

Solución de problemas y depuración de llamadas VoIP

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Antecedentes](#)

[Flujo de llamadas en la red](#)

[Flujo de llamadas del router](#)

[Arquitectura de interfaz de telefonía](#)

[Verificar la señalización digital y analógica \(tramo de llamada de POTS\)](#)

[show controllers T1 / E1 \(digital\)](#)

[show voice port](#)

[debug vpm \(módulo del procesador de voz\)](#)

[Verificar los dígitos recibidos y los enviados \(tramo de llamada POTS\)](#)

[show dialplan number](#)

[debug vtsp session](#)

[Verificación de principio a fin de la señalización de VoIP \(Tramo de llamada VOIP\)](#)

[debug voip ccapi inout](#)

[Comprender los problemas de calidad del servicio \(QoS\) de voz sobre IP](#)

[Detalles de códigos de causas y valores de depuración para VoIP](#)

[Causas de desconexión de llamada Q.931 \(cause_codes from debug voip ccapi inout\)](#)

[Valores de negociación de codec \(desde el comando debug voip ccapi inout\)](#)

['Tipos de tonos'](#)

[Valores de capacidades FAX-Rate y VAD](#)

[Información Relacionada](#)

Introducción

Este documento describe las técnicas y comandos básicos para resolver problemas y depurar redes VoIP.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Configuración de VoIP
- QoS de voz
- Diseño e implementación de redes VoIP

Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware. No obstante, los resultados mostrados se basan en Cisco IOS® versión 12.3(8).

