

# VoIP com o PPP sobre a linha alugada da largura de banda elevada e o LLQ

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Informações de Apoio](#)

[Configurar](#)

[Diagrama de Rede](#)

[Configurações](#)

[Verificar](#)

[Verificação do roteador San Jose](#)

[Verificação do roteador Raleigh](#)

[Troubleshooting](#)

[Comandos de solução de problemas](#)

[Informações Relacionadas](#)

## [Introdução](#)

Este documento fornece configurações de amostra para dois Cisco 3640 Router. As configurações permitem o Roteadores de comunicar-se com VoIP com PPP sobre uma linha alugada da largura de banda elevada com o Low Latency Queuing (LLQ). Para obter mais informações sobre do LLQ, refira o documento [VoIP sobre links de PPP com Qualidade de Serviço \(prioridade RTP LLQ/IP, LFI, cRTP\)](#).

**Note:** Quando este documento discute a largura de banda elevada em termos de VoIP e de QoS, a largura de banda elevada é qualquer largura de banda acima de 768 kbps.

## [Pré-requisitos](#)

### [Requisitos](#)

Não existem requisitos específicos para este documento.

### [Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software e hardware:

- IP Plus do Software Release 12.2(19a) de Cisco IOS® ou algum outro Cisco IOS Software Release de 12.2, 12.2T, 12.3, ou 12.3T
- Dois Cisco 3640 Router com pelo menos a 48 DRAM e 16 Mb da memória Flash
- Dois módulos de rede de slot da placa de interface de voz/fax de Cisco NM-2V mais duas placas de interface do VIC-2FXS
- Duas interfaces serial Neste exemplo, as duas interfaces serial são NM-1E2W, com um WAN Interface Card cada um WIC-1T.
- Os telefones analógicos para o acessório à estação de câmbio internacional (FXO) movem para chamadas de voz

**Note:** Os módulos de rede NM-1E2W, NM-1E1R2W, e NM-2E2W não têm bastante potência do desempenho apoiar o WIC-2T. A falta do apoio é devido às limitações do hardware.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

## [Convenções](#)

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

## [Informações de Apoio](#)

Se o tempo necessário enviar um pacote 1500-byte para fora no fio é maior do que a Senhora 10, você precisa pacotes de fragmento. Este documento apresenta uma configuração sem fragmentação. A configuração é para um link 1544-kilobit para que o retardo de transmissão para um pacote 1500-byte é menos do que a Senhora 10.

**Note:** Em alguns casos em quais você tem um dedicado, a conexão T1 completa, uns recursos de fragmentação pode ser desnecessária. Mas, você ainda precisa um mecanismo de QoS. Use o LLQ neste caso. Se a quantidade de tempo necessária enviar um pacote 1500-byte para fora no fio é menos do que a Senhora 10, você não precisa pacotes de fragmento. O T1 completo oferece largura de banda suficiente para permitir que pacotes de voz entrem e saiam da fila sem problemas de retardo.

**Note:** Se você permitiu a fragmentação no roteador, há habilitação do mecanismo de filas 100 por cento do tempo. Se você configurou o LLQ, o valor você configurou limites o tráfego para a fila de prioridade. Quando você não permitiu a fragmentação, o roteador aplica somente a política de QoS no caso da congestão.

Também, no caso da linha taxas que é maior de 768 kbps, o protocolo compressed real-time transport (cRTP) pode ser desnecessário. Refira o documento [VoIP sobre links de PPP com o \[LLQ/IP RTP Priority, LFI, cRTP\] de Qualidade de Serviço](#). O uso do cRTP ajuda a salvar a largura de banda porque o cRTP comprime cabeçalhos de IP RTP. Na seção de [configurações](#) deste documento, a habilitação do cRTP é desnecessária. O T1 permite que a largura de banda suficiente para que os pacotes de voz fluam, sem compressão, no fio sem edição.

**Caution:** Se você decide usar o cRTP, esteja ciente que o cRTP usa recursos do CPU. O cRTP pode overtax um roteador que tenha uma carga pesada do tráfego de voz.

