

Risoluzione dei problemi e debug delle chiamate VoIP

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[Premesse](#)

[Flusso di chiamata nella rete](#)

[Flusso delle chiamate del router](#)

[Architettura dell'interfaccia di telefonia](#)

[Verifica della segnalazione digitale e analogica \(POTS Call-Leg\)](#)

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

[show voice port](#)

[debug vpm \(voice processor module\)](#)

[Verifica delle cifre ricevute e inviate \(terminale di chiamata POTS\)](#)

[mostra numero dial plan](#)

[sessione vtsp di debug](#)

[Verifica della segnalazione VoIP end-to-end \(VoIP Call-Leg\)](#)

[debug voip ccapi inout](#)

[Comprendere i problemi di qualità del servizio \(QoS\) VoIP](#)

[Dettagli dei codici causa e dei valori di debug per VoIP](#)

[Q.931 Cause disconnessione chiamata \(cause_codes from debug voip ccapi inout\)](#)

[Valori di negoziazione codec \(da debug voip capi inout\)](#)

[Tipi di tono](#)

[Valori della velocità di trasmissione e delle capacità VAD](#)

[Informazioni correlate](#)

Introduzione

In questo documento vengono descritte le tecniche e i comandi di base per la risoluzione dei problemi e il debug delle reti VoIP.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Configurazione VoIP
- QoS voce
- Progettazione e installazione di reti VoIP

Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware. Tuttavia, gli output mostrati sono basati sul software Cisco IOS® versione 12.3(8).

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni nei suggerimenti tecnici.

Premesse

In questo documento vengono illustrate le tecniche e i comandi base per risolvere i problemi ed effettuare il debug sulle reti VoIP. Viene presentata una panoramica dell'architettura Voice Call Flow and Telephony sui router Cisco e una procedura dettagliata per risolvere i problemi VoIP:

1. [Verificare la segnalazione digitale e analogica.](#)
2. [Verificare le cifre ricevute e inviate dalle porte vocali analogiche e digitali.](#)
3. [Verificare la segnalazione VoIP end-to-end.](#)
4. [Comprendere i problemi relativi alla qualità del servizio \(QoS\) VoIP.](#)
5. [Comprendere i dettagli dei codici causa e dei valori di debug per VoIP.](#)



Nota: questo documento non spiega tutti gli aspetti dell'architettura Cisco IOS utilizzata nei gateway e gatekeeper VoIP Cisco. Al contrario, viene creato per mostrare quali comandi possono essere utilizzati e quali campi dagli output del comando possono essere più utili.



Attenzione: il debug su Cisco IOS richiede molte risorse del processore. procedere con cautela quando si utilizzano i debug elencati in questo documento. Per ulteriori informazioni, consultare le [informazioni importanti sui comandi di debug](#).

I debug devono essere eseguiti con l'opzione timestamp abilitata nel log. Abilitare l'indicatore orario con i seguenti comandi: `service timestamp debug datetime msec`, `service timestampslog datetime msec` in modalità abilitazione. I timestamp consentono di determinare l'intervallo di tempo tra le modifiche dello stato.

Flusso di chiamata nella rete

Un fattore importante da considerare prima di avviare la risoluzione dei problemi o il debug VoIP è che le chiamate VoIP sono costituite da tre componenti di chiamata. I terminali di chiamata sono POTS (Plain Old Telephone Systems) di origine, VoIP e POTS di destinazione. Lo schema seguente ne dà una rappresentazione grafica. La risoluzione dei problemi e il debug devono concentrarsi prima su ogni tappa in modo indipendente e poi sulla chiamata VoIP nel suo

complesso.

Flusso delle chiamate del router

Queste definizioni spiegano la funzione dei componenti principali visualizzati nel diagramma del flusso di chiamate del router:

API di controllo delle chiamate (Application Programming Interface): tre client utilizzano l'API di controllo delle chiamate. I tre client sono CLI, l'agente SNMP (Simple Network Management Protocol) e l'applicazione di sessione. Le principali funzioni dell'API di controllo delle chiamate (detta anche CCAPI) sono:

- Identificare le gambe di chiamata (ad esempio, qual è il peer da comporre? da dove proviene?).
- Decidere quale applicazione di sessione accetta la chiamata (ad esempio, chi la gestisce?).
- Richiama il gestore di pacchetti.
- Convinciamo le gambe di richiamo.
- Inizia a registrare le statistiche sulle chiamate.

