

Técnicas básicas para resolver problemas y depurar llamadas VoIP

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[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

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. [Verifique la señalización analógica y digital.](#)
2. [Verifique los dígitos recibidos y enviados de los puertos de la voz analógica y digital.](#)
3. [Verifique la señalización del VoIP de extremo a extremo.](#)
4. [Entienda los problemas de la Calidad de Servicio de VoIP \(QoS\).](#)

5. [Entienda los detalles de los códigos de la causa y haga el debug de los valores para el VoIP.](#)

Nota: Este documento no explica cada faceta de la arquitectura de Cisco IOS® usada en los gateways VoIP de Cisco y los porteros. En lugar, se piensa para mostrar qué comandos pueden ser utilizados y qué campos de las salidas de comando pueden ser los más valiosos.

Precaución: Hacer el debug del Cisco IOS puede ser hace un uso intensivo del procesador. Ejercite la precaución cuando usted utiliza los debugs enumerados en este documento. Para más información, refiera a la [información importante en los comandos Debug](#).

Los debugs necesitan ser ejecutarse con los grupos fecha/hora habilitados en el registro. Active la indicación de fecha y hora agregando los comandos: [service timestamps debug datetime msec](#), *registro de las indicaciones de fecha y hora de servicio datetime msec* en el enable mode. La ayuda de los grupos fecha/hora determina el intervalo del tiempo entre los cambios de estado.

[prerrequisitos](#)

[Requisitos](#)

Este documento se piensa para el personal de redes implicado en el diseño y el despliegue de las redes VoIP. Quienes lean este documento deben tener conocimiento de los siguientes temas:

- Configuración de VoIP
- Voz QoS

[Componentes Utilizados](#)

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware. Sin embargo, las salidas mostradas se basan en el Software Release 12.3(8) de Cisco IOS®.

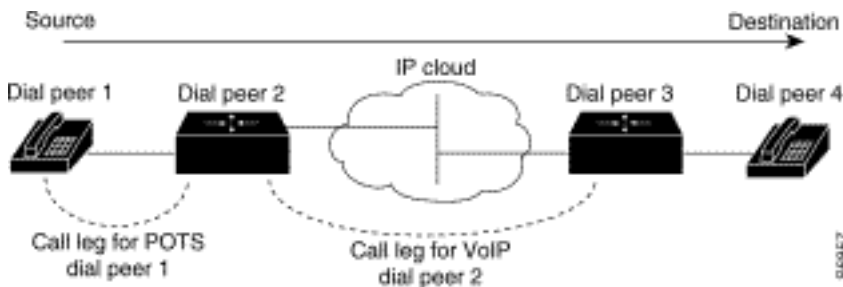
La información que se presenta en este documento se originó a partir de dispositivos dentro de un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener un comando antes de ejecutarlo.

[Convenciones](#)

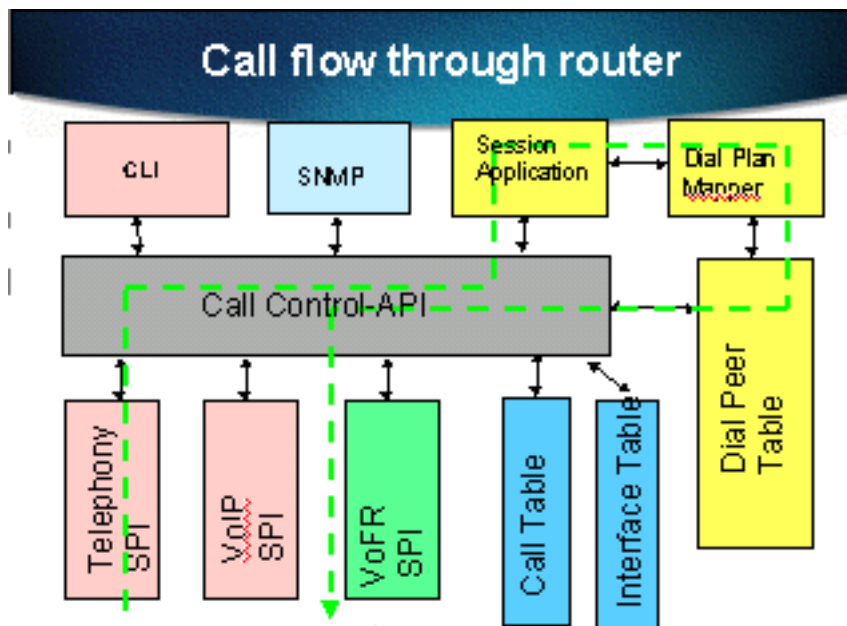
Para obtener más información sobre las convenciones del documento, consulte las [Convenciones de Consejos Técnicos de Cisco](#).