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Convenciones

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

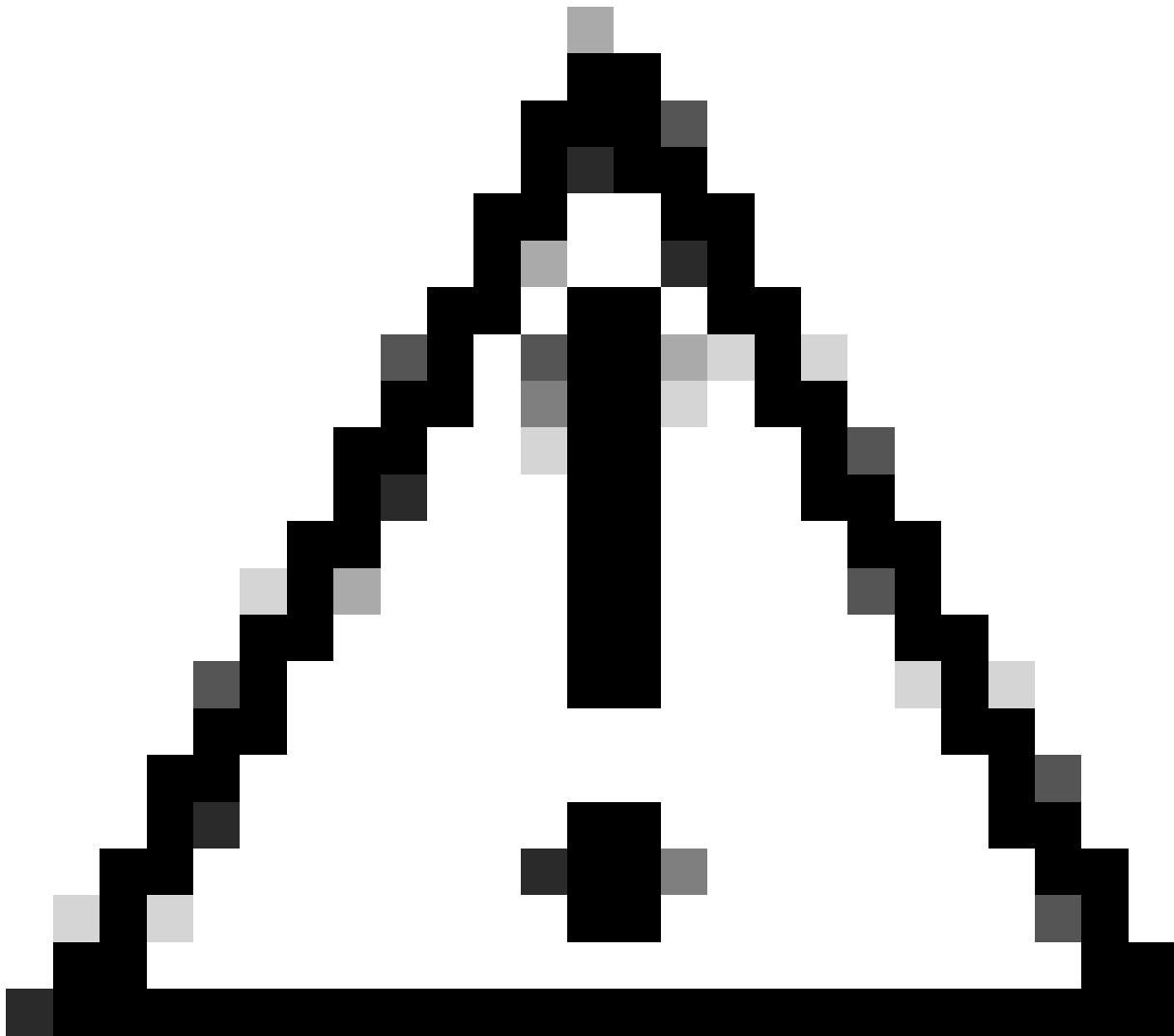
Antecedentes

En este documento se muestran técnicas y comandos básicos de Troubleshooting y debug de redes VoIP. Se incluye una introducción a la arquitectura de telefonía y flujo de llamadas de voz en un router Cisco, seguida de un enfoque de Troubleshooting de VoIP paso a paso presentado en los pasos siguientes:

1. [Verificar la señalización digital y analógica](#)
2. [Verificar los dígitos recibidos y enviados desde puertos de voz analógicos y digitales](#)
3. [Verificar la señalización de VoIP de extremo a extremo](#)
4. [Comprender los problemas de calidad de servicio \(QoS\) de VoIP](#)
5. [Comprender los detalles de códigos de causas y valores de depuración de VoIP](#)



Nota: Este documento no explica todas las facetas de la arquitectura de Cisco IOS utilizada en gateways y gatekeepers de Cisco VoIP. sino que tiene como objetivo mostrar qué comandos se pueden utilizar y qué campos de los resultados de los comandos son más valiosos.



Precaución: es intensivo para el procesador depurar Cisco IOS. Sea precavido a la hora de utilizar las depuraciones indicadas en este documento. Para obtener más información, consulte Información importante sobre comandos de depuración.

Las depuraciones deben ejecutarse con la indicación de fecha y hora en el registro. Habilite la marca de tiempo con los comandos: `service timestamps debug datetime msec`, `service timestampslog datetime msec` en el modo de habilitación. Las indicaciones de fecha y hora ayudan a determinar el intervalo de tiempo entre cambios de estado.

Flujo de llamadas en la red

Un factor importante que hay que tener en cuenta antes de empezar a solucionar problemas de VoIP o a depurar VoIP es que las llamadas VoIP están compuestas por tres tramos de llamada. Dichos tramos de llamada son los sistemas telefónicos convencionales de origen (POTS), VoIP y POTS de destino. Esto se ilustra en el diagrama siguiente: La resolución de problemas y la depuración primero tienen que centrarse independientemente en cada tramo y luego en la

llamada VoIP en general.

