

Dépannage et débogage des appels VoIP

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Conventions](#)

[Informations générales](#)

[Flux d'appel dans le réseau](#)

[Flux d'appels du routeur](#)

[Architecture d'interface de téléphonie](#)

[Vérifier signalisation numérique et analogique \(le segment d'appel de POTS\)](#)

[montrer contrôleurs t1/E1 \(numérique\)](#)

[montrer port voix](#)

[vpm de débogage \(module de processeur vocal\)](#)

[Vérifier les chiffres reçus et envoyés \(le segment d'appel de POTS\)](#)

[montrer numéro de plan de numérotation](#)

[débogage de session vtsp](#)

[Vérifier la signalisation de bout en bout VoIP \(le segment d'appel de VOIP\)](#)

[debug voip ccapi inout](#)

[Comprendre les problèmes de qualité de service \(QoS\) VoIP](#)

[Détails de codes de motif et des valeurs de débogage pour le VoIP](#)

[Causes de la déconnexion d'appel Q.931 \(cause codes de debug voip ccapi inout\)](#)

[Valeurs de négociation de codecs \(du debug voip ccapi inout\)](#)

[Types de tonalité](#)

[Débit de télécopie et valeurs de fonctionnalités VAD](#)

[Informations connexes](#)

Introduction

Ce document décrit les techniques et les commandes de base pour dépanner et déboguer les réseaux VoIP.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Configuration VoIP

- QoS voix
- Conception et déploiement de réseaux VoIP

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques. Cependant, les sorties montrées sont basées sur la version logicielle 12.3(8) de Cisco IOS®.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à Conventions relatives aux conseils techniques Cisco.