Applicazione sessione e mappatore dial plan: l'applicazione sessione utilizza il mappatore dial plan per mappare un numero a un dial-peer (POTS locale o VoIP remoto). Dial Plan Mapper utilizza Dial Peer Table per trovare i dial-peer attivi.

Telephony and VoIP Service Provider Interface (SPI): lo SPI di telefonia comunica con i dial-peer POTS (analogici: fxs, fxo, e&m digitali: isdn, qsig, e&m, e così via). VoIP SPI è l'interfaccia specifica dei peer VoIP. I driver di telefonia/DSP forniscono servizi all'SPI di telefonia, mentre l'SPI di VoIP si basa sui protocolli di sessione.

Architettura dell'interfaccia di telefonia

Il diagramma mostra l'architettura dei componenti di telefonia dei router Cisco e come interagiscono tra loro.

L'elenco descrive le funzioni e le definizioni dei principali componenti del diagramma:

- Call Control Application Programming Interface (CCAPI): entità software che stabilisce, termina e crea un bridge per i componenti di chiamata.
- Voice Telephony Service Provider (VTSP): processo Cisco IOS che gestisce le richieste dall'API di controllo delle chiamate e formula le richieste appropriate al DSP (Digital Signal Processor) o al VPM.
- Voice Processor Module (VPM): il VPM è responsabile del bridging e del coordinamento dei processi di segnalazione tra il computer a stato di segnalazione delle porte di telefonia

(SSM), DSP Resource Manager e VTSP.

- Gestione risorse DSP: DSPRM fornisce interfacce tramite le quali il VTSP può inviare e ricevere messaggi da e verso i DSP.
- Gestore pacchetti: il gestore di pacchetti inoltra i pacchetti tra i DSP e le gambe delle chiamate peer.
- Peer chiamata (Call Peer) - Il Peer chiamata è la gamba di chiamata opposta. Questa può essere un'altra connessione telefonica voce (POTS), VoFR, VoATM o una connessione VoIP.

Verifica della segnalazione digitale e analogica (POTS Call-Leg)

La verifica della segnalazione digitale e analogica ha lo scopo di:

- Verificare di aver ricevuto i segnali digitali o analogici appropriati on-hook e off-hook.
- Verificare che la segnalazione E&M, FXO e FXS corretta sia configurata sul router e sullo switch (CO o PBX).
- Verificare che i DSP siano in modalità raccolta cifre.

I comandi descritti in queste sezioni possono essere utilizzati per verificare la segnalazione.

show controllers T1 / E1 (digitale)

show controllers t1 [slot/port] - Utilizzare questo comando per primo. Indica se la connessione digitale T1 tra il router e lo switch (CO o PBX) è attiva o inattiva e se funziona correttamente.

L'output di questo comando è simile al seguente:

```
<#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 si utilizza E1, usare il comando show controllers e1. Per ulteriori informazioni, consultare:

- [Risoluzione dei problemi di T1 Layer 1](#)
- [Diagramma di flusso per la risoluzione dei problemi T1](#)
- [Risoluzione dei problemi della linea seriale](#)

show voice port

show voice port slot-number/port: consente di visualizzare lo stato della porta e i parametri configurati sulla porta voce delle schede di interfaccia voce (VIC) Cisco. Come tutti i comandi Cisco IOS, i valori predefiniti non vengono visualizzati in show running-config, ma vengono visualizzati con questo comando.

Di seguito viene riportato un esempio di output per una porta voce di 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 (voice processor module)

Questi comandi vengono utilizzati per eseguire il debug dell'interfaccia di telefonia VPM:

- debug vpm signal: questo comando viene utilizzato per raccogliere informazioni di debug per segnalare eventi e può essere utile per risolvere problemi di segnalazione a un PBX.
- debug vpm spi: questo comando traccia il modo in cui l'interfaccia SPI (Voice Port Module Service Provider Interface) si interfaccia con l'API di controllo delle chiamate. Questo comando debug visualizza informazioni sulla gestione di ciascuna indicazione di rete e richiesta di applicazione.
- debug vpm dsp: questo comando visualizza i messaggi dal DSP sul VPM al router e può essere utile se si sospetta che il VPM non funzioni. È un modo semplice per controllare se il VPM risponde alle indicazioni off-hook e per valutare la temporizzazione dei messaggi di segnalazione dall'interfaccia.
- debug vpm all: questo comando di esecuzione abilita tutti i comandi debug vpm: debug vpm spi, debug vpm signal e debug vpm dsp.
- debug vpm port: utilizzare questo comando per limitare l'output del debug a una porta specifica. Ad esempio, questo output visualizza i messaggi debug vpm dspmessage solo per la porta 1/0/0:

```
debug vpm dsp
```

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

Output di esempio per 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
```

Se l'on-hook e lo off-hook non segnalano correttamente, controllare i seguenti elementi:

- Verificare che il cablaggio sia corretto.
- Verificare che la messa a terra del router e dello switch (CO o PBX) sia corretta.
- Verificare che entrambe le estremità della connessione dispongano di configurazioni di segnalazione equivalenti. Configurazioni non corrispondenti possono causare una

segnalazione incompleta o unidirezionale.

Per ulteriori informazioni sulla risoluzione dei problemi relativi a E&M, consultare il documento [sulla descrizione e la risoluzione dei problemi relativi ai tipi di interfaccia E&M analogica e alle disposizioni dei cavi.](#)

Output di esempio per 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
```

Verifica delle cifre ricevute e inviate (terminale di chiamata POTS)

Dopo aver verificato che i segnali on-hook e off-hook funzionino correttamente, verificare che le cifre corrette siano state ricevute o inviate sulla porta voce (digitale o analogica). Un dial-peer non corrisponde oppure lo switch (CO o PBX) non è in grado di far squillare la stazione corretta se vengono inviate o ricevute cifre incomplete o errate. Di seguito sono riportati alcuni comandi che è possibile utilizzare per verificare le cifre ricevute/inviate:

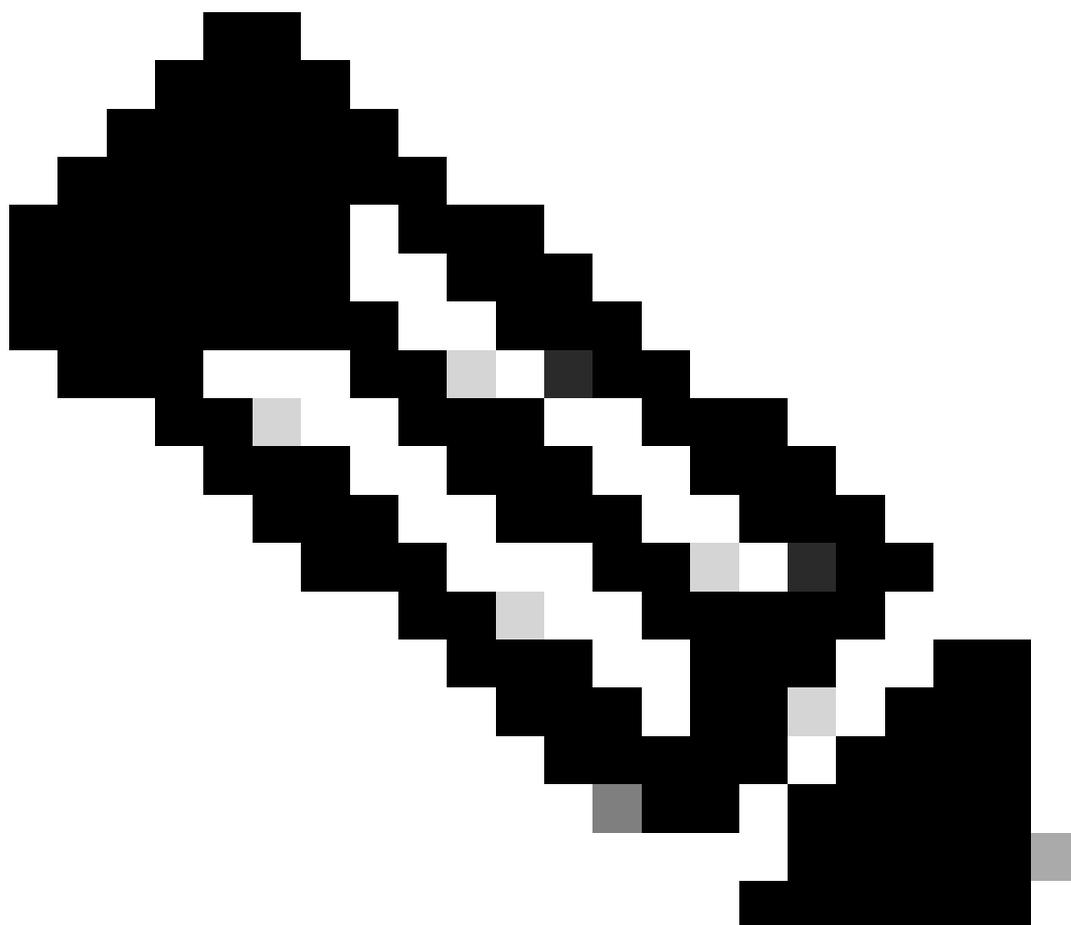
- `show dialplan number`: questo comando viene utilizzato per visualizzare il peer di composizione raggiunto quando viene composto un determinato numero di telefono.
- `debug vtsp session`: questo comando visualizza informazioni su come vengono elaborate le indicazioni di rete e le richieste di applicazione, le indicazioni di segnalazione e i messaggi di controllo DSP.
- `debug vtsp dsp`: versione precedente al software Cisco IOS versione 12.3, questo comando visualizza le cifre ricevute dalla porta voce. Tuttavia, nel software Cisco IOS versione 12.3 e successive, l'output del comando debug non visualizza più le cifre. La combinazione dei dettagli hpi di debug e della notifica hpi di debug può essere utilizzata per visualizzare le

cifre in arrivo.

- debug vtsp all: questo comando abilita i seguenti comandi debug voice telephony service provider (VTSP): debug vtsp session, debug vtsp error, debug vtsp dsp.

mostra numero dial plan

show dialplan number <digit_string>: questo comando visualizza il dial-peer corrispondente a una stringa di cifre. Se è possibile abbinare più peer di connessione, questi verranno visualizzati nell'ordine in cui sono stati abbinati.



Nota: è necessario utilizzare il simbolo # alla fine dei numeri di telefono per i peer della chiamata di lunghezza variabile in modo da trovare una corrispondenza con i modelli di destinazione che terminano con T.

L'output di questo comando è simile al seguente:

```
<#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
```

sessione vtsp di debug

Il comando debug vtsp session mostra le informazioni su come il router interagisce con il DSP in base alle indicazioni di segnalazione dello stack di segnalazione e alle richieste dell'applicazione. Questo comando debug visualizza informazioni sulla gestione di ciascuna indicazione di rete e richiesta di applicazione, sulle indicazioni di segnalazione e sui messaggi di controllo 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_payout
: [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 si determina che le cifre non sono inviate o ricevute correttamente, allora può essere necessario utilizzare un digt-grabber (strumento di prova) o un tester T1 per verificare che le cifre siano inviate alla frequenza e all'intervallo di tempo corretti. Se i dati vengono inviati "in modo non corretto" per lo switch (CO o PBX), alcuni valori sul router o sullo switch (CO o PBX) potrebbero richiedere una regolazione per consentire la corrispondenza e il funzionamento. Si tratta in genere di valori di durata e di durata tra cifre. Un altro elemento da esaminare se le cifre vengono inviate correttamente è rappresentato da tabelle di conversione dei numeri nello switch (CO o PBX) in grado di aggiungere o rimuovere cifre.

Verifica della segnalazione VoIP end-to-end (VoIP Call-Leg)

Dopo aver verificato che la segnalazione della porta voce funzioni correttamente e che vengano ricevute le cifre corrette, passare alla sezione Risoluzione dei problemi e debug del controllo delle chiamate VoIP. Questi fattori spiegano perché il debug del controllo delle chiamate può diventare un processo complesso:

- I gateway VoIP Cisco utilizzano la segnalazione H.323 per completare le chiamate. H.323 è costituito da tre livelli di negoziazione delle chiamate e determinazione delle chiamate:

H.225, H.245 e H.323. Questi protocolli utilizzano una combinazione di TCP e UDP per configurare e stabilire una chiamata.

- Il debug VoIP end-to-end mostra una serie di macchine a stato Cisco IOS. Problemi con qualsiasi macchina a stati possono causare il mancato completamento di una chiamata.
- Il debug VoIP end-to-end può essere molto dettagliato e creare un elevato output di debug.

debug voip ccapi inout

Il comando principale per il debug delle chiamate VoIP end-to-end è debug voip capi inout. In questo output viene visualizzato l'output del debug di una chiamata.

