas](#)

Dado que es poco o nada bueno observar la salida del comando **show call active voice** o **show voice call summary** para las llamadas troncales, necesita un método simple para determinar qué

truncos de voz admiten las llamadas activas. Una de las maneras más sencillas de hacerlo es ejecutar el comando **show voice trunk-conditioning** junto con el *parámetro include* y utilizar *ABCD* como la cadena incluida, como se muestra 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=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 una llamada activa en el intervalo de tiempo 10 y otra llamada que se está iniciando en el intervalo de tiempo ocho. Si lo utiliza mucho, desea crear un alias para este comando bastante largo.

Resolución de problemas de DTMF

Aparte de la señalización de descolgado y colgado, la única otra cosa que los routers pasan entre los PBX (además de la voz) son los tonos DTMF. También hay una ruta de audio, por lo que esto no suele ser un problema, pero hay un problema. El problema surge cuando se hace audio a través de esa ruta. A veces es preferible utilizar los códecs de baja velocidad de bits para ahorrar ancho de banda. Se plantea el problema de que estos códecs de baja velocidad de bits están diseñados mediante algoritmos escritos para el habla humana. Los tonos DTMF no se ajustan muy bien a estos algoritmos y necesitan algún otro método para transmitir a menos que el cliente utilice el códec g711. La respuesta reside en el comando **dtmf-relay**. Esta función permite a los DSP al final, iniciar el tono, reconocer el tono DTMF y separarlo de la secuencia de audio normal. Según cómo se configure, el DSP codifica este tono como un tipo diferente de paquete de protocolo en tiempo real (RTP) o como un mensaje h245 que se envía a través del link independientemente de la secuencia de audio. Este es el mismo proceso detrás de los comandos fax-relay y modem-relay.

Esta función plantea otro problema de depuración para la resolución de problemas troncales. ¿Cómo verifica qué dígitos se pasan si no hay configuración de llamada y tiene que extraer esa información del flujo de paquetes entre los routers? Cómo hacerlo depende del tipo de comando **dtmf-relay** empleado.

Como se muestra en este ejemplo, el comando [dtmf-relay cisco-rtp](#), utiliza un tipo de carga útil de Cisco propietario, por lo que debe mirar hacia abajo los DSP para ver esto. Puede ejecutar el comando **debug vpm signal** junto con el comando **debug vpm port x/x:y.z** (para limitar la salida al puerto en cuestión) para ver los dígitos pasados a los DSPs en el lado de origen. Esta salida se muestra en el lado de origen, no en el lado de terminación.

```
*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.

```

Puede verificar qué dígitos se envían desde el lado de origen con el comando [dtmf-relay h245-alphanumeric](#). El comando **dtmf-relay h245-alfanumérico** utiliza la parte alfanumérica de h.245 para transmitir los tonos. Como se muestra en este ejemplo, los dígitos se pueden ver fácilmente en los lados de origen y de terminación del tronco cuando el comando **debug h245 asn1** está habilitado:

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 se puede ver cuando se utilizan las mismas depuraciones que el comando **dtmf-relay h245-alphanumeric**. En general, resolver problemas de los troncales de conexión con el comando **dtmf-relay** es bastante difícil sin las depuraciones mencionadas.

[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](#)