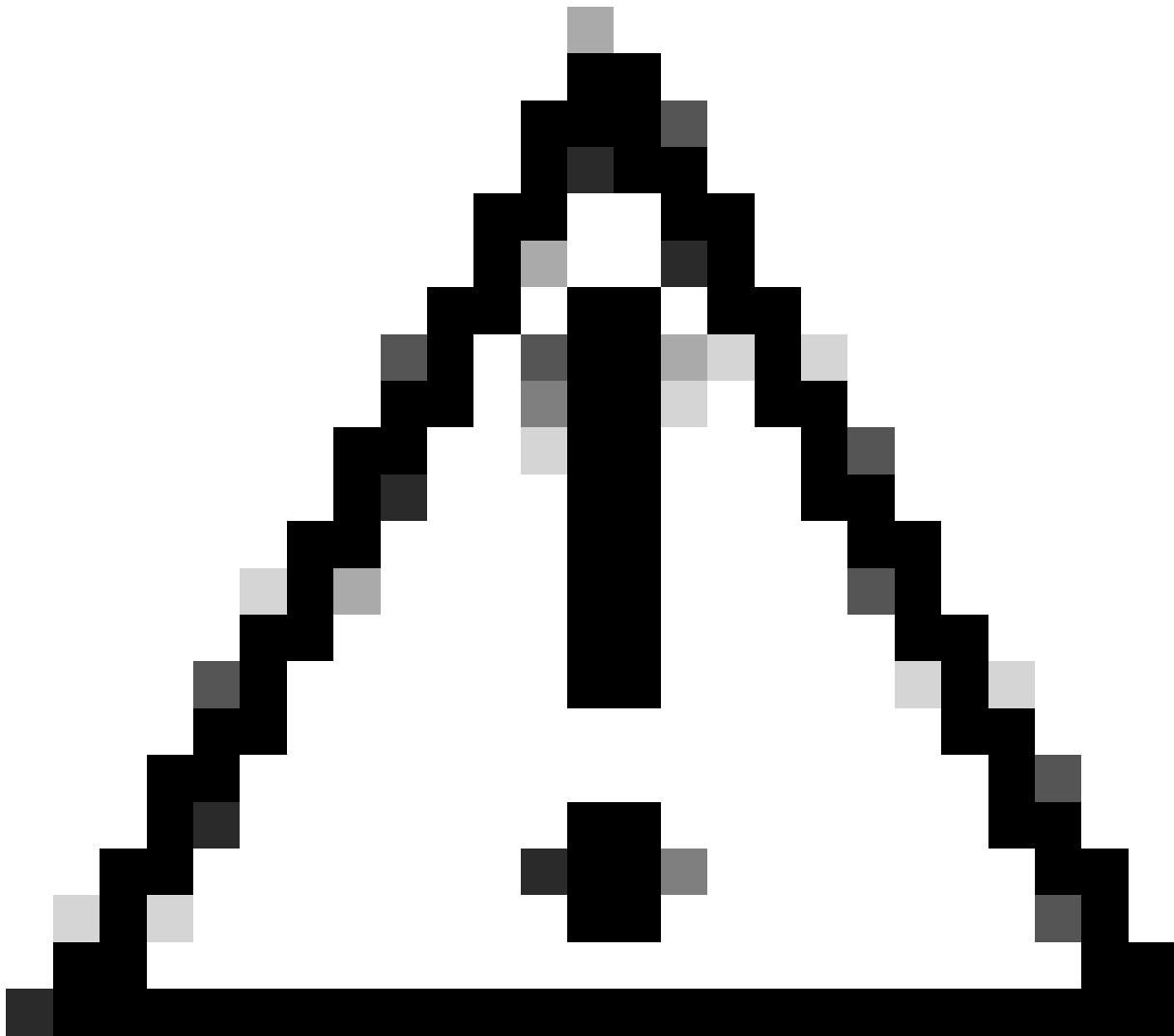
Informations générales

Ce document explique des techniques et des commandes de base pour dépanner et déboguer des réseaux VoIP. Une vue d'ensemble de l'architecture de flux et de téléphonie d'appel vocal dans un routeur Cisco est présentée, suivie d'une approche de dépannage du pas à pas VoIP présentée dans ces étapes :

1. [Vérifier la signalisation numérique et analogique.](#)
2. [Vérifier les chiffres reçus et envoyés de l'analogie et des ports vocaux numériques.](#)
3. [Vérifier la signalisation de bout en bout VoIP.](#)
4. [Comprendre les problèmes de Qualité de service \(QoS\) VoIP](#)
5. [Comprendre les détails de codes de motif et des valeurs de débogage pour le VoIP](#)



Remarque : ce document n'explique pas toutes les facettes de l'architecture Cisco IOS utilisée dans les passerelles VoIP et les gardes-barrière Cisco. Au lieu de cela, il montrera quelles commandes peuvent être utilisées et quels champs de résultats peuvent être les plus utiles.



Attention : le débogage de Cisco IOS sollicite énormément le processeur. Exercez la précaution quand vous utilisez les débogages listés dans ce document. Pour plus d'informations, consultez les informations importantes sur des commandes de débogage.

Les débogages doivent être exécutés avec l'horodatage activé dans le journal. Activez l'horodatage avec les commandes : `service timestamps debug datetime msec`, `service timestampslog datetime msec` en mode enable. L'horodatage aide à déterminer l'intervalle du temps entre les changements d'état.

Flux d'appel dans le réseau

Un facteur important à considérer avant que vous commenciez un dépannage ou débogage VoIP est que des appels VoIP se composent de trois signaux d'appel. Ces signaux d'appel sont des systèmes téléphoniques commutés traditionnels de source (POTS), VoIP et POTS de destination. Ceci est montré dans le schéma suivant : Le dépannage et le débogage doit se concentrer d'abord indépendamment sur chaque signal d'appel VoIP dans son ensemble.

