Troubleshooting de Troncos de Conexão de Voz

Índice

Introdução

Os troncos de conexão de voz estabelecem permanentemente chamadas de voz, Voz sobre IP (VoIP), voz sobre o Frame Relay (VOFR), ou Voz sobre ATM (VoATM). As chamadas serão estabelecidas assim que o roteador for ligado e a configuração for concluída. Assim que as portas de voz forem giradas acima, as portas de voz discam automaticamente o número de telefone do manequim especificado sob a porta de voz e colocam um atendimento ao lugar. As portas de voz terminam o atendimento à outra extremidade através dos dial peer correspondentes. Uma vez que esta conexão é estabelecida, tanto quanto o roteador, a chamada de voz está na sessão e é conectada.

Pré-requisitos

Requisitos

Não existem requisitos específicos para este documento.

Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se seu l rede está vivo, certifique-se de que você compreende o impacto potencial do comando any antes que você o use.

Convenções

Para obter mais informações sobre convenções de documento, consulte as Convenções de dicas técnicas Cisco.

Problema

Os problemas comuns que se referem os troncos são transparentes ao roteador e muito difíceis de pesquisar defeitos. Os problemas comuns considerados com os troncos da Voz estão manifestados quando um atendimento está colocado sobre os troncos e nada está ouvido. Este é um dos problemas conhecidos com os troncos de conexão e é causado por muitas edições diferentes. O outro problema é os tons Multifrequency do tom dual (DTMF) que não são passados corretamente, e a sinalização do central telefônica privada (PBX) ao PBX não é transportada corretamente. Este documento apresenta soluções para esses problemas.

Quando os caminhões da Voz são ascendentes e ativos, os sinais comportam-se diferentemente nos caminhões da conexão. Os comandos any que você emite normalmente sob a porta de voz para características de sinalização não são relevantes e úteis. O tronco da Voz transforma-se uma conduíte da sinalização e retransmite-se o sinal através do link de VoIP. Quando você usa os troncos da Voz, a sinalização de PBX deve combinar fim-a-fim. Tanto quanto as duas máquinas PBX, o objetivo é fazer o olhar da conexão de tronco da Voz idêntico a uma linha T1 alugada ao PBX, com o Roteadores completamente transparente quando um link claro for estabelecido entre os dois PBX no processo inteiro.

Quando o tronco vem acima, o tronco transforma-se um cabo do software e o tipo de sinal é considerado um tipo de conector. O tronco não se importa com o tipo de sinal que é usado. O tronco ainda vem acima mesmo se o sinal não combina no ambas as extremidades. Enquanto os PBX no ambas as extremidades fazem a mesma sinalização, os troncos funcionam corretamente.

Solução

A aproximação a tomar quando você pesquisa defeitos edições do tronco de conexão é diferente do que aquele é usada para chamadas comutadas. Para ver o que acontece realmente depois que os troncos são verificados, você precisa de olhar à sinalização de PBX. Antes que você continue olhar a sinalização, verifique que os troncos são ascendentes e que os processadores do sinal digital (DSP) processam os pacotes de voz.

Nota: Você quer provavelmente desabilitar a detecção de atividade da Voz (VAD) a fim pesquisar defeitos. Uma vez que se verifica que os troncos funcionam corretamente, você precisa de olhar a Sinalização de telefonia a fim pesquisar defeitos mais.

Se os troncos estão estabelecidos, e ninguém tenta fazer um atendimento, as mensagens do keepalive de tronco estão enviadas para a frente e para trás entre as caixas remotas. Este Keepalives verifica a conectividade de tronco e leva a informação de sinalização de fim-a-fim. Para verificar este Keepalives, emita o comando debug vpm signal. Se há muitos troncos, a saída dos **comandos debug vpm**, você pode limitar a saída a uma porta única se você introdução da opção do **comando debug vpm port x**, onde “x” é a porta de voz na pergunta. Esta é a saída do **comando debug vpm signal** emitido quando você olha todas as portas:

```
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
```

Se você limita este, com o *comando debug vpm port x*, debuga muito mais fácil interpretar, segundo as indicações deste exemplo:

```
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
```

O Keepalives é enviado e recebeu cada cinco segundos. Os termos “enviaram ao dsp” e “recebido do dsp” seja do ponto de vista do ® do Cisco IOS. Substitua o PBX para que o DSP faça-o mais compreensível. Estas são as mensagens que são consideradas quando não houver nenhuma atividade nos troncos. As mensagens de manutenção de atividade permitem que os roteadores em cada extremidade do circuito percebam que os troncos ainda estão ativos. Quando cinco destas mensagens são faltadas em seguido, o tronco vai para baixo. Uma das causas é se os troncos batem constantemente em uma rede. Para verificar se os keepalives de tronco da Voz estão enviados e recebidos, emita o **comando debug vpm trunk-sc**. Isto debuga não gerencie nenhuma saída até que os keepalives de tronco estejam faltados. Este é um exemplo do **comando debug vpm trunk-sc** output quando o Keepalives é faltado:

```
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
```

Se nenhuma saída é considerada quando o comando debug vpm trunk-sc está emitido, a seguir nenhum Keepalives está faltado. Mesmo se o Keepalives é faltado, o tronco fica acima até que cinco mensagens sequencial estejam faltados. Isto significa que uma conexão deve estar para baixo para 25 segundos antes que os troncos vão para baixo.