Flujo de llamadas del router

Las definiciones siguientes explican la función de los principales componentes que se muestran en el diagrama de flujo de la llamada del router:

API de control de llamadas (interfaz de programación de aplicaciones): Tres clientes utilizan la API de control de llamadas. Los tres clientes son: la CLI, el agente del protocolo de administración de red simple (SNMP) y la aplicación de sesión. Las funciones principales de la API de control de llamadas (también denominada CCAPI) son las siguientes:

- Identifique los tramos de llamada (por ejemplo, ¿de qué par de marcado se trata? ¿de dónde procede?).
- Decidir qué aplicación de sesión acepta la llamada (por ejemplo, ¿quién la maneja?).
- Invocar el administrador del paquete.
- Poner juntos en conferencia los tramos de llamada.
- Empezar a registrar las estadísticas de llamadas.

Aplicación de sesión y mapeador del plan de marcado: La aplicación de sesión utiliza el mapeador del plan de marcado para asociar un número a un par de marcado (POTS local o VoIP remoto) El Mapeador del plan de marcado utiliza la Tabla de par de marcado para buscar pares de marcado activos.

Telephony and VoIP Service Provider Interface (SPI): el SPI de telefonía se comunica con los dial-peers POTS (analógicos: fxs, fxo, e&m Digital: isdn, qsig, e&m, etc.). VoIP SPI es la interfaz específica para los pares de VoIP. Los controladores Telephony/DSP prestan servicios al SPI de telefonía mientras el SPI de VoIP cuenta con los protocolos de sesión.

Arquitectura de interfaz de telefonía

Este diagrama muestra la arquitectura de los bloques de construcción del router de Cisco Telephony y cómo interactúan entre sí.

La lista siguiente describe las funciones y definiciones de los principales componentes del diagrama:

- Interfaz de programación de aplicaciones de control de llamadas (CCAPI) — Entidad de software que establece, finaliza y conecta en puente los segmentos de llamada.
- Proveedor de servicios de telefonía de voz (VTSP): proceso de Cisco IOS que atiende las solicitudes de la API de control de llamadas y formula las solicitudes adecuadas al procesador de señales digitales (DSP) o al VPM.

- Módulo del procesador de voz (VPM): — El VPM se encarga de conectar en puente y coordinar los procesos de señalización entre la máquina de estado de señalización (SSM) de los puertos de telefonía, el administrador de recursos de DSP y el VTSP.
- Administrador de recursos de DSP — El DSPRM proporciona interfaces a través de las cuales el VTSP puede enviar mensajes a los DSP y recibir mensajes de éstos.
- Administrador de paquetes — El administrador de paquetes reenvía paquetes entre los DSP y los segmentos de llamadas de pares.
- Par de llamada — El par de llamada es el segmento de llamada opuesto. Puede tratarse de otra conexión de voz de telefonía (POTS), VoFR, VoATM o una conexión VoIP.

Verificar la señalización digital y analógica (tramo de llamada de POTS)

La verificación de la señalización digital y analógica es para:

- Determinar que se reciba la señalización digital o analógica activada o desactivada adecuada.
- Determinar que esté configurada la señalización E&M, FXO y FXS apropiada en ambos lados del router y del switch (CO o PBX).
- Verifique que los DSP estén en el modo de recolección de dígitos.

Los comandos indicados en estas secciones se pueden utilizar para verificar la señalización.

show controllers T1 / E1 (digital)

show controllers t1 [slot/port] — Utilice este comando primero. Muestra si la conexión T1 digital entre el router y el switch (CO o PBX) está activada o desactivada y si funciona correctamente. El resultado de este comando es como se indica a continuación:

```
<#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
```

Si utiliza E1, utilice el comando `show controllers e1`. Para obtener más información, consulte:

- [Resolución de problemas de capa 1 de T1](#)
- [Diagrama de flujo de resolución de problemas de T1](#)
- [Resolución de problemas de línea serial](#)

show voice port

`show voice port slot-number/port`: utilice este comando para visualizar el estado del puerto y los parámetros configurados en el puerto de voz de las tarjetas de interfaz de voz (VIC) de Cisco. Al igual que todos los comandos de Cisco IOS, los valores predeterminados no se muestran en `show running-config`, pero sí se muestran con este comando.