```
<#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)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_app1:
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_app1:
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 la chiamata ha esito negativo e la causa sembra trovarsi nella parte VoIP della configurazione della chiamata, è possibile che sia necessario controllare la parte TCP H.225 o H.245 della configurazione della chiamata, in contrapposizione alla sola parte UDP della configurazione

H.323. I comandi che possono essere utilizzati per eseguire il debug della configurazione delle chiamate H.225 o H.245 sono:

- debug ip tcp transaction e debug ip tcp packet: questi comandi esaminano la parte TCP della negoziazione H.225 e H.245. Restituiscono gli indirizzi IP, le porte TCP e gli stati delle connessioni TCP.
- debug cch323 h225: questo comando esamina la parte H.225 della negoziazione delle chiamate e traccia la transizione di stato della macchina a stati H.225 in base all'evento elaborato. Consideratelo come la parte di livello 1 della configurazione della chiamata H.323 in tre parti.
- debug cch323 h245: questo comando esamina la parte H.245 della negoziazione delle chiamate e traccia la transizione di stato della macchina a stati H.245 in base agli eventi elaborati. Consideratelo come la parte di livello 2 della configurazione della chiamata H.323 in tre parti.

Comprendere i problemi di qualità del servizio (QoS) VoIP

Quando le chiamate VoIP sono stabilite correttamente, il passo successivo è verificare che la qualità della voce sia buona. Anche se la risoluzione dei problemi QoS non è trattata in questo documento, per ottenere una buona qualità vocale è necessario considerare le seguenti linee guida:

- Comprendere la larghezza di banda utilizzata da una chiamata VoIP con ciascun codec. Tra queste, le intestazioni Layer 2 e IP/UDP/RTP. Per ulteriori informazioni, fare riferimento a [Modifica del calcolo del consumo della larghezza di banda per le chiamate vocali](#).
- Comprendere le caratteristiche della rete IP su cui viaggiano le chiamate. Ad esempio, la larghezza di banda di una rete frame relay in CIR è molto diversa da quella sopra CIR (o burst), dove i pacchetti possono essere scartati o accodati nel cloud Frame Relay. Assicuratevi che ritardo e jitter siano controllati ed eliminati il più possibile. Il ritardo di trasmissione unidirezionale non deve superare i 150 ms (per raccomandazione G.114).
- Utilizzare una tecnica di accodamento che consente di identificare il traffico VoIP e di assegnare le priorità.
- Quando si trasmette il protocollo VoIP su collegamenti a bassa velocità, usare le tecniche di frammentazione dei pacchetti di layer 2, ad esempio MLPPP con LFI (Link Fragmentation and Interleaving) su collegamenti point-to-point o FRF.12 sui collegamenti Frame-Relay. La frammentazione di pacchetti dati più grandi consente di ridurre il tremolio e il ritardo nella trasmissione del traffico VoIP in quanto i pacchetti VoIP possono essere interlacciati sul collegamento.
- Provare a utilizzare un codec diverso e provare la chiamata con VAD abilitato e disabilitato per limitare il problema al DSP, in contrapposizione alla rete IP.

Con il VoIP, gli elementi principali da cercare quando si risolvono i problemi relativi al QoS sono i

pacchetti scartati e i colli di bottiglia della rete che possono causare ritardo e jitter.

Cerca:

- interfacce non elaborate
- buffer drops
- congestione dell'interfaccia
- congestione dei collegamenti

È necessario esaminare ogni interfaccia nel percorso della chiamata VoIP. Inoltre, eliminare le cadute e la congestione. Inoltre, il ritardo di andata e ritorno deve essere ridotto il più possibile. I ping tra gli endpoint VoIP danno un'indicazione del ritardo di andata e ritorno di un collegamento. Il ritardo di andata e ritorno non deve essere superiore a 300 ms, ove possibile. Se il ritardo non deve superare questo valore, è necessario fare in modo che il ritardo sia costante, in modo da non introdurre jitter o un ritardo variabile.

Inoltre, è necessario verificare che il meccanismo di coda Cisco IOS inserisca i pacchetti VoIP nelle code appropriate. I comandi Cisco IOS, ad esempio show queue interface o show priority, possono essere utili per la verifica dell'accodamento.

Dettagli dei codici causa e dei valori di debug per VoIP

Utilizzare queste tabelle quando si leggono i debug e i valori associati all'interno dei debug.

