Conceitos Básicos de Troubleshooting e Depuração de Chamadas VoIP

Índice

Introdução

Este documento demonstra técnicas básicas e comandos para resolver problemas e debugar redes VoIP. Uma visão geral da Arquitetura de Fluxo de Chamada por Voz e Telefonia em um Cisco Router é apresentada, seguido por uma abordagem passo a passo de troubleshooting do VoIP, apresentado nas seguintes etapas:

1. Verifique a sinalização digital e analógica.
2. Verifique os dígitos recebidos e enviados das portas da voz analógica e digital.
3. Verifique a sinalização do voip de ponta a ponta.
4. Compreenda edições do Qualidade de Serviço (QoS) de VoIP.

5. Compreenda detalhes de códigos de causa e debugar valores para VoIP.

Nota: Este documento não explica cada faceta da arquitetura de Cisco IOS® usada nos Gateway Cisco VoIP e nos porteiros. Em lugar de, pretende-se mostrar que comandos podem ser usados e que campos das saídas do comando podem ser os mais valiosos.

Cuidado: Debugar o Cisco IOS pode ser utilização de processador. Exercite o cuidado quando você usa debuga listado neste documento. Para mais informação, refira a informação importante em comandos Debug.

Debuga a necessidade de ser sido executado com os timestamps permitidos no log. Habilite o rótulo de tempo adicionando os comandos: **o service timestamps debug datetime msec**, **rótulos de tempo de serviço** *registra o datetime msec no* modo enable. Os timestamps ajudam a determinar o intervalo do tempo entre mudanças de estado.

Pré-requisitos

Requisitos

Este documento é pretendido para os equipes de rede de comunicação envolvidos no projeto e no desenvolvimento das redes voip. Leitores deste documento devem estar cientes destes tópicos:

- Configuração de VoIP
- Voz QoS

Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas. Contudo, as saídas mostradas são baseadas no Software Release 12.3(8) de Cisco IOS®.

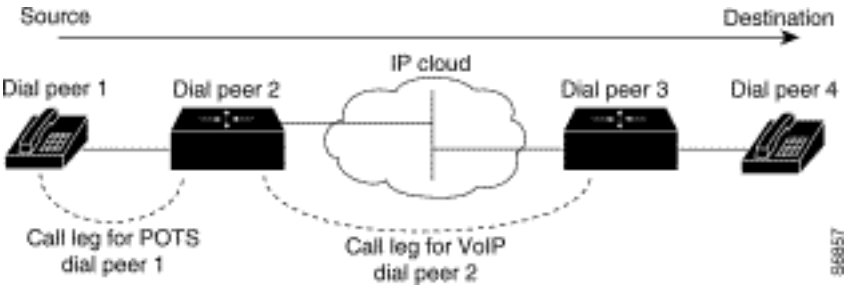
As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se você estiver trabalhando em uma rede ativa, certifique-se de que entende o impacto potencial de qualquer comando antes de utilizá-lo.

Convenções

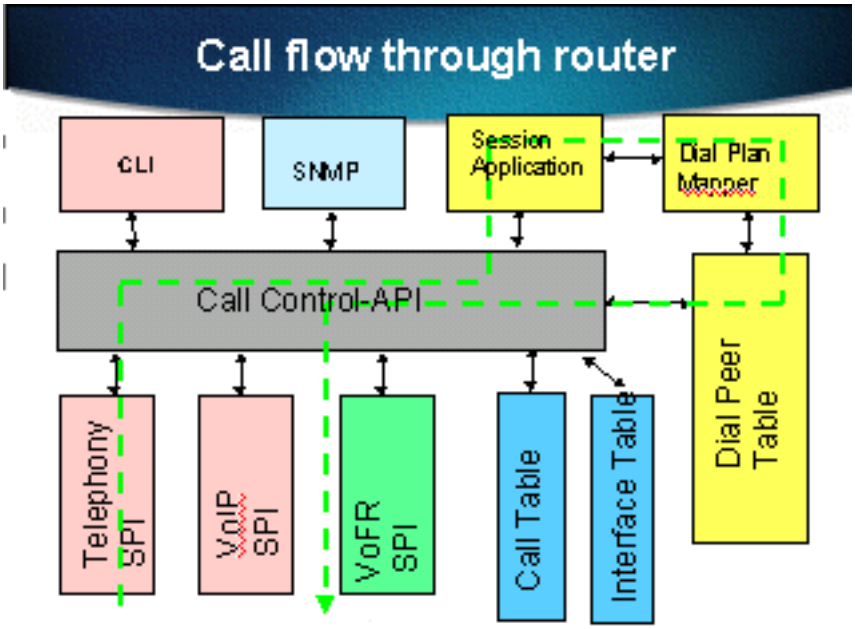
Para obter mais informações sobre convenções de documento, consulte as Convenções de dicas técnicas Cisco.

Fluxo de chamada na rede

Um fator importante a considerar antes que você comece todo o Troubleshooting de VoIP ou eliminação de erros é que as chamadas VoIP estão compostas de três trechos de chamada. Estes trechos de chamada são sistemas antigos de telefone simples da fonte (POTENCIÔMETROS), VoIP, e POTENCIÔMETROS do destino. Isto é mostrado neste diagrama. O Troubleshooting e a eliminação de erros precisam de centrar-se primeiramente independentemente e então sobre cada pé na chamada VoIP no conjunto.



