

# Dépannage des troncs de connexion voix

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Problème](#)

[Solution](#)

[Problèmes courants pour des jonctions de connexion](#)

[Commencez à dépanner](#)

[Déterminez quels appels sont en hausse](#)

[DTMF dépannent](#)

[Informations connexes](#)

## [Introduction](#)

Les joncteurs réseau de connexion vocale établissent de manière permanente des communications voix, la voix sur ip (VoIP), le voix sur relais de trame (VOFR), ou le Voix sur ATM (VoATM). Les appels sont établis dès que le routeur sera tourné et la configuration est complète. Dès que les ports vocaux seront tournés, les ports vocaux composent automatiquement le numéro de téléphone factice spécifié sous le port vocal et placent un appel à l'emplacement. Les ports vocaux se terminent l'appel à l'autre extrémité par les pairs de cadran correspondants. Une fois que cette connexion est établie, en ce qui concerne le routeur, la communication voix est en session et est connectée.

## [Conditions préalables](#)

### [Conditions requises](#)

Aucune condition préalable spécifique n'est requise pour ce document.

### [Composants utilisés](#)

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

Les informations présentées dans ce document ont été créées à partir de périphériques dans un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est vivant, assurez-vous que vous comprenez l'impact potentiel de n'importe quelle commande avant que vous l'utilisiez.

## Conventions

Pour plus d'informations sur les conventions de documents, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

## Problème

Les problèmes courants qui concernent les joncteurs réseau sont transparents au routeur et très difficiles à dépanner. Les problèmes courants vus avec les joncteurs réseau de Voix sont manifestés quand un appel est placé au-dessus des joncteurs réseau et rien n'est entendu. C'est l'un des problèmes connus avec les jonctions de connexion et est provoqué par beaucoup de différentes questions. L'autre sujet est les tonalités multifréquences de double tonalité (DTMF) qui ne sont pas passées correctement, et la signalisation de l'autocommutateur privé (PBX) au PBX ne sont pas transportées correctement. Ce document dépanne ces problèmes.

Quand les camions de Voix sont hauts et en activité, les signaux se comportent différemment dans les camions de connexion. Aucune commande que vous émettez normalement sous le port vocal pour des caractéristiques de signalisation n'est appropriée et utile. Le joncteur réseau de Voix devient un conduit de signalisation et transmet par relais le signal à travers le lien VoIP. Quand vous utilisez les joncteurs réseau de Voix, la signalisation PBX doit apparier de bout en bout. En ce qui concerne les deux ordinateurs PBX, le but est de rendre le sembler de connexion de jonction de Voix identique à une ligne louée de t1 au PBX, avec des Routeurs complètement transparents tandis qu'un lien clair est établi entre les deux PBX dans le processus entier.

Quand le joncteur réseau monte, le joncteur réseau devient un câble de logiciel et le type de signal est considéré un type de connecteur. Le joncteur réseau ne s'inquiète pas du type de signal qui est utilisé. Le joncteur réseau monte toujours même si le signal ne s'assortit pas aux deux extrémités. Tant que les PBX aux deux extrémités font la même signalisation, les joncteurs réseau fonctionnent correctement.

## Solution

L'approche à prendre quand vous dépannez des questions de jonction de connexion est différent que cela est utilisée pour des appels sur réseau commuté. Pour voir ce qui se produit vraiment après que les joncteurs réseau soient vérifiés, vous devez regarder à la signalisation PBX. Avant que vous poursuiviez pour regarder la signalisation, vérifiez que les joncteurs réseau sont hauts et que les processeurs de signaux numériques (DSP) traitent les paquets vocaux.

**Remarque:** Vous voulez probablement désactiver la détection d'activité vocale (VAD) afin de dépanner. Une fois qu'on le vérifie que les joncteurs réseau fonctionnent correctement, vous devez regarder la Signalisation téléphonique afin de dépanner plus loin.