Flux d'appels du routeur

Ces définitions expliquent la fonction des composants principaux affichés dans l'organigramme de flux d'appels du routeur :

API de contrôle d'appel (interface de programmation d'application) - Trois clients se servent de l'API de contrôle d'appel. Les trois clients sont CLI, agent de Protocole de gestion de réseau simple (SNMP) et l'application de la session. Les principales fonctions de l'API de contrôle des appels (également appelée CCAPI) sont les suivantes :

- Identifiez les segments d'appel (par exemple, quel terminal de numérotation dial-peer est-ce ? d'où provient-il ?).
- Décider quelle application de session prend l'appel (par exemple, qui la prend en charge ?).
- Impliquer le dispositif de traitement de paquets.
- Associer les segments d'appel.
- Débuter d'enregistrer la statistique d'appels.

Application de session et mappeur du plan de numérotation - L'application de session utilise le mappeur du plan de numérotation pour mapper un numéro à un homologue de numérotation (POTS local ou VoIP distant). Le mappeur du plan de numérotation emploie la table des homologues de numérotation pour rechercher l'homologue de numérotation actif.

Telephony and VoIP Service Provider Interface (SPI) : le SPI de téléphonie communique avec les terminaux de numérotation dial-peer POTS (analogiques : fxs, fxo, e&m Digital : isdn, qsig, e&m, etc.). Le VoIP SPI est l'interface spécifique des homologues de numérotation VoIP. Les gestionnaires Telephony/DSP fournissent des services au SPI de téléphonie tandis que le VoIP SPI se fonde sur des protocoles de session.

Architecture d'interface de téléphonie

Ce schéma montre l'architecture des éléments constitutifs de téléphonie de routeur Cisco et comment ils interagissent les uns avec les autres.

Cette liste décrit les fonctions et les définitions des composants principaux de schéma :

- Application programming interface de Contrôle d'appel (CCAPI) - L'entité de logiciel qui établit, termine et pontage des signaux d'appel.
- Voice Telephony Service Provider (VTSP) : processus Cisco IOS qui traite les demandes provenant de l'API de contrôle d'appel et formule les demandes appropriées au processeur de signal numérique (DSP) ou au VPM.
- Module de processeur vocal (VPM) - Le VPM est responsable du pontage et des processus coordonnés de signalisation entre la machine d'état de signalisation de ports de téléphonie

(SSM), le manager de ressource DSP et le VTSP.

- Manager de ressource DSP - Le DSPRM fournit les interfaces par lesquelles le VTSP peut envoyer et recevoir des messages à et du DSP.
- Dispositif de traitement de paquets - Le dispositif de traitement de paquets transfère des paquets entre le DSP et les segments d'appel du partenaire.
- Homologue d'appel - L'homologue d'appel est le signal d'appel opposé. Ceci peut être une autre connexion vocale téléphonique (POTS), un VoFR, un VoATM, ou une connexion VoIP.

Vérifier signalisation numérique et analogique (le segment d'appel de POTS)

La vérification de la signalisation numérique et analogique consiste à :

- Déterminer que la signalisation analogique ou numérique appropriée avec combiné raccroché et décroché est reçue.
- Déterminer que la signalisation appropriée E&M, FXO et FXS est configurée sur le routeur et le commutateur (Co ou PBX).
- Vérifier que les DSP sont en mode de collecte de chiffres.

Les commandes tracées dans ces sections peuvent être utilisées pour vérifier la signalisation.

montrer contrôleurs t1/E1 (numérique)

montrer contrôleurs t1 [emplacement/port] - Utilisez cette commande d'abord. Elle montre si la connexion numérique de t1 entre le routeur et le commutateur (Co ou PBX) est en active ou non et si elle fonctionne correctement. Le résultat de cette commande ressemble à ceci :

```
<#root>
router#
show controllers T1 1/0
T1 1/0 is up.
Applique type is Channelized T1
Cablelength is short 133
No alarms detected.
Framing is ESF, Line Code is B8ZS, Clock Source is Line
Primary.
Data in current interval (6 seconds elapsed):
    0 Line Code Violations, 0 Path Code Violations
    0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs,
    0 Degraded Mins
```

```
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs,  
0 Unavail Secs
```

Si vous utilisez E1, utilisez la commande `show controllers e1`. Pour plus d'informations à ce sujet, consultez :

- [Dépannage de la couche 1 de T1](#)
- [Diagramme du dépannage T1](#)
- [Dépannage des problèmes de ligne série](#)

montrer port voix

`show voice portslot-number/port` : utilisez cette commande pour afficher l'état du port et les paramètres configurés sur le port vocal des cartes d'interface vocale (VIC) Cisco. Comme toutes les commandes Cisco IOS, les valeurs par défaut ne s'affichent pas dans `show running-config`, mais elles s'affichent avec cette commande.