Fluxo de chamada do roteador



Estas definições explicam a função dos componentes principais indicados no diagrama de fluxo de chamadas de roteador:

Controle de chamada API (interface de programação de aplicativo) — Três clientes utilizam o controle de chamada API. Os três clientes são CLI, agente do Simple Network Management Protocol (SNMP), e o aplicativo de sessão. As funções principal do controle de chamada API (igualmente referido como o CCAPI) estão a:

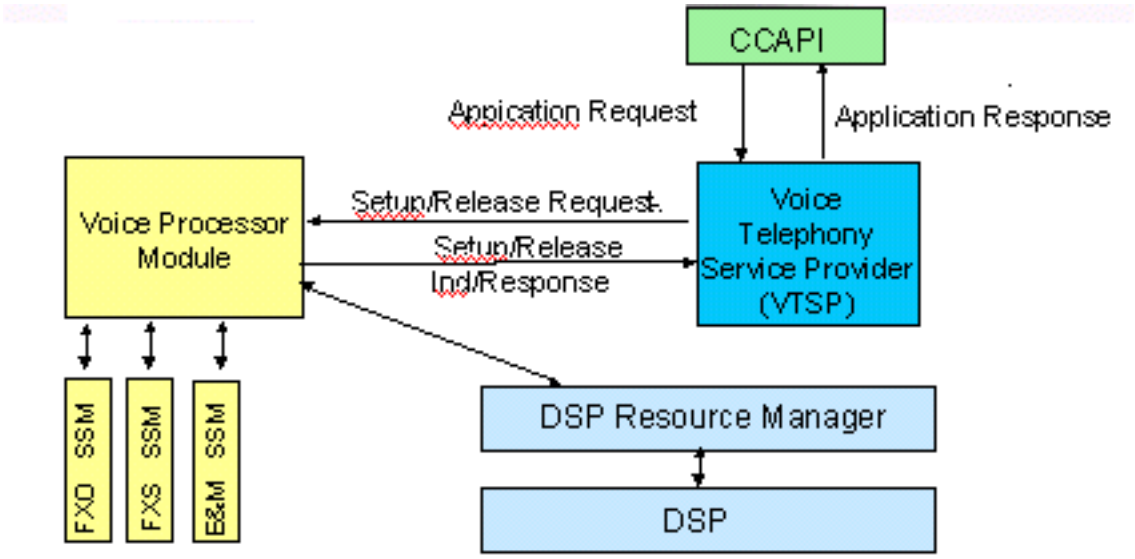
- Identifique os trechos de chamada (por exemplo, que o dial peer é ele? onde fez venha de?).
- Decida que aplicativo de sessão toma o atendimento (por exemplo, que o segura?).
- Invoque o alimentador de pacote de informação.
- Conferência os trechos de chamada junto.
- Comece gravar estatísticas de chamada.

Aplicativo de sessão e dial plan mapper — O aplicativo de sessão usa o dial plan mapper para traçar um número a um dial-peer (POTENCIÔMETROS ou voip remoto local). O Mapeador de Plano de Discagem usa a Tabela de Peer de Discagem para localizar os peers de discagem ativos.

A relação do fornecedor da telefonia e de serviço voip (SPI) — o SPI de telefonia comunica-se com os POTENCIÔMETROS (analógico: fxs, fxo, e&m Digital: isdn, qsig, e&m, e assim por diante) dial peers. O VoIP SPI é uma interface específica para os peers de VoIP. Drivers de telefonia/DSP entregam serviços para o SPI de telefonia, enquanto o SPI de VoIP conta com protocolos de sessão.

Arquitetura de interface de telefonia

Esse diagrama mostra a arquitetura dos blocos de criação de Telefonia do Cisco Router e como eles interagem uns com os outros.



Esta lista descreve as funções e as definições dos principais componentes do diagrama:

- **Interface de programação de aplicativo do Controle de chamadas (CCAPI)** — A entidade de software que estabelece, termina e constrói uma ponte sobre os trechos de chamada.
- **Provedor de serviços de telefonia por voz (VTSP)** — Um processo de IOS que preste serviços de manutenção a pedidos do controle de chamada API e formule as requisições apropriadas ao processador do sinal digital (DSP) ou ao VPM.
- **Módulo de processador de voz (VPM)** — O VPM é responsável dos processos de sinalização de construção de uma ponte sobre e de coordenação entre a máquina de estado de sinalização das portas de telefonia (SS), o gerente de recursos de DSP, e o VTSP.
- **Gerente de recursos de DSP** — O DSPRM fornece as relações por que o VTSP pode enviar e receber mensagens a e dos DSP.
- **Alimentador de pacote de informação** — Do alimentador de pacote de informação os pacotes para a frente entre os DSP e os trechos de chamada de peer.
- **Call peer** — O call peer é o trecho de chamada oposto. Esta pode ser uma outra conexão de voz de telefonia (POTENCIÔMETROS), um VoFR, um VoATM, ou uma conexão voip.

