Voice Connection-trunks voor probleemoplossing

Inhoud

[Inleiding](#)
[Voorwaarden](#)
[Vereisten](#)
[Gebruikte componenten](#)
[Conventies](#)
[Probleem](#)
[Oplossing](#)
[Gemeenschappelijke problemen voor verbindingstrunks](#)
[Probleemoplossing starten](#)
[Bepaal wat de oproepen zijn](#)
[DTMF-probleemoplossing](#)
[Gerelateerde informatie](#)

Inleiding

De stemverbinding vormt permanent spraakoproepen, of Voice-over-IP (VoIP), Voice-over-Frame Relay (VoFR) of Voice-over-ATM (VoATM). De oproepen worden ingesteld zodra de router is ingeschakeld en de configuratie is voltooid. Zodra de spraakpoorten zijn geactiveerd, bellen de spraakpoorten automatisch het telefoonnummer dat onder de spraakpoort is opgegeven en zetten ze een oproep naar de locatie. De spraakpoorten voltooien de oproep aan het andere uiteinde door de corresponderende kiespeers. Zodra deze verbinding wordt gevestigd, wat de router betreft, is de spraakoproep in zitting en is verbonden.

Voorwaarden

Vereisten

Er zijn geen specifieke voorwaarden van toepassing op dit document.

Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies.

De informatie in dit document is gebaseerd op apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk leeft, zorg ervoor dat u de potentiële impact van om het even welke opdracht begrijpt alvorens u het gebruikt.

Conventies

Raadpleeg Cisco Technical Tips Conventions (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

Probleem

De gemeenschappelijke problemen die met de trunks te maken hebben zijn transparant voor de router en zeer moeilijk in te lossen. De gemeenschappelijke kwesties die met de stemstammen te maken hebben, worden duidelijk als er een oproep boven de stammen wordt geplaatst en er niets wordt gehoord. Dit is een van de bekende problemen met de verbindingsstammen en wordt veroorzaakt door vele verschillende problemen. Een ander probleem is de Dual Tone Multiflex (DTMF)-tonen die niet goed worden doorgegeven, en signalering van Private Branch Exchange (PBX) naar PBX wordt niet goed getransporteerd. Dit document lost deze problemen op.

Wanneer de vrachtwagens actief zijn, gedragen de signalen zich anders in de verbindingswagens. Alle opdrachten die u normaal onder de spraakpoort geeft voor het signaleren van kenmerken zijn niet relevant en behulpzaam. De stemstam wordt een signaleringsgeleider en geeft het signaal over de VoIP-link terug. Wanneer u de spraakstammen gebruikt, moet de PBX-signalering overeenkomen met end-to-end. Wat de twee PBX-machines betreft, is het doel om de spraakverbinding identiek te maken met een geleasede T1-lijn naar de PBX-systemen, met routers die volledig transparant zijn, terwijl er een duidelijke link wordt gelegd tussen de twee PBX-systemen in het gehele proces.

Wanneer de romp omhoog komt, wordt de romp een softwarekabel en het signaaltype wordt als een connector type beschouwd. De romp geeft niet om het signaaltype dat wordt gebruikt. De romp komt nog steeds naar boven, zelfs als het signaal niet aan beide uiteinden overeenkomt. Zolang de PBX's aan beide uiteinden dezelfde signalering uitvoeren, werken de trunks goed.

Oplossing

De benadering om te nemen wanneer u problemen met de verbindingsromp van de probleemoplossing hebt is verschillend dan die voor geschakelde oproepen wordt gebruikt. Om te zien wat er echt gebeurt nadat de stammen zijn geverifieerd, moet je naar de PBX-signalering kijken. Voordat u doorgaat om naar de signalering te kijken, controleert u of de stammen omhoog zijn en of de digitale signaalprocessors (DSP's) de spraakpakketten verwerken.

Opmerking: U wilt waarschijnlijk VAD (Voice Action Detection) uitschakelen om een oplossing te vinden. Zodra het wordt geverifieerd dat de stammen correct functioneren, moet u het signaleren van de Telephony bekijken om verder oplossing te zoeken.