[Flujo de llamadas en la red](#)

Un factor importante a considerar antes de que usted comience cualquier Troubleshooting de VoIP o debugging es que las llamadas VoIP están compuestas de tres tramos de llamada. Estos tramos de llamada son servicios telefónicos convencionales de la fuente (CRISOLES), VoIP, y los POT de destinos. Esto se muestra en este diagrama. El troubleshooting y el debugging necesita primero independientemente y después centrarse en cada pierna en la llamada VoIP en conjunto.



Flujo de llamadas del router



Estas definiciones explican la función de los componentes principales visualizados en el diagrama del flujo de llamadas del router:

Control de llamada API (interfaz de programación de aplicaciones) — Tres clientes hacen uso del control de llamada API. Los tres clientes son CLI, agente del Simple Network Management Protocol (SNMP), y la aplicación de sesión. Las funciones principales del control de llamada API (también designado el CCAPI) están a:

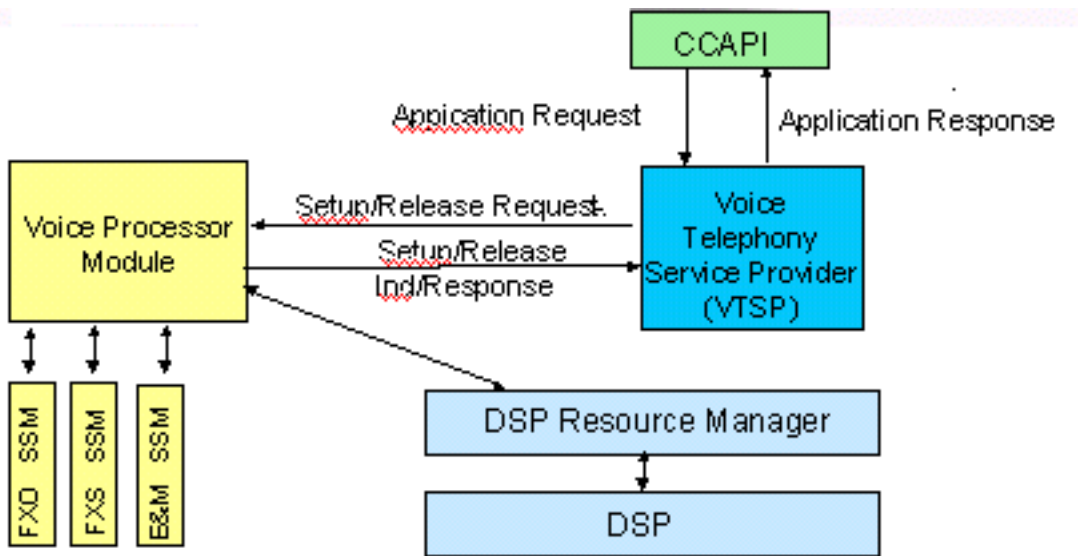
- ¿Identifique los tramos de llamada (por ejemplo, que el dial peer es él? donde lo hizo venga de?).
- Decida qué aplicación de sesión toma la llamada (por ejemplo, que la maneja?).
- Invoque al manejador de paquetes.
- Conferencia los tramos de llamada junto.
- Comience a registrar las Estadísticas de las llamadas.

Aplicación de sesión y asignador de plan de marcado — La aplicación de sesión utiliza el asignador de plan de marcado para asociar un número a un dial-peer (los POT locales o VoIP remoto). El Mapeador del plan de marcado utiliza la Tabla de par de marcado para buscar pares de marcado activos.

La interfaz del proveedor de la telefonía y del servicio de VoIP (SPI) — el SPI de telefonía comunica con los CRISOLES (análogo: fxs, fxo, e&m Digital: isdn, qsig, e&m, y así sucesivamente) dial-peers. 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.



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

- **Interfaz de programación de aplicación de control de llamada (CCAPI)** — La entidad de software que establece, termina y interliga los tramos de llamada.
- **Proveedor de servicio de telefonía de voz (VTSP)** — Un proceso IOS que mantiene las peticiones del control de llamada API y formula los pedidos apropiados al procesador de señales digitales (DSP) o al VPM.
- **Módulo del procesador de voz (VPM)** — El VPM está responsable del bridging y de los procesos de señalización de coordinación entre la máquina de estado de señalización de los puertos de telefonía (SS), el administrador de recursos DSP, y el VTSP.
- **Administrador de recursos DSP** — El DSPRM proporciona las interfaces por las cuales el VTSP puede enviar y recibir los mensajes a y desde el DSPs.
- **Manejador de paquetes** — Del manejador de paquetes los paquetes adelante entre el DSPs y los tramos de llamada de peer.
- **Par de la llamada** — El par de la llamada es el tramo opuesto de la llamada. Esto puede ser otra Conexión telefónica de voz (CRISOLES), VoFR, VoATM, o una conexión VoIP.