Verificar sinalização digital e analógica (trecho de chamada POTS)

Os objetivos para verificar a sinalização digital e analógica estão a:

- Determine que o em-gancho e o analógico ou a sinalização digital apropriada do fora-gancho estão recebidos.
- Determine que a sinalização apropriada do E&M, FXO e FXS está configurada no roteador e no interruptor (CO ou PBX).
- Verifique se os DSPs estão no modo de coleta de dígitos.

Os comandos esboçados nestas seções podem ser usados para verificar a sinalização.

show controllers T1 / E1 (digital)

mostre a controladores o [slot/port] T1 — Use este comando primeiro. Mostra se a conexão T1 digital entre o roteador e o interruptor (CO ou PBX) é para cima ou para baixo e se funciona corretamente. A saída deste comando olha como esta:

```
router# show controllers T1 1/0 T1 1/0 is up. Applique
type is Channelized T1 Cablelength is short 133 No
alarms detected. Framing is ESF, Line Code is B8ZS,
Clock Source is Line Primary. Data in current interval
(6 seconds elapsed): 0 Line Code Violations, 0 Path Code
Violations 0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs,
0 Degraded Mins 0 Errored Secs, 0 Bursty Err Secs, 0
Severely Err Secs, 0 Unavail Secs
```

Se usando o uso E1 o comando show controllers e1. Para obter mais informações, consulte:

- Troubleshooting de T1 Layer 1
- Fluxograma de Troubleshooting T1
- Troubleshooting Problemas de Linha Serial

show voice port

show VOICE port slot-number/port — Use este comando indicar o estado de porta e os parâmetros configurados na porta de voz das placas de interface de voz de Cisco (VIC). Como todos os comandos ios, os padrões não indicam na executar-**configuração da mostra**, mas indicam com este comando.

Este é exemplo de saída para uma porta da voz de E&M:

```
router# show voice port 1/0:1 recEive and transMit Slot
is 1, Sub-unit is 0, Port is 1 Type of VoicePort is E&M
Operation State is DORMANT Administrative State is UP No
Interface Down Failure Description is not set Noise
Regeneration is enabled Non Linear Processing is enabled
Music On Hold Threshold is Set to -38 dBm In Gain is Set
to 0 dB Out Attenuation is Set to 0 dB Echo Cancellation
is enabled Echo Cancel Coverage is set to 16 ms
Connection Mode is normal Connection Number is not set
Initial Time Out is set to 10 s Interdigit Time Out is
set to 10 s Call-Disconnect Time Out is set to 60 s
Region Tone is set for US Voice card specific Info
Follows: Out Attenuation is Set to 0 dB Echo
Cancellation is enabled Echo Cancel Coverage is set to
16 ms Connection Mode is normal (could be trunk or plar)
Connection Number is not set Initial Time Out is set to
10 s Interdigit Time Out is set to 10 s Call-Disconnect
Time Out is set to 60 s Region Tone is set for US Voice
card specific Info Follows: Signal Type is wink-start
Operation Type is 2-wire E&M Type is 1 Dial Type is dtmf
In Seizure is inactive Out Seizure is inactive Digit
Duration Timing is set to 100 ms InterDigit Duration
Timing is set to 100 ms Pulse Rate Timing is set to 10
pulses/second InterDigit Pulse Duration Timing is set to
500 ms Clear Wait Duration Timing is set to 400 ms Wink
```

```
Wait Duration Timing is set to 200 ms Wink Duration
Timing is set to 200 ms Delay Start Timing is set to 300
ms Delay Duration Timing is set to 2000 ms Dial Pulse
Min. Delay is set to 140 ms
```

debug vpm (módulo de processamento de voz)

Estes comandos são usados debugar a interface de telefonia VPM:

- debugar o sinal do vpm — Este comando é usado recolher debuga a informação sobre os eventos de sinalização e pode ser útil em problemas de resolução com sinalização a um PBX.
- debugar o spi do vpm — Este comando segue como a interface do provedor de serviços do módulo da porta de voz (SPI) conecta com o controle de chamada API. Esse comando de depuração exibe informações sobre como cada indicação de rede e solicitação de aplicativo é manipulada.
- debug vpm dsp — Este comando indica mensagens do DSP no VPM ao roteador e pode ser útil se você suspeita que o VPM não é funcional. É uma maneira simples verificar se o VPM responde indicações do fora-gancho e avaliar o sincronismo para mensagens de sinalização da relação.
- debugar o vpm todo — Este comando exec permite todos os comandos debug vpm: **debugar o spi do vpm**, **debugar o sinal do vpm**, e o **debug vpm dsp**.
- debug vpm port — Use este comando limitar o resultado do debug a uma porta particular. Por exemplo, esta saída mostra mensagens do **debug vpm dsp** somente para a porta 1/0/0:`debug vpm dsp`

  `debug vpm port 1/0/0` Para mais informação, refira comandos Debug de VoIP.