Ceci est un exemple de sortie pour un port de voix 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
```

vpm de débogage (module de processeur vocal)

Ces commandes sont utilisées pour déboguer l'interface de téléphonie VPM :

- debug vpm signal - Cette commande est utilisée pour collecter des informations de débogage pour les événements de signalisation et peut être utile pour résoudre les problèmes de signalisation à un PBX.
- debug vpm spi - Cette commande vérifie comment l'interface du fournisseur de service de module de routage de port vocal (SPI) se connecte par interface à l'API de contrôle d'appel. Cette commande de débogage affiche des informations au sujet de la façon dont chaque indication et requête d'application de réseau est traitée.
- debug vpm dsp - Cette commande affiche des messages du protocole de système d'annuaire sur le VPM au routeur et peut être utile si vous suspectez que le VPM ne soit pas fonctionnel. C'est un moyen simple de vérifier si le VPM réagit aux indications de combiné raccroché et d'évaluer la temporisation pour des messages de signalisation de l'interface.
- debug vpm all - Cette commande EXEC active toutes les commandes debug vpm : debug vpm spi, debug vpm signal, et debug vpm dsp.

- debug vpm port - Utilisez cette commande pour limiter le résultat du débogage à un port particulier. Par exemple, cette sortie montre debug vpm dspmessages seulement pour le port 1/0/0 :

```
debug vpm dsp
```

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

Exemple de résultat pour debug vpm signalCommand

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

!--- FXS port 1/0/0 goes from the "on-hook" to "off-hook" !--- state.
htsp_process_event: [1/0/0, 1.2 , 36]
fxs1s_onhook_offhook htsp_setup_ind
*Mar 10 16:08:55.958: htsp_process_event:
[1/0/0, 1.3 , 8]

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

!--- End of phone call, port goes "on-hook".
*Mar 10 16:09:11.966: htsp_process_event:
[1/0/0, 1.3 , 6]
*Mar 10 16:09:17.218: htsp_process_event:
[1/0/0, 1.3 , 28]
fxs1s_offhook_onhook
*Mar 10 16:09:17.370: htsp_process_event:
[1/0/0, 1.3 , 41] fxs1s_offhook_timer
*Mar 10 16:09:17.382: htsp_process_event:
[1/0/0, 1.2 , 7]
fxs1s_onhook_release
```

Si le combiné raccroché et le combiné décroché ne signalent pas correctement, vérifiez ces

éléments :

- Vérifiez le câblage est correct.
- Vérifiez si le routeur et le commutateur (Co ou PBX) sont correctement fondus.
- Vérifiez que les deux extrémités de la connexion ont des configurations de signalisation équivalentes. Les configurations mal adaptées peuvent entraîner une signalisation inachevée ou à sens unique.

Pour plus d'informations sur le dépannage E&M, consultez Comprendre et dépanner les types d'interface d'E & M analogique et câbler des configurations.

