

Fehlerbehebung und Debuggen von VoIP-Anrufen

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Hintergrundinformationen](#)

[Anrufablauf im Netzwerk](#)

[Anrufablauf des Routers](#)

[Architektur der Telefonieschnittstelle](#)

[Überprüfen der digitalen und analogen Signalisierung \(POTS-Anrufweiterleitung\)](#)

[Show-Controller T1 / E1 \(digital\)](#)

[Sprachport anzeigen](#)

[debug vpm \(Sprachprozessormodul\)](#)

[Überprüft empfangene und gesendete Ziffern \(POTS-Rufzeichen\)](#)

[Wählplannummer anzeigen](#)

[debuggen vtsp-Sitzung](#)

[Überprüfen der End-to-End-VoIP-Signalisierung \(VOIP-Anrufstrecke\)](#)

[debuggen voip ccapi inout](#)

[Probleme mit der VoIP-QoS \(Quality of Service\)](#)

[Details zu Ursachencodes und Fehlerbehebungswerten für VoIP](#)

[Q.931 Ursachen für Verbindungstrennung \(Cause_codes vom debug voip ccapi inout\)](#)

[Codec-Verhandlungswerte \(aus debug voip ccapi inout\)](#)

[Tonarten](#)

[FAX-Rate- und VAD-Funktionswerte](#)

[Zugehörige Informationen](#)

Einleitung

In diesem Dokument werden die grundlegenden Techniken und Befehle zum Beheben und Debuggen von VoIP-Netzwerken beschrieben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- VoIP-Konfiguration
- Sprach-QoS
- Design und Bereitstellung von VoIP-Netzwerken

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardware-Versionen beschränkt. Die angezeigten Ergebnisse basieren jedoch auf der Cisco IOS® Software-Version 12.3(8).

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter Cisco Technical Tips Conventions (Technische Tipps von Cisco zu Konventionen).