Exemplo de saída para o comando debug vpm signal

```
maui-voip-austin#debug vpm signal !--- FXS port 1/0/0
goes from the "on-hook" to "off-hook" !--- state.
htsp_process_event: [1/0/0, 1.2 , 36]
fxsls_onhook_offhook htsp_setup_ind *Mar 10
16:08:55.958: htsp_process_event: [1/0/0, 1.3 , 8] !---
Sends ringing alert to the called phone. *Mar 10
16:09:02.410: htsp_process_event: [1/0/0, 1.3 , 10]
htsp_alert_notify *Mar 10 16:09:03.378:
htsp_process_event: [1/0/0, 1.3 , 11] !--- End of phone
call, port goes "on-hook". *Mar 10 16:09:11.966:
htsp_process_event: [1/0/0, 1.3 , 6] *Mar 10
16:09:17.218: htsp_process_event: [1/0/0, 1.3 , 28]
fxsls_offhook_onhook *Mar 10 16:09:17.370:
htsp_process_event: [1/0/0, 1.3 , 41]
fxsls_offhook_timer *Mar 10 16:09:17.382:
htsp_process_event: [1/0/0, 1.2 , 7]
fxsls_onhook_release
```

Se o em-gancho e o fora-gancho não estão sinalizando corretamente, verifique estes artigos:

- Verifique se o cabeamento está correto.
- Verifique que o roteador e o interruptor (CO ou PBX) estão aterrados corretamente.
- Verifique se as duas extremidades da conexão têm configurações equivalentes de sinalização. As configurações incompatíveis podem causar incompleto ou a sinalização unidirecional.

Para obter mais informações sobre do E&M que pesquisa defeitos, refira a compreensão e pesquisando defeitos tipos de interface e arranjos de rede do E & M analógico.

Exemplo de Saída do Comando debug vpm spi

```
maui-voip-austin#debug vpm spi Voice Port Module Session
debugging is enabled !--- The DSP is put into digit
collection mode. *Mar 10 16:48:55.710:
dsp_digit_collect_on: [1/0/0] packet_len=20
channel_id=128 packet_id=35 min_inter_delay=290
max_inter_delay=3200 mim_make_time=18 max_make _time=75
min_brake_time=18 max_brake_time=75
```

Verificar dígitos recebidos e enviados (trecho de chamada do POTS)

Uma vez que a sinalização conectada e desconectada é verificada e trabalha corretamente, verifique que os dígitos correto estão recebidos ou enviados sobre a porta de voz (digital ou análogo). Um dial-peer não é combinado ou o interruptor (CO ou PBX) não pode soar a estação correspondente se incompleto ou dígitos incorreto são enviados ou recebidos. Alguns comandos que podem ser utilizados para verificar os dígitos recebidos/enviados são:

- dialplan number da mostra — Este comando é usado mostrar que dial peer é alcançado quando um número de telefone particular é discado.
- debug vtsp session — Este comando indica a informação em como cada indicação de rede e pedido de aplicativo são processados, sinalizando mensagens das indicações, e do controle DSP.
- debugar o dsp do vtsp — Mais cedo do que o Cisco IOS Software Release 12.3, este comando indica os dígitos enquanto são recebidos pela porta de voz. Contudo, no Cisco IOS Software Release 12.3 e Mais Recente, a saída do **comando debug** já não indica os dígitos. A combinação de debuga o *detalhe do* hpi e **debuga a** *notificação do* **hpi** pode ser usada para considerar os dígitos recebidos.
- debugar o vtsp todo — Este comando permite estes debuga comandos do provedor de serviços de telefonia por voz (VTSP): **o debug vtsp session**, debuga o erro do vtsp, e **debuga o dsp do vtsp**.

Para mais informação, refira comandos Debug de VoIP.

show dialplan number

mostre o dialplan number que <digit_string > — Este comando indica o dial-peer que é combinado por uma série de dígito. Se os dial peer múltiplos podem ser combinados, todos estão mostrados na ordem em que são combinados.

Nota: Você precisa de usar # sinal na extremidade dos números de telefone para o dial peers com comprimento variável a fim combinar no padrõeso de destino que termina com T.

A saída deste comando olha como esta:

```
maui-voip-austin#show dialplan number 5000 Dial string
terminator: # Macro Exp.: 5000 VoiceOverIpPeer2
information type = voice, tag = 2, destination-pattern =
```

```
`5000', answer-address = `', preference=0, group = 2,
Admin state is up, Operation state is up, incoming
called-number = `', connections/maximum = 0/unlimited,
application associated: type = voip, session-target =
`ipv4:192.168.10.2', technology prefix: ip precedence =
5, UDP checksum = disabled, session-protocol = cisco,
req-qos = best-effort, acc-qos = best-effort, dtmf-relay
= cisco-rtp, fax-rate = voice, payload size = 20 bytes
codec = g729r8, payload size = 20 bytes, Expect factor =
10, Icpif = 30, signaling-type = cas, VAD = enabled,
Poor QOV Trap = disabled, Connect Time = 25630, Charged
Units = 0, Successful Calls = 25, Failed Calls = 0,
Accepted Calls = 25, Refused Calls = 0, Last Disconnect
Cause is "10 ", Last Disconnect Text is "normal call
clearing.", Last Setup Time = 84427934. Matched: 5000
Digits: 4 Target: ipv4:192.168.10.2
```