Problemas comuns sobre truncamentos de conexão

Há diversos erros associados com as conexões de tronco da Voz. Verifique estes erros se você vê qualquer coisa incomum. Antes que o Cisco IOS Software 12.2 fosse liberado, a maioria destas edições tinham sido endereçadas e integradas. Você pode olhar através dos erros para fazer-se ciente que estas são causas dos problemas com código mais velho. Um da maioria de problemas comuns é conseguir os PBX sinalizar corretamente sobre a conexão de tronco. Parece como uma boa ideia derrubar os troncos e configurar o Roteadores de modo que trabalhe em cada extremidade, mas a aproximação é realmente ineficaz desde que qualquer coisa que você

mudou agora se torna sede os troncos é estabelecida uma vez. A melhor maneira de pesquisar defeitos é com os troncos ascendentes e funcionais.

Comece a pesquisar defeitos

Énecessário olhar os princípios para estabelecer que estes funcionam corretamente:

- Os troncos estão estabelecidos? Emita o **comando show voice call summary**, e certifique-se de que os troncos estão no estado `S_CONNECTED`.
- Os DSPs estão processando pacotes? Emita o **comando show voice dsp** verificar isto. Se você não vê que os pacotes estão processados pelos DSP, é porque o VAD é permitido e é suprime os pacotes. Desative o VAD, restabeleça os troncos e pesquise novamente. Também, certifique-se dos contadores de pacote de informação incrementem quando o o **comando show call ative voice brief** é emitido. Este comando igualmente mostra se o VAD está permitido para a pergunta do início de uma sessão do atendimento.

Se os troncos conectam aos portos analógicos em qualquer local, é o melhor verificar a operação do PBX no modo do NON-em tronco. Para pesquisar defeitos problemas de conectividade análogos do E&M, refira a compreensão e a pesquisa de defeitos de tipos de interface e de arranjos de rede análogos do E&M. Uma vez que tudo é verificado e funciona corretamente, traga os troncos acima e olhe a sinalização que é passada entre os PBX.

A maneira ideal pesquisar defeitos problemas de conexão de tronco da Voz é examinar a sinalização que é passada entre os PBX. É o melhor ter uma sessão de Telnet a cada roteador na pergunta de modo que a sinalização possa ser observada como ela seja passada de uma extremidade à outro. Este documento usa a sinalização da piscadela do E&M desde que é razoavelmente popular e o sincronismo de permissão tem que ser tomado na consideração.

Esta é a saída do roteador conectado ao PBX que origina o atendimento:

```
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
from dsp SIG DCBA state 0xF May 22 19:39:40.089: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May
22 19:39:40.097: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.109: [3/0:0(1)] send
to dsp SIG DCBA state 0x0 May 22 19:39:40.117: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF !---
Receiving wink from remote side, and passes to PBX. May 22 19:39:40.129: [3/0:0(1)] send to dsp
SIG DCBA state 0xF May 22 19:39:40.137: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22
```

19:39:40.149: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.157: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.169: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.177: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.189: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.197: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.213: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.217: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.229: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.237: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.249: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.257: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:40.269: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.289: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.309: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.329: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:40.349: [3/0:0(1)] send to dsp SIG DCBA state 0x0 *!--- Wink ended from remote side, and passes to PBX.* May 22 19:39:40.369: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.389: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.409: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.429: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.449: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.469: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.493: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.509: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.529: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.549: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.569: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.589: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.613: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.629: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.649: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.669: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.689: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.709: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.729: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.749: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:40.769: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:45.773: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:50.081: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:50.101: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:39:50.121: [3/0:0(1)] send to dsp SIG DCBA state 0xF *!--- Wink ends, the remote end is now off-hook, the conversation happens.* May 22 19:39:50.141: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.161: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.181: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.197: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.221: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.241: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.261: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.261: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:50.281: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.301: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.321: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.341: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.361: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.381: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.401: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.421: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.441: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.461: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.481: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.501: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.521: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.541: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:50.561: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:39:55.265: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:39:55.561: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:00.269: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:00.565: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:05.268: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:05.564: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:10.272: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:10.568: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:15.276: [3/0:0(1)] rcv from dsp SIG DCBA state 0xF May 22 19:40:15.572: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.676: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.696: [3/0:0(1)] send to dsp SIG DCBA state 0xF May 22 19:40:19.716: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.736: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.756: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.776: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.796: [3/0:0(1)] send to dsp SIG DCBA state 0x0 May 22 19:40:19.796: [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