**Note:** Nesta configuração, os dois Roteadores conectam lado a lado sobre uma linha alugada. Mas, na maioria de topologias, o Roteadores com habilitação da Voz pode existir em qualquer lugar. Geralmente, o Roteadores da Voz conecta com a conectividade de LAN ao outro Roteadores que conecta a WAN. Se seu Roteadores da Voz não conecta através do PPP sobre uma linha alugada, você precisa de configurar todos os comandos configuration da conectividade de WAN naquele Roteadores que conecta a WAN; você não configura os comandos no Roteadores da Voz, que as [configurações](#) neste documento mostram.

**Note:** Esta configuração pode trabalhar para o Cisco 1700, os [2600, os 3600, e os 3700 Series Router](#).

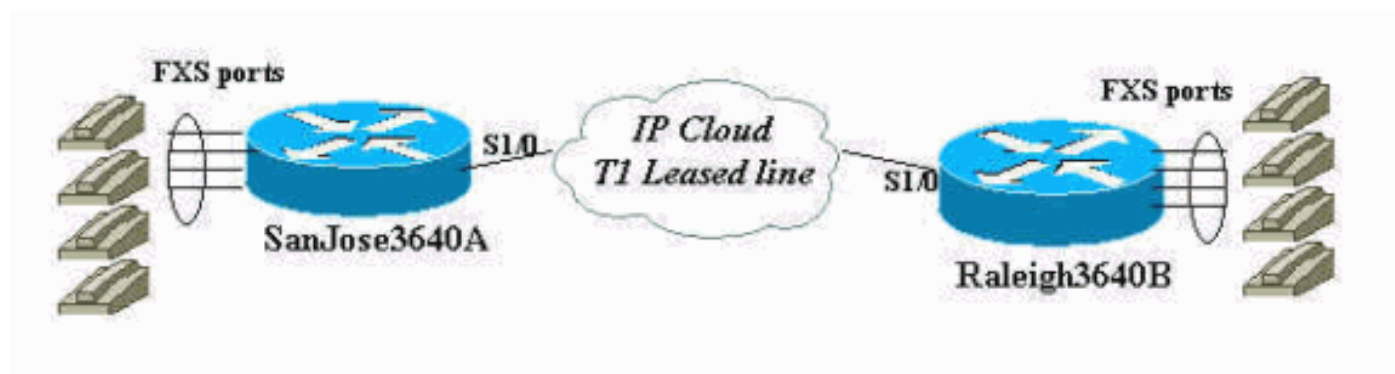
## [Configurar](#)

Nesta seção, você encontrará informações para configurar os recursos descritos neste documento.

**Note:** Para localizar informações adicionais sobre os comandos usados neste documento, utilize a Ferramenta Command Lookup (somente clientes [registrados](#)).

## [Diagrama de Rede](#)

Este documento utiliza a seguinte configuração de rede:



## [Configurações](#)

Este documento utiliza as seguintes configurações:

