Solucionar problemas e depurar chamadas VoIP

Contents

Introdução

Este documento descreve as técnicas e os comandos básicos para solucionar problemas e depurar redes VoIP.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Configuração de VoIP

- QoS de voz
- Projeto e implantação de redes VoIP

Componentes Utilizados

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

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 a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

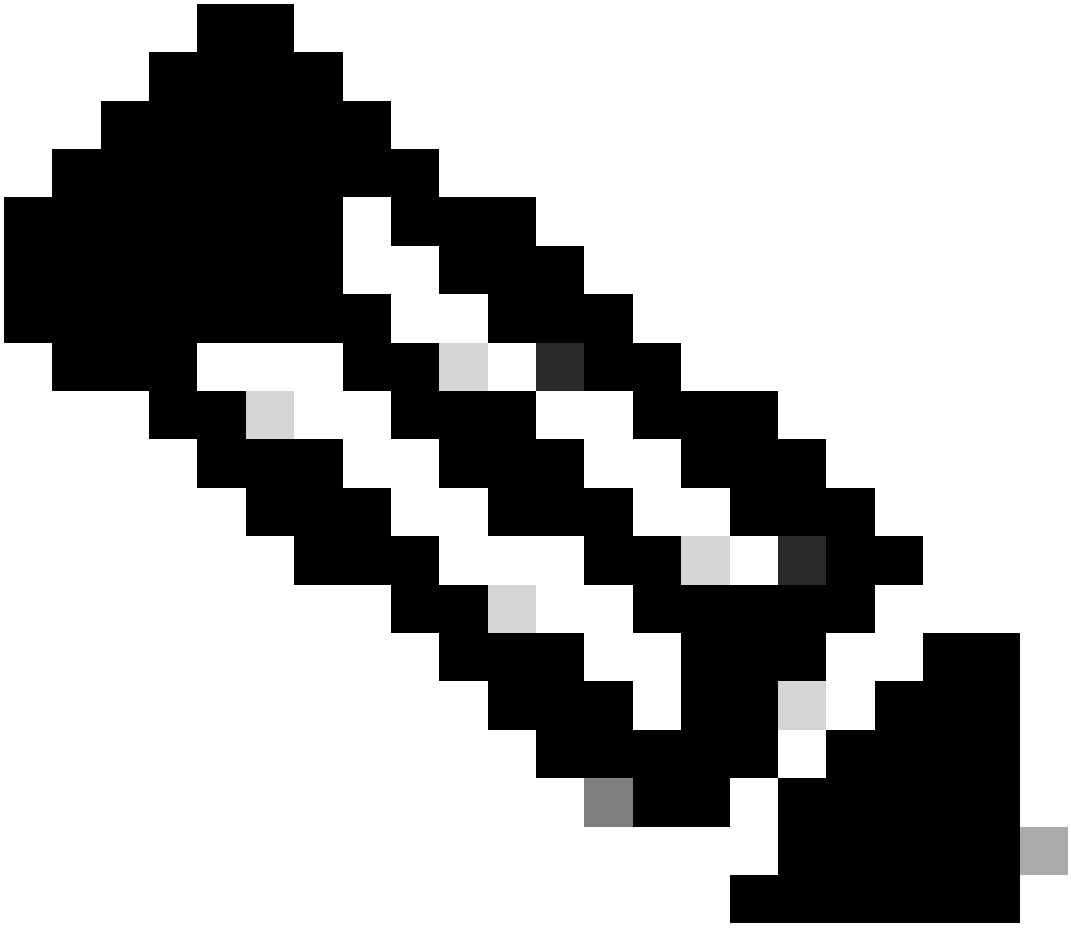
Conventions

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

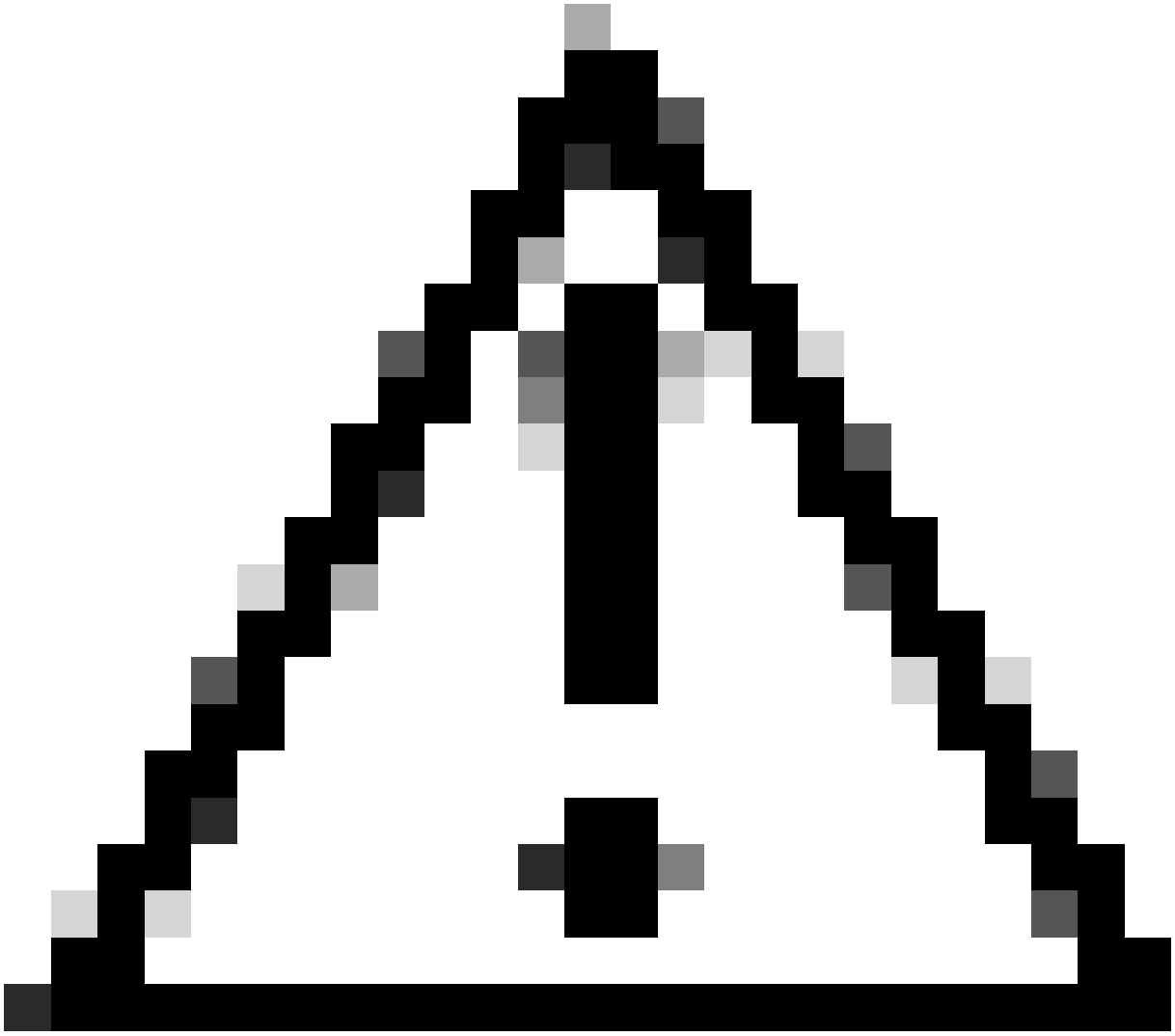
Informações de Apoio

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 de voz analógica e digital.
3. Verifique a sinalização de VoIP de ponta a ponta.
4. Entenda os problemas de Qualidade de Serviço (QoS) de VoIP.
5. Entenda os detalhes dos códigos de causa e valores de depuração para VoIP.



Observação: este documento não explica cada faceta da arquitetura do Cisco IOS usada nos gateways e gatekeepers Cisco VoIP. Em vez disso, o documento é destinado a mostrar os comandos que podem ser usados e os campos das saídas de comando que podem ser mais úteis.



Cuidado: a depuração do Cisco IOS exige muito do processador. Tenha cuidado ao usar as depurações listadas neste documento. Para obter mais informações, consulte Informações importantes sobre comandos debug.

As depurações precisam ser executadas com os carimbos de hora ativados no log. Ative o registro de data e hora com os comandos: service timestamps debug datetime msec, service timestampslog datetime msec no modo de ativação. Os carimbos de hora ajudam a determinar o intervalo de tempo entre as alterações de estado.

Fluxo de chamada na rede

Um fator importante a considerar antes de iniciar qualquer solução de problemas ou depuração de VoIP é que as chamadas de VoIP são compostas por três segmentos de chamada. Esses segmentos de chamada são POTS (Plain Old Telephone Systems) de origem, VoIP e POTS de destino. Isso é mostrado neste diagrama. A solução de problemas e a depuração precisam se concentrar primeiro em cada segmento de forma independente e depois na chamada de VoIP

como um todo.

