

Dépannage des troncs de connexion voix

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Problème](#)

[Solution](#)

[Problèmes courants pour les agrégations de connexion](#)

[Début du dépannage](#)

[Déterminer les appels actifs](#)

[Dépannage DTMF](#)

[Informations connexes](#)

[Introduction](#)

Les liaisons de connexion vocale établissent de manière permanente des appels vocaux, soit par VoIP (Voice over IP), soit par VoFR (Voice over Frame Relay), soit par VoATM (Voice over ATM). Les appels sont établis dès que le routeur est activé et que la configuration est terminée. Dès que les ports vocaux sont activés, les ports vocaux composent automatiquement le numéro de téléphone factice spécifié sous le port vocal et passent un appel à l'emplacement. Les ports vocaux terminent l'appel vers l'autre extrémité via les homologues de numérotation correspondants. Une fois cette connexion établie, l'appel vocal est en session et connecté.

[Conditions préalables](#)

[Conditions requises](#)

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

[Components Used](#)

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. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est actif, assurez-vous de bien comprendre l'impact potentiel d'une commande avant de l'utiliser.

[Conventions](#)

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

Problème

Les problèmes courants liés aux agrégations sont transparents pour le routeur et très difficiles à résoudre. Les problèmes courants observés avec les agrégations vocales se manifestent lorsqu'un appel est passé sur les agrégations et qu'aucun appel n'est entendu. Il s'agit d'un des problèmes connus avec les agrégations de connexion et est causé par de nombreux problèmes différents. Un autre problème concerne les tonalités DTMF (Dual Tone Multifrequency) qui ne sont pas transmises correctement et la signalisation de PBX (Private Branch Exchange) vers PBX n'est pas transportée correctement. Ce document résout ces problèmes.

Lorsque les camions vocaux sont activés, les signaux se comportent différemment dans les camions de connexion. Les commandes que vous émettez normalement sous le port vocal pour les caractéristiques de signalisation ne sont pas pertinentes et utiles. La liaison vocale devient un conduit de signalisation et transmet le signal via la liaison VoIP. Lorsque vous utilisez les liaisons vocales, la signalisation PBX doit correspondre de bout en bout. En ce qui concerne les deux PBX, l'objectif est de rendre la connexion de liaison vocale identique à une ligne T1 louée au PBX, avec des routeurs totalement transparents tandis qu'une liaison claire est établie entre les deux PBX dans l'ensemble du processus.

Lorsque la liaison est activée, elle devient un câble logiciel et le type de signal est considéré comme un type de connecteur. Le trunk ne se soucie pas du type de signal utilisé. L'agrégation s'active même si le signal ne correspond pas aux deux extrémités. Tant que les PBX aux deux extrémités font la même signalisation, les agrégations fonctionnent correctement.

Solution

L'approche à adopter lors du dépannage des problèmes de liaison de connexion est différente de celle utilisée pour les appels commutés. Pour voir ce qui se passe réellement après la vérification des agrégations, vous devez consulter la signalisation PBX. Avant de continuer à examiner la signalisation, vérifiez que les trunks sont actifs et que les processeurs de signal numérique (DSP) traitent les paquets vocaux.

Remarque : Vous devez probablement désactiver la détection d'activité vocale (VAD) afin de résoudre les problèmes. Une fois qu'il a été vérifié que les agrégations fonctionnent correctement, vous devez examiner la signalisation téléphonique afin de dépanner plus loin.

