

VoIP avec PPP sur ligne louée à large bande passante et LLQ

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Informations générales](#)

[Configurez](#)

[Diagramme du réseau](#)

[Configurations](#)

[Vérifiez](#)

[Vérification pour le routeur de San Jose](#)

[Vérification pour le routeur de Raleigh](#)

[Dépannez](#)

[Dépannage des commandes](#)

[Informations connexes](#)

[Introduction](#)

Ce document fournit des configurations d'échantillon pour deux Routeurs de Cisco 3640. Les configurations permettent aux Routeurs de communiquer avec le VoIP avec le PPP au-dessus d'une ligne louée de bande passante élevée avec le Fonction Low Latency Queuing (LLQ). Pour plus d'informations sur LLQ, référez-vous au document [VoIP au-dessus des liens de PPP avec la qualité de service \(LLQ/IP RTP Priority, LFI, cRTP\)](#).

Remarque: Quand ce document discute la bande passante élevée en termes de VoIP et QoS, la bande passante élevée est n'importe quelle bande passante au-dessus de 768 Kbps.

[Conditions préalables](#)

[Conditions requises](#)

Aucune spécification déterminée n'est requise pour ce document.

[Composants utilisés](#)

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- IP Plus de version de logiciel 12.2(19a) de Cisco IOS® ou toute autre version du logiciel Cisco IOS de 12.2, 12.2T, 12.3, ou 12.3T
- Deux Routeurs de Cisco 3640 avec au moins mi-bande 48 la mémoire vive dynamique et 16 de mémoire flash
- Deux modules réseau d'emplacement de carte d'interface de voix/télécopie de Cisco NM-2V plus deux cartes d'interface VIC-2FXS
- Deux interfaces série Dans cet exemple, les deux interfaces série sont NM-1E2Ws, avec une carte d'interface WAN chacune WIC-1T.
- Les téléphones analogiques pour la connexion au Foreign Exchange Station (FXS) met en communication pour des communications voix

Remarque: Les modules réseau NM-1E2W, NM-1E1R2W, et NM-2E2W n'ont pas assez d'alimentation de représentation de prendre en charge le WIC-2T. Le manque de support est dû aux limitations matérielles.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Conventions

Pour plus d'informations sur les conventions de documents, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

Informations générales

Si le temps nécessaire pour envoyer un paquet 1500-byte sur le fil est plus grand que 10 ms, vous devez fragmenter des paquets. Ce document présente une configuration sans fragmentation. La configuration est pour un lien 1544-kilobit pour lequel le retard de transmission pour un paquet 1500-byte est moins de 10 ms.

Remarque: Dans certains cas dans ce que vous avez une connexion dédiée et pleine de t1, une fonctionnalité de fragmentation peut être inutile. Mais, vous avez besoin toujours d'un mécanisme de QoS. Utilisation LLQ dans ce cas. Si la durée nécessaire pour envoyer un paquet 1500-byte sur le fil est moins de 10 ms, vous n'avez pas besoin de fragmenter des paquets. Le plein t1 offre l'assez de bande passante pour permettre à des paquets vocaux pour écrire et laisser les questions de file d'attente sans tarder.

Remarque: Si vous avez activé la fragmentation sur le routeur, il y a d'activation du mécanisme de mise en file d'attente 100 pour cent du temps. Si vous avez configuré LLQ, la valeur vous avez configuré des limites le trafic pour la file d'attente prioritaire. Quand vous n'avez pas activé la fragmentation, le routeur applique seulement la stratégie QoS dans le cas de l'encombrement.

En outre, dans le cas de ligne débits qui sont plus grande que 768 Kbps, le Real-Time Transport Protocol compressé (cRTP) peut être inutile. Référez-vous au document [VoIP au-dessus des liens de PPP avec la qualité de service \[LLQ/IP RTP Priority, LFI, cRTP\]](#). L'utilisation du cRTP aide à sauvegarder la bande passante parce que le cRTP compresse des en-têtes de RTP IP. Dans la section de [configurations de](#) ce document, l'activation du cRTP est inutile. Le t1 permet à l'assez de bande passante pour que les paquets vocaux coulent, sans compactage, sur le fil sans question.

Attention : Si vous décidez d'utiliser le cRTP, rendez-vous compte que le cRTP utilise des ressources CPU. Le cRTP peut surcharger un routeur qui a une charge lourde du trafic vocal.

Remarque: Dans cette configuration, les deux Routeurs connectent dos à dos au-dessus d'une ligne louée. Mais, dans la plupart des topologies, les Routeurs avec l'activation de Voix peuvent exister n'importe où. Habituellement, les Routeurs de Voix se connectent à la Connectivité de RÉSEAU LOCAL à d'autres Routeurs qui se connectent au WAN. Si vos Routeurs de Voix ne se connectent pas par l'intermédiaire du PPP au-dessus d'une ligne louée, vous devez configurer toutes les commandes de configuration de connectivité WAN sur ces Routeurs qui se connectent au WAN ; vous ne configurez pas les commandes sur les Routeurs de Voix, que les [configurations](#) dans ce document affichent.

Remarque: Cette configuration peut fonctionner pour Cisco 1700, [2600, 3600, et des](#) Routeurs de [gamme 3700](#).

[Configurez](#)

Cette section vous fournit des informations pour configurer les fonctionnalités décrites dans ce document.

Remarque: Pour obtenir des informations supplémentaires sur les commandes utilisées dans ce document, utilisez l'[Outil de recherche de commande](#) ([clients enregistrés](#) seulement).

[Diagramme du réseau](#)

Ce document utilise la configuration réseau suivante :

[Configurations](#)

Ce document utilise les configurations suivantes :

- [San Jose](#)
- [Raleigh](#)

San Jose

```
SanJose3640A# show run Building configuration... Current
configuration : 1425 bytes ! version 12.2 service
timestamps debug datetime msec service timestamps log
datetime msec no service password-encryption ! hostname
SanJose3640A ! logging buffered 50000 debugging ! ip
subnet-zero ! ! no ip domain-lookup ! call rsvp-sync ! !
! ! ! ! ! class-map match-all voice-signaling match
access-group 103 class-map match-all voice-traffic match
access-group 102 ! ! policy-map voice-policy class
voice-traffic priority 51 !--- These are two
uncompressed G729 VoIP calls at 24 kpbs each !--- that
have voice activity detection (VAD) disablement. You
also need !--- to consider the Layer 2 (L2) overhead.
class voice-signaling bandwidth 16 !--- This assigns a
queue for voice signaling traffic that ensures 8 kbps.
!--- Note: This action is optional and has nothing to do
with good voice !--- quality. This queue assignment is a
way to secure signaling. class class-default fair-queue
```

```

!--- The class-default class classifies traffic that
does !--- not fall into one of the class definitions.
The fair-queue command !--- associates the default class
weighted fair queuing (WFQ). !!! interface Ethernet1/0
ip address 10.89.251.158 255.255.255.192 half-duplex !
interface Serial1/0 bandwidth 1544 ip address
192.168.1.1 255.255.255.0 service-policy output voice-
policy encapsulation ppp load-interval 30 clockrate
2000000 ! ip classless ip route 0.0.0.0 0.0.0.0
10.89.251.129 no ip http server ! access-list 102 permit
udp any any range 16384 32767 access-list 103 permit tcp
any eq 1720 any access-list 103 permit tcp any any eq
1720 ! voice-port 3/0/0 ! voice-port 3/0/1 ! voice-port
3/1/0 ! voice-port 3/1/1 ! dial-peer cor custom !!!
dial-peer voice 1 voip incoming called-number .
destination-pattern 2... session target ipv4:192.168.1.2
dtmf-relay h245-alphanumeric no vad ! dial-peer voice 2
pots destination-pattern 1001 port 3/0/0 ! dial-peer
voice 3 pots destination-pattern 1002 port 3/0/1 !!
line con 0 line aux 0 line vty 0 4 password cisco login
! end SanJose3640A# SanJose3640A# SanJose3640A# show
version Cisco Internetwork Operating System Software IOS
(tm) 3600 Software (C3640-IS-M), Version 12.2(19a),
RELEASE SOFTWARE (fc2) Copyright (c) 1986-2003 by cisco
Systems, Inc. Compiled Mon 29-Sep-03 23:45 by pwade
Image text-base: 0x60008930, data-base: 0x61134000 ROM:
System Bootstrap, Version 11.1(20)AA2, EARLY DEPLOYMENT
RELEASE SOFTWARE (fc1) SanJose3640A uptime is 5 minutes
System returned to ROM by reload System image file is
"flash:c3640-is-mz.122-19a.bin" cisco 3640 (R4700)
processor (revision 0x00) with 126976K/4096K bytes of
memory. Processor board ID 15636516 R4700 CPU at 100Mhz,
Implementation 33, Rev 1.0 Bridging software. X.25
software, Version 3.0.0. SuperLAT software (copyright
1990 by Meridian Technology Corp). 1 Ethernet/IEEE 802.3
interface(s) 1 Serial network interface(s) 2 Voice FXO
interface(s) 2 Voice FXS interface(s) DRAM configuration
is 64 bits wide with parity disabled. 125K bytes of non-
volatile configuration memory. 32768K bytes of processor
board System flash (Read/Write) 16384K bytes of
processor board PCMCIA Slot1 flash (Read/Write)
Configuration register is 0x2102 SanJose3640A#