Fluxo de chamada do roteador

Essas definições explicam a função dos principais componentes exibidos no diagrama de fluxo de chamadas do roteador:

API (Interface de programação de aplicativos) do controle de chamadas — Três clientes usam a API do controle de chamadas. Os três clientes são CLI, agente Simple Network Management Protocol (SNMP) e o aplicativo de sessão. As principais funções da API de controle de chamadas (também conhecida como CCAPI) são:

- Identifique os segmentos de chamada (por exemplo, qual é o peer de discagem? de onde ele veio?).
- Decidir qual aplicativo de sessão recebe a chamada (por exemplo, quem lida com isso?).
- Chamar o manipulador de pacotes.
- Reunir os segmentos de chamada em conferência.
- Começar a registrar as estatísticas da chamada.

Aplicativo de sessão e mapeador do plano de discagem — O aplicativo de sessão usa o mapeador do plano de discagem para mapear um número para um par de discagem (POTS local ou VoIP remoto). O Mapeador de Plano de Discagem usa a Tabela de Peer de Discagem para localizar os peers de discagem ativos.

Telefonia e interface de provedor de serviços (SPI - Service Provider Interface) de VoIP—O SPI de telefonia se comunica com os correspondentes de discagem POTS (analógico: fxs, fxo, e&m Digital: isdn, qsig, e&m e assim por diante). 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 definições dos principais componentes do diagrama:

- Interface de programação de aplicativos do controle de chamadas (CCAPI) — Entidade de software que estabelece, termina e conecta os segmentos de chamada.
- Voice Telephony Service Provider (VTSP) — Um processo do Cisco IOS que atende solicitações da API de controle de chamadas e formula as solicitações apropriadas para o processador de sinal digital (DSP) ou o VPM.

- Módulo de processador de voz (VPM) — O VPM é responsável pela ponte e coordenação dos processos de sinalização entre a máquina do estado de sinalização (SSM), o gerenciador de recursos DSP e o VTSP das portas de telefonia.
- Gerenciador de recursos DSP — O DSPRM fornece interfaces pelas quais o VTSP pode enviar e receber mensagens de e para os DSPs.
- Manipulador de pacotes — O manipulador de pacotes encaminha pacotes entre os DSPs e os segmentos de chamada dos pares.
- Par de chamadas — O par de chamadas é o segmento de chamada oposto. Pode ser outra conexão de voz de telefonia (POTS), VoFR, VoATM ou uma conexão VoIP.

Verificar sinalização digital e analógica (trecho de chamada POTS)

Verificar se a sinalização digital e analógica é para:

- Determinar se a sinalização analógica ou digital no gancho ou fora do gancho é recebida.
- Determinar se a sinalização adequada de E&M, FXO e FXS está configurada no roteador e no switch (CO ou PBX).
- Verifique se os DSPs estão no modo de coleta de dígitos.

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

show controllers T1 / E1 (digital)

show controllers t1 [slot/port] — Use este comando primeiro. Ele mostra se a conexão T1 digital entre o roteador e o switch (CO ou PBX) está ativa ou inativa e se funciona corretamente. A saída deste comando é assim:

```
<#root>

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 você usar E1, use 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 portslot-number/port — Use este comando para exibir o estado da porta e os parâmetros configurados na porta de voz das placas de interface de voz (VIC) da Cisco. Como todos os comandos do Cisco IOS, os padrões não são exibidos em show running-config , mas são exibidos com esse comando.

Esta é a saída de amostra para uma porta de voz E&M:

```
<#root>

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 para depurar a interface de telefonia VPM:

- debug vpm signal — Este comando é usado para coletar informações de depuração para eventos de sinalização e pode ser útil para resolver problemas de sinalização para um PBX.
- debug vpm spi — Este comando determina como a interface do provedor de serviços (SPI) do módulo de porta de voz é conectada à API do controle de chamadas. Esse comando de depuração exibe informações sobre como cada indicação de rede e solicitação de aplicativo é manipulada.
- debug vpm dsp — Este comando exibe as mensagens do DSP no VPM para o roteador e pode ser útil se você suspeitar que o VPM não está funcionando. É uma maneira simples de verificar se o VPM responde a indicações fora do gancho e de avaliar o tempo de sinalização das mensagens da interface.
- debug vpm all — Este comando EXEC habilita todos os comandos debug vpm: debug vpm spi, debug vpm signal e debug vpm dsp.
- debug vpm port — Use este comando para limitar a saída de depuração a uma porta

específica. Por exemplo, esta saída mostra debug vpm dspmessages somente para a porta 1/0/0:

```
debug vpm dsp

debug vpm port 1/0/0
```