debug vtsp session

O comando debug vtsp session mostra a informação em como o roteador interage com o DSP baseado nas indicações da sinalização da pilha de sinalização e em pedidos do aplicativo. Este **comando debug** indica a informação sobre como cada indicação de rede e pedido de aplicativo são segurados, sinalizando mensagens das indicações, e do controle DSP.

```
maui-voip-austin#debug vtsp session Voice telephony call
control session debugging is on !--- Output is
suppressed. !--- ACTION: Caller picked up handset. !---
The DSP is allocated, jitter buffers, VAD !---
thresholds, and signal levels are set. *Mar 10
18:14:22.865: dsp_set_playout: [1/0/0 (69)]
packet_len=18 channel_id=1 packet_id=76 mode=1
initial=60 min=4 max=200 fax_nom=300 *Mar 10
18:14:22.865: dsp_echo_canceller_control: [1/0/0 (69)]
packet_len=10 channel_id=1 packet_id=66 flags=0x0 *Mar
10 18:14:22.865: dsp_set_gains: [1/0/0 (69)]
packet_len=12 channel_id=1 packet_id=91 in_gain=0
out_gain=65506 *Mar 10 18:14:22.865: dsp_vad_enable:
[1/0/0 (69)] packet_len=10 channel_id=1 packet_id=78
thresh=-38act_setup_ind_ack *Mar 10 18:14:22.869:
dsp_voice_mode: [1/0/0 (69)] packet_len=24 channel_id=1
packet_id=73 coding_type=1 voice_field_size=80
VAD_flag=0 echo_length=64 comfort_noise=1
inband_detect=1 digit_relay=2
AGC_flag=0act_setup_ind_ack(): dsp_dtmf_mod
e()act_setup_ind_ack: passthru_mode = 0,
no_auto_switchover = 0dsp_dtmf_mode
(VTSP_TONE_DTMF_MODE) !--- The DSP is put into "voice
mode" and dial-tone is !--- generated. *Mar 10
18:14:22.873: dsp_cp_tone_on: [1/0/0 (69)] packet_len=30
channel_id=1 packet_id=72 tone_id=4 n_freq=2
freq_of_first=350 freq_of_second=440 amp_of_first= 4000
amp_of_second=4000 direction=1 on_time_first=65535
off_time_first=0 on_time _second=65535 off_time_second=0
```

Se se determina os dígitos não estão sendo enviados nem estão sendo recebidos corretamente, a seguir pôde possivelmente ser necessário usar ou um dígito-grabber (ferramenta de teste) ou o verificador T1 para verificar os dígitos está sendo enviado na frequência correta e no intervalo cronometrando. Se estão sendo enviados “incorretamente” para o interruptor (CO ou PBX), alguns valores no roteador ou o interruptor (CO ou PBX) pôde possivelmente precisar de ser ajustado de modo que combinassem e pudessem interoperar. Esses valores são geralmente de

duração de dígito e interdígito. Um outro artigo a examinar se os dígitos parecem ser enviados corretamente é todas as tabelas de tradução do número no interruptor (CO ou PBX) que pode adicionar ou remover dígitos.

Verificar a sinalização VoIP de ponta-a-ponta (trecho de chamada VOIP)

Depois que você verifica que os trabalhos da sinalização da porta de voz corretamente e os dígitos correto estão recebidos, mova-se para o Troubleshooting e a eliminação de erros do controle de chamada VoIP. Estes fatores explicam porque o debugging de controle de chamada pode se transformar um trabalho complexo:

- Gateways VoIP da Cisco utilizam sinalização H.323 para completar chamadas. H.323 é composto de três camadas de negociação de chamada e de estabelecimento de chamada: H.225, H.245, e H.323. Esse protocolos usam uma combinação de TCP e UDP para configurar e estabelecer uma chamada.
- O end-to-end voip debugging mostra a um número de IO máquinas de estado. Os problemas com toda a estado-máquina podem fazer com que um atendimento falhe.
- O end-to-end voip debugging pode ser muito verboso e criar muito resultado do debug.

debug voip ccapi inout

O comando primary debugar chamadas voip de ponta a ponta é debuga o inout do ccapi do voip. A saída de um atendimento debuga é mostrada nesta saída.

```
!--- Action: A VoIP call is originated through the !---
Telephony SPI (pots leg) to extension 5000. !--- Some
output is omitted. maui-voip-austin#debug voip ccapi
inout voip ccAPI function enter/exit debugging is on !--
- Call leg identification, source peer: Call !---
originated from dial-peer 1 pots !--- (extension 4000).
*Mar 15 22:07:11.959: cc_api_call_setup_ind
(vdbPtr=0x81B09EFC, callInfo={called=, calling=4000,
fdest=0 peer_tag=1}, callID=0x81B628F0) !--- CCAPI
invokes the session application. *Mar 15 22:07:11.963:
cc_process_call_setup_ind (event=0x81B67E44) handed call
to app "SESSION" *Mar 15 22:07:11.963: sess_appl:
ev(23=CC_EV_CALL_SETUP_IND), cid(88), disp(0) !---
Allocate call leg identifiers "callid = 0x59" *Mar 15
22:07:11.963: ccCallSetContext (callID=0x58,
context=0x81BAF154) *Mar 15 22:07:11.963: ccCallSetupAck
(callID=0x58) !--- Instruct VTSP to generate dialtone .
*Mar 15 22:07:11.963: ccGenerateTone (callID=0x58
tone=8) !--- VTSP passes digits to CCAPI. *Mar 15
22:07:20.275:cc_api_call_digit_begin
(vdbPtr=0x81B09EFC,callID=0x58,digit=5, flags=0x1,
timestamp=0xC2E63BB7, expiration=0x0) *Mar 15
22:07:20.279: sess_appl: ev(10=CC_EV_CALL_DIGIT_BEGIN),
cid(88), disp(0) *Mar 15 22:07:20.279: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0) *Mar
15 22:07:20.279: ssaIgnore cid(88), st(0),oldst(0),
ev(10) *Mar 15 22:07:20.327: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=5 , duration=100)
*Mar 15 22:07:20.327: sess_appl: ev(9=CC_EV_CALL_DIGIT),
```