Si les joncteurs réseau sont établis, et personne des essais pour faire un appel, des messages de keepalive de joncteur réseau sont envoyés dans les deux sens entre les cases distantes. Ce Keepalives vérifie la connectivité de jonction et diffuse les informations de signalisation de bout en bout. Pour vérifier ce Keepalives, émettez la commande de [debug vpm signal](#). S'il y a beaucoup de joncteurs réseau, la sortie de **mettent au point des** commandes de **vpm**, vous peut limiter la sortie à un port unique si vous question de l'option de commande du **debug vpm port X**, où « x » est le port vocal en question. C'est la sortie de la commande émise de **debug vpm signal** quand vous regardez tous les ports :

```
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 vous limitez ceci, avec la commande du [debug vpm port X](#), met au point beaucoup plus facile à interpréter, suivant les indications de cet exemple :

```
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
```

Le Keepalives est envoyé et a reçu toutes les cinq secondes. Les termes « ont envoyé au dsp » et « reçu du dsp » soyez du point de vue de Cisco IOS®. PBX de remplacement pour que le DSP le rende plus compréhensible. Ce sont les messages qui sont vus tandis qu'il n'y a aucune activité sur les joncteurs réseau. Les messages de keepalive font les Routeurs sur chaque extrémité du circuit savoir que les joncteurs réseau sont toujours. Quand cinq de ces messages sont manqués dans une ligne, le joncteur réseau descend. Une des causes est si les joncteurs réseau s'agitent constamment dans un réseau. Pour vérifier si le Keepalives de joncteur réseau de Voix est envoyé et reçu, émettez la commande joncteur réseau-**Sc de vpm de débogage**. Ceci mettent au point ne génère pas n'importe quelle sortie jusqu'à ce que le Keepalives de joncteur réseau soit manqué. C'est un exemple de la sortie de commande joncteur réseau-**Sc de vpm de débogage** quand le Keepalives est manqué :

```
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 aucune sortie n'est vue quand la commande joncteur réseau-**Sc de vpm de débogage** est émise, alors aucun Keepalives n'est manqué. Même si le Keepalives est manqué, le joncteur réseau reste jusqu'à ce que cinq messages séquentiels soient manqués. Ceci signifie qu'une connexion doit être en baisse pour 25 secondes avant que les joncteurs réseau descendent.

## [Problèmes courants pour des jonctions de connexion](#)

Il y a plusieurs bogues associées avec les connexions de jonction de Voix. Vérifiez ces bogues si vous voyez n'importe quoi peu commun. Avant que le logiciel 12.2 de Cisco IOS ait été libéré, la plupart de ces questions avaient été abordées et intégrées. Vous pouvez regarder par les bogues pour se mettre au courant que ce sont des causes des problèmes avec un code plus ancien. Un

des la plupart des problèmes courants est d'obtenir les PBX pour signaler correctement au-dessus de la connexion de jonction. Il semble comme une bonne idée de réduire les joncteurs réseau et de configurer les Routeurs de sorte qu'ils travaillent à chaque extrémité, mais l'approche est vraiment contre-productive puisque quelque chose que vous avez changée maintenant devient discutable une fois les joncteurs réseau sont établies. La meilleure manière de dépanner est avec les joncteurs réseau hauts et fonctionnels.

## Commencez à dépanner

Il est nécessaire de regarder les fondements établir que ceux-ci fonctionnent correctement :

- Les joncteurs réseau sont-ils établis ? Émettez la commande **récapitulative de show voice call**, et assurez-vous que les joncteurs réseau sont dans l'état `S_CONNECTED`.
- Les DSP traitent-ils des paquets ? Émettez la commande de **show voice dsp** de vérifier ceci. Si vous ne voyez pas que des paquets sont traités par les DSP, il est parce que VAD est activé et est supprime les paquets. Arrêtez VAD, rétablissez les joncteurs réseau et regardez de nouveau. En outre, contrôlez que le paquet pare l'incrément quand la la **commande brief de show call active voice** est émise. Cette commande affiche également si VAD est activé pour la question de procédure de connexion d'appel.