Si les agrégations sont établies et que personne n'essaie de passer un appel, les messages de test d'activité des agrégations sont renvoyés entre les boîtes distantes. Ces keepalives vérifient la connectivité de liaison et transportent les informations de signalisation de bout en bout. Pour vérifier ces keepalives, émettez la commande [debug vpm signal](#). S'il y a beaucoup de liaisons, la sortie des commandes **debug vpm**, vous pouvez limiter la sortie à un seul port si vous émettez l'option de commande **debug vpm port x**, où « x » est le port vocal en question. Ceci est la sortie de la commande **debug vpm signal** émise lorsque 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 [debug vpm port x](#), *les débogages sont beaucoup plus faciles à interpréter, comme illustré dans 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
```

Les messages de test d'activité sont envoyés et reçus toutes les cinq secondes. Les termes « envoyé à dsp » et « reçu de dsp » proviennent du point de vue de Cisco IOS®. Remplacez PBX par DSP pour le rendre plus compréhensible. Il s'agit des messages qui sont affichés alors qu'il n'y a aucune activité sur les agrégations. Les messages keepalive permettent aux routeurs de chaque extrémité du circuit de savoir que les agrégations sont toujours actives. Lorsque cinq de ces messages sont manqués dans une ligne, l'agrégation tombe en panne. L'une des causes est que les agrégations battent constamment dans un réseau. Pour vérifier si les keepalives de liaison vocale sont envoyés et reçus, émettez la commande **debug vpm trunk-sc**. Ce débogage ne génère aucune sortie tant que les keepalives de trunk ne sont pas manqués. Voici un exemple de la sortie de commande **debug vpm trunk-sc** lorsque des messages de test d'activité sont manqués :

```
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 visible lorsque la commande [debug vpm trunk-sc](#) est exécutée, aucune activité de test d'activité n'est manquée. Même si des messages de test d'activité sont manqués, la liaison reste active jusqu'à ce que cinq messages séquentiels soient manqués. Cela signifie qu'une connexion doit être interrompue pendant 25 secondes avant que les agrégations ne s'arrêtent.

[Problèmes courants pour les agrégations de connexion](#)

Plusieurs bogues sont associés aux connexions de liaison vocale. Vérifiez ces bogues si vous voyez quelque chose d'inhabituel. Au moment de la sortie du logiciel Cisco IOS 12.2, la plupart de ces problèmes avaient été traités et intégrés. Vous pouvez parcourir les bogues pour vous rendre compte que ce sont des causes de problèmes avec le code plus ancien. L'un des problèmes les plus courants est d'obtenir que les PBX signalent correctement via la connexion de liaison. Il

semble judicieux d'arrêter les agrégations et de configurer les routeurs de sorte qu'ils fonctionnent à chaque extrémité, mais l'approche est vraiment contre-productive car tout ce que vous avez changé devient désormais sans objet une fois les agrégations établies. Le meilleur moyen de dépanner est d'installer et de faire fonctionner les agrégations.

Début du dépannage

Il est nécessaire d'examiner les éléments de base pour établir que ces éléments fonctionnent correctement :

- Les trunks sont-ils établis ? Exécutez la commande **show voice call summary** et assurez-vous que les faisceaux sont à l'état `S_CONNECTED`.
- Les DSP traitent-ils des paquets ? Émettez la commande **show voice dsp** pour vérifier ceci. Si vous ne voyez pas que les paquets sont traités par les DSP, c'est parce que VAD est activé et supprime les paquets. Désactivez VAD, rétablissez les trunks et examinez à nouveau. Vérifiez également que les compteurs de paquets s'incrémentent lorsque la commande **show call active voice brief** est exécutée. Cette commande indique également si VAD est activé pour le journal des appels en question.

Si les trunks se connectent aux ports analogiques de n'importe quel site, il est préférable de vérifier le fonctionnement du PBX en mode non agrégé. Pour résoudre les problèmes de connectivité E&M analogique, reportez-vous à [Présentation et dépannage des types d'interface E&M analogique et des arrangements de câblage](#) . Une fois que tout est vérifié et fonctionne correctement, activez les agrégations et examinez la signalisation qui est transmise entre les PBX.

La manière idéale de dépanner les problèmes de connexion de liaison vocale consiste à examiner la signalisation qui est transmise entre les PBX. Il est préférable d'avoir une session Telnet sur chaque routeur en question afin que la signalisation puisse être observée lors de son passage d'une extrémité à l'autre. Ce document utilise la signalisation de liaison E&M, car elle est assez populaire et il faut tenir compte du moment de la liaison.