A continuación, mostramos un resultado de ejemplo de un puerto 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 del procesador de voz)

Los comandos siguientes se utilizan para depurar la interfaz de telefonía VPM:

- `debug vpm signal`: este comando se utiliza para recopilar información de depuración para los eventos de señalización y puede ser útil para resolver problemas con la señalización a un PBX.
- `debug vpm spi` — Este comando hace un seguimiento del modo en que la interfaz del proveedor de servicios (SPI) del módulo del puerto de voz interactúa con la API de control de llamadas. Este comando depurador muestra información sobre cómo es manejada cada indicación de red y petición de aplicación.
- `debug vpm dsp` — Este comando muestra mensajes del DSP, en el VPM, al router. Además, puede ser útil si sospecha que el VPM no es funcional. Es una forma sencilla de comprobar si el VPM responde a las indicaciones de descolgar y para evaluar el tiempo de los mensajes de señalización desde la interfaz.
- `debug vpm all`: este comando EXEC habilita todos los comandos `debug vpm`: `debug vpm spi`, `debug vpm signal` y `debug vpm dsp`.

- debug vpm port — Este comando sirve para limitar el resultado de la depuración a un puerto concreto. Por ejemplo, este resultado muestra debug vpm dspmessages solamente para el puerto 1/0/0:

```
debug vpm dsp
```

```
debug vpm port 1/0/0
```

Ejemplo de resultado 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]
fxs1s_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]
fxs1s_offhook_onhook
*Mar 10 16:09:17.370: htsp_process_event:
[1/0/0, 1.3 , 41] fxs1s_offhook_timer
*Mar 10 16:09:17.382: htsp_process_event:
[1/0/0, 1.2 , 7]
fxs1s_onhook_release
```

Si el estado desactivado y activado no señalizan correctamente, compruebe los elementos

siguientes:

- Verifique que el cableado sea correcto.
- Verifique que tanto el router como el switch (CO o PBX) estén correctamente conectados a tierra.
- Verifique que ambos extremos de la conexión tengan configuraciones de señalización equivalentes. Las configuraciones que no sean coincidentes pueden producir una señalización incompleta o unidireccional.

Para obtener más información acerca de la resolución de problemas E&M, consulte Comprensión y solución de problemas de tipos de interfaces E&M analógicas y disposición del cableado.

Ejemplo de resultado para debug vpm spiCommand