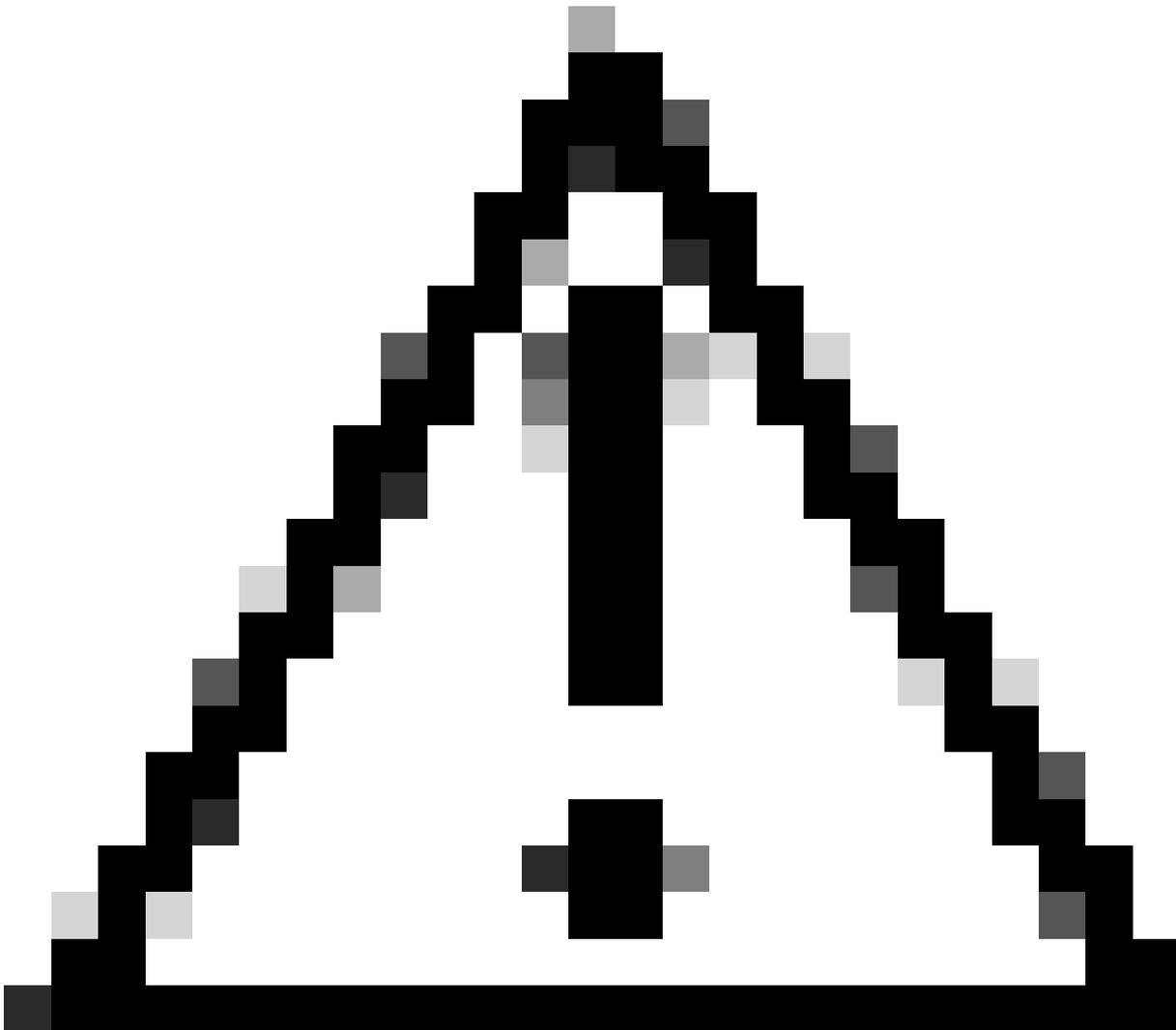
Hintergrundinformationen

Dieses Dokument zeigt grundlegende Techniken und Befehle zur Fehlerbehebung und -suche in VoIP-Netzwerken. Es wird ein Überblick über den Sprachanrufverlauf und die Telefoniearchitektur in einem Cisco Router gegeben, gefolgt von einem Schritt-für-Schritt-Ansatz zur VoIP-Fehlerbehebung, der in diesen Schritten vorgestellt wird:

1. [Überprüfen Sie die digitale und analoge Signalisierung.](#)
2. [Überprüfen Sie die Ziffern, die von den analogen und digitalen Sprach-Ports empfangen und gesendet wurden.](#)
3. [Überprüfen der End-to-End-VoIP-Signalisierung](#)
4. [Probleme mit der VoIP-QoS \(Quality of Service\) verstehen](#)
5. [Details zu Ursachencodes und Debugwerten für VoIP verstehen](#)



Hinweis: In diesem Dokument werden nicht alle Aspekte der Cisco IOS-Architektur erläutert, die in Cisco VoIP-Gateways und -Gatekeepern verwendet werden. Stattdessen soll gezeigt werden, welche Befehle verwendet werden können und welche Felder aus den Befehlsausgaben am wertvollsten sein können.



Vorsicht: Die Fehlerbehebung von Cisco IOS ist prozessorintensiv. Seien Sie vorsichtig, wenn Sie die in diesem Dokument aufgeführten Debugs verwenden. Weitere Informationen finden Sie unter [Wichtige Informationen zu Debugbefehlen](#).

Debugs müssen mit Zeitstempeln ausgeführt werden, die im Protokoll aktiviert sind. Aktivieren Sie Zeitstempel mit den Befehlen: `service timestamps debug datetime msec`, `service timestampslog datetime msec` im privilegierten Modus. Die Zeitstempel helfen dabei, das Zeitintervall zwischen Zustandsänderungen zu bestimmen.

Anrufablauf im Netzwerk

Bevor Sie mit der Fehlerbehebung oder Fehlerbehebung für VoIP beginnen, sollten Sie unbedingt beachten, dass VoIP-Anrufe aus drei Anrufabschnitten bestehen. Bei diesen Anrufabschnitten handelt es sich um Quelldienste für Plain Old Telephone Systems (POTS), VoIP und Ziel-POTS. Dies wird im folgenden Diagramm dargestellt. Die Fehlerbehebung und das Debugging müssen sich zunächst auf die einzelnen Levels und dann auf den VoIP-Anruf als Ganzes konzentrieren.