- [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
kpbs 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#
```

## Verificar

Depois que você incorpora estas [configurações em](#) seu Roteadores, verifique que trabalham corretamente. Os comandos e a saída respectiva aqui demonstram uma implementação bem sucedida das configurações.

A [Output Interpreter Tool \(somente clientes registrados\)](#) oferece suporte a determinados comandos show, o que permite exibir uma análise da saída do comando show.

- **mostre a série 1/0 da relação** — Permite que você verifique o estado de sua interface serial.
- **show call active voice brief** — Permite que você ver a informação de chamada durante um atendimento.
- **mostre a voz ativa do atendimento** — Permite que você ver a informação de chamada durante um atendimento.



- **mostre a relação do mapa de política** — Permite que você verifique a política de QoS que a relação usa.
- **mostre a lista de acesso 102** — Permite que você verifique a seleção de pacote pela lista de acessos para a classe da Voz. Emita o comando um a segunda vez depois que alguns segundos e verifique que há um aumento no contagem de pacote de informação. Emita o **comando clear access-list counters 102**, caso necessário.
- **mostre o sumário da chamada de voz** — Permite que você verifique o estado dos atendimentos. O comando mostra-lhe se os atendimentos têm a conexão.
- **mostre o sumário da porta de voz** — Permite que você verifique o estado das portas de voz. O comando mostra as portas de voz como o em-gancho ou o fora-gancho.
- **DSP de voz da mostra** — Permite que você verifique o status do processador de sinal digital (DSP) e o codificador-decodificador (codec) que cada atendimento usa.

## Verificação do roteador San Jose

Antes que você execute a verificação, verifique as relações para assegurar-se de que você tenha a Conectividade necessária colocar atendimentos. Emita o **comando show interface serial 1/0** verificar o estado de sua interface serial. Com as [configurações n](#) neste documento, seja certo que suas série e interfaces multilink estão em um protocolo de linha acima do estado. Igualmente seja certo que você vê este:

- **LCP aberto, multilink aberto** — Indica o estabelecimento da conexão PPP.
- **Abra: IPCP, CDPCP** — Diz-lhe que a emissão do tráfego IP é possível através do link de PPP.
- **Estratégia de enfileiramento: feira tornada mais pesada** — Corresponde ao comando `line interface(cli)` da saída da serviço-política sob a interface serial. A estratégia é para a configuração do LLQ para dar a prioridade à Voz sobre dados.

```
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#

Esta saída mostra a Conectividade bem sucedida entre o Roteadores. Se você não vê que o protocolo de linha está acima, verifique o Clock Rate que está na relação DCE. Algumas interfaces serial não apoiam a alta velocidade, tal como o NM-8A/S. Também, verifique que os parâmetros em ambos os lados combinam e, o mais importante, que o encapsulamento combina.

A saída do comando **show call active voice brief** aqui mostra duas chamadas bem sucedidas. Um atendimento é do roteador Raleigh ao roteador san jose, e o outro é de SÃO JOSÉ a Raleigh. Esta lista explica a saída que aparece no negrito:

- **Active da resposta 1001** — Significa que SÃO JOSÉ é o roteador de que o atendimento origina.
- **3/0/0 Tele** — Significa que este é o trecho da chamada telefônica.
- **Origine o active 2001** — Significa que um telefone no lado de Raleigh recebe o atendimento.
- **IP 192.168.1.2** — Significa que este é o trecho de chamada IP.
- **Active da resposta 2002** — Significa que Raleigh é o roteador a que o atendimento envia.
- **IP 192.168.1.2** — Significa que este é o trecho de chamada IP.
- **Origine o active 1002** — Significa que um telefone no lado de SÃO JOSÉ recebe o atendimento.
- **3/0/1 Tele** — Significa que este é o trecho da chamada telefônica.

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/0:-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/0:-40/-40 dBm

Total call-legs: 4

SanJose3640A#

Esta saída do comando **show call active voice** fornece mais detalhe sobre a chamada ativa:

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  
OnTimeRvPlayout=300810  
GapFillWithSilence=0 ms  
GapFillWithPrediction=0 ms  
GapFillWithInterpolation=0 ms  
GapFillWithRedundancy=0 ms  
HiWaterPlayoutDelay=70 ms  
LoWaterPlayoutDelay=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=

OnTimeRvPlayout=295580

GapFillWithSilence=0 ms

GapFillWithPrediction=0 ms

GapFillWithInterpolation=0 ms

GapFillWithRedundancy=0 ms

HiWaterPlayoutDelay=70 ms

LoWaterPlayoutDelay=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:

A saída do comando **show policy-map interface** inclui esta instrução em negrito:

- **30 segundos taxa** oferecida 51000 bps — Mostra a largura de banda que os dois atendimentos exigem, 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#

```

A saída do comando **show access-lists 102** inclui esta instrução em negrito:

- **100676 fósforos** — Mostra que a prioridade dos pacotes RTP ocorre porque os pacotes alcançam a lista de acessos 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#
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#
SanJose3640A# show access-lists 102