Exemplo de saída para debug vpm signalCommand

```
<#root>

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 a sinalização no gancho e fora do gancho não estiver correta, verifique estes itens:

- Verifique se o cabeamento está correto.
- Verifique se o roteador e o switch (CO ou PBX) estão devidamente aterrados.
- Verifique se ambas as extremidades da conexão têm configurações de sinalização equivalentes. Configurações incompatíveis podem causar sinalização incompleta ou unidirecional.

Para obter mais informações sobre a solução de problemas de E&M, consulte Noções básicas e solução de problemas dos tipos de interface de E&M analógica e arranjos de cabeamento.

Exemplo de saída do comando debug vpm spi

```
<#root>

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)

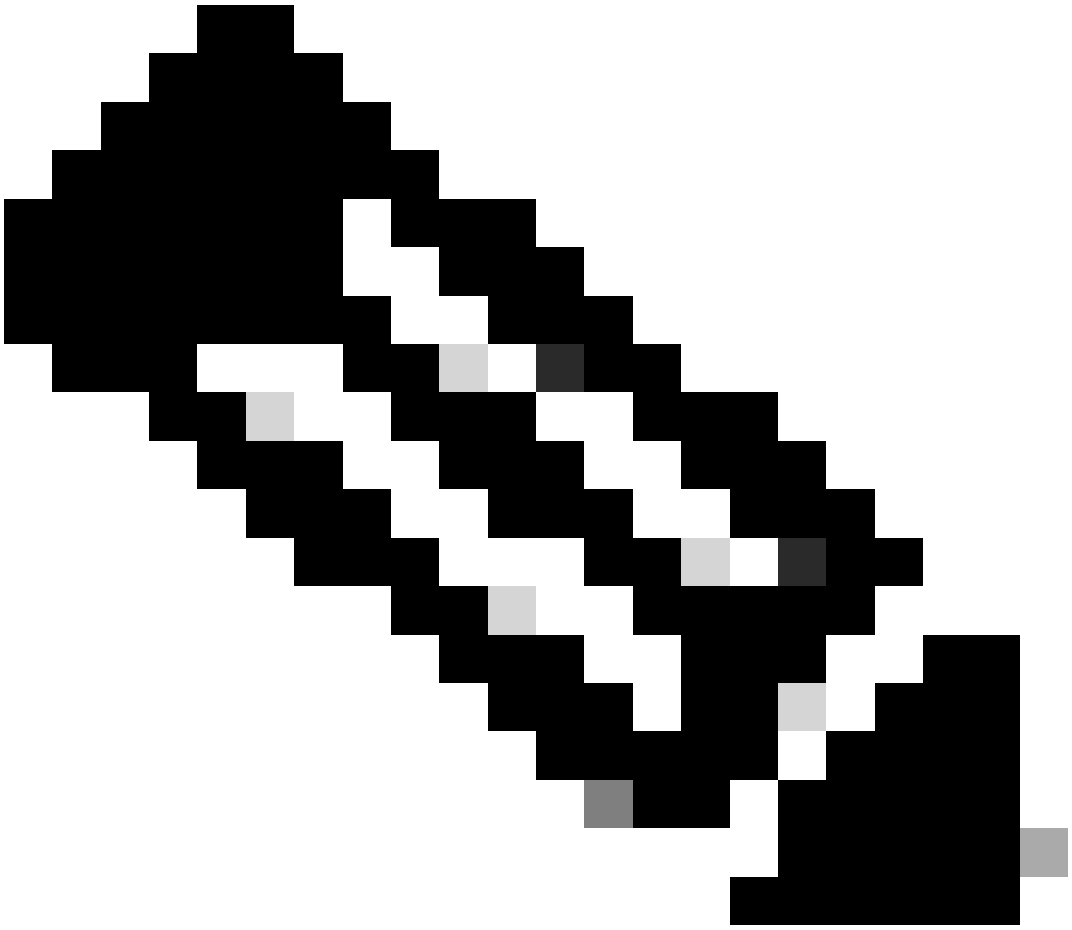
Depois que a sinalização no gancho e fora do gancho for verificada e funcionar corretamente, verifique se os dígitos corretos foram recebidos ou enviados no voice-port (digital ou analógico). Um par de discagem não corresponde e o switch (CO ou PBX) não pode tocar na estação correta, se dígitos incompletos ou incorretos são enviados ou recebidos. Alguns comandos que podem ser utilizados para verificar os dígitos recebidos/enviados são:

- show dialplan number — Este comando é usado para mostrar o par de discagem que é alcançado quando determinado número de telefone é discado.
- debug vtsp session — Este comando exibe as informações sobre como todas as indicações de rede e solicitações de aplicativo são processadas, indicações de sinalização e mensagens de controle DSP.