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

Los objetivos para verificar la señalización analógica y digital están a:

- Determine que el en-gancho y el análogo o la señalización digital apropiado del apagado-gancho está recibido.
- Determine esa señalización apropiada E&M, FXO y FXS se configura en el router y el Switch (CO o PBX).
- Verifique que los DSP estén en el modo de recolección de dígitos.

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

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

[muestre a reguladores el T1](#) [slot/port] — Utilice este comando primer. Muestra si la conexión digital T1 entre el router y el Switch (CO o PBX) está hacia arriba o hacia abajo y si funciona correctamente. La salida de este comando parece esto:

```
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 usa el uso del e1 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 de visualizar el estado de puerto y los parámetros configurados en el puerto de voz de las placas interfaz de voz de Cisco (VIC). Como todos los comandos ios, los valores por defecto no visualizan en los ejecutar-config de la **demonstración**, sino que visualizan con este comando.

Ésta es salida de muestra para un puerto de la voz 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 del procesador de voz)

Se utilizan estos comandos de hacer el debug de la interfaz de telefonía VPM:

- [señal del vpm del debug](#) — Este comando se utiliza de recoger la información para eventos de señalización del debug y puede ser útil en los problemas de resolución con la señalización a un PBX.
- [spi del vpm del debug](#) — Trazas de este comando cómo la interfaz del proveedor de servicios del módulo del puerto de voz (SPI) interconecta con el control de llamada API. Este comando depurador muestra información sobre cómo es manejada cada indicación de red y petición de aplicación.
- [debug vpm dsp](#) — Este comando visualiza los mensajes del DSP en el VPM al router y puede ser útil si usted sospecha que el VPM no es funcional. Es un método simple marcar si el VPM responde a las indicaciones descolgadas y evaluar la sincronización para los mensajes de señalización de la interfaz.
- [vpm todo del debug](#) — Este comando exec habilita todos los comandos debug vpm: **haga el debug del spi del vpm, la señal del vpm del debug, y el debug vpm dsp.**
- [debug vpm port](#) — Utilice este comando de limitar la salida de los debugs a un puerto determinado. Por ejemplo, esta salida muestra los mensajes del **debug vpm dsp** solamente para el puerto 1/0/0:debug vpm dsp

debug vpm port 1/0/0 Para más información, refiera a los [comandos Debug VoIP](#).

Salida de muestra para el 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]  
fxspls_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]  
fxspls_offhook_onhook *Mar 10 16:09:17.370:  
htsp_process_event: [1/0/0, 1.3 , 41]  
fxspls_offhook_timer *Mar 10 16:09:17.382:  
htsp_process_event: [1/0/0, 1.2 , 7]  
fxspls_onhook_release
```

Si el en-gancho y descolgado no está señalando correctamente, marca estos elementos:

- Verifique que el cableado sea correcto.
- Verifique que pongan a tierra al router y el Switch (CO o PBX) correctamente.
- Verifique que ambos extremos de la conexión cuenten con configuraciones de señalización coincidentes. Las configuraciones no coincidentes pueden causar incompleto o la

señalización unidireccional.

Para más información sobre el troubleshooting E&M, refiera a [entender y a resolver problemas los tipos de interfaz y las disposiciones de cableado del E y M analógico](#).

Salida de muestra del 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 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)

Señalización activada y desactivada se verifican una vez y trabajan correctamente, verifican los dígitos correctos se reciben o se envían encendido el puerto de voz (digital o analógico). Un dial-peer no se corresponde con o el Switch (CO o PBX) no puede sonar la estación correcta si es incompleto o los dígitos incorrectos se envían o se reciben. Algunos comandos que se pueden utilizar para verificar los dígitos recibidos/enviados son:

- [dialplan number de la demostración](#) — Se utiliza este comando de mostrar alcanzan a qué dial peer cuando se marca un número de teléfono particular.
- [debug vtsp session](#) — Este comando visualiza la información sobre cómo cada indicación de la red y pedido de aplicación se procesa, señalando las indicaciones, y los mensajes de control DSP.
- [dsp del vtsp del debug](#) — Anterior que el Cisco IOS Software Release 12.3, este comando visualiza los dígitos mientras que son recibidos por el puerto de voz. Sin embargo, en el Cisco IOS Software Release 12.3 y Posterior, la salida del **comando debug** visualiza no más los dígitos. La combinación de *detalle del* [hpi del debug](#) y de [hpi notification del debug](#) se puede utilizar para considerar los dígitos entrantes.
- [vtsp todo del debug](#) — Este comando habilita estos comandos del proveedor de servicio de telefonía de voz del debug (VTSP): **debug vtsp session**, [vtsp error del debug](#), y **dsp del vtsp del debug**.