```
<#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 min_make_time=18 max_make
_time=75 min_brake_time=18 max_brake_time=75
```

Verificar los dígitos recibidos y los enviados (tramo de llamada POTS)

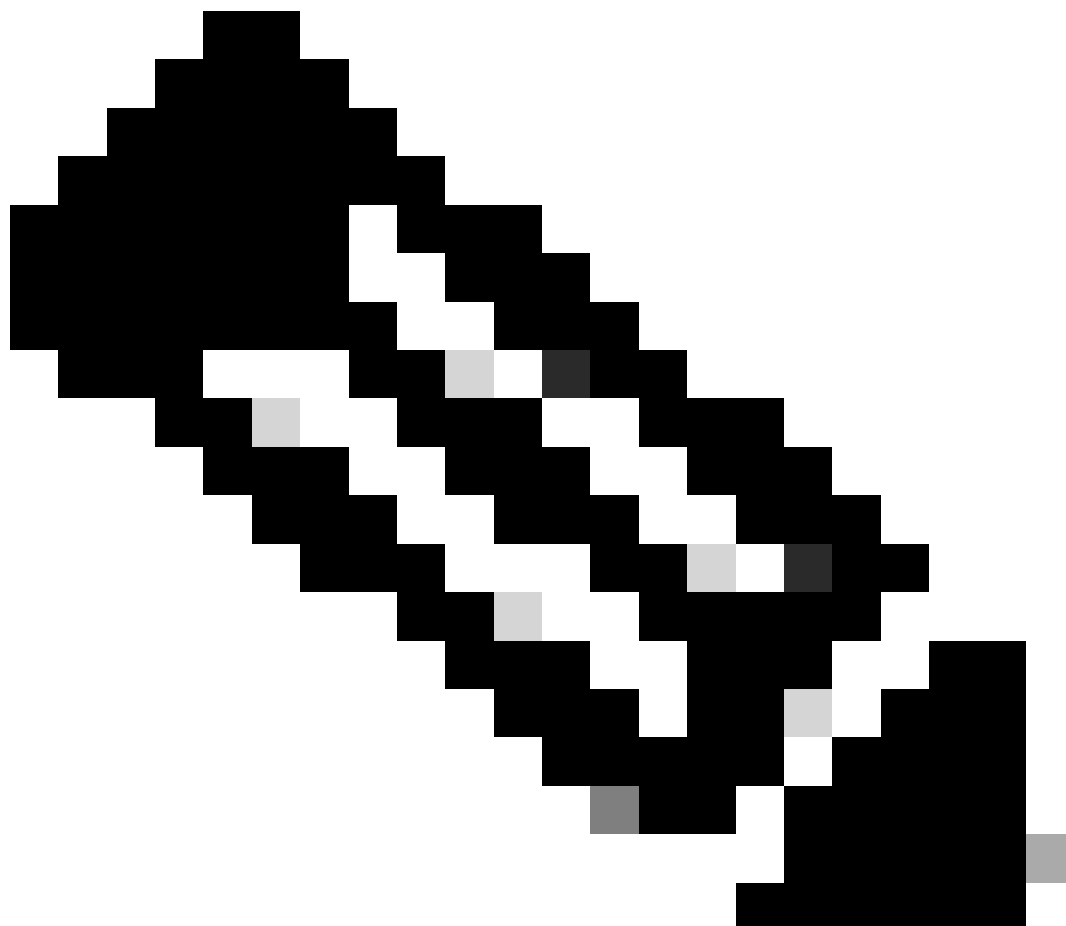
Una vez que se haya verificado la señalización del estado desactivado y el estado activado (off-hook y on-hook) y se sabe que éstos funcionan correctamente, verifique que se hayan recibido o enviado los dígitos correctos en el puerto de voz (digital o analógico). No se podrá asociar un par de marcado o un switch (CO o PBX) no podrá llamar a la estación correcta si se envían o reciben dígitos incorrectos. Algunos comandos que se pueden utilizar para verificar los dígitos recibidos/enviados son:

- show dialplan number — Este comando se utiliza para mostrar qué par de marcado se alcanza al marcar un número de teléfono concreto.

- `debug vtsp session` — Este comando muestra información acerca de cómo se procesa cada aplicación e indicación de red, las indicaciones de señalización y los mensajes de control de DSP.
- `debug vtsp dsp` — En las versiones anteriores a la versión del software Cisco IOS 12.3, este comando mostraba los dígitos a medida que el puerto de voz los recibía. No obstante, en la versión del software Cisco IOS 12.3 y posteriores, el resultado del comando `debug` ya no muestra los dígitos. La combinación de `debug hpi detail` y `debug hpinotification` se puede utilizar para ver los dígitos entrantes.
- `debug vtsp all`: este comando habilita estos comandos `debug voice telephony service provider (VTSP)`: `debug vtsp session`, `debug vtsp error` y `debug vtsp dsp`.

`show dialplan number`

`show dialplan number <digit_string>` — Este comando muestra el par de marcado que una cadena de dígitos asocia. Si se pueden asociar varios pares de marcado, se mostrarán todos en el orden en el que se asocian.



Nota: Debe utilizar el signo # al final de los números de teléfono de los pares de marcado con una longitud variable, a fin de asociar los patrones de destino que terminan con una T.

El resultado de este comando es como se indica a continuación:

```
<#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

El comando `debug vtsp session` muestra información sobre cómo el router interactúa con el DSP basándose en las indicaciones de señalización de la pila de señalización y las solicitudes de la aplicación. El comando `debug` muestra información acerca de cómo se manejan las solicitudes de aplicación y de indicación de red, las indicaciones de señalización y los mensajes de control de 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
```

Si se determina que los dígitos no se envían o reciben correctamente, entonces posiblemente puede ser necesario utilizar un capturador de dígitos (herramienta de prueba) o un probador T1 para verificar que los dígitos se envían con la frecuencia y el intervalo de tiempo correctos. Si se envían "incorrectamente" para el switch (CO o PBX), es posible que sea necesario ajustar algunos valores del router o switch (CO o PBX) para que coincidan y puedan interoperar. En general, estos son valores de duración de dígitos y de duración de interdígitos. Otro elemento que debe examinarse si parece que los dígitos se envían correctamente son las tablas de traducción

del switch (CO o PBX) que pueden agregar o eliminar dígitos.

Verificación de principio a fin de la señalización de VoIP (Tramo de llamada VOIP)

Después de verificar que la señalización del puerto de voz funciona correctamente y que se reciben los dígitos correctos, pase a la resolución de problemas y depuración del control de llamadas de VoIP. Los factores siguientes explican por qué la depuración del control de llamadas puede convertirse en una tarea compleja:

- Los gateways VoIP de Cisco utilizan la señalización H.323 para realizar llamadas. H.323 se compone de tres capas de negociación y establecimiento de llamadas: H.225, H.245 y H.323. Estos protocolos utilizan una combinación de TCP y UDP para configurar y establecer una llamada.
- La depuración VoIP de extremo a extremo muestra varias máquinas de estado de Cisco IOS. Si se producen problemas con alguna máquina de estado, una llamada puede fallar.
- La depuración VoIP de extremo a extremo puede ser muy verbosa y crear un resultado de depuración de gran volumen.

debug voip ccapi inout

El principal comando de depuración de llamadas VoIP de extremo a extremo es `debug voip ccapi inout`. A continuación, mostramos el resultado de una depuración de llamada.