Anrufablauf des Routers

Diese Definitionen erläutern die Funktion der Hauptkomponenten, die im Anrufflussdiagramm des Routers angezeigt werden:

Anrufsteuerungs-API (Application Programming Interface) - Drei Clients verwenden die Anrufsteuerungs-API. Die drei Clients sind CLI, Simple Network Management Protocol (SNMP)-Agent und die Sitzungsanwendung. Die Hauptfunktionen der Anrufsteuerungs-API (auch CCAPI genannt) sind:

- Identifizieren Sie die Anrufabschnitte (z. B. welchen DFÜ-Peer handelt es sich? Woher stammen die Abschnitte?).
- Entscheiden Sie, welche Sitzungsanwendung den Anruf entgegennimmt (z. B. wer kümmert sich darum?).
- Rufen Sie den Pakethandler auf.
- Führen Sie alle erforderlichen Schritte gemeinsam durch.
- Starten der Aufzeichnung der Anrufstatistik

Session Application (Sitzungsanwendung) und Dial Plan Mapper (Wählplanzuordnung): Die Sitzungsanwendung verwendet den Dial Plan Mapper, um einem Dial-Peer (lokaler POTS oder Remote-VoIP) eine Nummer zuzuordnen. Die Wählplanzuordnung sucht mithilfe der DFÜ-Peer-Tabelle nach aktiven DFÜ-Peers.

Telephony and VoIP Service Provider Interface (SPI) (Telefonie-SPI): Der Telefonie-SPI kommuniziert mit den POTS-DFÜ-Peers (analog: fxs, fxo, e&m Digital: isdn, qsig, e&m usw.). Der VoIP-SPI ist die spezifische Schnittstelle zu den VoIP-Peers. Telefonie-/DSP-Treiber stellen Dienste für den Telefonie-SPI bereit, während der VoIP-SPI auf Sitzungsprotokollen beruht.

Architektur der Telefoneschnittstelle

Dieses Diagramm zeigt die Architektur der Telefonie-Bausteine des Cisco Routers und ihre Interaktion untereinander.

Diese Liste beschreibt die Funktionen und Definitionen der Hauptdiagrammkomponenten:

- CCAPI (Call Control Application Programming Interface) - Softwareeinheit, die Anrufabschnitte erstellt, terminiert und überbrückt.
- Voice Telephony Service Provider (VTSP): Ein Cisco IOS-Prozess, der Anforderungen der Anrufsteuerungs-API bedient und die entsprechenden Anforderungen an den digitalen Signalprozessor (DSP) oder das VPM formuliert.
- Voice Processor Module (VPM) - Das VPM überbrückt und koordiniert Signalisierungsprozesse zwischen den Telefonie-Ports des Signaling State Machine (SSM),

dem DSP Resource Manager und dem VTSP.

- DSP Resource Manager (DSP-Ressourcenmanager) - Das DSPRM stellt Schnittstellen bereit, über die der VTSP Nachrichten an die DSPs senden und von diesen empfangen kann.
- Packet Handler (Paketbehandler): Der Paketbehandler leitet Pakete zwischen den DSPs und den Peer-Call-Abschnitten weiter.
- Call Peer (Beigeordneter Anruf): Der Beigeordnete Anruf ist der entgegengesetzte Anrufabschnitt. Dies kann eine andere Telefonie-Sprachverbindung (POTS), VoFR, VoATM oder eine VoIP-Verbindung sein.

Überprüfen der digitalen und analogen Signalisierung (POTS-Anrufweiterleitung)

Für die Überprüfung der digitalen und analogen Signalisierung ist Folgendes erforderlich:

- Prüfen Sie, ob die richtige analoge oder digitale Signalisierung bei aufgelegtem und abgehobenem Hörer empfangen wird.
- Prüfen Sie, ob die richtige E&M-, FXO- und FXS-Signalisierung sowohl auf dem Router als auch auf dem Switch (CO oder PBX) konfiguriert ist.
- Überprüfen Sie, ob sich die DSPs im Ziffernsammelmodus befinden.

Die in diesen Abschnitten beschriebenen Befehle können zum Überprüfen der Signalisierung verwendet werden.

Show-Controller T1 / E1 (digital)

`show controllers t1 [slot/port]` (Controller t1 anzeigen [Steckplatz/Port]): Verwenden Sie zuerst diesen Befehl. Es wird angezeigt, ob die digitale T1-Verbindung zwischen Router und Switch (CO oder PBX) aktiv oder inaktiv ist und ob sie ordnungsgemäß funktioniert. Die Ausgabe dieses Befehls sieht wie folgt aus:

```
<#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
```

Wenn Sie E1 verwenden, verwenden Sie den Befehl `show controllers e1`. Weitere Informationen finden Sie unter:

- [T1 Layer 1 - Fehlerbehebung](#)
- [T1 Fehlerbehebung - Flussdiagramm](#)
- [Fehlerbehebung bei Problemen mit seriellen Leitungen](#)

Sprachport anzeigen

`show voice ports` Steckplatznummer/Port: Mit diesem Befehl können Sie den Port-Status und die am Sprach-Port der Cisco Voice Interface Cards (VIC) konfigurierten Parameter anzeigen. Wie alle Cisco IOS-Befehle werden die Standardeinstellungen nicht in `show running-config` angezeigt, sondern mit diesem Befehl.