Esta saída mostra que o roteador termina o atendimento. O NTP (Protocolo de tempo de rede) foi 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 saída mostra a sinalização que ocorre em ambos os lados de um tronco da Voz que sinalização da piscadela do E&M dos usos. Outros tipos de sinalização podem ser considerados que que usa estes mesma debuga. Se você vê os atendimentos estabelecidos corretamente (como mostrado aqui), a obrigação do áudio de duas vias esta presente. Isto pode ser verificado se você olha o **DSP de voz da mostra** ou a saída do **comando show call ative voice brief**. Se tudo olha para ser fino lá, e você está obtendo problemas de áudio (não audio ou de sentido único) com conexões analógicas, verifique estas conexões outra vez.

Determine que atendimentos estão acima

Desde que faz quase nenhum bom olhar o **comando show call ative voice ou show voice call summary** output para chamadas truncadas, você precisa um método simples de determinar que troncos da Voz apoiam chamadas ativa. Uma das maneiras as mais fáceis de fazer isto é emitir o **comando show voice trunk-conditioning signaling** conjuntamente com o parâmetro *incluir* e usar o *ABCD* como a corda incluída, segundo as indicações de aqui:

```
Phoenix#show voice trunk-conditioning signaling | include ABCD last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-
ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=1111, last-RX-ABCD=0000 !---
Timeslot 8. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=1111, last-RX-ABCD=1111 !---
Timeslot 10. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-
ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000
```

Nota: Esta saída mostra um active do atendimento no intervalo de tempo 10 e em um outro atendimento que está sendo começado no intervalo de tempo oito. Você quer fazer um pseudônimo para este comando um pouco longo se você o usa muito.

O DTMF pesquisa defeitos

Independentemente do fora-gancho e da sinalização conectada, o únicos a outra coisa que o Roteadores passa entre os PBX (além da Voz) são toms DMTF. Há igualmente um caminho de áudio assim que este não é geralmente um problema, mas há um problema. O problema elevara com como você faz o áudio sobre esse trajeto. É às vezes preferível usar os codecs do Low Bit Rate a fim salvar a largura de banda. A edição vem acima do esse estes codecs do Low Bit Rate é projetada por meio dos algoritmos que foram escritos para o discurso humano. Os toms DMTF não se conformam a estes algoritmos muito bem e precisam-se algum outro método de transportar a menos que o cliente usar o codec g711. A resposta encontra-se no **comando dtmf-relay**. Esta característica permite os DSP na extremidade, começa o tom, reconhecer o tom DMTF e separá-lo do fluxo de áudio regular. Baseado em como é configurado, o DSP codifica então este tom como um tipo diferente de pacote do Real-Time Protocol (RTP) ou como uma mensagem h245 a ser enviada separadamente através do link do fluxo de áudio. Esse é o mesmo processo por trás dos comandos fax-relay e modem-relay.

Esta característica levanta outra debuga o problema para o Troubleshooting de tronco. Como você verifica o que os dígitos estão passados se não há nenhuma configuração de chamada e você tem que extrair essa informação da corrente de pacote de informação entre o Roteadores? Como fazer isto depende em cima de que tipo de **comando dtmf-relay** é empregado.

Segundo as indicações deste exemplo, o comando dtmf-relay cisco-rtp, usa um tipo de payload proprietário de Cisco, assim que você deve olhar para baixo nos DSP para ver este. Você pode emitir o **comando debug vpm signal** conjuntamente com o **debug vpm port x/x: comando y.z** (para limitar a saída à porta na pergunta) ver os dígitos passados aos DSP no lado de origem. Esta saída é indicada no lado de origem, não no lado de terminação.

```
*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.
```

Você pode verificar que dígitos são enviados do lado de origem com o comando dtmf-relay h245-alphanumeric. **O comando dtmf-relay h245-alphanumeric** usa a parcela alfanumérica de h.245 para transportar os tons. Segundo as indicações deste exemplo, os dígitos podem facilmente ser considerados na origem e nos lados de terminação do tronco quando o **comando debug h245 asn1** é permitido:

Lado de origem:

```
*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 terminação:

```
*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"
```

O comando dtmf-relay h245-signal é muito similar e pode ser considerado quando o mesmo debuga enquanto o **comando dtmf-relay h245-alphanumeric** está usado. Total, pesquisar defeitos os troncos de conexão com o **comando dtmf-relay** é um pouco difícil sem debuga mencionado.

Informações Relacionadas

- Configurando e Troubleshooting de Transparent CCS
- Suporte à Tecnologia de Voz
- Suporte de Produtos de Comunicação de Voz e de IP
- Troubleshooting da Telefonia IP Cisco
- Suporte Técnico - Cisco Systems