```
<#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)csz(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)csz(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)csz(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)csz(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)csz(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)csz(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)csz(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)csz(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)csz(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)csz(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)
```

Si la llamada falla y la causa parece estar en la parte de VoIP de la configuración de la llamada, posiblemente necesite observar la parte de TCP H.225 o H.245 de la configuración de la llamada, en lugar de solo la parte de UDP de la configuración H.323. Los comandos que pueden usarse para depurar la configuración de llamada H.225 o H.245 son:

- `debug ip tcp transactions and debug ip tcp packet` — Estos comandos examinan la parte de TCP de la negociación de H.225 y H.245. Devuelven las direcciones IP, los puertos TCP y los estados de las conexiones TCP.
- `debug cch323 h225` — Este comando examina la parte de H.225 de la negociación de la llamada y rastrea la transición de estado de la máquina de estado H.225 en función del evento procesado. Considérelo como la parte de la capa 1 de la configuración de llamada H.323 de tres partes.
- `debug cch323 h245` — Este comando examina la parte de H.245 de la negociación de la llamada y rastrea la transición de estado de la máquina de estado H.245 en función de los eventos procesados. Considérelo como la parte de la capa 2 de la configuración de la llamada H.323 de tres partes.

Comprender los problemas de calidad del servicio (QoS) de voz sobre IP

Cuando las llamadas VoIP se establecen adecuadamente, el paso siguiente consiste en verificar que la calidad de voz sea buena. Aunque en este documento no se cubre la resolución de problemas de QoS, tenga en cuenta las directrices que exponemos a continuación para conseguir una buena calidad de voz:

- Comprenda cuánto ancho de banda consume una llamada VoIP con cada códec. Esto incluye la capa 2 y los encabezados IP/UDP/RTP. Para obtener más información, consulte [Modificación del cálculo del consumo de ancho de banda para llamadas de voz](#).
- Comprenda las características de la red IP sobre la que viajan las llamadas. Por ejemplo, el ancho de banda de una red de retransmisión de tramas en CIR es muy diferente del que está por encima de CIR (o ráfaga), en el que los paquetes se pueden abandonar o poner en cola en la nube de Frame Relay. Asegúrese de que el retraso y las fluctuaciones estén

controlados y eliminados en lo máximo posible. La demora de transmisión unidireccional no debe superar los 150 ms (según la recomendación de G.114).

- Utilice una técnica de envío a cola que permita identificar y priorizar el tráfico VoIP.
- Cuando transmite VoIP a través de links de baja velocidad, utilice técnicas de fragmentación de paquetes de Capa 2, como MLPPP con fragmentación e intercalación de enlaces (LFI) en links punto a punto, o FRF.12 en links Frame Relay. La fragmentación de paquetes de datos más grandes permite una menor fluctuación y menos retraso en la transmisión de tráfico VoIP debido a que los paquetes VoIP pueden ser entrelazados en el link.
- Pruebe con otro códec e intente llamar con VAD habilitado e inhabilitado para en lo posible acotar el problema al DSP, en contraposición con la red IP.

En la resolución de problemas de la Calidad de servicio (QoS), deben tenerse en cuenta principalmente los paquetes descartados y los embotellamientos de red que pueden causar retrasos e inestabilidad.

Busque:

- caídas de la interfaz
- caídas del búfer
- congestión de la interfaz
- congestión de link

Debe examinar cada interfaz del trayecto de la llamada VoIP. Asimismo, elimine los paquetes abandonados y la congestión. Además, el retraso de ida y vuelta debe reducirse lo máximo posible. Los pings entre los puntos finales de VoIP proporcionan una indicación del retraso de ida y vuelta de un enlace. Siempre que sea posible, la demora de ida y vuelta no debe superar los 300 ms. Si es preciso que el retraso supere este valor, deberán realizarse esfuerzos para asegurarse de que dicho retraso sea constante, de modo que no se introduzca ninguna fluctuación ni retraso variable.