- debug vtsp dsp — Anterior ao software Cisco IOS versão 12.3, este comando exibe os dígitos como são recebidos pelo voice-port. No entanto, no software Cisco IOS versão 12.3 e posterior, a saída do comando debug não exibe mais os dígitos. A combinação de debug hpi detail e debug hpinotification pode ser usada para ver os dígitos de entrada.
- debug vtsp all — Este comando habilita estes comandos debug voice telephony service provider (VTSP): debug vtsp session, debug vtsp error , e debug vtsp dsp.

show dialplan number

show dialplan number <digit_string> — Este comando exibe o par de discagem correspondente a uma string de dígitos. Se vários pares de discagem corresponderem, todos eles serão mostrados na ordem em que correspondem.



Observação: você precisa usar o sinal # no final dos números de telefone para correspondentes de discagem com comprimento variável para corresponder em padrões de destino que terminam com T.

A saída deste comando é assim:

```
<#root>

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 as informações sobre como o roteador interage com o DSP com base nas indicações de sinalização da pilha de sinalização e das solicitações do aplicativo. Este comando debug exibe as informações sobre como todas as indicações de rede e solicitações de aplicativo são resolvidas, indicações de sinalização e mensagens de controle DSP.

```
<#root>

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=-38

act_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 for determinado que os dígitos não são enviados ou recebidos corretamente, então possivelmente pode ser necessário usar um capturador de dígitos (ferramenta de teste) ou testador T1 para verificar se os dígitos são enviados na frequência e intervalo de tempo corretos. Se forem enviados "incorretamente" para o switch (CO ou PBX), alguns valores no roteador ou switch (CO ou PBX) possivelmente precisarão ser ajustados para que correspondam e possam interoperar. Esses valores são geralmente de duração de dígito e interdígito. Outro item para examinar se os dígitos parecem ser enviados corretamente é qualquer tabela de conversão de números no switch (CO ou PBX) que possa adicionar ou remover dígitos.

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

Depois de verificar se a sinalização de voice-port funciona corretamente e se os dígitos corretos são recebidos, vá para a solução de problemas e depuração do controle de chamadas de VoIP. Esses fatores explicam por que a depuração do controle de chamadas pode se tornar um trabalho complexo:

- Os gateways Cisco VoIP usam a sinalização H.323 para completar chamadas. O H.323 é composto de três camadas de negociação e estabelecimento de chamadas: H.225, H.245 e H.323. Esse protocolos usam uma combinação de TCP e UDP para configurar e estabelecer uma chamada.
- A depuração VoIP de ponta a ponta mostra várias máquinas de estado do Cisco IOS. Os problemas com qualquer state-machine podem causar falha de chamada.
- A depuração de VoIP de ponta a ponta pode ser muito detalhada e criar um grande volume de saída de depuração.