Dies ist eine Beispielausgabe für einen E&M-Sprach-Port:

```
<#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 (Sprachprozessormodul)

Die folgenden Befehle werden zum Debuggen der VPM-Telefonieschnittstelle verwendet:

- debug vpm signal - Dieser Befehl dient zum Sammeln von Debuginformationen für Signalisierungsereignisse und kann beim Lösen von Problemen mit der Signalisierung an ein PBX-System hilfreich sein.
- debug vpm spi - Dieser Befehl verfolgt, wie die Sprachport-Modul-Service Provider-Schnittstelle (SPI) mit der Anrufsteuerungs-API interagiert. Dieser Debug-Befehl zeigt Informationen über die Behandlung der einzelnen Netzwerkindikationen und Anwendungsanforderungen an.
- debug vpm dsp - Dieser Befehl zeigt Meldungen des DSP auf dem VPM an den Router an und kann nützlich sein, wenn Sie vermuten, dass das VPM nicht funktioniert. Es ist eine einfache Möglichkeit, zu überprüfen, ob das VPM auf Off-Hook-Anzeigen reagiert, und das Timing für Signalisierungsnachrichten von der Schnittstelle auszuwerten.

- debug vpm all - Dieser EXEC-Befehl aktiviert alle debug vpm-Befehle: debug vpm spi, debug vpm signal und debug vpm dsp.
- debug vpm port - Verwenden Sie diesen Befehl, um die Debug-Ausgabe auf einen bestimmten Port zu beschränken. Diese Ausgabe zeigt beispielsweise debug vpm dspmessages only for port 1/0/0:

```
debug vpm dsp
```

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

Beispielausgabe für debug vpm signalCommand

```
<#root>
maui-voip-austin#
debug vpm signal

!--- FXS port 1/0/0 goes from the "on-hook" to "off-hook" !--- state.
htsp_process_event: [1/0/0, 1.2 , 36]
fxsls_onhook_offhook htsp_setup_ind
*Mar 10 16:08:55.958: htsp_process_event:
[1/0/0, 1.3 , 8]

!--- Sends ringing alert to the called phone.
*Mar 10 16:09:02.410: htsp_process_event:
[1/0/0, 1.3 , 10] htsp_alert_notify
*Mar 10 16:09:03.378: htsp_process_event:
[1/0/0, 1.3 , 11]

!--- End of phone call, port goes "on-hook".
*Mar 10 16:09:11.966: htsp_process_event:
[1/0/0, 1.3 , 6]
*Mar 10 16:09:17.218: htsp_process_event:
[1/0/0, 1.3 , 28]
fxsls_offhook_onhook
*Mar 10 16:09:17.370: htsp_process_event:
[1/0/0, 1.3 , 41] fxsls_offhook_timer
*Mar 10 16:09:17.382: htsp_process_event:
[1/0/0, 1.2 , 7]
fxsls_onhook_release
```

Wenn die Signale bei aufgelegtem und abgehobenem Hörer nicht richtig ausgegeben werden, überprüfen Sie folgende Punkte:

- Überprüfen Sie die korrekte Verkabelung.
- Prüfen Sie, ob Router und Switch (CO oder PBX) ordnungsgemäß geerdet sind.
- Stellen Sie sicher, dass beide Enden der Verbindung über äquivalente Signalisierungskonfigurationen verfügen. Nicht übereinstimmende Konfigurationen können eine unvollständige oder unidirektionale Signalisierung verursachen.

Weitere Informationen zur Fehlerbehebung bei E&M finden Sie unter [Verstehen und Beheben von Problemen bei analogen E&M-Schnittstellentypen und Verkabelungsanordnungen](#).

Beispielausgabe für 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
```

Überprüft empfangene und gesendete Ziffern (POTS-Rufzeichen)