cid(88), disp(0) *Mar 15 22:07:20.327: ssaTraceSct: cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDes t(0) *Mar 15 22:07:21.975:cc_api_call_digit_begin (vdbPtr=0x81B09EFC,callID=0x58,digit=0, flags=0x1, timestamp=0xC2E63BB7, expiration=0x0) *Mar 15 22:07:21.979: sess_appl: ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0) *Mar 15 22:07:21.979: ssaTraceSct: cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDes t(0) *Mar 15 22:07:21.979: ssaIgnore cid(88), st(0),oldst(0), ev(10) ***Mar 15 22:07:22.075: cc_api_call_digit (vdbPtr=0x81B09EFC, callID=0x58, digit=0** , duration=150) *Mar 15 22:07:22.079: sess_appl: ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0) *Mar 15 22:07:22.079: ssaTraceSct: cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0) *Mar 15 22:07:23.235: cc_api_call_digit_begin (vdbPtr=0x81B09EFC, callID=0x58, dgit=0, flags=0x1, timestamp=0xC2E63BB7, expiration=0x0) *Mar 15 22:07:23.239: sess_appl: ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0) *Mar 15 22:07:23.239: ssaTraceSct: cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0) *Mar 15 22:07:23.239: ssaIgnore cid(88), st(0),oldst(0), ev(10) ***Mar 15 22:07:23.335: cc_api_call_digit (vdbPtr=0x81B09EFC, callID=0x58, digit=0** , duration=150) *Mar 15 22:07:23.339: sess_appl: ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0) *Mar 15 22:07:23.339: ssaTraceSct: cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDes t(0) *Mar 15 22:07:25.147: cc_api_call_digit_begin (vdbPtr=0x81B09EFC, callID=0x58, d igit=0, flags=0x1, timestamp=0xC2E63BB7, expiration=0x0) *Mar 15 22:07:25.147: sess_appl: ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0) *Mar 15 22:07:25.147: ssaTraceSct: cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0) *Mar 15 22:07:25.147: ssaIgnore cid(88), st(0),oldst(0), ev(10) ***Mar 15 22:07:25.255: cc_api_call_digit (vdbPtr=0x81B09EFC, callID=0x58, digit=0** , duration=160) *Mar 15 22:07:25.259: sess_appl: ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0) *Mar 15 22:07:25.259: ssaTraceSct: cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0) *!--- Matched dial-peer 2 voip. Destination number !--- 5000* ***Mar 15 22:07:25.259: ssaSetupPeer cid(88) peer list:tag(2) called number(5000)** *Mar 15 22:07:25.259: ssaSetupPeer cid(88), **destPat(5000)**, matched(4), prefix(), peer(81C04A10) *!--- Continue to call an interface and start the !--- next call leg.* ***Mar 15 22:07:25.259: ccCallProceeding (callID=0x58**, prog_ind=0x0) ***Mar 15 22:07:25.259: ccCallSetupRequest (Inbound call = 0x58, outbound peer =2, dest=, params=0x81BAF168 mode=0, *callID=0x81B6DE58) *Mar 15 22:07:25.259: callingNumber=4000, calledNumber=5000**, redirectNumber= *!--- VoIP call setup.* ***Mar 15 22:07:25.263: ccIFCallSetupRequest:** (vdbPtr=0x81A75558, dest=, **callParams={called=5000, calling=4000, fdest=0, voice_peer_tag=2}**, mode=0x0) ***Mar 15 22:07:25.263: ccCallSetContext (callID=0x59**, context=0x81BAF3E4) *Mar 15 22:07:25.375: ccCallAlert (callID=0x58, prog_ind=0x8, sig_ind=0x1) *!--- POTS and VoIP call legs are tied together.* ***Mar 15 22:07:25.375: ccConferenceCreate (confID=0x81B6DEA0, callID1=0x58, callI D2=0x59, tag=0x0) *Mar 15 22:07:25.375: cc_api_bridge_done** (confID=0x1E, srcIF=0x81B09EFC, **srcCall ID=0x58, dstCallID=0x59**, disposition=0, tag=0x0) *!--- Exchange capability bitmasks with remote !--- the VoIP gateway !--- (Codec, VAD, VoIP or FAX, FAX-rate, and so forth).* ***Mar 15 22:07:26.127: cc_api_caps_ind**

```
(dstVdbPtr=0x81B09EFC, dstCallId=0x58, src
CallId=0x59,caps={codec=0x4, fax_rate=0x2, vad=0x2,
modem=0x1 codec_bytes=20, signal_type=0}) !--- Both
gateways agree on capabilities. *Mar 15 22:07:26.127:
cc_api_caps_ack (dstVdbPtr=0x81B09EFC, dstCallId=0x58,
src CallId=0x59, caps={codec=0x4, fax_rate=0x2, vad=0x2,
modem=0x1 codec_bytes=20, signal_type=0}) *Mar 15
22:07:26.139: cc_api_caps_ack (dstVdbPtr=0x81A75558,
dstCallId=0x59, src CallId=0x58, caps={codec=0x4,
fax_rate=0x2, vad=0x2, modem=0x1 codec_bytes=20,
signal_type=0}) *Mar 15 22:07:35.259: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=T , duration=0)
*Mar 15 22:07:35.259: sess_appl: ev(9=CC_EV_CALL_DIGIT),
cid(88), disp(0) *Mar 15 22:07:35.259: ssaTraceSct:
cid(88)st(4)oldst(3)cfid(30)csize(0)in(1) fDest(0)-
cid2(89)st2(4)oldst2(1) *Mar 15 22:07:35.399:
cc_api_call_connected (vdbPtr=0x81A75558, callID=0x59)
*Mar 15 22:07:35.399: sess_appl:
ev(8=CC_EV_CALL_CONNECTED), cid(89), disp(0) *Mar 15
22:07:35.399: ssaTraceSct:
cid(89)st(4)oldst(1)cfid(30)csize(0)in(0) fDest(0)-
cid2(88)st2(4)oldst2(4) !--- VoIP call is connected.
*Mar 15 22:07:35.399: ccCallConnect (callID=0x58) !---
VoIP call is disconnected. Cause = 0x10 *Mar 15
23:29:39.530: ccCallDisconnect (callID=0x5B, cause=0x10
tag=0x0)
```