Exemple de résultat pour 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
```

Vérifier les chiffres reçus et envoyés (le segment d'appel de POTS)

Une fois que la signalisation avec combiné raccroché et combiné décroché est vérifiée et si elle fonctionne correctement, vérifiez si les chiffres corrects sont reçus ou envoyés sur le port voix (numérique ou analogique). Un homologue de numérotation n'est pas mis en correspondance ou le commutateur (Co ou PBX) ne peut pas appeler la station correcte si des chiffres incorrects ou incomplets sont envoyés ou reçus. Certaines commandes peuvent être utilisées pour vérifier les chiffres reçus/envoyés, à savoir :

- show dialplan number - Cette commande est utilisée pour montrer quel partenaire de numérotation est atteint quand un numéro de téléphone particulier est composé.

- debug vtsp session - Cette commande montre les informations sur la façon dont chaque indication et requête d'application de réseau est traitée, signalant des messages d'indications et de contrôle de protocole de système d'annuaire.
- debug vtsp dsp - Dans les versions antérieures à Cisco IOS Version 12.3, cette commande affiche les chiffres pendant qu'ils sont reçus par le port voix. Cependant, dans le logiciel Cisco IOS version 12.3 et versions ultérieures, le résultat de la commande de débogage n'affiche plus les chiffres. La combinaison de debug hpi detail et debug hpinotification peut être utilisée pour voir les chiffres entrants.
- debug vtsp all - Cette commande active les commandes debug voice telephony service provider (VTSP) suivantes : debug vtsp session, debug vtsp error , et debug vtsp dsp.

montrer numéro de plan de numérotation

show dialplan number <digit_string> : cette commande affiche le terminal de numérotation dial-peer correspondant à une chaîne de chiffres. Si plusieurs homologues de numérotation peuvent être mis en correspondance, ils sont tous montrés dans la commande relative à la mise en correspondance.

Remarque : vous devez utiliser le signe # à la fin des numéros de téléphone pour les terminaux de numérotation dial-peer de longueur variable afin de correspondre aux modèles de destination qui se terminent par T.

Le résultat de cette commande ressemble à ceci :

```
<#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

```

débogage de session vtsp

La commande `debug vtsp session` montre les informations sur la façon dont le routeur interagit avec le protocole de système d'annuaire basé sur les indications de signalisation de la pile de signalisation et des requêtes de routage de l'application. Cette commande de débogage montre les informations au sujet de la façon dont chaque indication et requête d'application de réseau est traitée, signalant des messages d'indications et de contrôle de protocole de système d'annuaire.

<#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
```

S'il est déterminé que les chiffres ne sont pas envoyés ou reçus correctement, il peut être nécessaire d'utiliser un capteur de chiffres (outil de test) ou un testeur T1 pour vérifier que les chiffres sont envoyés à la fréquence et à l'intervalle de synchronisation corrects. S'ils sont envoyés « incorrectement » pour le commutateur (CO ou PBX), il peut être nécessaire d'ajuster certaines valeurs sur le routeur ou le commutateur (CO ou PBX) afin qu'elles correspondent et interagissent. Ce sont habituellement des valeurs de durée de chiffre et de durée d'inter-chiffre. Un autre élément pour examiner si les chiffres semblent être envoyés correctement sont les tables de traduction de numéros dans le commutateur (Co ou PBX) qui peut ajouter ou supprimer des chiffres.

Vérifier la signalisation de bout en bout VoIP (le segment d'appel de VOIP)

Après avoir vérifié que la signalisation du port voix fonctionne correctement et que les chiffres corrects sont reçus, essayez le dépannage et le débogage de contrôle d'appel VoIP. Ces facteurs expliquent pourquoi le débogage du contrôle d'appel peut devenir un travail complexe :

- Les passerelles Cisco VoIP utilisent la signalisation H.323 pour effectuer des appels. H.323 est constitué de trois couches de négociation et d'établissement d'appel : H.225, H.245 et H.323. Ces protocoles emploient une combinaison de TCP et UDP pour installer et établir un appel.
- Le débogage VoIP de bout en bout affiche un certain nombre de machines d'état Cisco IOS. Les problèmes de routage avec les machines - états peuvent faire échouer un appel.
- Le débogage de bout en bout VoIP peut être très bavard et créer beaucoup de résultat du

débogage.

debug voip ccapi inout

La commande principale de débogage des appels VoIP de bout en bout est debug voip ccapi inout. Le résultat d'un débogage d'appel est montré dans ce résultat.

```
<#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, digit=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)

```

Si l'appel échoue et que la cause semble être dans la partie VoIP de l'établissement d'appel, vous pouvez éventuellement avoir besoin d'examiner la partie H.225 ou H.245 TCP de l'établissement d'appel, par opposition à la partie UDP de l'établissement H.323. Les commandes qui peuvent être utilisées pour déboguer l'établissement H.225 ou H.245 sont :

- transactions de débogage ip tcp and paquet de débogage ip tcp — Ces commandes examinent la portion TCP de la négociation H.225 et H.245. Elles retournent les adresses IP, les ports TCP et les états des connexions TCP.