Wenn die Signale bei aufgelegtem und abgehobenem Hörer verifiziert wurden und richtig funktionieren, überprüfen Sie, ob die richtigen Ziffern am Sprach-Port empfangen oder gesendet werden (digital oder analog). Ein Dial-Peer wurde nicht zugeordnet, oder der Switch (CO oder PBX) kann den richtigen Sender nicht anrufen, wenn unvollständige oder falsche Ziffern gesendet oder empfangen werden. Die folgenden Befehle können verwendet werden, um die empfangenen/gesendeten Ziffern zu überprüfen:

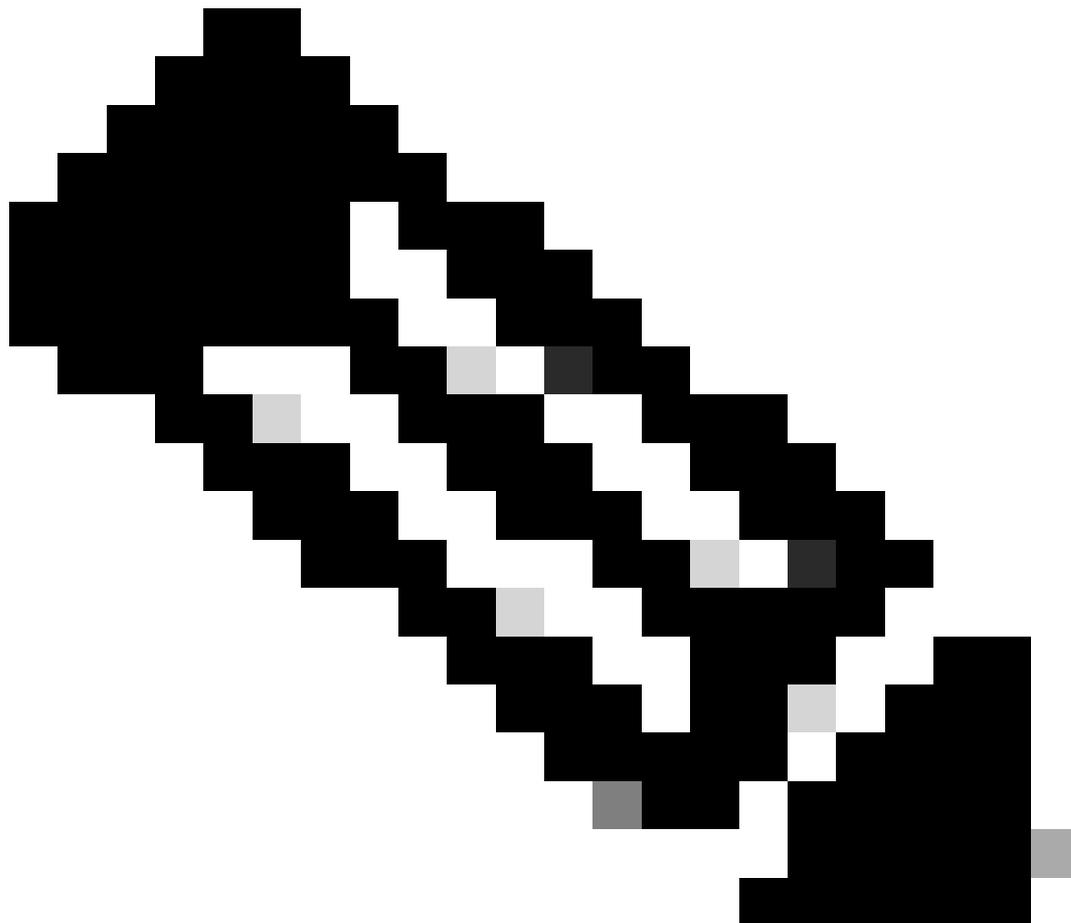
- show dialplan number (Nummernplan-Nummer anzeigen): Mit diesem Befehl wird angezeigt, welcher DFÜ-Peer beim Wählen einer bestimmten Telefonnummer erreicht wird.
- debug vtsp session - Dieser Befehl zeigt Informationen zur Verarbeitung der einzelnen

Netzwerkindikationen und Anwendungsanforderungen, Signalisierungshinweise und DSP-Kontrollmeldungen an.

