

Configurando o tronco de conexão para gateways VoIP

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[Configurar](#)

[Considerações e limitações do design do tronco de conexão](#)

[Diretrizes de configuração](#)

[Diagrama de Rede](#)

[Configurações](#)

[Verificar](#)

[Troubleshoot](#)

[Comandos para Troubleshooting](#)

[Informações Relacionadas](#)

Introduction

Um tronco (tie-line) é uma linha de comunicação ponto a ponto permanente entre duas portas de voz. O comando de tronco de conexão cria uma chamada permanente de Voz sobre IP (VoIP) entre dois gateways VoIP. Simula uma conexão de tronco através da criação de troncos virtuais (tie-lines) entre dois pontos finais de telefonia. Para os sistemas conectados, é como se um tronco T1 estivesse conectado diretamente entre eles.

Prerequisites

Requirements

Essas plataformas suportam um tronco de conexão VoIP:

- Interfaces digitais e analógicas das séries Cisco 2600, 3600 e 3700
- Interfaces digitais Cisco 7200/7500 Series
- Interfaces digitais e analógicas Cisco MC3810
- Cisco 1750/1751 e 1760

Observação: as plataformas AS5300/AS5400/AS5800 não suportam e não suportarão troncos de conexão, pois não são adequadas para conectividade WAN com volumes de tráfego intensos.

Componentes Utilizados

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

- Cisco IOS® Software Versão 12.2(10a) com Conjunto de Recursos de IP Plus
- Cisco 2610 Series Routers

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

Configurar

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

Observação: para encontrar informações adicionais sobre os comandos usados neste documento, use a [Command Lookup Tool](#) (somente clientes registrados)

Considerações e limitações do design do tronco de conexão

- O modo de tronco de conexão é suportado nas interfaces CAS (sinalização associada a canal) T1/E1. Um tronco de conexão não é suportado nas interfaces T1/E1 que estão usando a sinalização de canal comum (CCS); por exemplo, QSIG e PRI Q.931. Um tronco de conexão não é suportado nas portas FXO (Foreign Exchange Office) configuradas para início terra.
- O modo de tronco de conexão é uma conexão permanente; A chamada VoIP é sempre conectada independentemente da porta Plain Old Telephone Service (POTS) estar ou não no gancho. O Tronco de Conexão tem endpoints configurados estaticamente e não exige que um usuário disque para conectar chamadas. Também permite que a sinalização de chamada suplementar, como hookflash ou hoot-n-holler ponto a ponto, seja passada pela rede IP entre os dois dispositivos de telefonia.
- O modo de tronco de conexão é suportado com estas combinações de portas de voz:recEive e transMit (E & M) para E & M (mesmo tipo)FXO para Foreign Exchange Station (FXS)FXS para FXS (sem sinalização)**Observação:** essas combinações de portas de voz são permitidas entre interfaces analógicas e analógicas, digitais e digitais. Além disso, quando você está configurando FXS para FXS, a sinalização não pode ser transmitida porque não seria um caminho transparente. Os dispositivos conectados (FXOs) estariam tentando sinalizar um para o outro. É possível fazer com que este design funcione se você definir o caminho de voz para estar sempre aberto. Configure o `signal ext-signal` do tipo de `signal` para o peer de discagem VoIP, e o roteador não esperará mais pela sinalização antes de abrir o caminho de voz.
- Um mapeamento T1 CAS a E1 CAS do tronco de conexão não funciona por padrão. A manipulação de ordem de bits nos gateways deve ser executada e nem sempre pode funcionar, com base no suporte de PBX de várias sinalização de bit ABCD.

- Um tronco de conexão permite o tipo de funcionalidade de linha privada, ringdown-Off-Premise-Extension (PLAR-OPX) automática entre portas FXO e FXS. Isso permite que estações remotas (conectadas às portas FXS) apareçam no PBX como estações fisicamente conectadas. Se essa estação remota não atender uma chamada, ela poderá ser revertida para o correio de voz centralizado (se estiver configurada no PBX).
- Um tronco de conexão, como o PLAR, não exige que o roteador colete dígitos do dispositivo de telefonia. A chamada VoIP permanente é criada quando o roteador é inicializado e a conectividade IP é estabelecida. Por causa disso, o plano de discagem do cliente existente não precisa ser alterado.
- Um tronco de conexão pode transferir alguma sinalização telefônicas, como hookflash, mas não transmite sinalização PBX proprietária. Não é um recurso Transparent CCS (T-CCS).
- Um tronco de conexão, como o PLAR, é definido por porta de voz. Isso significa que a porta de voz não pode operar no modo Tronco de Conexão e no modo Coletar Dígitos Discados. A única instância em que isso pode não ser completamente desejável seria em um escritório remoto que também precisa discar entre ramais locais sem o uso de um PBX centralizado. Isso exigiria que o caminho da chamada passasse pela rede VoIP e voltasse, ao contrário de ela ser comutada dentro do roteador. Em geral, isso não é um problema.