- debug cch323 h225 - Cette commande examine la partie H.225 de la négociation d'appel et trace la transition d'état de la machine H.225 basée sur l'événement traité. Pensez à ceci comme le niveau 1 de l'établissement de H.323 en trois parties.
- debug cch323 h225 - Cette commande examine la partie H.225 de la négociation d'appel et trace la transition d'état de la machine H.225 basée sur l'événement traité. Pensez à ceci comme le niveau 2 de l'établissement de H.323 en trois parties.

Comprendre les problèmes de qualité de service (QoS) VoIP

Quand des appels VoIP sont correctement établis, l'étape suivante est de vérifier que la qualité vocale est bonne. Bien que le dépannage de QoS ne soit pas couvert dans ce document, ces directives doivent être considérées pour atteindre la bonne qualité de la voix :

- Vous devez savoir combien de largeur de bande un appel VoIP consomme avec chaque codec. Ceci inclut le niveau 2 et les en-têtes IP/UDP/RTP. Pour plus d'informations, référez-vous à [Modifier le calcul de consommation de bande passante pour les appels vocaux](#).
- Comprendre que les caractéristiques du réseau IP les appels voyage plus de. Par exemple, la largeur de bande d'un réseau à relais de trame au CIR est différente du CIR au-dessus (ou salve), où les paquets peuvent être déposés ou mis en attente dans l'entité en relais de trame. Assurez-vous que le délai et instabilité sont contrôlés et éliminés autant que possible. Le délai de transmission unidirectionnel ne doit pas dépasser 150 ms (conformément à la recommandation G.114).
- Utilisez une technique de mise en file d'attente qui permet d'identifier et de hiérarchiser le trafic VoIP.
- Lorsque vous transmettez la VoIP sur des liaisons à faible débit, utilisez des techniques de fragmentation de paquets de couche 2, telles que MLPPP avec fragmentation et entrelacement de liaison (LFI) sur des liaisons point à point ou FRF.12 sur des liaisons Frame Relay. La fragmentation d'un plus grand paquet de données donne moins de dispersions et de retard à transmettre le trafic VoIP parce que les paquets VoIP peuvent être intercalés sur la liaison.
- Essayez d'utiliser un codec différent et d'essayer l'appel avec VAD activé et désactivé pour cerner le problème au protocole de système d'annuaire, par opposition au réseau IP.

Avec le VoIP, les principales choses à rechercher en dépannant des problèmes de QoS sont des paquets abandonnés et des goulots d'étranglement du réseau qui peuvent entraîner retards et instabilité.

Cherchez :

- abandons d'interface
- abandons de mémoire tampon

- congestion d'interface
- encombrement de liaison

Chaque interface dans le chemin de l'appel VoIP doit être examinée. En outre, éliminez les abandons et l'encombrement. En outre, le délai d'aller-retour doit être réduit autant que possible. Les pings entre les points d'extrémité VoIP donnent une indication du délai d'aller-retour d'une liaison. Le délai de transmission aller-retour ne doit pas dépasser 300 ms dans la mesure du possible. Si le retard doit dépasser cette valeur, des efforts doivent être pris pour s'assurer que ce retard est constant, pour ne pas introduire le dispersement ou le retard variable.

Une vérification doit également être effectuée pour garantir que le mécanisme de mise en file d'attente Cisco IOS place les paquets VoIP dans les files d'attente appropriées. Les commandes Cisco IOS, telles que show queue interface ou show priority, peuvent aider à vérifier la mise en file d'attente.