```

Raleigh

```

Raleigh3640A# show run Building configuration... Current
configuration : 1406 bytes ! version 12.2 service
timestamps debug datetime msec service timestamps log
datetime msec no service password-encryption ! hostname
Raleigh3640A ! logging buffered 50000 debugging ! ip
subnet-zero !! no ip domain-lookup ! call rsvp-sync !!
!!!! class-map match-all voice-signaling match
access-group 103 class-map match-all voice-traffic match
access-group 102 !! policy-map voice-policy class
voice-traffic priority 51 !--- These are two
uncompressed G729 VoIP calls at 24 kbps each !--- that
have VAD disablement. You also need to consider !--- the
L2 overhead. class voice-signaling bandwidth 16 !---
This assigns a queue for voice signaling traffic that
ensures 8 kbps. !--- Note: This action is optional and
has nothing to do with good voice !--- quality. This
queue assignment is a way to secure signaling. class
class-default fair-queue !--- The class-default class
classifies traffic that does !--- not fall into one of
the class definitions. The fair-queue command !---

```

```

associates the default class WFQ. !!! interface
Ethernet1/0 ip address 10.89.251.159 255.255.255.192
half-duplex ! interface Serial1/0 bandwidth 1544 ip
address 192.168.1.2 255.255.255.0 service-policy output
voice-policy encapsulation ppp load-interval 30 ! ip
classless ip route 0.0.0.0 0.0.0.0 10.89.251.129 no ip
http server ! access-list 102 permit udp any any range
16384 32767 access-list 103 permit tcp any eq 1720 any
access-list 103 permit tcp any any eq 1720 ! voice-port
3/0/0 ! voice-port 3/0/1 ! voice-port 3/1/0 ! voice-port
3/1/1 ! dial-peer cor custom !!! dial-peer voice 1
voip incoming called-number . destination-pattern 1...
session target ipv4:192.168.1.1 dtmf-relay h245-
alphanumeric no vad ! dial-peer voice 2 pots
destination-pattern 2001 port 3/0/0 ! dial-peer voice 3
pots destination-pattern 2002 port 3/0/1 !! line con 0
line aux 0 line vty 0 4 password cisco login ! end
Raleigh3640A# Raleigh3640A# Raleigh3640A# show version
Cisco Internetwork Operating System Software IOS (tm)
3600 Software (C3640-IS-M), Version 12.2(19a), RELEASE
SOFTWARE (fc2) Copyright (c) 1986-2003 by cisco Systems,
Inc. Compiled Mon 29-Sep-03 23:45 by pwade Image text-
base: 0x60008930, data-base: 0x61134000 ROM: System
Bootstrap, Version 12.1(17r) [cmong 17r], RELEASE
SOFTWARE (fc1) Raleigh3640A uptime is 6 minutes System
returned to ROM by reload System image file is
"flash:c3640-is-mz.122-19a.bin" cisco 3640-A (R4700)
processor (revision 0x00) with 94208K/4096K bytes of
memory. Processor board ID 29851759 R4700 CPU at 100Mhz,
Implementation 33, Rev 1.0 Bridging software. X.25
software, Version 3.0.0. SuperLAT software (copyright
1990 by Meridian Technology Corp). 1 Ethernet/IEEE 802.3
interface(s) 1 Serial network interface(s) 2 Voice FXO
interface(s) 2 Voice FXS interface(s) DRAM configuration
is 64 bits wide with parity disabled. 123K bytes of non-
volatile configuration memory. 32768K bytes of processor
board System flash (Read/Write) 16384K bytes of
processor board PCMCIA Slot0 flash (Read/Write)
Configuration register is 0x2102 Raleigh3640A#

```

Vérifiez

Après que vous écriviez ces [configurations](#) dans vos Routeurs, vérifiez qu'elles fonctionnent correctement. Les commandes et la sortie respective ici expliquent une implémentation réussie des configurations.

Certaines commandes **show** sont prises en charge par l'[Output Interpreter Tool](#) ([clients enregistrés](#) uniquement), qui vous permet de voir une analyse de la sortie de la commande show.

- **affichez l'interface série 1/0 d'interface** — Te permet pour vérifier le statut de votre interface série.
- **brief de show call active voice** — Te permet pour visualiser les informations d'appel pendant un appel.
- **show call active voice** — Te permet pour visualiser les informations d'appel pendant un appel.
- **show policy-map interface** — Te permet pour vérifier la stratégie QoS que l'interface utilise.
- **affichez la liste d'accès 102** — Te permet pour vérifier la sélection de paquet par la liste d'accès pour la classe de Voix. Émettez la commande une deuxième fois après que quelques

secondes et la vérifiez qu'il y a une augmentation du compte de paquet. Émettez la commande du **clear access-list counters 102**, s'il y a lieu.

- **résumé de show voice call** — Te permet pour vérifier le statut des appels. La commande t'affiche si les appels ont la connexion.
- **résumé de show voice port** — Te permet pour vérifier l'état des ports vocaux. La commande affiche les ports vocaux en tant qu'avec combiné raccroché ou hors fonction-crochet.
- **show voice dsp** — Te permet pour vérifier l'état du processeur de signaux numériques (DSP) et le codeur-décodeur (codec) que chaque appel utilise.

Vérification pour le routeur de San Jose

Avant que vous exécutiez la vérification, vérifiez les interfaces pour s'assurer que vous avez la Connectivité nécessaire pour placer des appels. Émettez la commande de l'**interface série 1/0 d'interface d'exposition** de vérifier le statut de votre interface série. Avec les [configurations](#) dans ce document, soyez sûr que votre interface série et interfaces multiliason sont alignées en protocole vers le haut d'état. Soyez également sûr que vous voyez ceci :

- **LCP s'ouvrent, multilink ouvert** — Indique l'établissement de la connexion PPP.
- **Ouvrez-vous : IPCP, CDPCP** — Te dit que l'envoi du trafic IP est possible par le lien de PPP.
- **Stratégie de queue : weighted fair** — Correspond à l'interface de ligne de commande de sortie de service-stratégie (CLI) sous l'interface série. La stratégie est pour la configuration de LLQ pour donner la priorité à la Voix au-dessus des données.

```
SanJose3640A# show interface serial 1/0 Serial1/0 is up, line protocol is up Hardware is QUICC
Serial Internet address is 192.168.1.1/24 MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255 Encapsulation PPP, loopback not set Keepalive
set (10 sec) LCP Open Open: IPCP, CDPCP Last input 00:00:27, output 00:00:02, output hang never
Last clearing of "show interface" counters 00:00:05 Input queue: 0/75/0/0
(size/max/drops/flushes); Total output drops: 0 Queueing strategy: weighted fair Output queue:
0/1000/64/0 (size/max total/threshold/drops) Conversations 0/1/256 (active/max active/max total)
Reserved Conversations 1/1 (allocated/max allocated) Available Bandwidth 1091 kilobits/sec 30
second input rate 0 bits/sec, 0 packets/sec 30 second output rate 0 bits/sec, 0 packets/sec 1
packets input, 16 bytes, 0 no buffer Received 0 broadcasts, 0 runts, 0 giants, 0 throttles 0
input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort 1 packets output, 16 bytes, 0
underruns 0 output errors, 0 collisions, 0 interface resets 0 output buffer failures, 0 output
buffers swapped out 0 carrier transitions DCD=up DSR=up DTR=up RTS=up CTS=up SanJose3640A#
```

Cette sortie affiche la Connectivité réussie entre les Routeurs. Si vous ne voyez pas que la ligne protocole est en hausse, vérifiez le rythme d'horloge qui est sur l'interface DCE. Quelques interfaces série ne prennent en charge pas la grande vitesse, telle que le NM-8A/S. En outre, vérifiez que les paramètres des deux côtés s'assortissent et, le plus important, que l'encapsulation s'assortit.

La sortie de la commande brief de **show call active voice** ici affiche deux appels réussis. Un appel est du routeur de Raleigh au routeur de San Jose, et l'autre est de San Jose à Raleigh. Cette liste explique la sortie qui apparaît en caractères gras :

- **Active de la réponse 1001** — Signifie que San Jose est le routeur dont l'appel commence.
- **3/0/0 télé-** — Signifie que c'est le tronçon d'appel de téléphonie.
- **Lancez l'active 2001** — Signifie qu'un téléphone du côté de Raleigh reçoit l'appel.
- **IP 192.168.1.2** — Signifie que c'est le tronçon d'appel IP.
- **Active de la réponse 2002** — Signifie que Raleigh est le routeur auquel l'appel envoie.
- **IP 192.168.1.2** — Signifie que c'est le tronçon d'appel IP.