Q.931 Cause disconnessione chiamata (cause_codes from debug voip ccapi inout)

| Valore causa disconnessione chiamata (in esadecimale) | Significato e numero (in Decimale) |
|---|--|
| CC_CAUSE_UANUM = 0x1 | numero non assegnato. (1) |
| CC_CAUSE_NO_ROUTE = 0x3 | nessun percorso verso la destinazione. (3) |
| CC_CAUSE_NORM = 0x10 | normale cancellazione di chiamata. (16) |
| CC_CAUSE_BUSY = 0x11 | utente occupato. (17) |
| CC_CAUSE_NORS = 0x12 | nessuna risposta dell'utente. (18) |
| CC_CAUSE_NOAN = 0x13 | nessuna risposta dell'utente. (19) |
| CC_CAUSE_REJECT = 0x15 | chiamata rifiutata. (21) |
| CC_CAUSE_INVALID_NUMBER = 0x1C | numero non valido. (28) |
| CC_CAUSE_UNSP = 0x1F | normale, non specificato. (31) |
| CC_CAUSE_NO_CIRCUIT = 0x22 | nessun circuito. (34) |
| CC_CAUSE_NO_REQ_CIRCUIT = 0x2C | nessun circuito richiesto. (44) |
| CC_CAUSE_NO_RESOURCE = 0x2F | nessuna risorsa. (47) ¹ |
| CC_CAUSE_NOSV = 0x3F | servizio o opzione non disponibile o non |

| |
|-------------------|
| specificato. (63) |
|-------------------|

¹Questo problema può essere causato da una mancata corrispondenza del codec nella configurazione H323, quindi la prima fase di risoluzione dei problemi è codificare i dial-peer VoIP in modo da usare il codec corretto.

Valori di negoziazione codec (da debug voip capi inout)

Per ulteriori informazioni sui CODEC, fare riferimento a [Descrizione dei codec: Complessità, Supporto hardware, MOS e Negoziazione](#).

| Valore di negoziazione | Significato |
|------------------------|-------------------|
| 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=0x0002000 | G723ar63 |
| codec=0x00004000 | G723ar |
| codec=0x00008000 | CANALE_CHIARO |

Tipi di tono

| Tipi di tono | Significato |
|-------------------------|-------------------------------------|
| CC_TONE_RINGBACK 0x1 | Suoneria |
| CC_TONE_FAX 0x2 | Segnale fax |
| CC_TONE_BUSY 0x4 | Segnale di occupato |
| CC_TONE_DIALTONE 0x8 | Segnale acustico |
| C_TONE_OS 0x10 | Segnale di fuori servizio |
| CC_TONE_ADDR_ACK 0x20 | Segnale di riconoscimento indirizzo |
| CC_TONE_DISCONNECT 0x40 | Scollega segnale |

| | |
|---------------------------------|--|
| CC_TONE_OFF_HOOK_NOTICE 0x80 | Segnale acustico che indica che il telefono è stato sganciato |
| CC_TONE_OFF_HOOK_ALERT 0x100 | Una versione più urgente di CC_TONE_OFF_HOOK_NOTICE |
| CC_TONE_CUSTOM 0x200 | Tono personalizzato: utilizzato quando si specifica un tono personalizzato |
| CC_TONE_NULL 0x0 | Tono Null |

Valori della velocità di trasmissione e delle capacità VAD

| Valori | Significato |
|----------------------|-----------------------------------|
| CC_CAP_FAX_NONE 0x1 | Fax disattivato o non disponibile |
| CC_CAP_FAX_VOICE 0x2 | Chiamata vocale |
| CC_CAP_FAX_144 0x4 | 14,400 baud |
| CC_CAP_FAX_96 0x8 | 9,600 baud |
| CC_CAP_FAX_72 0x10 | 7,200 baud |
| CC_CAP_FAX_48 0x20 | 4,800 baud |
| CC_CAP_FAX_24:0x40 | 2,400 baud |
| CC_CAP_VAD_OFF 0x1 | VAD disattivato |
| CC_CAP_VAD_ON 0x2 | VAD abilitato |

Informazioni correlate

- [Risoluzione dei problemi di T1 Layer 1](#)
- [T1 - Risoluzione dei problemi](#)
- [Risoluzione dei problemi della linea seriale](#)
- [Risoluzione dei problemi di Cisco IP Telephony](#)
- [Supporto tecnico Cisco e download](#)

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).