Si les joncteurs réseau se connectent aux ports analogiques à n'importe quel site, il est le meilleur de vérifier l'exécution du PBX en mode non-trunked. Pour dépanner des problèmes analogiques de Connectivité E&M, référez-vous [en comprenant et dépannage des types d'interface analogiques E&M et câblage des organisations](#). Une fois que tout est vérifié et fonctionne correctement, apportez les joncteurs réseau et regardez la signalisation qui est passée entre les PBX.

La manière idéale de dépanner des problèmes de connexion de jonction de Voix est d'examiner la signalisation qui est passée entre les PBX. Il est le meilleur d'avoir une session de telnet à chaque routeur en question de sorte qu'on puisse observer la signalisation en tant qu'elle soit passée d'une extrémité à l'autre. Ce document utilise la signalisation de clin d'oeil E&M puisqu'il est assez populaire et synchronisation de clin d'oeil doit être pris en compte.

C'est la sortie du routeur connecté au PBX qui lance l'appel :

```
May 22 19:39:03.582: [3/0:0(1)] rcv from dsp sig DCBA state 0x0
!--- It is in idle state. May 22 19:39:07.774: [3/0:0(1)] send to dsp SIG DCBA state 0x0 !---
ABCD bits=0000. May 22 19:39:08.586: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
19:39:12.778: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:13.586: [3/0:0(1)] rcv from
dsp SIG DCBA state 0x0 May 22 19:39:17.777: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22
19:39:18.593: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:22.781: [3/0:0(1)] send to
dsp SIG DCBA state 0x0 May 22 19:39:23.593: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
19:39:27.781: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:28.597: [3/0:0(1)] rcv from
dsp SIG DCBA state 0x0 May 22 19:39:32.785: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22
19:39:33.597: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:37.789: [3/0:0(1)] send to
dsp SIG DCBA state 0x0 May 22 19:39:38.601: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22
19:39:39.777: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0 May 22 19:39:39.797: [3/0:0(1)] rcv
from dsp SIG DCBA state 0x0 May 22 19:39:39.817: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF !---
Receives off-hook from PBX, and passes to remote end. May 22 19:39:39.837: [3/0:0(1)] rcv from
dsp SIG DCBA state 0xF May 22 19:39:39.857: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22
19:39:39.877: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:39.897: [3/0:0(1)] rcv
from dsp SIG DCBA state 0xF May 22 19:39:39.917: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May
22 19:39:39.937: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:39.957: [3/0:0(1)] rcv
from dsp SIG DCBA state 0xF May 22 19:39:39.977: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May
22 19:39:39.997: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.017: [3/0:0(1)] rcv
from dsp SIG DCBA state 0xF May 22 19:39:40.037: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May
```



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

**Cette sortie affiche que le routeur termine l'appel. Le Protocole NTP (Network Time Protocol) synced.**

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

**Remarque:** Cette sortie affiche la signalisation qui se produit des deux côtés d'un joncteur réseau de Voix quelle signalisation de clin d'oeil des utilisations E&M. On peut voir d'autres types de signalisation que qui utilise des ces même met au point. Si vous voyez des appels établis correctement (comme affiché ici), l'audio bi-directionnel doit être présent. Ceci peut être vérifié si vous regardez le **show voice dsp** ou la sortie de **commande brief de show call active voice**. Si tout regarde pour être bon là, et vous obtenez des problèmes sonores (aucun sonore ou à sens unique) avec des connexions analogiques, vérifiez ces connexions de nouveau.