Se o atendimento falha e a causa parece estar na parcela de VoIP da configuração de chamada, você pôde possivelmente precisar de olhar parte do H.225 ou o H.245 TCP a configuração de chamada, ao contrário apenas da parcela UDP da instalação de H.323. Os comandos que podem ser usados para depurar a configuração de chamada H.225 ou H.245 são:

- debugar transações IP tcp e **debugar o pacote IP tcp** — estes comandos examinam a parcela TCP da negociação H.225 e H.245. Retornam os endereços IP de Um ou Mais Servidores Cisco ICM NT, as portas TCP, e os estados das conexões de TCP.
- debug cch323 h225 — Este comando examina a parcela H.225 da negociação de chamada e segue a transição de estado da máquina de estado H.225 baseada no evento processado. Pense desta como o Layer 1 parte da configuração de chamada de três partes de H.323.
- debugar cch323 h245 — Este comando examina a parcela H.245 da negociação de chamada e segue a transição de estado da máquina de estado H.245 baseada nos eventos processados. Pense desta como a camada 2 parte da configuração de chamada de três partes de H.323.

Entender problemas de Qualidade de Serviço (QoS) de VoIP

Quando chamadas de VoIP são apropriadamente estabelecidas, a próxima etapa é verificar se a qualidade da voz é boa. Embora o Troubleshooting de QoS não seja coberto neste documento, estas diretrizes precisam de ser consideradas para conseguir a boa qualidade de voz:

- Compreenda quanto largura de banda uma chamada VoIP consome com cada codec. Isto inclui a camada 2 e os encabeçamentos IP/UDP/RTP. Para obter mais informações, consulte Voz sobre IP - Consumo de largura de banda por chamada.
- Compreenda que as características da rede IP os atendimentos viajam sobre. Por exemplo, a largura de banda de uma rede do Frame Relay no CIR é muito diferente do que esse acima-

CIR (ou explosão), onde os pacotes podem ser deixados cair ou enfileirado na perturbação do Frame Relay. Assegure-se de que o retardo e tremulação seja controlado e eliminado tanto quanto possível. O One-way transmite o atraso não deve exceder a Senhora 150 (pela recomendação G.114).
- Use uma técnica de enfileiramento que permita que o tráfego voip seja identificado e dado a prioridade.
- Quando você transmite VoIP sobre enlaces de velocidade baixa, considere usar a camada 2 técnicas de fragmentação de pacote, tais como o MLPPP com o Link Fragmentation and Interleaving (LFI) nos link de ponto a ponto, ou o FRF.12 em Link do Frame Relay. A fragmentação de pacotes de dados maiores permite retardo e tremulação menores na transmissão de tráfego VoIP porque os pacotes VoIP podem ser intercalados no link.
- Tente usar um codec diferente e tentar o atendimento com vad enabled e desabilitado para reduzir possivelmente para baixo a edição ao DSP, ao contrário da rede IP.

Com o VoIP, os principais itens que devem ser observados durante o Troubleshooting de QoS são os pacotes descartados e os gargalos de rede que podem causar atrasos e jitter.

Procure:

- desconexão de interfaces
- quedas de buffer
- congestionamento de interface
- congestionamento de enlace

Cada relação no trajeto da chamada VoIP precisa de ser examinada. Também, elimine gotas e congestão. Também, o retardo round trip precisa de ser reduzido tanto quanto possível. Os sibilos entre os pontos finais de VoIP dão uma indicação do retardo de round trip de um link. O retardo de round trip não deve exceder a Senhora 300 sempre que possível. Se o atraso tem que exceder este valor, os esforços precisam de ser tomados para assegurar-se de que este atraso seja constante, de modo a para não introduzir o tremor ou o retardo variável.