Voici le résultat du routeur connecté au PBX qui est à l'origine de 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
22 19:39:40.057: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.077: [3/0:0(1)] rcv
```


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

Ce résultat montre que le routeur termine l'appel. Le protocole NTP (Network Time Protocol) est synchronisé.

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 montre la signalisation qui se produit des deux côtés d'une liaison vocale qui utilise la signalisation de liaison E&M. D'autres types de signalisation utilisent ces mêmes débogages. Si les appels sont correctement établis (comme indiqué ici), le son bidirectionnel doit être présent. Ceci peut être vérifié si vous regardez la sortie de commande **show voice dsp** ou **show call active voice brief**. Si tout semble aller bien là-bas et que vous avez des problèmes audio (pas d'audio ni d'aller simple) avec des connexions analogiques, vérifiez à nouveau ces connexions.

[Déterminer les appels actifs](#)

Étant donné qu'il ne sert à rien ou pas à examiner la sortie de commande **show call active voice** ou **show voice call summary** pour les appels agrégés, vous avez besoin d'une méthode simple pour déterminer quelles liaisons vocales prennent en charge les appels actifs. L'une des façons les plus simples d'y parvenir est d'émettre la commande **show voice trunk-conditionnement signalisation** en conjonction avec le paramètre *include* et d'utiliser *ABCD* comme chaîne incluse, comme illustré 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=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 : ce résultat montre un appel actif sur le lot de temps 10 et un autre appel en cours de démarrage sur le lot de temps 8. Vous voulez créer un alias pour cette commande assez longue si vous l'utilisez beaucoup.

Dépannage DTMF

En dehors de la signalisation décrochée et raccrochée, la seule autre chose que les routeurs passent entre les PBX (en plus de la voix) sont les tonalités DTMF. Il existe également un chemin audio, ce qui n'est généralement pas un problème, mais il y a un problème. Le problème est lié à la façon dont vous faites de l'audio sur ce chemin. Il est parfois préférable d'utiliser les codecs à faible débit binaire afin d'économiser de la bande passante. Le problème vient du fait que ces codecs à faible débit binaire sont conçus au moyen d'algorithmes écrits pour la parole humaine. Les tonalités DTMF ne sont pas très bien conformes à ces algorithmes et nécessitent une autre méthode de transmission, sauf si le client utilise le codec g711. La réponse se trouve dans la commande **dtmf-relay**. Cette fonction permet aux DSP de se terminer, de démarrer la tonalité, de reconnaître la tonalité DTMF et de la séparer du flux audio normal. En fonction de sa configuration, le DSP code ensuite cette tonalité comme un type différent de paquet RTP (Real Time Protocol) ou comme un message h245 à envoyer sur la liaison séparément du flux audio. Il s'agit du même processus derrière les commandes **fax-relay** et **modem-relay**.

Cette fonctionnalité pose un autre problème de débogage pour le dépannage de trunk. Comment vérifiez-vous les chiffres transmis s'il n'y a pas de configuration d'appel et que vous devez extraire ces informations du flux de paquets entre les routeurs ? La façon de procéder dépend du type de commande **dtmf-relay** utilisé.

Comme indiqué dans cet exemple, la commande [dtmf-relay cisco-rtp](#) utilise un type de données utiles Cisco propriétaire. Vous devez donc regarder les DSP en bas pour voir ceci. Vous pouvez émettre la commande **debug vpm signal** conjointement avec la commande **debug vpm port x/x:y.z** (pour limiter la sortie au port en question) pour voir les chiffres transmis aux DSP du côté d'origine. Cette sortie s'affiche du côté d'origine et non du 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 à l'aide de la commande [dtmf-relay h245-alphanumérique](#). La commande **dtmf-relay h245-alphanumérique** utilise la partie alphanumérique de h.245 pour transmettre les tonalités. Comme indiqué dans cet exemple, les chiffres peuvent être facilement visibles aux côtés d'origine et de fin de la liaison lorsque la commande **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

```

Côté terminaison :

```

*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 [dtmf-relay h245-signal](#) est très similaire et peut être vue lorsque les mêmes débogages que la commande **dtmf-relay h245-alphanumérique** sont utilisés. Globalement, le dépannage des agrégations de connexion avec la commande **dtmf-relay** est plutôt difficile sans les débogages mentionnés.

[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](#)