- **Lancez l'active 1002** — Signifie qu'un téléphone du côté de San Jose reçoit l'appel.
- **3/0/1 télé-** — Signifie que c'est le tronçon d'appel de téléphonie.

```
SanJose3640A# show call active voice brief <ID>: <start>hs.<index> +<connect> pid:<peer_id>
<dir> <addr> <state> dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> IP <ip>:<udp>
rtt:<time>ms pl:<play>/<gap>ms lost:<lost>/<early>/<late> delay:<last>/<min>/<max>ms <codec>
MODEMPASS <method> buf:<fills>/<drains> loss <overall%> <multipkt>/<corrected> last <buf event
time>s dur:<Min>/<Max>s FR <protocol> [int dlci cid] vad:<y/n> dtmf:<y/n> seq:<y/n> <codec>
(payload size) ATM <protocol> [int vpi/vci cid] vad:<y/n> dtmf:<y/n> seq:<y/n> <codec> (payload
size) Tele <int>: tx:<tot>/<v>/<fax>ms <codec> noise:<l> acom:<l> i/o:<l>/<l> dBm Proxy
<ip>:<audio udp>,<video udp>,<tcp0>,<tcp1>,<tcp2>,<tcp3> endpt: <type>/<manf> bw: <req>/<act>
codec: <audio>/<video> tx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120
pkts>/<t120 bytes> rx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120
bytes> Total call-legs: 4 11E8 : 115599hs.1 +318 pid:2 Answer 1001 active dur 00:00:29
tx:1545/30900 rx:1544/30880 Tele 3/0/0:20: tx:30890/30890/0ms g729r8 noise:0 acom:2 i/o:-35/-44
dBm 11E8 : 115823hs.1 +94 pid:1 Originate 2001 active dur 00:00:31 tx:1556/31120 rx:1602/32040
IP 192.168.1.2:17360 rtt:4ms pl:25590/0ms lost:0/1/0 delay:69/69/70ms g729r8 11F0 : 116855hs.1
+156 pid:1 Answer 2002 active dur 00:00:20 tx:1087/21740 rx:1009/20180 IP 192.168.1.2:16772
rtt:2ms pl:17270/0ms lost:0/0/0 delay:69/69/70ms g729r8 11F0 : 116855hs.2 +156 pid:3 Originate
1002 active dur 00:00:20 tx:1009/20180 rx:1087/21740 Tele 3/0/1 (23): tx:21740/21740/0ms g729r8
noise:0 acom:5 i/o:-40/-40 dBm Total call-legs: 4 SanJose3640A#
```

Cette sortie de la commande de **show call active voice** fournit plus de détail au sujet de l'appel actif :

```
SanJose3640A# show call active voice Total call-legs: 4 GENERIC: SetupTime=115599 ms Index=1
PeerAddress=1001 PeerSubAddress= PeerId=2 PeerIfIndex=9 LogicalIfIndex=4 ConnectTime=115917
CallDuration=00:05:05 CallState=4 CallOrigin=2 ChargedUnits=0 InfoType=2 TransmitPackets=15338
TransmitBytes=306760 ReceivePackets=15337 ReceiveBytes=306740 TELE: ConnectionId=[0x38D3783F
0x14F111CC 0x801CFDB1 0x2D0CC4A5] IncomingConnectionId=[0x38D3783F 0x14F111CC 0x801CFDB1
0x2D0CC4A5] TxDuration=306740 ms VoiceTxDuration=306740 ms FaxTxDuration=0 ms
CoderTypeRate=g729r8 NoiseLevel=0 ACOMLevel=5 OutSignalLevel=-43 InSignalLevel=-36
InfoActivity=2 ERLLevel=5 SessionTarget= ImgPages=0 GENERIC: SetupTime=115823 ms Index=1
PeerAddress=2001 PeerSubAddress= PeerId=1 PeerIfIndex=8 LogicalIfIndex=0 ConnectTime=115917
CallDuration=00:05:07 CallState=4 CallOrigin=1 ChargedUnits=0 InfoType=2 TransmitPackets=15357
TransmitBytes=307140 ReceivePackets=15403 ReceiveBytes=308060 VOIP: ConnectionId[0x38D3783F
0x14F111CC 0x801CFDB1 0x2D0CC4A5] IncomingConnectionId[0x38D3783F 0x14F111CC 0x801CFDB1
0x2D0CC4A5] RemoteIPAddress=192.168.1.2 RemoteUDPPort=17360
RemoteSignallingIPAddress=192.168.1.2 RemoteSignallingPort=1720 RemoteMediaIPAddress=192.168.1.2
RemoteMediaPort=17360 RoundTripDelay=1 ms SelectedQoS=best-effort tx_DtmfRelay=h245-alphanumeric
FastConnect=TRUE Separate H245 Connection=FALSE H245 Tunneling=TRUE SessionProtocol=cisco
SessionTarget=ipv4:192.168.1.2 OnTimeRvPayout=300810 GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms GapFillWithInterpolation=0 ms GapFillWithRedundancy=0 ms
HiWaterPayoutDelay=70 ms LoWaterPayoutDelay=69 ms ReceiveDelay=69 ms LostPackets=0
EarlyPackets=2 LatePackets=0 VAD = disabled CoderTypeRate=g729r8 CodecBytes=20 GENERIC:
SetupTime=116855 ms Index=1 PeerAddress=2002 PeerSubAddress= PeerId=1 PeerIfIndex=8
LogicalIfIndex=0 ConnectTime=117011 CallDuration=00:04:56 CallState=4 CallOrigin=2
ChargedUnits=0 InfoType=2 TransmitPackets=14915 TransmitBytes=298300 ReceivePackets=14837
ReceiveBytes=296740 VOIP: ConnectionId[0x6C135AD4 0x14F311CC 0x8024CE4C 0xAA60AB15]
IncomingConnectionId[0x6C135AD4 0x14F311CC 0x8024CE4C 0xAA60AB15] RemoteIPAddress=192.168.1.2
RemoteUDPPort=16772 RemoteSignallingIPAddress=192.168.1.2 RemoteSignallingPort=11004
RemoteMediaIPAddress=192.168.1.2 RemoteMediaPort=16772 RoundTripDelay=7 ms SelectedQoS=best-
effort tx_DtmfRelay=h245-alphanumeric FastConnect=TRUE Separate H245 Connection=FALSE H245
Tunneling=TRUE SessionProtocol=cisco SessionTarget= OnTimeRvPayout=295580 GapFillWithSilence=0
ms GapFillWithPrediction=0 ms GapFillWithInterpolation=0 ms GapFillWithRedundancy=0 ms
HiWaterPayoutDelay=70 ms LoWaterPayoutDelay=69 ms ReceiveDelay=69 ms LostPackets=0
EarlyPackets=0 LatePackets=0 VAD = disabled CoderTypeRate=g729r8 CodecBytes=20 GENERIC:
SetupTime=116855 ms Index=2 PeerAddress=1002 PeerSubAddress= PeerId=3 PeerIfIndex=10
LogicalIfIndex=5 ConnectTime=117011 CallDuration=00:04:59 CallState=4 CallOrigin=1
ChargedUnits=0 InfoType=2 TransmitPackets=14952 TransmitBytes=299040 ReceivePackets=15030
ReceiveBytes=300600 TELE: ConnectionId=[0x6C135AD4 0x14F311CC 0x8024CE4C 0xAA60AB15]
IncomingConnectionId=[0x6C135AD4 0x14F311CC 0x8024CE4C 0xAA60AB15] TxDuration=300600 ms
VoiceTxDuration=300600 ms FaxTxDuration=0 ms CoderTypeRate=g729r8 NoiseLevel=0 ACOMLevel=5
```

OutSignalLevel=-40 InSignalLevel=-41 InfoActivity=2 ERLLevel=5 SessionTarget= ImgPages=0Total call-legs: 4 SanJose3640A#\$ Other shows:

La sortie de la commande de **show policy-map interface** inclut cette déclaration grasse :

- **30 seconde débits offerts 51000 bps** — Affiche la bande passante que les deux appels exigent, 51 kpbs.

```
SanJose3640A# show policy-map interface Serial1/0 Service-policy output: voice-policy Class-map: voice-traffic (match-all) 99403 packets, 6401420 bytes 30 second offered rate 51000 bps, drop rate 0 bps Match: access-group 102 Queueing Strict Priority Output Queue: Conversation 264 Bandwidth 51 (kbps) Burst 1275 (Bytes) (pkts matched/bytes matched) 407/65676 (total drops/bytes drops) 0/0 Class-map: voice-signaling (match-all) 158 packets, 12926 bytes 30 second offered rate 0 bps, drop rate 0 bps Match: access-group 103 Queueing Output Queue: Conversation 265 Bandwidth 16 (kbps) Max Threshold 64 (packets) (pkts matched/bytes matched) 158/12926 (depth/total drops/no-buffer drops) 0/0/0 Class-map: class-default (match-any) 75 packets, 9221 bytes 30 second offered rate 0 bps, drop rate 0 bps Match: any Queueing Flow Based Fair Queueing Maximum Number of Hashed Queues 256 (total queued/total drops/no-buffer drops) 0/0/0 SanJose3640A#
```

Sortez du **show access-lists 102** que la commande inclut cette déclaration grasse :

- **100676 correspondances** — Prouve que la hiérarchisation des paquets de RTP se produit parce que les paquets atteignent la liste d'accès 102.

```
SanJose3640A# show access-lists 102 Extended IP access list 102 permit udp any any range 16384 32767 (100676 matches) SanJose3640A# SanJose3640A# SanJose3640A# SanJose3640A# SanJose3640A# show access-lists 102 Extended IP access list 102 permit udp any any range 16384 32767 (100930 matches) SanJose3640A# SanJose3640A# SanJose3640A# show access-lists 102 Extended IP access list 102 permit udp any any range 16384 32767 (101076 matches) SanJose3640A# SanJose3640A# SanJose3640A# show access-lists 102 Extended IP access list 102 permit udp any any range 16384 32767 (101198 matches) SanJose3640A# SanJose3640A# show access-lists 102 Extended IP access list 102 permit udp any any range 16384 32767 (101304 matches) SanJose3640A# SanJose3640A# SanJose3640A# SanJose3640A# show voice call sum PORT CODEC VAD VTSP STATE VPM STATE ===== 3/0/0 g729r8 n S_CONNECT FXSLS_CONNECT 3/0/1 g729r8 n S_CONNECT FXSLS_CONNECT 3/1/0 - - - FXOLS_ONHOOK 3/1/1 - - - FXOLS_ONHOOK SanJose3640A# SanJose3640A# SanJose3640A# SanJose3640A# show voice port sum IN OUT PORT CH SIG-TYPE ADMIN OPER STATUS STATUS EC ===== 3/0/0 -- fxs-ls up up off-hook idle y 3/0/1 -- fxs-ls up up off-hook idle y 3/1/0 -- fxo-ls up dorm idle on-hook y 3/1/1 -- fxo-ls up dorm idle on-hook y SanJose3640A# SanJose3640A# show voice dsp DSP DSP DSPWARE CURR BOOT PAK TX/RX TYPE NUM CH CODEC VERSION STATE STATE RST AI VOICEPORT TS ABORT PACK COUNT ===== C542 001 01 g729r8 3.4.55 busy idle 0 0 3/0/0 NA 0 62487/61902 C542 002 01 g729r8 3.4.55 busy idle 0 0 3/0/1 NA 0 44362/44194 C542 003 01 g711ulaw 3.4.55 IDLE idle 0 0 3/1/0 NA 0 541/546 C542 004 01 g711ulaw 3.4.55 IDLE idle 0 0 3/1/1 NA 0 535/532 SanJose3640A#
```