- `debug vtsp dsp` - Vor Version 12.3 der Cisco IOS-Software zeigt dieser Befehl die Ziffern an, die vom Sprach-Port empfangen werden. In Version 12.3 und höher der Cisco IOS-Software werden die Ziffern in der Ausgabe des Befehls `debug` nicht mehr angezeigt. Die Kombination aus `debug hpi detail` und `debug hpinotification` kann verwendet werden, um die eingehenden Ziffern anzuzeigen.
- `debug vtsp all` - Dieser Befehl aktiviert die folgenden Befehle zum Debuggen des Sprach-Telefonie-Diensteanbieters (VTSP): `debug vtsp session`, `debug vtsp error` and `debug vtsp dsp`.

Wählplannummer anzeigen

`show dialplan number <digit_string>` - Mit diesem Befehl wird der Dial-Peer angezeigt, dem eine Zeichenfolge entspricht. Wenn mehrere DFÜ-Peers zugeordnet werden können, werden sie alle in der Reihenfolge angezeigt, in der sie zugeordnet werden.



Hinweis: Sie müssen das #-Zeichen am Ende der Telefonnummern für DFÜ-Peers mit variabler Länge verwenden, um Zielmuster abzugleichen, die mit T enden.

Die Ausgabe dieses Befehls sieht wie folgt aus:

```
<#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-rtsp,
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
```

debuggen vtsp-Sitzung

Der Befehl `debug vtsp session` zeigt Informationen über die Interaktion des Routers mit dem DSP an, basierend auf den Signalisierungsangaben aus dem Signalisierungs-Stack und den Anforderungen der Anwendung. Dieser Befehl `debug` zeigt Informationen über die Behandlung der einzelnen Netzwerkindikationen und Anwendungsanforderungen, Signalisierungshinweise und DSP-Kontrollmeldungen an.

```
<#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

```

Wenn festgestellt wird, dass die Ziffern nicht richtig gesendet oder empfangen werden, kann es erforderlich sein, entweder einen Digit-Grabber (Testwerkzeug) oder einen T1-Tester zu verwenden, um zu überprüfen, ob die Ziffern mit der richtigen Frequenz und dem richtigen Zeitintervall gesendet werden. Wenn sie für den Switch "falsch" gesendet werden (CO oder PBX), müssen möglicherweise einige Werte auf dem Router oder Switch (CO oder PBX) angepasst werden, damit sie übereinstimmen und interagieren können. Dabei handelt es sich in der Regel um Werte für die Zifferndauer und die Interdigit-Dauer. Ein weiterer Punkt, bei dem geprüft werden

muss, ob die Ziffern korrekt gesendet wurden, sind alle Nummernumwandlungstabellen im Switch (CO oder PBX), die Ziffern hinzufügen oder entfernen können.

Überprüfen der End-to-End-VoIP-Signalisierung (VOIP-Anrufstrecke)

Nachdem Sie sichergestellt haben, dass die Sprach-Port-Signalisierung ordnungsgemäß funktioniert und die richtigen Ziffern empfangen wurden, fahren Sie mit der Fehlerbehebung und Fehlerbehebung für die VoIP-Anrufsteuerung fort. Diese Faktoren erklären, warum das Debuggen der Anrufsteuerung zu einer komplexen Aufgabe werden kann:

- Die Cisco VoIP-Gateways nutzen die H.323-Signalisierung für den Abschluss von Anrufen. H.323 besteht aus drei Ebenen: Anrufaushandlung und -einrichtung: H.225, H.245 und H.323. Diese Protokolle verwenden eine Kombination aus TCP und UDP, um einen Anruf einzurichten und einzurichten.
- Das End-to-End-VoIP-Debugging zeigt eine Reihe von Cisco IOS-Statussystemen. Probleme mit Statuscomputern können dazu führen, dass ein Anruf fehlschlägt.
- End-to-End VoIP-Debugging kann sehr ausführlich sein und eine Menge Debug-Ausgabe erzeugen.

debuggen voip ccapi inout

Der primäre Befehl zum Debuggen von End-to-End-VoIP-Anrufen ist `debug voip ccapi inout`. In dieser Ausgabe wird die Ausgabe eines Aufrufdebugs angezeigt.

```
<#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)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)
```

Wenn der Anruf fehlschlägt und die Ursache im VoIP-Bereich der Anrufeinrichtung liegt, müssen Sie möglicherweise den H.225- oder H.245-TCP-Bereich der Anrufeinrichtung betrachten, im Gegensatz zum UDP-Bereich der H.323-Einrichtung. Die folgenden Befehle können zum Debuggen der H.225- oder H.245-Anrufeinrichtung verwendet werden:

- `debug ip tcp transactions` and `debug ip tcp packet`: Diese Befehle untersuchen den TCP-Teil der H.225- und H.245-Aushandlung. Sie geben die IP-Adressen, TCP-Ports und den Status der TCP-Verbindungen zurück.
- `debug cch323 h225` - Dieser Befehl untersucht den H.225-Teil der Anrufaushandlung und verfolgt den Zustandsübergang des H.225-Zustandsmaschinensystems auf Basis des verarbeiteten Ereignisses. Stellen Sie sich dies als Layer-1-Komponente der dreiteiligen H.323-Anrufkonfiguration vor.
- `debug cch323 h245` - Dieser Befehl untersucht den H.245-Teil der Anrufaushandlung und verfolgt den Zustandsübergang des H.245-Zustandsmaschinensystems auf Basis der verarbeiteten Ereignisse. Stellen Sie sich dies als Layer-2-Komponente der dreiteiligen H.323-Anrufkonfiguration vor.

Probleme mit der VoIP-QoS (Quality of Service)

Wenn VoIP-Anrufe richtig eingerichtet wurden, besteht der nächste Schritt darin, die gute Sprachqualität zu überprüfen. QoS-Fehlerbehebung wird in diesem Dokument nicht behandelt. Um eine gute Sprachqualität zu erreichen, müssen jedoch die folgenden Richtlinien beachtet werden:

- Sie sollten wissen, wie viel Bandbreite ein VoIP-Anruf mit jedem Codec beansprucht. Dies umfasst Layer-2- und IP/UDP/RTP-Header. Weitere Informationen finden Sie unter [Modify Bandwidth Consumption Calculation for Voice Calls](#).
- Erfassen der Merkmale des IP-Netzwerks, über das die Anrufe übertragen werden. Beispielsweise ist die Bandbreite eines Frame-Relay-Netzwerks bei CIR sehr viel anders als die Bandbreite oberhalb von CIR (oder Burst), bei der Pakete verworfen oder in der Frame-Relay-Cloud in die Warteschlange gestellt werden können. Stellen Sie sicher, dass

Verzögerungen und Jitter so weit wie möglich kontrolliert und eliminiert werden. Die unidirektionale Übertragungsverzögerung darf 150 ms nicht überschreiten (gemäß G.114-Empfehlung).

- Verwenden Sie eine Warteschlangenmethode, mit der VoIP-Datenverkehr identifiziert und priorisiert werden kann.
- Wenn Sie VoIP über langsame Verbindungen übertragen, sollten Sie Layer-2-Paketfragmentierungstechniken wie MLPPP mit Link Fragmentation and Interleaving (LFI) auf Point-to-Point-Verbindungen oder FRF.12 auf Frame-Relay-Verbindungen verwenden. Die Fragmentierung größerer Datenpakete ermöglicht einen geringeren Jitter und geringere Verzögerungen bei der Übertragung von VoIP-Datenverkehr, da die VoIP-Pakete auf der Verbindung ineinander verschachtelt werden können.
- Versuchen Sie, einen anderen Codec zu verwenden, und versuchen Sie, den Anruf mit aktivierter und deaktivierter VAD zu tätigen, um das Problem möglicherweise auf den DSP einzugrenzen, im Gegensatz zum IP-Netzwerk.

Bei VoIP sind die wichtigsten Aspekte bei der Fehlerbehebung von QoS-Problemen verlorene Pakete und Netzwerkengpässe, die Verzögerungen und Jitter verursachen können.

Suchen nach:

- Schnittstellen-Drops
- Pufferverluste
- Überlastung der Schnittstelle
- Verbindungsstau

Jede Schnittstelle im Pfad des VoIP-Gesprächs muss überprüft werden. Vermeiden Sie außerdem Versäumnisse und Staus. Außerdem muss die Round-Trip-Verzögerung so weit wie möglich reduziert werden. Pings zwischen den VoIP-Endpunkten geben einen Hinweis auf die Round-Trip-Verzögerung einer Verbindung. Die Round-Trip-Verzögerung darf, wenn möglich, 300 ms nicht überschreiten. Wenn die Verzögerung diesen Wert nicht überschreiten muss, müssen Anstrengungen unternommen werden, um sicherzustellen, dass diese Verzögerung konstant ist, damit kein Jitter oder eine variable Verzögerung entsteht.