```

```

Extended IP access list 102
permit udp any any range 16384 32767 (101198 matches)
SanJose3640A#
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#

```

## [Verificação do roteador Raleigh](#)

O procedimento de verificação para o roteador Raleigh é similar ao procedimento para o roteador 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=  
OnTimeRvPlayout=497610  
GapFillWithSilence=0 ms  
GapFillWithPrediction=0 ms  
GapFillWithInterpolation=0 ms  
GapFillWithRedundancy=0 ms  
HiWaterPlayoutDelay=70 ms  
LoWaterPlayoutDelay=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#

## [Troubleshooting](#)

Esta seção fornece informações que podem ser usadas para o troubleshooting da sua configuração.

### [Comandos de solução de problemas](#)

A [Output Interpreter Tool \(somente clientes registrados\)](#) oferece suporte a determinados comandos show, o que permite exibir uma análise da saída do comando show.

**Note:** [Antes de emitir comandos de depuração, consulte Informações Importantes sobre Comandos de Depuração.](#)

- **debugar o inout do ccapi do voip** — Segue o trajeto da execução através do interface de programação de aplicativo de controle de chamadas (API).
- **debugar o vpm todo** — Permite a eliminação de erros em todas as áreas virtuais do módulo

da porta de voz (VPM).

- **log da mostra** — A saída das mostras do permitido debuga.

Desde que os lados de Raleigh e de SÃO JOSÉ são muito similares na configuração e setup, este documento mostra os **comandos debug voip ccapi inout e debug vpm all** para somente o roteador san jose.

Se o estabelecimento de chamada é um problema, emita os **comandos debug** lista dessa esta seção. Compare a saída com a informação aqui. Você pode usar o software, como compara-o ou compara-o além, para comparar os dois arquivos de texto e para encontrar as diferenças. A saída aqui serve como uma referência para uma chamada bem sucedida.

Primeiramente, determine o que ocorre no roteador durante o atendimento. Emita o **inout do ccapi do voip debugar** e os **comandos debug vpm all**. A saída da introdução do **comando show debug**, como aparece aqui, mostra a habilitação do **comando debug vpm all** no roteador san jose. Você pode determinar a habilitação do **comando debug vpm all** porque a saída mostra quatro comandos debug permitidos, além do **comando debug voip ccapi inout**. Estes quatro comandos têm a habilitação automática quando você emite o **comando debug vpm all**.

**Caution:** Você deve desabilitar estes **comandos debug** depois que você gerencie a saída que você precisa. Desabilite os **comandos debug** com a introdução do **comando undebug all**. Se você licença debuga a habilitação, você pode experimentar problemas de desempenho de roteador. Os comandos Debug com habilitação consomem recursos do 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)csize(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)csize(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)csize(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)csize(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)csize(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) vdbPtr 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)csize(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)csize(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)csize(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)csize(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)csize(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)csize(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)csize(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)csize(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)csize(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]fxsls\_

```
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#
*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)csize(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)csize(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)csize(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#

## [Informações Relacionadas](#)

- [Links de VoIP por PPP com qualidade de serviço \(LLQ / prioridade IP RTP, LFI, cRTP\)](#)
- [VoIP por Frame Relay com qualidade de serviço \(fragmentação, modelagem de tráfego, prioridade LLQ/IP RTP\)](#)
- [VoIP QoS para Frame Relay para Entrelaçamento de ATM com LLQ, PPP LFI e cRTP](#)
- [Entendendo os paridade de discagem e segmentos de chamada em plataformas Cisco IOS](#)
- [Conceitos Básicos de Troubleshooting e Depuração de Chamadas VoIP](#)
- [Suporte à Tecnologia de Voz](#)
- [Suporte de Produtos de Comunicação de Voz e de IP](#)
- [Troubleshooting da Telefonia IP Cisco](#)
- [Suporte técnico & documentação](#)