[Vérification pour le routeur de Raleigh](#)

La procédure de vérification pour le routeur de Raleigh est semblable à la procédure pour le routeur de San Jose.

```
Raleigh3640A# show interface serial 1/0 Serial1/0 is up, line protocol is up Hardware is QUICC Serial Internet address is 192.168.1.2/24 MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, reliability 255/255, txload 1/255, rxload 1/255 Encapsulation PPP, loopback not set Keepalive set (10 sec) LCP Open Open: IPCP, CDPCP Last input 00:00:15, output 00:00:00, output hang never Last clearing of "show interface" counters 00:12:33 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0 Queueing strategy: weighted fair Output queue: 0/1000/64/0 (size/max total/threshold/drops) Conversations 0/1/256 (active/max active/max total) Reserved Conversations 1/1 (allocated/max allocated) Available Bandwidth 1091 kilobits/sec 30 second input rate 0 bits/sec, 0 packets/sec 30 second output rate 0 bits/sec, 0 packets/sec 167 packets input, 6849 bytes, 0 no buffer Received 0 broadcasts, 0 runts, 0 giants, 0 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort 169 packets output, 6907 bytes, 0 underruns 0 output errors, 0 collisions, 0 interface resets 0 output buffer failures, 0 output
```


buffers swapped out 11 carrier transitions DCD=up DSR=up DTR=up RTS=up CTS=up Raleigh3640A#
Raleigh3640A# Raleigh3640A# Raleigh3640A# Raleigh3640A# **show call active voice** Total call-legs:
4 GENERIC: SetupTime=209451 ms Index=1 PeerAddress=1001 PeerSubAddress= PeerId=1 PeerIfIndex=8
LogicalIfIndex=0 ConnectTime=209543 CallDuration=00:08:20 CallState=4 CallOrigin=2
ChargedUnits=0 InfoType=2 TransmitPackets=25054 TransmitBytes=501080 ReceivePackets=25008
ReceiveBytes=500160 VOIP: ConnectionId[0x38D3783F 0x14F111CC 0x801CFDB1 0x2D0CC4A5]
IncomingConnectionId[0x38D3783F 0x14F111CC 0x801CFDB1 0x2D0CC4A5] RemoteIPAddress=192.168.1.1
RemoteUDPPort=17210 RemoteSignallingIPAddress=192.168.1.1 RemoteSignallingPort=11006
RemoteMediaIPAddress=192.168.1.1 RemoteMediaPort=17210 RoundTripDelay=3 ms SelectedQoS=best-
effort tx_DtmfRelay=h245-alphanumeric FastConnect=TRUE Separate H245 Connection=FALSE H245
Tunneling=TRUE SessionProtocol=cisco SessionTarget= OnTimeRvPayout=497610 GapFillWithSilence=0
ms GapFillWithPrediction=0 ms GapFillWithInterpolation=0 ms GapFillWithRedundancy=0 ms
HiWaterPayoutDelay=70 ms LoWaterPayoutDelay=69 ms ReceiveDelay=69 ms LostPackets=0
EarlyPackets=1 LatePackets=0 **VAD = disabled CoderTypeRate=g729r8** CodecBytes=20 GENERIC:
SetupTime=209451 ms Index=2 **PeerAddress=2001** PeerSubAddress= PeerId=2 PeerIfIndex=9
LogicalIfIndex=4 ConnectTime=209543 **CallDuration=00:08:21** CallState=4 CallOrigin=1
ChargedUnits=0 InfoType=2 TransmitPackets=25074 TransmitBytes=501480 ReceivePackets=25120
ReceiveBytes=502400 TELE: ConnectionId=[0x38D3783F 0x14F111CC 0x801CFDB1 0x2D0CC4A5]
IncomingConnectionId=[0x38D3783F 0x14F111CC 0x801CFDB1 0x2D0CC4A5] TxDuration=502410 ms
VoiceTxDuration=502410 ms FaxTxDuration=0 ms CoderTypeRate=g729r8 NoiseLevel=0 ACOMLevel=1
OutSignalLevel=-41 InSignalLevel=-37 InfoActivity=2 ERLLevel=1 SessionTarget= ImgPages=0
GENERIC: SetupTime=210097 ms Index=1 PeerAddress=2002 PeerSubAddress= PeerId=3 PeerIfIndex=10
LogicalIfIndex=5 ConnectTime=210638 **CallDuration=00:08:10** CallState=4 CallOrigin=2
ChargedUnits=0 InfoType=2 TransmitPackets=24606 TransmitBytes=492120 ReceivePackets=24605
ReceiveBytes=492100 TELE: ConnectionId=[0x6C135AD4 0x14F311CC 0x8024CE4C 0xAA60AB15]
IncomingConnectionId=[0x6C135AD4 0x14F311CC 0x8024CE4C 0xAA60AB15] TxDuration=492110 ms
VoiceTxDuration=492110 ms FaxTxDuration=0 ms CoderTypeRate=g729r8 NoiseLevel=0 ACOMLevel=0
OutSignalLevel=-46 InSignalLevel=-33 InfoActivity=2 ERLLevel=0 SessionTarget= ImgPages=0
GENERIC: SetupTime=210480 ms Index=1 **PeerAddress=1002** PeerSubAddress= PeerId=1 PeerIfIndex=8
LogicalIfIndex=0 ConnectTime=210638 **CallDuration=00:08:11** CallState=4 CallOrigin=1
ChargedUnits=0 InfoType=2 TransmitPackets=24587 TransmitBytes=491740 ReceivePackets=24664
ReceiveBytes=493280 VOIP: ConnectionId[0x6C135AD4 0x14F311CC 0x8024CE4C 0xAA60AB15]
IncomingConnectionId[0x6C135AD4 0x14F311CC 0x8024CE4C 0xAA60AB15] RemoteIPAddress=192.168.1.1
RemoteUDPPort=18884 RemoteSignallingIPAddress=192.168.1.1 RemoteSignallingPort=1720
RemoteMediaIPAddress=192.168.1.1 RemoteMediaPort=18884 **RoundTripDelay=4 ms** SelectedQoS=best-
effort tx_DtmfRelay=h245-alphanumeric FastConnect=TRUE Separate H245 Connection=FALSE H245
Tunneling=TRUE SessionProtocol=cisco SessionTarget=ipv4:192.168.1.1 OnTimeRvPayout=487570
GapFillWithSilence=0 ms GapFillWithPrediction=0 ms GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms HiWaterPayoutDelay=70 ms LoWaterPayoutDelay=69 ms ReceiveDelay=69
ms **LostPackets=0 EarlyPackets=1 LatePackets=0 VAD = disabled CoderTypeRate=g729r8**
CodecBytes=20Total call-legs: 4 Raleigh3640A# Raleigh3640A# Raleigh3640A# **show policy interface**
Serial1/0 Service-policy output: voice-policy Class-map: voice-traffic (match-all) 113186
packets, 7289624 bytes **30 second offered rate 51000 bps, drop rate 0 bps** Match: access-group 102
Queueing Strict Priority Output Queue: Conversation 264 **Bandwidth 51 (kbps) Burst 1275 (Bytes)**
(pkts matched/bytes matched) 471/75864 (total drops/bytes drops) 0/0 Class-map: voice-signaling
(match-all) 162 packets, 13339 bytes 30 second offered rate 0 bps, drop rate 0 bps Match:
access-group 103 Queueing Output Queue: Conversation 265 Bandwidth 16 (kbps) Max Threshold 64
(packets) (pkts matched/bytes matched) 162/13339 (depth/total drops/no-buffer drops) 0/0/0
Class-map: class-default (match-any) 194 packets, 16761 bytes 30 second offered rate 0 bps, drop
rate 0 bps Match: any Queueing Flow Based Fair Queueing Maximum Number of Hashed Queues 256
(total queued/total drops/no-buffer drops) 0/0/0 Raleigh3640A# Raleigh3640A# **show access-lists**
102 Extended IP access list 102 permit udp any any range 16384 32767 (**113963 matches**)
Raleigh3640A# Raleigh3640A# Raleigh3640A# **show access-lists 102** Extended IP access list 102
permit udp any any range 16384 32767 (**114093 matches**) Raleigh3640A# Raleigh3640A# Raleigh3640A#
show access-lists 102 Extended IP access list 102 permit udp any any range 16384 32767 (**114188**
matches) Raleigh3640A# Raleigh3640A# Raleigh3640A# **show access-lists 102** Extended IP access list
102 permit udp any any range 16384 32767 (**114404 matches**) Raleigh3640A# Raleigh3640A#
Raleigh3640A# Raleigh3640A# **show voice call sum** PORT CODEC VAD VTSP STATE VPM STATE =====
===== == =====
3/0/0 g729r8 n S_CONNECT FXSLS_CONNECT
3/0/1 g729r8 n S_CONNECT FXSLS_CONNECT 3/1/0 - - - FXOLS_ONHOOK 3/1/1 - - - FXOLS_ONHOOK
Raleigh3640A# Raleigh3640A# **show voice port sum** IN OUT PORT CH SIG-TYPE ADMIN OPER STATUS STATUS
EC ===== == =====
3/0/0 -- fxs-ls up up off-hook idle y
3/0/1 -- fxs-ls up up off-hook idle y 3/1/0 -- fxo-ls up dorm idle on-hook y 3/1/1 -- fxo-ls up
dorm idle on-hook y Raleigh3640A# Raleigh3640A# Raleigh3640A# **show voice dsp** DSP DSP DSPWARE

```
CURR BOOT PAK TX/RX TYPE NUM CH CODEC VERSION STATE STATE RST AI VOICEPORT TS ABORT PACK COUNT
==== == == ===== ===== ===== ===== == == ===== == ===== C542 001 01
g729r8 3.4.55 busy idle 0 0 3/0/0 NA 0 69615/68771 C542 002 01 g729r8 3.4.55 busy idle 0 0 3/0/1
NA 0 51511/51520 C542 003 01 g711ulaw 3.4.55 IDLE idle 0 0 3/1/0 NA 0 541/546 C542 004 01
g711ulaw 3.4.55 IDLE idle 0 0 3/1/1 NA 0 535/532 Raleigh3640A#
```