Außerdem muss sichergestellt werden, dass die VoIP-Pakete mithilfe des Cisco IOS-Warteschlangenmechanismus in die richtigen Warteschlangen gestellt werden. Cisco IOS-Befehle wie `show queue interface` oder `show priority` können bei der Überprüfung der Warteschlange helfen.

Details zu Ursachencodes und Fehlerbehebungswerten für VoIP

Verwenden Sie diese Tabellen, wenn Sie Debugs und die zugehörigen Werte innerhalb der Debugs lesen.

Q.931 Ursachen für Verbindungstrennung (Cause_codes vom debug voip ccapi inout)

Wert für die Anrufabtrennungsursache (in Hex)	Bedeutung und Zahl (dezimal)
CC_CAUSE_UANUM = 0x1	nicht zugewiesene Nummer (1)
CC_CAUSE_NO_ROUTE = 0x3	keine Route zum Ziel. (3)
CC_CAUSE_NORM = 0x10	normale Anrufabwicklung. (16)
CC_CAUSE_BUSY = 0x11	Benutzer ist beschäftigt. (17)
CC_CAUSE_NORS = 0x12	Keine Benutzerantwort. (18)
CC_CAUSE_NOAN = 0x13	Keine Benutzerantwort. (19)
CC_CAUSE_REJECT = 0x15	Anruf abgelehnt. (21)
CC_CAUSE_INVALID_NUMBER = 0x1C	Ungültige Nummer. (28)
CC_CAUSE_UNSP = 0x1F	normal, nicht angegeben. (31)
CC_CAUSE_NO_CIRCUIT = 0x22	kein Stromkreis. (34)
CC_CAUSE_NO_REQ_CIRCUIT = 0x2C	Keine angeforderte Schaltung. (44)
CC_CAUSE_NO_RESOURCE = 0x2F	keine Ressource. (47) ¹
CC_CAUSE_NOSV = 0x3F	Dienst oder Option nicht verfügbar oder Nicht angegeben. (63)

¹Dieses Problem kann aufgrund einer Codec-Diskrepanz innerhalb der H323-Konfiguration auftreten. Der erste Schritt zur Fehlerbehebung besteht daher darin, die VoIP-DFÜ-Peers fest zu codieren, um den richtigen Codec zu verwenden.

Codec-Verhandlungswerte (aus debug voip ccapi inout)

Weitere Informationen zu CODECs finden Sie unter [Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation.](#)

Verhandlungswert	Bedeutung
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=0x0000200	G723r53
Codec=0x0000400	GSMFR
Codec=0x0000800	G729b
Codec=0x00001000	G729ab
Codec=0x00002000	G723ar63
Codec=0x00004000	G723ar53
Codec=0x00008000	CLEAR_CHANNEL

Tonarten

Tonarten	Bedeutung
CC_TONE_RINGBACK 0x1	Klingelton
CC_TONE_FAX 0x2	Faxton
CC_TONE_BUSY 0x4	Besetztzeichen
CC_TONE_DIALTONE 0x8	Freizeichen
CC_TONE_OOS 0x10	Ton bei fehlendem Betrieb
CC_TONE_ADDR_ACK 0x20	Ton für Adressbestätigung
CC_TONE_DISCONNECT 0x40	Ton trennen
CC_TONE_OFF_HOOK_NOTICE 0x80	Ton, der anzeigt, dass der Hörer abgehoben wurde
CC_TONE_OFF_HOOK_ALERT 0x100	Eine dringendere Version von CC_TONE_OFF_HOOK_NOTICE
CC_TONE_CUSTOM 0x200	Benutzerdefinierter Ton: Dieser Ton wird verwendet, wenn Sie einen benutzerdefinierten Ton angeben.
CC_TONE_NULL 0x0	Nullton

FAX-Rate- und VAD-Funktionswerte

Werte	Bedeutung
CC_CAP_FAX_NONE 0x1	Fax deaktiviert oder nicht verfügbar
CC_CAP_FAX_VOICE 0x2	Sprachanruf
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 0 x 40	2.400 Baud
CC_CAP_VAD_OFF 0x1	VAD deaktiviert
CC_CAP_VAD_ON 0x2	VAD aktiviert

Zugehörige Informationen

- [T1 Layer 1 - Fehlerbehebung](#)
- [T1 - Fehlerbehebung](#)
- [Fehlerbehebung bei Problemen mit seriellen Leitungen](#)
- [Fehlerbehebung bei Cisco IP-Telefonie](#)
- [Technischer Support und Downloads von Cisco](#)

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.