Diretrizes de configuração

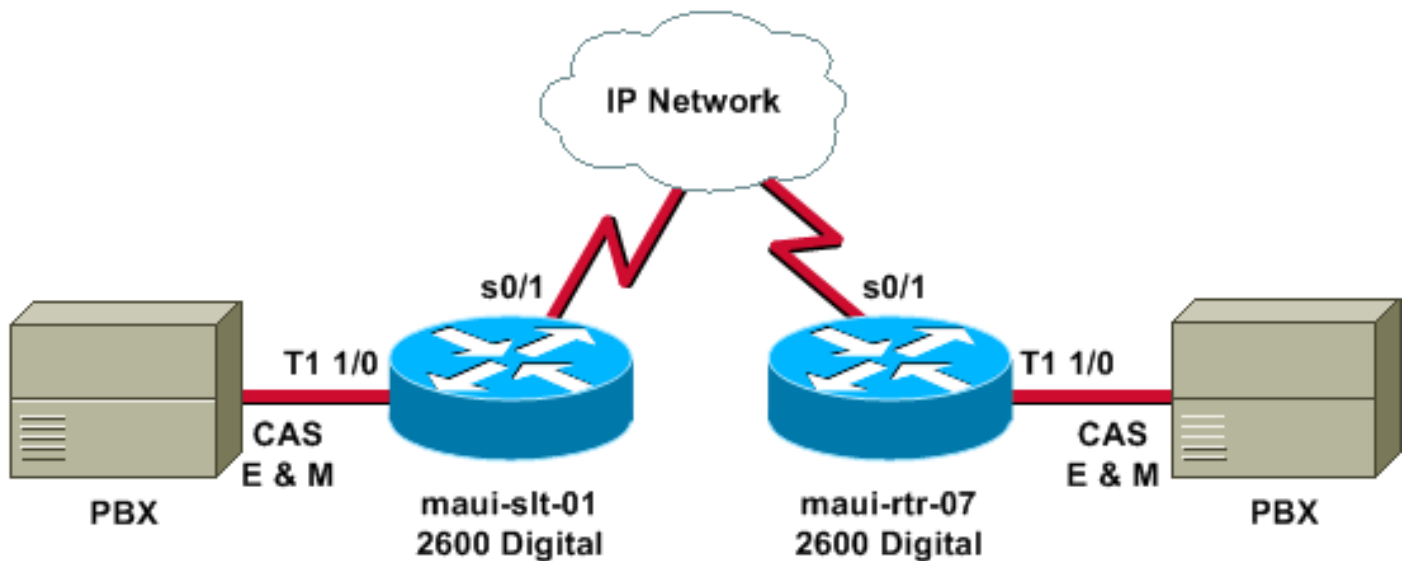
É necessário configurar o tronco de conexão em suas duas extremidades. Ao configurar um tronco de conexão com interfaces analógicas, ele deve ser definido por porta de voz. Ao configurar um tronco de conexão com interfaces digitais, há várias opções:

- Você pode definir um comando [ds0-group](#) separado para cada DS0 (cada timeslot) e pode usar o comando [connection trunk](#) para definir cada porta de voz criada. Isso garante que o mapeamento DS0 a DS0 seja mantido em troncos digitais.
- Você pode definir um único comando [ds0-group](#) para manipular todos os DS0s e pode definir um único comando [de tronco de conexão na porta de voz](#). Isso reduz a quantidade de configuração manual necessária, mas não há garantia de mapeamento um para um de DS0s em qualquer extremidade do tronco. Além disso, cada vez que o roteador é recarregado, o mapeamento pode ser diferente da última vez. Além disso, essa configuração complica a solução de problemas, pois você não pode isolar o problema em um único (ou mesmo em alguns) timeslots sem derrubar todo o grupo de troncos. Essa configuração também não é recomendada para T-CCS com sinalização proprietária em qualquer extremidade dos PBXs, porque não forneceria o canal de sinalização de forma confiável sem mapeamento um a um.
- Recomenda-se que um lado da conexão seja configurado com a palavra-chave **answer-mode** especificada após o comando **connection trunk string**. Isso torna um lado do tronco o "lado mestre". O gateway (roteador) com a palavra-chave **do modo de resposta** é então o "lado escravo". O comando **answer-mode** especifica que o gateway não tentará iniciar uma conexão de tronco, mas, em vez disso, esperará uma chamada recebida antes de estabelecer o tronco. Esse esquema de configuração minimiza o tempo que os roteadores levam para ativar troncos e para garantir que os troncos fiquem inativos quando as conexões são perdidas entre dois gateways. Caso contrário, os gateways podem não tentar restabelecer o tronco quando a conexão estiver ativa novamente.