Dépannez

Cette section fournit des informations que vous pouvez utiliser pour dépanner votre configuration.

Dépannage des commandes

Certaines commandes **show** sont prises en charge par l'[Output Interpreter Tool](#) ([clients enregistrés](#) uniquement), qui vous permet de voir une analyse de la sortie de la commande show.

Remarque: Avant d'émettre des commandes **debug**, reportez-vous aux [Informations importantes sur les commandes de débogage](#).

- **debug voip ccapi inout** — Trace le chemin d'exécution par l'interface de programmation de Contrôle d'appel (API).
- **debug vpm all** — Enables mettant au point sur toutes les zones virtuelles du module de port vocal (VPM).
- **show log** — La sortie d'expositions d'activer met au point.

Puisque les côtés de Raleigh et de San Jose sont très semblables dans la configuration et l'installation, ce document affiche le **debug voip ccapi inout** et les commandes de **debug vpm all** pour seulement le routeur de San Jose.

Si l'établissement d'appel est un problème, émettez les commandes de **débogage** que cette section répertorie. Comparez la sortie aux informations ici. Vous pouvez utiliser le logiciel, comme le comparez ou comparez au-delà, pour comparer les deux fichiers texte et pour trouver les différences. La sortie ici sert de référence à un appel réussi.

D'abord, déterminez ce qui se produit dans le routeur pendant l'appel. Émettez le **debug voip ccapi inout** et les commandes de **debug vpm all**. La sortie de la question de la commande de **show debug**, comme apparaît ici, affiche l'activation de la commande de **debug vpm all** dans le routeur de San Jose. Vous pouvez déterminer l'activation de la commande de **debug vpm all** parce que la sortie affiche quatre commandes de débogage activées, sans compter que la commande de **debug voip ccapi inout**. Ces quatre commandes ont l'activation automatique quand vous émettez la commande de **debug vpm all**.

Attention : Vous devez désactiver ces commandes de **débogage** après que vous génériez la sortie dont vous avez besoin. Désactivez les commandes de **débogage** avec la question de la commande **undebug all**. Si vous partez pour mettre au point l'activation, vous pouvez rencontrer des problèmes de performances de routeur. Les commandes de debug avec l'activation consomment des ressources CPU.

```
SanJose3640A# show debug voip: voip ccAPI function enter/exit debugging is on Voice Port Module
session debugging is on Voice Port Module DSP message debugging is on Voice Port Module error
debugging is on Voice Port Module signaling debugging is on Voice Port Module voaal2 debugging
is on Voice Port Module trunk conditioning is on SanJose3640A# SanJose3640A# SanJose3640A#
SanJose3640A# SanJose3640A#! Call from 1001 to 2001 SanJose3640A# SanJose3640A# SanJose3640A#
SanJose3640A# *Mar 1 00:05:07.675: htsp_dsp_message: SEND/RESP_SIG_STATUS: state=0xC
timestamp=33146 systime=30767 *Mar 1 00:05:07.679: htsp_process_event: [3/0/0, FXSLS_ONHOOK,
E_DSP_SIG_1100] fxsls_onhook_offhook htsp_setup_ind *Mar 1 00:05:07.679: [3/0/0]
```

get_local_station_id calling num= calling name= calling time=00/00 00:00 *Mar 1 00:05:07.679:
cc_api_call_setup_ind (vdbPtr=0x6217C270, callInfo={called=,called_
oct3=0x81,calling=,calling_oct3=0x0,calling_oct3a=0x0,calling_xlated=false,
subscriber_type_str=RegularLine,fdest=0,peer_tag=2, prog_ind=3,callingIE_present 0},
callID=0x61DAB4F4) *Mar 1 00:05:07.679: cc_api_call_setup_ind calling number is null, answer
addr dest pattern 1001 e164_ans_addr 0 e164_dest_pattern 1 *Mar 1 00:05:07.679:
cc_api_call_setup_ind valid dest pattern, copying 1001 to calling number *Mar 1 00:05:07.679:
cc_api_call_setup_ind type 3 , prot 0 *Mar 1 00:05:07.683: cc_process_call_setup_ind
(event=0x62107860) *Mar 1 00:05:07.683: >>>CCAPI handed cid 5 with tag 2 to app "DEFAULT" *Mar
1 00:05:07.683: sess_appl: ev(24=CC_EV_CALL_SETUP_IND), cid(5), disp(0) *Mar 1 00:05:07.683:
sess_appl: ev(SSA_EV_CALL_SETUP_IND), cid(5), disp(0) *Mar 1 00:05:07.683: ssaCallSetupInd *Mar
1 00:05:07.683: ccCallSetContext (callID=0x5, context=0x620005E8) *Mar 1 00:05:07.683:
ssaCallSetupInd cid(5), st(SSA_CS_MAPPING),oldst(0), ev(24)ev-
>e.evCallSetupInd.nCallInfo.finalDestFlag = 0 *Mar 1 00:05:07.683: ccCallSetupAck (callID=0x5)
*Mar 1 00:05:07.683: ccCallReportDigits (callID=0x5, enable=0x1) *Mar 1 00:05:07.683:
cc_api_call_report_digits_done (vdbPtr=0x6217C270, callID=0x5, disp=0) *Mar 1 00:05:07.683:
sess_appl: ev(53=CC_EV_CALL_REPORT_DIGITS_DONE), cid(5), disp(0) *Mar 1 00:05:07.683:
cid(5)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_REPORT_DIGITS_DONE) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) *Mar 1 00:05:07.683: ssaReportDigitsDone cid(5) peer list: (empty) *Mar
1 00:05:07.683: ssaReportDigitsDone callid=5 Enable succeeded *Mar 1 00:05:07.687:
ccGenerateTone (callID=0x5 tone=8) *Mar 1 00:05:07.687: dsp_digit_collect_on: [3/0/0]
packet_len=20 channel_id=128 packet_id= 35 min_inter_delay=240 max_inter_delay=9760
mim_make_time=10 max_make_time=100 min_brake_time=10 max_brake_time=100 *Mar 1 00:05:07.687:
dsp_soutput: [3/0/0] *Mar 1 00:05:07.687: dsp_digit_collect_on: [3/0/0] packet_len=20
channel_id=128 packet_id= 35 min_inter_delay=240 max_inter_delay=9760 mim_make_time=10
max_make_time=100 min_brake_time=10 max_brake_time=100 *Mar 1 00:05:07.687: dsp_soutput: [3/0/0]
*Mar 1 00:05:07.687: htsp_process_event: [3/0/0, FXSLS_WAIT_SETUP_ACK, E_HTSP_SETUP_ACK] *Mar 1
00:05:09.455: cc_api_call_digit_begin (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF, srcCallId=0x5,
digit=2, digit_begin_flags=0x1, rtp_timestamp=0xEB32A6E0 rtp_expiration=0x0, dest_mask=0x1) *Mar
1 00:05:09.455: sess_appl: ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(5), disp(0) *Mar 1 00:05:09.455:
cid(5)st(SSA_CS_MAPPING)ev(SSA_EV_DIGIT_BEGIN) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) *Mar 1 00:05:09.455: ssaIgnore cid(5), st(SSA_CS_MAPPING),oldst(0),
ev(10) *Mar 1 00:05:09.515: cc_api_call_digit_end (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF,
srcCallId=0x5,digit=2,duration=95,xruleCallingTag=0,xruleCalledTag=0, dest_mask=0x1),
digit_tone_mode=0 *Mar 1 00:05:09.515: sess_appl: ev(9=CC_EV_CALL_DIGIT_END), cid(5), disp(0)
*Mar 1 00:05:09.515: cid(5)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_DIGIT) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) *Mar 1 00:05:09.515: ssaDigit *Mar 1 00:05:09.515: ssaDigit, 0. sct-
>digit , sct->digit len 0, usrDigit 2, digit_tone_mode=0 *Mar 1 00:05:09.515: ssaDigit,1.
callinfo.called , digit 2, callinfo.calling 1001, xrulecallingtag 0, xrulecalledtag 0 *Mar 1
00:05:09.515: ssaDigit, 7. callinfo.calling 1001, sct->digit 2, result 1 *Mar 1 00:05:09.635:
cc_api_call_digit_begin (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF, srcCallId=0x5, digit=0,
digit_begin_flags=0x1, rtp_timestamp=0xEB32A6E0 rtp_expiration=0x0, dest_mask=0x1) *Mar 1
00:05:09.635: sess_appl: ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(5), disp(0) *Mar 1 00:05:09.635:
cid(5)st(SSA_CS_MAPPING)ev(SSA_EV_DIGIT_BEGIN) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) *Mar 1 00:05:09.635: ssaIgnore cid(5), st(SSA_CS_MAPPING),oldst(0),
ev(10) *Mar 1 00:05:09.695: cc_api_call_digit_end (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF,
srcCallId=0x5,digit=0,duration=95,xruleCallingTag=0,xruleCalledTag=0, dest_mask=0x1),
digit_tone_mode=0 *Mar 1 00:05:09.695: sess_appl: ev(9=CC_EV_CALL_DIGIT_END), cid(5), disp(0)
*Mar 1 00:05:09.695: cid(5)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_DIGIT) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) *Mar 1 00:05:09.695: ssaDigit *Mar 1 00:05:09.695: ssaDigit, 0. sct-
>digit 2, sct->digit len 1, usrDigit 0, digit_tone_mode=0 *Mar 1 00:05:09.695: ssaDigit,1.
callinfo.called , digit 20, callinfo.calling 1001, xrulecallingtag 0, xrulecalledtag 0 *Mar 1
00:05:09.695: ssaDigit, 7. callinfo.calling 1001, sct->digit 20, result 1 *Mar 1 00:05:09.815:
cc_api_call_digit_begin (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF, srcCallId=0x5, digit=0,
digit_begin_flags=0x1, rtp_timestamp=0xEB32A6E0 rtp_expiration=0x0, dest_mask=0x1) *Mar 1
00:05:09.815: sess_appl: ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(5), disp(0) *Mar 1 00:05:09.815:
cid(5)st(SSA_CS_MAPPING)ev(SSA_EV_DIGIT_BEGIN) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) *Mar 1 00:05:09.815: ssaIgnore cid(5), st(SSA_CS_MAPPING),oldst(0),
ev(10) *Mar 1 00:05:09.875: cc_api_call_digit_end (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF,
srcCallId=0x5,digit=0,duration=95,xruleCallingTag=0,xruleCalledTag=0, dest_mask=0x1),
digit_tone_mode=0 *Mar 1 00:05:09.875: sess_appl: ev(9=CC_EV_CALL_DIGIT_END), cid(5), disp(0)
*Mar 1 00:05:09.875: cid(5)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_DIGIT) oldst(SSA_CS_MAPPING)cfid(-
1)csz(0)in(1)fDest(0) *Mar 1 00:05:09.875: ssaDigit *Mar 1 00:05:09.875: ssaDigit, 0. sct-
>digit 20, sct->digit len 2, usrDigit 0, digit_tone_mode=0 *Mar 1 00:05:09.875: ssaDigit,1.