Als de stammen worden ingericht, en niemand probeert een oproep te doen, dan worden de boodschappen van de boomstam tussen de verafgelegen dozen heen en weer gestuurd. Deze keepalives verifiëren de boomstamverbinding en dragen de signaleringsinformatie van eind aan eind. Om deze keepalives te verifiëren, geeft u de opdracht **debug vpm signaal** uit. Als er veel stammen zijn, kan de output van **debug vpm** opdrachten, u de uitvoer beperken tot één poort als u de **debug vpm port *x* opdracht optie** geeft, waar "x" de spraakpoort in kwestie is. Dit is de output van het **debug vpm signaalopdracht** die wordt uitgegeven wanneer u alle poorten bekijkt:

```
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
```

Als u dit beperkt, met het **debug vpm port *x*** opdracht, is het veel makkelijker om te interpreteren, zoals in dit voorbeeld wordt getoond:

```
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
```

De keepaliven worden gestuurd en ontvangen elke vijf seconden. De termen "verzonden naar dsp" en "ontvangen van dsp" komen uit het gezichtspunt van Cisco IOS. Vervang PBX voor DSP om dit begrijpelijker te maken. Dit zijn de boodschappen die worden gezien terwijl er geen activiteit is op de stammen. Met de aanhoudende berichten kunnen de routers op elk einde van het circuit weten dat de stammen nog steeds omhoog zijn. Als vijf van deze berichten op een rij gemist worden, gaat de kofferbak omlaag. Een van de oorzaken is dat de stammen voortdurend in een netwerk opflakkeren. Om te verifiëren of de de boomstamkeepalives van de stem worden verzonden en ontvangen, geef het **debug vpm boomstam-sc** bevel uit. Dit debug genereert geen uitvoer tot de hoofdkisten zijn gemist. Dit is een voorbeeld van de **debug vpm boomstam-sc** opdrachtoutput wanneer keepalives worden gemist:

```
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
```

Als er geen output wordt gezien wanneer het **debug vpm boomstam-sc** bevel wordt uitgegeven, dan worden geen keepalives gemist. Zelfs als keepalives worden gemist, blijft de stam omhoog tot vijf opeenvolgende boodschappen worden gemist. Dit betekent dat de verbinding 25 seconden ingedrukt moet zijn voordat de stammen naar beneden gaan.

Gemeenschappelijke problemen voor verbindingstrunks

Er zijn verschillende beestjes verbonden met de spraakboomverbindingen. Controleer deze insecten als u iets ongebruikelijks ziet. Tegen de tijd dat Cisco IOS-softwarerelease 12.2 werd vrijgegeven, waren de meeste van deze problemen aangepakt en geïntegreerd. Je kunt door de insecten kijken om jezelf te laten weten dat dit problemen zijn met oudere code. Een van de meest voorkomende problemen is om de PBX's goed te laten signaleren via de verbinding met de romp. Het lijkt een goed idee om de stammen omlaag te brengen en de routers zo te configureren dat ze

aan elk eind werken, maar de aanpak is echt contraproductief omdat alles wat je nu verandert veel voorkomt zodra de stammen zijn gevestigd. De beste manier om problemen op te lossen is met de trunks omhoog en functioneel.

Probleemoplossing starten

Het is nodig om basisgegevens te bekijken om vast te stellen dat deze correct werken:

- Zijn de stammen opgericht? Maak de **show voice call summiere** opdracht uit en zorg dat de trunks in de S_CONNECTED status zijn.
- Verwerken de DSP's pakketjes? Geef de **show voice dsp** opdracht uit om dit te verifiëren. Als u ziet dat pakketten niet door de DSP's worden verwerkt, is het omdat VAD is ingeschakeld en de pakketten onderdrukt. Zet de VAD uit, herhaal stammen en kijk weer. Controleer ook dat het pakkettellers toename wanneer de **show actieve stembriefje** opdracht wordt verstrekt. Deze opdracht toont ook aan of VAD is ingeschakeld voor het betreffende telefoonlogboek.

Als de trunks op een willekeurige plaats verbinding maken met analoge poorten, is het best om de werking van PBX in de niet-getrunkeerde modus te controleren. Als u problemen met analoge E&M-connectiviteit wilt oplossen, raadpleegt u Analoge E&M-interfacetypen en -aansluitingen voor begrip en probleemoplossing. Zodra alles op de juiste manier is geverifieerd en functioneert, brengt u de stammen omhoog en kijkt u naar de signalering die tussen de PBX-systemen wordt doorgegeven.

De ideale manier om problemen op te lossen met spraakverbinding is door het signaleren te onderzoeken dat tussen PBX's wordt doorgegeven. Het is best om een zitting van Telnet aan elke router in kwestie te hebben zodat het signaleren kan worden waargenomen aangezien het van één eind aan het andere wordt doorgegeven. In dit document wordt gebruikgemaakt van E&M-signalering, aangezien dit tamelijk populair is en rekening moet worden gehouden met het tijdstip van knipperen.