debug voip ccapi inout

O comando principal para depurar chamadas de VoIP de ponta a ponta é debug voip ccapi inout. A saída de uma depuração de chamada é mostrada nesta saída.

```
<#root>

!--- 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 a chamada falhar e a causa parecer estar na parte de VoIP da configuração da chamada, você possivelmente precisará examinar a parte de TCP H.225 ou H.245 da configuração da chamada, ao contrário apenas da parte de UDP da configuração de H.323. Os comandos que podem ser usados para depurar a configuração de chamada H.225 ou H.245 são:

- debug ip tcp transactions e debug ip tcp packet — Estes comandos examinam a parte de TCP da negociação H.225 e H.245. Eles retornam os endereços IP, portas TCP e estados das conexões TCP.
- debug cch323 h225 — Este comando examina a parte H.225 da negociação de chamadas e determina a transição de estado da máquina de estado H.225 com base no evento processado. Pense nisso como a parte da camada 1 da configuração de chamada H.323 de três partes.
- debug cch323 h245 — Este comando examina a parte H.245 da negociação de chamadas e determina a transição de estado da máquina de estado H.245 com base nos eventos processados. Pense nisso como a parte da camada 2 da configuração de chamada H.323 de três partes.

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 a solução de problemas da QoS não seja contemplada neste documento, essas diretrizes precisam ser consideradas para alcançar uma boa qualidade de voz:

- Entenda a quantidade de largura de banda que uma chamada de VoIP consome com cada codec. Isso inclui cabeçalhos da camada 2 e IP/UDP/RTP. Para obter mais informações, consulte Modificar Cálculo de Consumo de Largura de Banda para Chamadas de Voz.
- Entenda as características da rede IP em que as chamadas percorrem. Por exemplo, a largura de banda de uma rede Frame-Relay no CIR é muito diferente do CIR acima (ou pico), em que os pacotes podem ser descartados ou enfileirados na nuvem Frame-Relay. Assegure que o atraso e a instabilidade sejam controlados e eliminados o máximo possível. O atraso de transmissão unidirecional não deve exceder 150 ms (por recomendação G.114).
- Use uma técnica de enfileiramento que permita que o tráfego VoIP seja identificado e priorizado.
- Ao transmitir VoIP em links de baixa velocidade, use técnicas de fragmentação de pacote de

Camada 2, como MLPPP com LFI (Fragmentação e Intercalação de Link) em links ponto-a-ponto, ou FRF.12 em links 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 tente a chamada com o VAD ativado e desativado para reduzir o problema ao DSP, em 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 interface no caminho da chamada de VoIP precisa ser examinada. Além disso, elimine os descartes e o congestionamento. Além disso, o atraso de ida e volta precisa ser reduzido o máximo possível. Os pings entre os endpoints de VoIP fazem uma indicação do atraso de ida e volta de um link. O atraso de ida e volta não deve exceder 300 ms sempre que possível. Se o atraso tiver de exceder esse valor, é necessário tomar providências para garantir que esse atraso seja constante, de modo a não introduzir instabilidade ou atraso variável.

A verificação também deve ser feita para garantir que o mecanismo de enfileiramento do Cisco IOS coloque os pacotes VoIP dentro das filas apropriadas. Os comandos do Cisco IOS, como show queue interface ou show priority podem ajudar na verificação do enfileiramento.

Detalhes dos códigos de causa e valores de depuração de VoIP

Use estas tabelas ao ler as depurações e os valores associados dentro das depurações.

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

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 para o destino. (3)
CC_CAUSE_NORM = 0x10	limpeza de chamadas normais. (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	chamada rejeitada. (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	nenhum circuito solicitado. (44)
CC_CAUSE_NO_RESOURCE = 0x2F	sem recurso. (47) 1
CC_CAUSE_NOSV = 0x3F	serviço ou opção não disponível ou não especificado. (63)

[1]Esse problema pode ocorrer devido a uma incompatibilidade de codec na configuração H323, portanto, a primeira etapa de solução de problemas é codificar os peers de discagem VoIP para usar o codec correto.

Valores da negociação Codec (de debug voip ccapi inout)

Para obter mais informações sobre CODECs, consulte Noções Básicas sobre Codecs: Complexidade, Suporte de Hardware, 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	Toque do telefone
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	Toque de confirmação de endereço
CC_TONE_DISCONNECT 0x40	Toque de finalização de chamada
CC_TONE_OFF_HOOK_NOTICE 0x80	Tom que indica que o telefone foi deixado fora do gancho
CC_TONE_OFF_HOOK_ALERT 0x100	Uma versão mais urgente do CC_TONE_OFF_HOOK_NOTICE
CC_TONE_CUSTOM 0x200	Tom Personalizado - usado quando você especifica um tom personalizado
CC_TONE_NULL 0x0	Toque nulo

Valores das capacidades de taxa de FAX e VAD

Valores	Significado
CC_CAP_FAX_NONE 0x1	O fax é desativado ou não está disponível
CC_CAP_FAX_VOICE 0x2	Chamada de Voz
CC_CAP_FAX_144 0x4	14.400 bauds
CC_CAP_FAX_96 0x8	9.600 bauds
CC_CAP_FAX_72 0x10	7.200 bauds
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

- Troubleshooting de T1 Layer 1
- Troubleshooting de T1
- Troubleshooting Problemas de Linha Serial
- Troubleshooting da Telefonia IP Cisco
- Suporte técnico e downloads da Cisco

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.