callinfo.called , digit 200, callinfo.calling 1001, xrulecallingtag 0, xrulecalledtag 0 *Mar 1 00:05:09.875: ssaDigit, 7. callinfo.calling 1001, sct->digit 200, result 1 *Mar 1 00:05:09.995: cc_api_call_digit_begin (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF, srcCallId=0x5, digit=1, digit_begin_flags=0x1, rtp_timestamp=0xEB32A6E0 rtp_expiration=0x0, dest_mask=0x1) *Mar 1 00:05:09.995: sess_appl: ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(5), disp(0) *Mar 1 00:05:09.995: cid(5)st(SSA_CS_MAPPING)ev(SSA_EV_DIGIT_BEGIN) oldst(SSA_CS_MAPPING)cfid(-1)csz(0)in(1)fDest(0) *Mar 1 00:05:09.995: ssaIgnore cid(5), st(SSA_CS_MAPPING),oldst(0), ev(10) *Mar 1 00:05:10.055: cc_api_call_digit_end (dstVdbPtr=0x0, dstCallId=0xFFFFFFFF, srcCallId=0x5,digit=1,duration=95,xruleCallingTag=0,xruleCalledTag=0, dest_mask=0x1), digit_tone_mode=0 *Mar 1 00:05:10.055: sess_appl: ev(9=CC_EV_CALL_DIGIT_END), cid(5), disp(0) *Mar 1 00:05:10.055: cid(5)st(SSA_CS_MAPPING)ev(SSA_EV_CALL_DIGIT) oldst(SSA_CS_MAPPING)cfid(-1)csz(0)in(1)fDest(0) *Mar 1 00:05:10.055: ssaDigit *Mar 1 00:05:10.055: ssaDigit, 0. sct->digit 200, sct->digit len 3, usrDigit 1, digit_tone_mode=0 *Mar 1 00:05:10.055: ssaDigit,1. callinfo.called , digit 2001, callinfo.calling 1001, xrulecallingtag 0, xrulecalledtag 0 *Mar 1 00:05:10.055: ssaDigit, 7. callinfo.calling 1001, sct->digit 2001, result 0 *Mar 1 00:05:10.055: ccCallReportDigits (callID=0x5, enable=0x0) *Mar 1 00:05:10.055: cc_api_call_report_digits_done (vdbPtr=0x6217C270, callID=0x5, disp=0) *Mar 1 00:05:10.055: ssaSetupPeer cid(5) peer list: tag(1) called number (2001) *Mar 1 00:05:10.055: ssaSetupPeer cid(5), destPat(2001), matched(1), prefix(), peer(622FB888), peer->encapType (2) *Mar 1 00:05:10.055: ccCallProceeding (callID=0x5, prog_ind=0x0) *Mar 1 00:05:10.059: ccCallSetupRequest (Inbound call = 0x5, outbound peer =1, dest=, params=0x621129C8 mode=0, *callID=0x6 2112D38, prog_ind = 3) callingIE_present 0 *Mar 1 00:05:10.059: ccCallSetupRequest numbering_type 0x81 *Mar 1 00:05:10.059: ccCallSetupRequest encapType 2 clid_restrict_disable 1 null_orig_clg 1 clid_transparent 0 callingNumber 1001 *Mar 1 00:05:10.059: dest pattern 2..., called 2001, digit_strip 0 *Mar 1 00:05:10.059: callingNumber=1001, calledNumber=2001, redirectNumber= display_info= calling_oct3a=0 *Mar 1 00:05:10.059: accountNumber=, finalDestFlag=0, guid=3f30.bb8e.14ef.11cc.8008.fdb1.2d0c.c4a5 *Mar 1 00:05:10.059: peer_tag=1 *Mar 1 00:05:10.059: ccIFCallSetupRequestPrivate: (vdbPtr=0x620BCAF0, dest=, callParams={called=2001,called_oct3=0x81, calling=1001,calling_oct3=0x0, calling_xlated=false, subscriber_type_str=RegularLine, fdest=0, voice_peer_tag=1},mode=0x0) vdbP tr type = 1 *Mar 1 00:05:10.059: ccIFCallSetupRequestPrivate: (vdbPtr=0x620BCAF0, dest=, callParams={called=2001, called_oct3 0x81, calling=1001,calling_oct3 0x0, calling_xlated=false, fdest=0, voice_peer_tag=1}, mode=0x0, xltrc=-5) *Mar 1 00:05:10.059: ccSaveDialpeerTag (callID=0x5, dialpeer_tag=0x1) *Mar 1 00:05:10.059: ccCallSetContext (callID=0x6, context=0x61DAD8A0) *Mar 1 00:05:10.059: sess_appl: ev(53=CC_EV_CALL_REPORT_DIGITS_DONE), cid(5), disp(0) *Mar 1 00:05:10.059: cid(5)st(SSA_CS_CALL_SETTING)ev(SSA_EV_CALL_REPORT_DIGITS_DONE) oldst(SSA_CS_MAPPING)cfid(-1)csz(0)in(1)fDest(0) *Mar 1 00:05:10.059: - cid2(6)st2(SSA_CS_CALL_SETTING)oldst2(SSA_CS_MAPPING) *Mar 1 00:05:10.059: ssaReportDigitsDone cid(5) peer list: (empty) *Mar 1 00:05:10.059: ssaReportDigitsDone callid=5 Reporting disabled. *Mar 1 00:05:10.063: dsp_digit_collect_off: [3/0/0] packet_len=8 channel_id=128 packet_id= 36 *Mar 1 00:05:10.063: dsp_soutput: [3/0/0] *Mar 1 00:05:10.063: htsp_process_event: [3/0/0, FXSLS_OFFHOOK, E_HTSP_PROCEEDING] *Mar 1 00:05:10.095: cc_api_call_proceeding(vdbPtr=0x620BCAF0, callID=0x6, prog_ind=0x0) *Mar 1 00:05:10.099: sess_appl: ev(21=CC_EV_CALL_PROCEEDING), cid(6), disp(0) *Mar 1 00:05:10.099: cid(6)st(SSA_CS_CALL_SETTING)ev(SSA_EV_CALL_PROCEEDING) oldst(SSA_CS_MAPPING)cfid(-1)csz(0)in(0)fDest(0) *Mar 1 00:05:10.099: - cid2(5)st2(SSA_CS_CALL_SETTING)oldst2(SSA_CS_CALL_SETTING) *Mar 1 00:05:10.099: ssaCallProc *Mar 1 00:05:10.099: ccGetDialpeerTag (callID=0x5) *Mar 1 00:05:10.099: ssaIgnore cid(6), st(SSA_CS_CALL_SETTING),oldst(1), ev(21) *Mar 1 00:05:10.103: cc_api_call_cut_progress(vdbPtr=0x620BCAF0, callID=0x6, prog_ind=0x8, sig_ind=0x1) *Mar 1 00:05:10.103: sess_appl: ev(22=CC_EV_CALL_PROGRESS), cid(6), disp(0) *Mar 1 00:05:10.107: cid(6)st(SSA_CS_CALL_SETTING)ev(SSA_EV_CALL_PROGRESS) oldst(SSA_CS_CALL_SETTING)cfid(-1)csz(0)in(0)fDest(0) *Mar 1 00:05:10.107: - cid2(5)st2(SSA_CS_CALL_SETTING)oldst2(SSA_CS_CALL_SETTING) *Mar 1 00:05:10.107: ssaCutProgress *Mar 1 00:05:10.107: ccGetDialpeerTag (callID=0x5) *Mar 1 00:05:10.107: ccCallCutProgress (callID=0x5, prog_ind=0x8, sig_ind=0x1) *Mar 1 00:05:10.107: **ccConferenceCreate** (confID=0x6211310C, callID1=0x5, callID2=0x6, tag=0x0) *Mar 1 00:05:10.107: cc_api_bridge_done (confID=0x3, srcIF=0x620BCAF0, srcCallID=0x6, dstCallID=0x5, disposition=0, tag=0x0)htsp_alert_notify *Mar 1 00:05:10.107: cc_api_bridge_done (confID=0x3, srcIF=0x6217C270, srcCallID=0x5, dstCallID=0x6, disposition=0, tag=0x0) *Mar 1 00:05:10.107: cc_api_caps_ind (dstVdbPtr=0x620BCAF0, dstCallId=0x6, srcCallId=0x5, caps={codec=0x2EBFB, fax_rate=0x7F, vad=0x3, modem=0x2 codec_bytes=0, signal_type=3}) *Mar 1 00:05:10.107: cc_api_caps_ind (Playout: mode 1, initial 60,min 40, max 200) *Mar 1 00:05:10.111: cc_api_caps_ind (dstVdbPtr=0x6217C270, dstCallId=0x5, srcCallId=0x6, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2}) *Mar 1 00:05:10.111: cc_api_caps_ind (Playout: mode 1, initial 60,min 40, max 200) *Mar 1 00:05:10.111: cc_api_caps_ack (dstVdbPtr=0x6217C270, dstCallId=0x5, srcCallId=0x6,

caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2, seq_num_start=9062}) *Mar 1 00:05:10.111: cc_api_caps_ack (dstVdbPtr=0x620BCAF0, dstCallId=0x6, srcCallId=0x5, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2, seq_num_start=9062}) *Mar 1 00:05:10.111: cc_api_voice_mode_event , callID=0x5 *Mar 1 00:05:10.111: Call Pointer =620005E8 *Mar 1 00:05:10.115: cc_api_caps_ind (dstVdbPtr=0x6217C270, dstCallId=0x5, srcCallId=0x6, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2}) *Mar 1 00:05:10.115: cc_api_caps_ind (Playout: mode 1, initial 60,min 40, max 200) *Mar 1 00:05:10.115: cc_api_caps_ack (dstVdbPtr=0x6217C270, dstCallId=0x5, srcCallId=0x6, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2, seq_num_start=9062}) *Mar 1 00:05:10.123: cc_api_caps_ack (dstVdbPtr=0x620BCAF0, dstCallId=0x6, srcCallId=0x5, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2, seq_num_start=9062}) *Mar 1 00:05:10.123: cc_api_voice_mode_event , callID=0x5 *Mar 1 00:05:10.123: Call Pointer =620005E8 *Mar 1 00:05:10.123: htsp_process_event: [3/0/0, FXSLS_OFFHOOK, E_HTSP_VOICE_CUT_THROUGH] *Mar 1 00:05:10.123: htsp_process_event: [3/0/0, FXSLS_OFFHOOK, E_HTSP_VOICE_CUT_THROUGH] *Mar 1 00:05:10.123: sess_appl: ev(29=CC_EV_CONF_CREATE_DONE), cid(5), disp(0) *Mar 1 00:05:10.123: cid(5)st(SSA_CS_CONFERENCING_PROGRESS)ev(SSA_EV_CONF_CREATE_DONE) oldst(SSA_CS_CALL_SETTING)cfid(3)csz(0)in(1)fDest(0) *Mar 1 00:05:10.127: - cid2(6)st2(SSA_CS_CONFERENCING_PROGRESS)oldst2(SSA_CS_CALL_SETTING) *Mar 1 00:05:10.127: ssaConfCreateDoneAlert *Mar 1 00:05:10.127: sess_appl: ev(51=CC_EV_VOICE_MODE_DONE), cid(5), disp(0) *Mar 1 00:05:10.127: cid(5)st(SSA_CS_CONFERENCED_ALERT)ev(SSA_EV_VOICE_MODE_DONE) oldst(SSA_CS_CONFERENCING_PROGRESS)cfid(3)csz(0)in(1)fDest(0) *Mar 1 00:05:10.127: - cid2(6)st2(SSA_CS_CONFERENCED_ALERT)oldst2(SSA_CS_CALL_SETTING) *Mar 1 00:05:10.127: ssaIgnore cid(5), st(SSA_CS_CONFERENCED_ALERT),oldst(4), ev(51) *Mar 1 00:05:10.127: sess_appl: ev(51=CC_EV_VOICE_MODE_DONE), cid(5), disp(2) *Mar 1 00:05:10.127: cid(5)st(SSA_CS_CONFERENCED_ALERT)ev(SSA_EV_VOICE_MODE_DONE) oldst(SSA_CS_CONFERENCED_ALERT)cfid(3)csz(0)in(1)fDest(0) *Mar 1 00:05:10.127: - cid2(6)st2(SSA_CS_CONFERENCED_ALERT)oldst2(SSA_CS_CALL_SETTING) *Mar 1 00:05:10.127: ssaIgnore cid(5), st(SSA_CS_CONFERENCED_ALERT),oldst(4), ev(51) *Mar 1 00:05:10.127: cc_process_notify_bridge_done (event=0x6210BDB8) *Mar 1 00:05:10.131: cc_api_caps_ind (dstVdbPtr=0x6217C270, dstCallId=0x5, srcCallId=0x6, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2}) *Mar 1 00:05:10.131: cc_api_caps_ind (Playout: mode 1, initial 60,min 40, max 200) *Mar 1 00:05:10.131: cc_api_caps_ack (dstVdbPtr=0x6217C270, dstCallId=0x5, srcCallId=0x6, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2, seq_num_start=9063}) *Mar 1 00:05:10.131: cc_api_caps_ind (dstVdbPtr=0x6217C270, dstCallId=0x5, srcCallId=0x6, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2}) *Mar 1 00:05:10.131: cc_api_caps_ind (Playout: mode 1, initial 60,min 40, max 200) *Mar 1 00:05:10.131: cc_api_caps_ack (dstVdbPtr=0x6217C270, dstCallId=0x5, srcCallId=0x6, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2, seq_num_start=9063}) *Mar 1 00:05:10.135: cc_api_caps_ack (dstVdbPtr=0x620BCAF0, dstCallId=0x6, srcCallId=0x5, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2, seq_num_start=9063}) *Mar 1 00:05:10.135: cc_api_voice_mode_event , callID=0x5 *Mar 1 00:05:10.135: Call Pointer =620005E8 *Mar 1 00:05:10.135: cc_api_caps_ack (dstVdbPtr=0x620BCAF0, dstCallId=0x6, srcCallId=0x5, caps={codec=0x4, fax_rate=0x2, vad=0x1, modem=0x0 codec_bytes=20, signal_type=2, seq_num_start=9063}) *Mar 1 00:05:10.135: cc_api_voice_mode_event , callID=0x5 *Mar 1 00:05:10.135: Call Pointer =620005E8 *Mar 1 00:05:10.135: htsp_process_event: [3/0/0, FXSLS_OFFHOOK, E_HTSP_VOICE_CUT_THROUGH] *Mar 1 00:05:10.135: htsp_process_event: [3/0/0, FXSLS_OFFHOOK, E_HTSP_VOICE_CUT_THROUGH] *Mar 1 00:05:10.135: sess_appl: ev(51=CC_EV_VOICE_MODE_DONE), cid(5), disp(0) *Mar 1 00:05:10.135: cid(5)st(SSA_CS_CONFERENCED_ALERT)ev(SSA_EV_VOICE_MODE_DONE) oldst(SSA_CS_CONFERENCED_ALERT)cfid(3)csz(0)in(1)fDest(0) *Mar 1 00:05:10.135: - cid2(6)st2(SSA_CS_CONFERENCED_ALERT)oldst2(SSA_CS_CALL_SETTING) *Mar 1 00:05:10.135: ssaIgnore cid(5), st(SSA_CS_CONFERENCED_ALERT),oldst(4), ev(51) *Mar 1 00:05:10.135: sess_appl: ev(51=CC_EV_VOICE_MODE_DONE), cid(5), disp(0) *Mar 1 00:05:10.135: cid(5)st(SSA_CS_CONFERENCED_ALERT)ev(SSA_EV_VOICE_MODE_DONE) oldst(SSA_CS_CONFERENCED_ALERT)cfid(3)csz(0)in(1)fDest(0) *Mar 1 00:05:10.139: - cid2(6)st2(SSA_CS_CONFERENCED_ALERT)oldst2(SSA_CS_CALL_SETTING) *Mar 1 00:05:10.139: ssaIgnore cid(5), st(SSA_CS_CONFERENCED_ALERT),oldst(4), ev(51) *Mar 1 00:05:18.303: cc_api_call_connected(vdbPtr=0x620BCAF0, callID=0x6), prog_ind = 2cc_api_call_connected: setting callEntry->connected to TRUE *Mar 1 00:05:18.303: sess_appl: ev(8=CC_EV_CALL_CONNECTED), cid(6), disp(0) *Mar 1 00:05:18.303: cid(6)st(SSA_CS_CONFERENCED_ALERT)ev(SSA_EV_CALL_CONNECTED) oldst(SSA_CS_CALL_SETTING)cfid(3)csz(0)in(0)fDest(0) *Mar 1 00:05:18.307: - cid2(5)st2(SSA_CS_CONFERENCED_ALERT)oldst2(SSA_CS_CONFERENCED_ALERT) *Mar 1 00:05:18.307: ssaConnectAlert *Mar 1 00:05:18.307: ccGetDialpeerTag (callID=0x5) *Mar 1 00:05:18.307:

```

ccCallConnect (callID=0x5), prog_ind = 2ccCallConnect: setting callEntry->connected to TRUE *Mar
1 00:05:18.307: ssaFlushPeerTagQueue cid(5) peer list: (empty)htsp_connect: no_offhook 0 *Mar 1
00:05:18.307: htsp_process_event: [3/0/0, FXSLS_OFFHOOK, E_HTSP_CONNECT]fxspls_offhook_connect
*Mar 1 00:05:18.307: [3/0/0] set signal state = 0x6 timestamp = 0 *Mar 1 00:05:18.307:
dsp_set_sig_state: [3/0/0] packet_len=12 channel_id=128 packet_id=39 state=0x6 timestamp=0x0
*Mar 1 00:05:18.307: dsp_soutput: [3/0/0] SanJose3640A# SanJose3640A# SanJose3640A#
SanJose3640A#! call connected SanJose3640A# SanJose3640A# SanJose3640A# SanJose3640A#
SanJose3640A#! 1001 disconnecting the call SanJose3640A# SanJose3640A# SanJose3640A#
SanJose3640A# SanJose3640A# *Mar 1 00:05:57.019: htsp_dsp_message: SEND/RESP_SIG_STATUS:
state=0x4 timestamp=16952 systime=35702 *Mar 1 00:05:57.019: htsp_process_event: [3/0/0,
FXSLS_CONNECT, E_DSP_SIG_0100]fxspls_offhook_onhook, HF duration=500 *Mar 1 00:05:57.023:
htsp_timer - 500 msec *Mar 1 00:05:57.523: htsp_process_event: [3/0/0, FXSLS_CONNECT,
E_HTSP_EVENT_TIMER]fxspls_connect_wait_release_req *Mar 1 00:05:57.523: htsp_timer_stop *Mar 1
00:05:57.523: cc_api_call_disconnected(vdbPtr=0x6217C270, callID=0x5, cause=0x10) *Mar 1
00:05:57.523: sess_appl: ev(11=CC_EV_CALL_DISCONNECTED), cid(5), disp(0) *Mar 1 00:05:57.523:
cid(5)st(SSA_CS_ACTIVE)ev(SSA_EV_CALL_DISCONNECTED)
oldst(SSA_CS_CONFERENCED_ALERT)cfid(3)csz(0)in(1)fDest(0) *Mar 1 00:05:57.523: -
cid2(6)st2(SSA_CS_ACTIVE)oldst2(SSA_CS_CONFERENCED_ALERT) *Mar 1 00:05:57.523: ssa: Disconnected
cid(5) state(5) cause(0x10) *Mar 1 00:05:57.523: ccConferenceDestroy (confID=0x3, tag=0x0) *Mar
1 00:05:57.523: cc_api_bridge_drop_done (confID=0x3, srcIF=0x620BCAF0, srcCallID=0x6,
dstCallID=0x5, disposition=0 tag=0x0) *Mar 1 00:05:57.523: cc_api_bridge_drop_done (confID=0x3,
srcIF=0x6217C270, srcCallID=0x5, dstCallID=0x6, disposition=0 tag=0x0) *Mar 1 00:05:57.523:
sess_appl: ev(30=CC_EV_CONF_DESTROY_DONE), cid(5), disp(0) *Mar 1 00:05:57.523:
cid(5)st(SSA_CS_CONF_DESTROYING)ev(SSA_EV_CONF_DESTROY_DONE) oldst(SSA_CS_ACTIVE)cfid(-
1)csz(0)in(1)fDest(0) *Mar 1 00:05:57.527: -
cid2(6)st2(SSA_CS_CONF_DESTROYING)oldst2(SSA_CS_CONFERENCED_ALERT) *Mar 1 00:05:57.527:
ssaConfDestroyDone *Mar 1 00:05:57.527: ccCallDisconnect (callID=0x5, cause=0x10 tag=0x0) *Mar 1
00:05:57.527: ccCallDisconnect: existing_cause = 0x0, new_cause = 0x10 *Mar 1 00:05:57.527:
ccCallDisconnect (callID=0x6, cause=0x10 tag=0x0) *Mar 1 00:05:57.527: ccCallDisconnect:
existing_cause = 0x0, new_cause = 0x10htsp_release_req: cause 16, no_onhook 0 *Mar 1
00:05:57.531: htsp_process_event: [3/0/0, FXSLS_WAIT_RELEASE_REQ, E_HTSP_RELEASE_REQ]
fxspls_waitrls_req_rls *Mar 1 00:05:57.531: [3/0/0] set signal state = 0x4 timestamp = 0 *Mar 1
00:05:57.531: dsp_set_sig_state: [3/0/0] packet_len=12 channel_id=128 packet_id=39 state=0x4
timestamp=0x0 *Mar 1 00:05:57.531: dsp_soutput: [3/0/0]htsp_report_onhook_sig *Mar 1
00:05:57.531: cc_api_call_feature: (vdbPtr=0x6217C270, callID=0x5, feature_ind.type=5 *Mar 1
00:05:57.535: cc_api_call_disconnect_done(vdbPtr=0x6217C270, callID=0x5, disp=0, tag=0x0) *Mar 1
00:05:57.535: hdsprm_close_cleanup *Mar 1 00:05:57.535: sess_appl: ev(28=CC_EV_CALL_FEATURE),
cid(5), disp(0) *Mar 1 00:05:57.535: cid(5)st(SSA_CS_DISCONNECTING)ev(SSA_EV_CALL_FEATURE)
oldst(SSA_CS_CONF_DESTROYING)cfid(-1)csz(0)in(1)fDest(0) *Mar 1 00:05:57.535: -
cid2(6)st2(SSA_CS_DISCONNECTING)oldst2(SSA_CS_CONFERENCED_ALERT) *Mar 1 00:05:57.535: ssaIgnore
cid(5), st(SSA_CS_DISCONNECTING),oldst(7), ev(28) *Mar 1 00:05:57.539: sess_appl:
ev(12=CC_EV_CALL_DISCONNECT_DONE), cid(5), disp(0) *Mar 1 00:05:57.539:
cid(5)st(SSA_CS_DISCONNECTING)ev(SSA_EV_CALL_DISCONNECT_DONE) oldst(SSA_CS_DISCONNECTING)cfid(-
1)csz(0)in(1)fDest(0) *Mar 1 00:05:57.539: -
cid2(6)st2(SSA_CS_DISCONNECTING)oldst2(SSA_CS_CONFERENCED_ALERT) *Mar 1 00:05:57.539:
ssaDisconnectDone *Mar 1 00:05:57.543: cc_api_icpif: expect factor = 0 *Mar 1 00:05:57.543:
g113_calculate_impairment (delay=101,loss=0), Io=0 Iq=0 Idte=0 Idd=0 Ie=9 Itot=9 *Mar 1
00:05:57.543: cc_api_call_disconnect_done(vdbPtr=0x620BCAF0, callID=0x6, disp=0, tag=0x0) *Mar 1
00:05:57.547: sess_appl: ev(12=CC_EV_CALL_DISCONNECT_DONE), cid(6), disp(0) *Mar 1 00:05:57.547:
cid(6)st(SSA_CS_DISCONNECTING)ev(SSA_EV_CALL_DISCONNECT_DONE)
oldst(SSA_CS_CONFERENCED_ALERT)cfid(-1)csz(1)in(0)fDest(0) *Mar 1 00:05:57.547:
ssaDisconnectDone SanJose3640A# SanJose3640A#

```

[Informations connexes](#)

- [VoIP sur liaisons PPP avec qualité de service \(LLQ / IP RTP Priority, LFI, cRTP\)](#)
- [VoIP sur relais de trame avec qualité de service \(fragmentation, formatage du trafic, LLQ / IP RTP Priority\)](#)
- [QoS \(Qualité de service\) VoIP pour interopérabilité Frame Relay et ATM avec LLQ, PPP LFI et cRTP](#)
- [Compréhension des homologues de numérotation et des signaux d'appel sur des plates-](#)

formes Cisco IOS

- Dépannage et débogage des appels VoIP – Notions élémentaires
- Assistance technique concernant la technologie vocale
- Support produit pour Voix et Communications IP
- Dépannage des problèmes de téléphonie IP Cisco
- Soutien technique et documentation