## [Déterminez quels appels sont en hausse](#)

Puisque cela le fait peu ou pas de bien de regarder la sortie de commande **récapitulative de show call active voice** ou de **show voice call** pour des appels trunked, vous avez besoin d'une méthode simple pour déterminer quel des appels actifs de support de joncteurs réseau de Voix. Un des moyens les plus simples de faire ceci est d'émettre la commande de **show voice trunk-conditioning signaling** en même temps que le paramètre d'*inclure* et d'utiliser *ABCD* comme chaîne incluse, suivant les indications d'ici :

```
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
```

**Remarque:** Cette sortie affiche un appel est actif sur un créneau horaire 10 et un appel différent étant commencé sur le créneau horaire huit. Vous voulez faire un pseudonyme pour cette commande plutôt longue si vous l'utilisez beaucoup.

## DTMF dépannement

Indépendamment du hors fonction-crochet et de la signalisation avec combiné raccroché, les seuls l'autre chose que les Routeurs passent entre les PBX (sans compter que la Voix) sont des tonalités DTMF. Il y a également un chemin audio ainsi ce n'est habituellement pas un problème, mais il y a un problème. Le problème surgit avec la façon dont vous faites l'audio au-dessus de ce chemin. Il est parfois préférable d'employer les bas codecs de débit binaire afin de sauvegarder la bande passante. La question est soulevée que ces bas codecs de débit binaire sont conçus au moyen d'algorithmes qui ont été écrits pour le discours humain. Les tonalités DTMF ne se conforment pas à ces algorithmes très bien et ont besoin d'une autre méthode pour transporter à moins que le client utilise les codecs g711. La réponse se situe dans la commande de **dtmf-relay**. Cette caractéristique permet les DSP à l'extrémité, commence la tonalité, pour identifier la tonalité DTMF et pour la séparer du flux audio régulier. Basé sur la façon dont il est configuré, le DSP code alors cette tonalité comme type différent de paquet de Protocole RTP (Real-Time Protocol) ou comme message h245 à envoyer à travers le lien séparément du flux audio. C'est le même processus derrière les commandes de **télécopie-relais** et de **modem-relais**.

Cette caractéristique pose des autres mettent au point le problème pour le dépannage de joncteur réseau. Comment vérifiez-vous ce que des chiffres sont passés s'il n'y a aucun établissement d'appel et vous devez extraire ces informations du flux de paquets entre les Routeurs ? Comment faire ceci dépend de quel type de commande de **dtmf-relay** est utilisé.

Suivant les indications de cet exemple, la commande de Cisco-[rtp de dtmf-relay](#), utilise un type de charge utile de Cisco de classe des propriétaires, ainsi vous devez regarder vers le bas les DSP pour voir ceci. Vous pouvez émettre la commande de **debug vpm signal** en même temps que le **debug vpm port x/x : la commande y.z** (pour limiter la sortie au port en question) de voir les chiffres a passé aux DSP au côté d'origine. Cette sortie est affichée au côté d'origine, pas sur le côté de terminaison.

```
*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.

```

Vous pouvez vérifier quels chiffres sont envoyés du côté d'origine avec la commande du [dtmf-relay h245-alphanumeric](#). La commande du **dtmf-relay h245-alphanumeric** emploie la partie alphanumérique de h.245 pour donner les tonalités. Suivant les indications de cet exemple, les chiffres peuvent facilement être vus sur les côtés d'origine et de terminaisons du joncteur réseau quand la commande de **debug h245 asn1** est activée :

### Côté d'origine :

```

*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

```

### Terminaison du côté :

```

*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"

```

La commande du [dtmf-relay h245-signal](#) est très semblable et peut être vue quand la même met au point pendant que la commande du **dtmf-relay h245-alphanumeric** sont utilisées. De façon générale, il est plutôt difficile de dépanner les jonctions de connexion avec la commande de **dtmf-relay** sans met au point mentionné.

## Informations connexes

- [Configuration et résolution des problèmes associés à la signalisation CCS transparente](#)
- [Assistance technique concernant la technologie vocale](#)
- [Support produit pour Voix et Communications IP](#)
- [Dépannage des problèmes de téléphonie IP Cisco](#)
- [Support technique - Cisco Systems](#)