Para más información, refiera a los [comandos Debug VoIP](#).

show dialplan number

muestre el dialplan number <digit_string > — Este comando visualiza el dial-peer que es correspondido con por una cadena de dígitos. Si los dial-peer múltiples pueden ser correspondidos con, les todos muestran en la orden en la cual los corresponden con.

Nota: Usted necesita utilizar # muestra en el extremo de los números de teléfono para el dial-peers con la Longitud variable para hacer juego en los diagramas de destinos que terminan con el T.

La salida de este comando parece esto:


```

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 la información sobre cómo el router obra recíprocamente con el DSP basado en las indicaciones de la señalización de la pila de señalización y las peticiones de la aplicación. Este **comando debug** visualiza la información sobre cómo cada indicación de la red y pedido de aplicación se maneja, señalando las indicaciones, y los mensajes de control 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

```

Si se determina los dígitos no se están enviando ni se están recibiendo correctamente, después puede ser que sea posiblemente necesario utilizar o a un dígito-capturador (herramienta de

evaluación) o el probador T1 para verificar los dígitos se está enviando en la frecuencia correcta y el intervalo que mide el tiempo. Si se están enviando "incorrectamente" para el Switch (CO o PBX), algunos valores en el router o el Switch (CO o PBX) pudo necesitar posiblemente ser ajustado de modo que correspondan con y puedan interoperar. En general, estos son valores de duración de dígitos y de duración de interdígitos. Otro elemento a examinar si los dígitos aparecen ser enviados correctamente es cualquier tabla de traducción del número en el Switch (CO o PBX) que puede agregar o quitar los dígitos.

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

Después de que usted verifique que los trabajos de la señalización del puerto de voz correctamente y los dígitos correctos estén recibidos, muévase al troubleshooting y al debugging del control de llamada VoIP. Estos factores explican porqué el debugging del control de llamados puede convertirse en una tarea compleja:

- los gateways de VoIP de Cisco utilizan señalización H.323 para completar las llamadas. H.323 se compone de tres capas de negociación de la llamada y de establecimiento de llamada: H.225, H.245, y H.323. Estos protocolos utilizan una combinación de TCP y UDP para configurar y establecer una llamada.
- El debugging VoIP de extremo a extremo muestra varias máquinas de estado IOS. Los problemas con cualquier estado-máquina pueden hacer una llamada fallar.
- El debugging VoIP de extremo a extremo puede ser muy prolijo y crear mucha salida de los debugs.

debug voip ccapi inout

El comando primary de hacer el debug de las llamadas VoIP de extremo a extremo es [inout del ccapi del voip del debug](#). La salida de un debug de la llamada se muestra en esta salida.

```
!--- 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, srcCallID=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)

```

Si la llamada falla y la causa parece estar en la porción VoIP de la configuración de la llamada, usted puede ser que necesite posiblemente mirar a la parte de H.225 o H.245 TCP la configuración de la llamada, en comparación con apenas la porción UDP de la configuración de H.323. Los comandos que pueden usarse para depurar la configuración de llamada H.225 o H.245 son:

- paquete tcp del IP de las [transacciones](#) y del debug [tcp del IP del debug](#) — estos comandos examinan la porción TCP de la negociación H.225 y H.245. Vuelven los IP Addresses, los puertos TCP, y los estados de las conexiones TCP.
- [debug cch323 h225](#) — Este comando examina la porción H.225 de la negociación de la llamada y localiza la transición de estado de la máquina de estado H.225 basada en el evento procesado. Piense en esto como la parte del Layer 1 la configuración de la llamada de H.323 de tres porciones.
- [haga el debug de cch323 h245](#) — Este comando examina la porción H.245 de la negociación de la llamada y localiza la transición de estado de la máquina de estado H.245 basada en los eventos procesados. Piense en esto como la capa a 2 partes de la configuración de la llamada de H.323 de tres porciones.