A verificação deve ser feita também para assegurar que o mecanismo de enfileiramento do IOS está colocando pacotes VoIP nas filas adequadas. Os comandos ios, tais como a *relação de* **fila da mostra** ou a **prioridade da mostra** podem ajudar na verificação do enfileiramento.

Detalhes dos códigos de causa e valores de depuração de VoIP

Use estas tabelas quando você lê debuga e os valores associados dentro do debugam.

Causas de desconexão de chamadas Q.931 (cause_codes em debug voip ccapi inout)

Para obter mais informações sobre dos códigos de causa Q.931 e dos valores, refira tipos de switch ISDN, códigos, e valores

Valor de causa de desconexão de chamada (em hex)	Significado e Número (em Decimais)
CC_CAUSE_UANUM = 0x1	número não-atribuído. (1)
CC_CAUSE_NO_ROUTE = 0x3	nenhuma rota ao

	destino. (3)
CC_CAUSE_NORM = 0x10	esclarecimento de chamada normal. (16)
CC_CAUSE_BUSY = 0x11	usuário ocupado. (17)
CC_CAUSE_NORS = 0x12	nenhuma resposta de usuário. (18)
CC_CAUSE_NOAN = 0x13	nenhuma resposta do usuário. (19)
CC_CAUSE_REJECT = 0x15	atendimento rejeitado. (21)
CC_CAUSE_INVALID_NUMBER = 0x1C	número inválido. (28)
CC_CAUSE_UNSP = 0x1F	Normal, não especificado. (31)
CC_CAUSE_NO_CIRCUIT = 0x22	sem circuito. (34)
CC_CAUSE_NO_REQ_CIRCUIT = 0x2C	nenhuns circuitos solicitados. (44)
CC_CAUSE_NO_RESOURCE = 0x2F	sem recurso. (47) [1]
CC_CAUSE_NOSV = 0x3F	serviço ou opção não disponível ou não especificado. (63)

[1] esta edição pode ocorrer devido a uma incompatibilidade de codec dentro da instalação de H323, assim que o primeiro passo de Troubleshooting é codificar os VoIP dial-peer para usar o codec correto.

Valores da negociação Codec (de debug voip ccapi inout)

Para obter mais informações sobre dos CODEC, refira VoIP - Compreendendo codecs: Complexidade, apoio, MOS, e negociação.

Valor de negociação	Significado
codec=0x00000001	G711 ULAW 64K PCM
codec=0x00000002	G711 ALAW 64K PCM
codec=0x00000004	G729
codec=0x00000004	G729IETF
codec=0x00000008	G729a
codec=0x00000010	G726r16
codec=0x00000020	G726r24
codec=0x00000040	G726r32
codec=0x00000080	G728
codec=0x00000100	G723r63
codec=0x00000200	G723r53

codec=0x00000400	GSMFR
codec=0x00000800	G729b
codec=0x00001000	G729ab
codec=0x00002000	G723ar63
codec=0x00004000	G723ar53
codec=0x00008000	CLEAR_CHANNEL

Tipos de tom

Tipos de tom	Significado
CC_TONE_RINGBACK 0x1	Tom de toque
CC_TONE_FAX 0x2	Tom de Fax
CC_TONE_BUSY 0x4	Tom de Ocupado
CC_TONE_DIALTONE 0x8	Tom de discagem
CC_TONE_OOS 0x10	Tom de Sem Serviço
CC_TONE_ADDR_ACK 0x20	Tom do reconhecimento do endereço
CC_TONE_DISCONNECT 0x40	Tom desconectado
CC_TONE_OFF_HOOK_NOTICE 0x80	Tom que indica que o telefone foi deixado fora do gancho
CC_TONE_OFF_HOOK_ALERT 0x100	Mais versão urgente de CC_TONE_OFF_HOOK_NOTICE
CC_TONE_CUSTOM 0x200	Tom personalizado - usado ao especificar um tom personalizado
CC_TONE_NULL 0x0	Tom nulo

Valores das capacidades de taxa de FAX e VAD

Valores	Significado
CC_CAP_FAX_NONE 0x1	Inutilizações do fax ou não disponível
CC_CAP_FAX_VOICE 0x2	Chamada de Voz
CC_CAP_FAX_144 0x4	14,400 baud
CC_CAP_FAX_96 0x8	9,600 baud
CC_CAP_FAX_72 0x10	7,200 baud
CC_CAP_FAX_48 0x20	4.800 de baud
CC_CAP_FAX_24 0x40	2.400 de baud
CC_CAP_VAD_OFF 0x1	VAD desabilitado
CC_CAP_VAD_ON 0x2	VAD Habilitado

Informações Relacionadas

- Configurando planos de discagem, correspondentes de discagem e manipulação de dígitos
- Comandos de debug VoIP
- Troubleshooting de T1 Layer 1
- Fluxograma de Troubleshooting T1
- Troubleshooting Problemas de Linha Serial
- Suporte à Tecnologia de Voz
- Suporte ao Produto de Voz e Comunicações Unificadas
- Troubleshooting da Telefonia IP Cisco
- Suporte Técnico - Cisco Systems