También se debe realizar una verificación para garantizar que el mecanismo de cola del IOS de Cisco coloque los paquetes VoIP dentro de las colas adecuadas. Los comandos de Cisco IOS, como `show queue interface` o `show priority` pueden ayudar en la verificación de la colocación en cola.

Detalles de códigos de causas y valores de depuración para VoIP

Use estas tablas cuando lea las depuraciones y los valores asociados de las depuraciones.

Causas de desconexión de llamada Q.931 (`cause_codes` from `debug voip ccapi inout`)

| Valor de la causa de desconexión de la llamada (en hex) | Significado y número (en decimales) |
|---|---|
| CC_CAUSE_UANUM = 0x1 | número sin asignar. (1) |
| CC_CAUSE_NO_ROUTE = 0x3 | no hay ruta para el destino. (3) |
| CC_CAUSE_NORM = 0x10 | verificación normal de llamadas. apartado |
| CC_CAUSE_BUSY = 0x11 | usuario ocupado. (17) |
| CC_CAUSE_NORS = 0x12 | sin respuesta del usuario. (18) |
| CC_CAUSE_NOAN = 0x13 | sin respuesta del usuario. (19) |
| CC_CAUSE_REJECT = 0x15 | llamada rechazada. 210 |
| CC_CAUSE_INVALID_NUMBER = 0x1C | número no válido 28 |
| CC_CAUSE_UNSP = 0x1F | normal, sin especificar. (31) |
| CC_CAUSE_NO_CIRCUIT = 0x22 | sin circuito. (34) |
| CC_CAUSE_NO_REQ_CIRCUIT = 0x2C | ningún circuito solicitado. (44) |
| CC_CAUSE_NO_RESOURCE = 0x2F | sin recursos. (47) 1 |
| CC_CAUSE_NOSV = 0x3F | servicio u opción no disponible, o sin especificar (63) |

¹Este problema puede ocurrir debido a una discordancia de códec dentro de la configuración de H323, por lo que el primer paso de troubleshooting es codificar los pares de marcado VoIP para utilizar el códec correcto.

Valores de negociación de codec (desde el comando debug voip ccapi inout)

Para obtener más información sobre los CÓDEC, consulte [Introducción a los Códec: Complejidad, Soporte de Hardware, MOS y Negociación.](#)

| Valor de negociación | Significado |
|----------------------|-------------------|
| Códec = 0x00000001 | G711 ULAW 64K PCM |
| Códec = 0x00000002 | G711 ALAW 64K PCM |
| Códec = 0x00000004 | G729 |
| Códec = 0x00000004 | G729IETF |
| Códec = 0x00000008 | G729a |
| Códec = 0x00000010 | G726r16 |
| Códec = 0x00000020 | G726r24 |
| Códec = 0x00000040 | G726r32 |
| Códec = 0x00000080 | G728 |
| Códec = 0x00000100 | G723r63 |
| Códec = 0x00000200 | G723r53 |
| Códec = 0x00000400 | GSMFR |
| Códec = 0x00000800 | G729b |

| | |
|--------------------|---------------|
| Códec = 0x00001000 | G729ab |
| Códec = 0x00002000 | G723ar63 |
| Códec = 0x00004000 | G723ar53 |
| Códec = 0x00008000 | CLEAR_CHANNEL |

'Tipos de tonos'

| 'Tipos de tonos' | Significado |
|------------------------------|---|
| CC_TONE_RINGBACK 0x1 | Tono |
| CC_TONE_FAX 0x2 | Tono de fax |
| CC_TONE_BUSY 0x4 | Tono de ocupado |
| CC_TONE_DIALTONE 0x8 | Tono de marcado |
| CC_TONE_OOS 0x10 | Tono de fuera de servicio |
| CC_TONE_ADDR_ACK 0x20 | Tono de acuse de recibo de dirección |
| CC_TONE_DISCONNECT 0x40 | Tono de desconexión |
| CC_TONE_OFF_HOOK_NOTICE 0x80 | Tono que indica que el teléfono está descolgado |
| CC_TONE_OFF_HOOK_ALERT 0x100 | Versión más urgente de CC_TONE_OFF_HOOK_NOTICE |
| CC_TONE_CUSTOM 0x200 | Tono personalizado: se utiliza cuando se especifica un tono personalizado |
| CC_TONE_NULL 0x0 | Tono nulo |

Valores de capacidades FAX-Rate y VAD

| Valores | Significado |
|----------------------|-------------------------------------|
| CC_CAP_FAX_NONE 0x1 | Inhabilita fax o no está disponible |
| CC_CAP_FAX_VOICE 0x2 | Llamado de voz |
| CC_CAP_FAX_144 0x4 | 14.400 baudios |
| CC_CAP_FAX_96 0x8 | 9.600 baudios |
| CC_CAP_FAX_72 0x10 | 7.200 baudios |
| CC_CAP_FAX_48 0x20 | 4,800 baudios |
| CC_CAP_FAX_24 0x40 | 2.400 baudios |
| CC_CAP_VAD_OFF 0x1 | VAD desactivado |
| CC_CAP_VAD_ON 0x2 | VAD habilitado |

Información Relacionada

- [Resolución de problemas de capa 1 de T1](#)

- [Resolución de problemas de T1](#)
- [Resolución de problemas de línea serial](#)
- [Troubleshooting de Cisco IP Telephony](#)
- [Soporte técnico y descargas de Cisco](#)

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).