[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 el troubleshooting de QoS no se cubra en este documento, estas guías de consulta necesitan ser consideradas para alcanzar la buena calidad de voz:

- Entienda cuánto ancho de banda consume una llamada VoIP con cada codificador-decodificador. Esto incluye la capa 2 y los encabezados IP/UDP/RTP. [Para obtener más información, consulte Voz sobre IP – Consumo de ancho de banda por llamada.](#)
- Entienda que viajan las características de la red del IP las llamadas encima. Por ejemplo, el ancho de banda de una red Frame Relay en el CIR es mucho diferente que ese sobre-CIR (o la explosión), donde los paquetes se pueden caer o hacer cola en la nube de Frame Relay. Asegúrese de que la fluctuación y retraso sea controlada y eliminada tanto cuanto sea posible. Una forma transmite el retardo no debe exceder al ms 150 (por la recomendación G.114).
- Utilice una técnica de colocación en cola que permita que dan prioridad el tráfico de VoIP sea identificado y.
- Cuando usted transmite el VoIP sobre los links de baja velocidad, considere usando las técnicas de fragmentación de paquetes de la capa 2, tales como MLPPP con el Link Fragmentation and Interleaving (LFI) en los enlaces punto a punto, o el FRF.12 en los links de 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.
- Intente utilizar un diverso codificador-decodificador e intentar la llamada con el vad enabled y inhabilitada para estrechar posiblemente abajo el problema al DSP, en comparación con la red del 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

Cada interfaz en la trayectoria de la llamada VoIP necesita ser examinada. También, elimine los descensos y la congestión. También, el retardo de ida y vuelta necesita ser reducido tanto cuanto sea posible. Los ping entre los puntos extremos VoIP dan una indicación del retardo de ida y vuelta de un link. El retardo de ida y vuelta no debe exceder al ms 300 siempre que sea posible. Si el retardo tiene que exceder este valor, esfuerzos necesitan ser tomados para asegurarse que este retardo es constante, para no introducir el jitter o el Retraso variable.

Se deberá hacer la verificación para asegurarse de que el mecanismo de envío a cola del IOS coloque los paquetes VoIP dentro de las colas adecuadas. Los comandos ios, tales como *interfaz de cola de la demostración* o *prioridad de la demostración* pueden ayudar a la verificación de la espera.

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

Utilice estas tablas cuando usted lee los debugs y los valores asociados dentro de los debugs.

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

Para más información sobre los códigos de la causa y los valores del q.931, refiera a los [tipos del switch de ISDN](#), [a los códigos](#), [y a los valores](#)

Valor de la causa de desconexión de la llamada (en hex)	Significado y número (en decimales)
CC_CAUSE_UANUM = 0x1	número no asignado. (1)
CC_CAUSE_NO_ROUTE = 0x3	ninguna ruta al destino. (3)
CC_CAUSE_NORM = 0x10	Verificación normal de llamadas. (16)
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. (21)
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)

¹ este problema puede ocurrir debido a una discrepancia de códec dentro de la configuración H323, así que el primer paso de Troubleshooting es poner en hard-code los VoIP dial-peer para utilizar el codificador-decodificador correcto.

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

Para más información sobre los CODEC, refiera al [VoIP - comprensión del codecs: Complejidad, soporte, MOS, y negociación.](#)

Valor de negociación	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 tonos

'Tipos de tonos'	Significado
CC_TONE_RINGBACK 0x1	Tono de llamada
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 del acuse de recibo del direccionamiento
CC_TONE_DISCONNECT 0x40	Tono de desconexión
CC_TONE_OFF_HOOK_N OTICE 0x80	El tono indica que el teléfono fue dejado desactivado
CC_TONE_OFF_HOOK_AL ERT 0x100	Una más versión urgente de CC_TONE_OFF_HOOK_N OTICE
CC_TONE_CUSTOM 0x200	Tono personalizado - usado al especificar un tono personalizado
CC_TONE_NULL 0x0	Tono nulo

Valores de capacidades FAX-Rate y VAD

Valores	Significado
CC_CAP_FAX_NONE 0x1	Neutralizaciones del fax

	o no 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](#)

- [Configuración de planes de marcado, pares de marcado y manipulación de dígitos](#)
- [Comandos de depuración de VoIP](#)
- [Resolución de problemas de capa 1 de T1](#)
- [Diagrama de flujo de resolución de problemas de T1](#)
- [Resolución de problemas de línea serial](#)
- [Soporte de tecnología de voz](#)
- [Soporte de Productos de Voice and Unified Communications](#)
- [Troubleshooting de Cisco IP Telephony](#)
- [Soporte Técnico - Cisco Systems](#)