Dit is de uitvoer van de router die op PBX is aangesloten die van de vraag afkomstig is:

```
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

Deze uitvoer toont de router eindigt de vraag. Network Time Protocol (NTP) wordt gesynchroniseerd.

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
```

Opmerking: Deze uitvoer toont het signaleren dat aan beide zijden van een stembus voorkomt die E&M-knippering gebruikt. Andere typen signalering die gebruik maken van dezelfde debugs. Als u correct ingevoerde oproepen ziet (zoals hier wordt getoond), moet bidirectionele audio aanwezig zijn. Dit kan worden geverifieerd als u kijkt naar de **show voice dsp** of de **show call active voice short** commando uitvoer. Als alles er goed uitziet en u problemen met audio krijgt (geen audio of één manier) met analoge verbindingen, controleer deze verbindingen opnieuw.

Bepaal wat de oproepen zijn

Aangezien het weinig of geen goed doet om naar de **show te** kijken **actieve stem** of **de samenvatting** van spraakoproepen **te tonen** beveluitvoer voor gerangschikte vraag te tonen hebt u een eenvoudige methode nodig om te bepalen welke stemstammen actieve vraag steunen. Een van de makkelijkste manieren om dit te doen is om de **show** van de **autostam-conditionerende signalering** uit te geven in combinatie met de *omvat **parameter en ABCD als de opgenomen string*** gebruiken**, zoals hier getoond wordt:**

```
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
```

Opmerking: Deze uitvoer toont een vraag actief op timeslot 10 en een andere vraag die op timeslot acht wordt gestart. Je wilt een alias maken voor deze vrij lange opdracht als je het veel gebruikt.

DTMF-probleemoplossing

Naast de off-haak- en on-haak-signalering is het enige andere ding dat de routers tussen de PBX-systemen (behalve spraak) passeren DTMF-tonen. Er is ook een audio path, dus dit is meestal geen probleem, maar er is een probleem. Het probleem doet zich voor hoe je audio maakt over dat pad. Het is soms te verkiezen om de lage rentecodecs te gebruiken om bandbreedte op te slaan. Het probleem is dat deze lage-bits-codecs ontworpen zijn met algoritmen die geschreven zijn voor menselijke spraak. DTMF-tonen komen niet goed overeen met deze algoritmen en hebben een andere methode nodig om dit over te brengen, tenzij de klant g711 codec gebruikt. Het antwoord ligt in de opdracht **dtmf-relais**. Deze eigenschap staat DSPs aan het eind toe, begint de toon, om de DTMF toon te herkennen en het van de regelmatige audio stroom te scheiden. Op basis van hoe deze wordt geconfigureerd codeert de DSP deze tint vervolgens als een ander type Real Time Protocol (RTP)-pakket of als een h245-bericht dat afzonderlijk via de link wordt verzonden vanuit de audiobron. Dit is hetzelfde proces achter de opdrachten **fax-relay** en **modemrelais**.

Deze optie stelt een ander debug-probleem voor problemen met de romp. Hoe verifieert u welke cijfers worden doorgegeven als er geen aanroep is ingesteld en u moet die informatie uit de pakketstroom tussen de routers halen? Hoe dit te doen hangt af van het type **dtmf-relais** opdracht gebruikt wordt.

Zoals in dit voorbeeld wordt getoond, gebruikt de **dtmf-relais cisco-rtp** opdracht, een bedrijfseigen type van de lading van Cisco, zodat u naar beneden op de DSPs moet kijken om dit te zien. U kunt de opdracht **debug vpm-signaal** uitgeven in combinatie met de opdracht **debug vpm poort x/x:y.z** (om uitvoer naar de betrokken poort te beperken) om de cijfers te zien die aan de DSPs aan de vanaf-kant worden doorgegeven. Deze uitvoer wordt aan de oorspronkelijke kant weergegeven, niet aan de eindzijde.

```
*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.
```

U kunt verifiëren welke cijfers van de oorsprong kant met de dtmf-relais h245-alfanumerieke opdracht worden verzonden. De opdracht **dtmf-relais h245-alfanumeriek** gebruikt het alfanumerieke gedeelte van h.245 om de tonen over te brengen. Zoals in dit voorbeeld wordt getoond, kunnen de cijfers gemakkelijk zowel aan de van oorsprong als aan het eind van de boomstam worden gezien wanneer de **debug h245 n1** opdracht wordt geactiveerd:

Oorspronkelijke zijde:

```
*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
```

Afuiteinde:

```
*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"
```

De opdracht dtmf-relais h245-signaal is sterk vergelijkbaar en kan worden gezien wanneer

dezelfde debugs als de **dtmf-relais h245-alfanumerieke** opdracht worden gebruikt. Als u een oplossing wilt vinden voor de verbindingsstukken met de opdracht **dtmf-relais**, is het nogal moeilijk zonder de genoemde details.

Gerelateerde informatie

- Transparante CCS configureren en probleemoplossing
- Ondersteuning voor spraaktechnologie
- Productondersteuning voor spraak- en IP-communicatie
- Probleemoplossing voor Cisco IP-telefonie
- Technische ondersteuning - Cisco-systemen