Détails de codes de motif et des valeurs de débogage pour le VoIP

Utilisez ces tableaux quand vous lisez des débogages et les valeurs associées dans les débogages.

Causes de la déconnexion d'appel Q.931 (cause_codes de debug voip ccapi inout)

Valeur de la cause de déconnexion d'appel (dans l'hexa)	Signification et numéro (dans la décimale)
CC_CAUSE_UANUM = 0x1	numéro non affecté. (1)
CC_CAUSE_NO_ROUTE = 0x3	aucune route à la destination. (3)
CC_CAUSE_NORM = 0x10	effacement d'appel normal. (16)
CC_CAUSE_BUSY = 0x11	utilisateur occupé (17)
CC_CAUSE_NORS = 0x12	aucune réponse de l'utilisateur. (18)
CC_CAUSE_NOAN = 0x13	aucune réponse d'utilisateur. (19)
CC_CAUSE_REJECT = 0x15	appel rejeté. (21)
CC_CAUSE_INVALID_NUMBER = 0x1C	nombre incorrect. (28)
CC_CAUSE_UNSP = 0x1F	normal, non spécifié. (31)
CC_CAUSE_NO_CIRCUIT = 0x22	aucun circuit. (34)
CC_CAUSE_NO_REQ_CIRCUIT = 0x2C	aucun circuit demandé. (44)
CC_CAUSE_NO_RESOURCE = 0x2F	aucune ressource. (47) 1
CC_CAUSE_NOSV = 0x3F	service ou option non disponible, ou non spécifiée. (63)

¹Ce problème peut se produire en raison d'une non-correspondance de codecs dans la

configuration H323. La première étape de dépannage consiste donc à coder en dur les terminaux de numérotation dial-peer VoIP pour utiliser le codec correct.

Valeurs de négociation de codecs (du debug voip ccapi inout)

Pour plus d'informations sur les CODEC, référez-vous à [Comprendre les codecs : Complexité, Support matériel, MOS et Négociation](#).

Valeur de négociation	Signification
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

Types de tonalité

Types de tonalité	Signification
CC_TONE_RINGBACK 0x1	Tonalité de sonnerie
CC_TONE_FAX 0x2	Tonalité de télécopie
CC_TONE_BUSY 0x4	Signal d'occupation
CC_TONE_DIALTONE 0x8	Tonalité
CC_TONE_OOS 0x10	Tonalité hors service
CC_TONE_ADDR_ACK 0x20	Tonalité d'accusé de réception d'adresse
CC_TONE_DISCONNECT 0x40	Tonalité de déconnexion
CC_TONE_OFF_HOOK_NOTICE 0x80	Tonalité indiquant que le téléphone a été décroché

CC_TONE_OFF_HOOK_ALERT 0x100	Une version plus récente de CC_TONE_OFF_HOOK_NOTICE
CC_TONE_CUSTOM 0x200	Tonalité personnalisée : utilisée lorsque vous spécifiez une tonalité personnalisée
CC_TONE_NULL 0x0	Tonalité nulle

Débit de télécopie et valeurs de fonctionnalités VAD

Valeurs	Signification
CC_CAP_FAX_NONE 0x1	Débranchement de télécopie ou non disponible
CC_CAP_FAX_VOICE 0x2	Appel vocal
CC_CAP_FAX_144 0x4	14,400 bauds
CC_CAP_FAX_96 0x8	9600 bauds
CC_CAP_FAX_72 0x10	7,200 bauds
CC_CAP_FAX_48 0x20	4,800 bauds
CC_CAP_FAX_24 0x40	2,400 bauds
CC_CAP_VAD_OFF 0x1	VAD désactivé
CC_CAP_VAD_ON 0x2	vad activé

Informations connexes

- [Dépannage de la couche 1 de T1](#)
- [Dépannage de T1](#)
- [Dépannage des problèmes de ligne série](#)
- [Dépannage des problèmes de téléphonie IP Cisco](#)
- [Assistance technique de Cisco et téléchargements](#)

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.