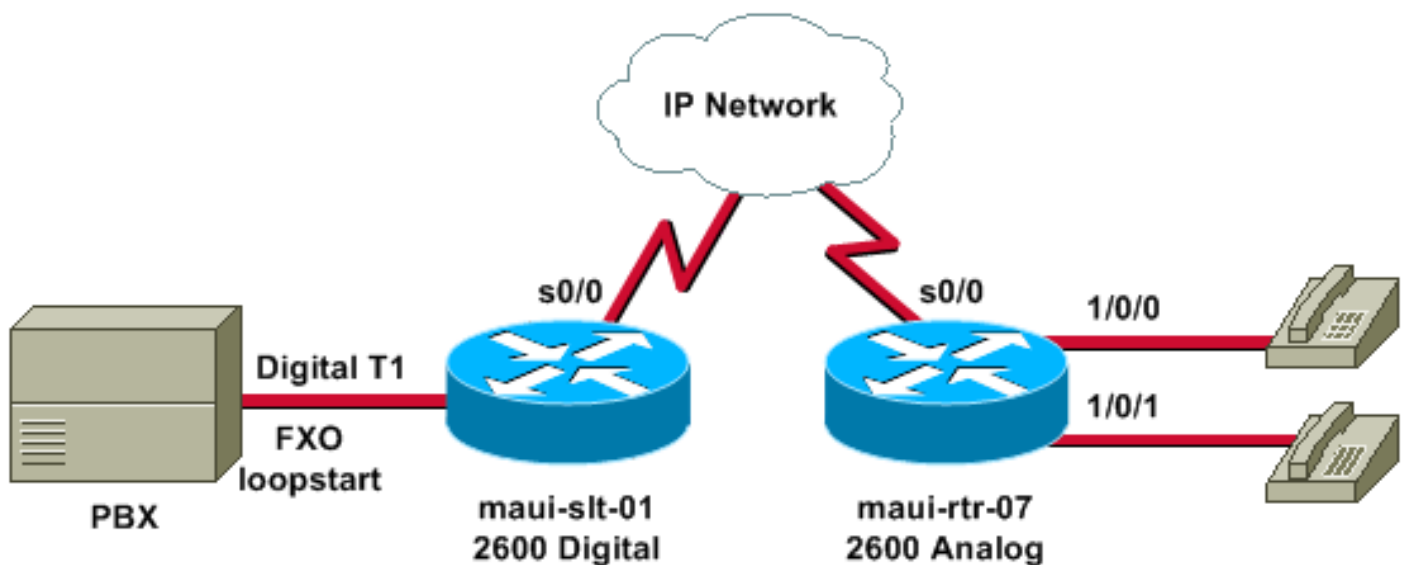
Observação: ao emitir o comando **connection trunk**, você deve executar uma sequência de comandos **shutdown/no shutdown** na porta de voz.

Diagrama de Rede

Este documento usa estas duas configurações de rede:



O diagrama anterior ilustra o cenário digital-digital, em que ambos os lados do roteador têm links digitais.



O diagrama anterior ilustra o cenário digital-analógico, com digital em uma extremidade e analógico na outra.

Configurações

Este documento utiliza as seguintes configurações:

- Digital para digital [maui-slt-01maui-rtr-07](#)
- Digital para analógico [maui-slt-01maui-rtr-07](#)

A primeira configuração (digital-para-digital) mostra uma configuração típica para um tronco de conexão entre dois roteadores com interfaces digitais T1. Neste exemplo, os roteadores estão fornecendo uma verdadeira substituição de linha de ligação entre os PBXs.

Digital para digital - maui-slt-01

```

version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname maui-slt-01
!
voice-card 1
!
controller T1 1/0
framing esf
linecode b8zs
ds0-group 1 timeslots 1 type e & m-wink-start
ds0-group 2 timeslots 2 type e & m-wink-start
clock source line
!--- The ds0-group command creates the logical voice-
ports: !--- voice-port 1/0:1 and voice-port 1/0:2.

!
voice-port 1/0:1
connection trunk 2000
!--- "master side" !--- This starts the trunk connection
using digits 2000 to match !--- a VoIP dial-peer. The
digits are generated internally by the !--- router and
are not received from the voice-port. ! voice-port
1/0:2
connection trunk 2001
!
dial-peer voice 2 voip
destination-pattern 200.
!--- Matches connection trunk string 2000 and 2001.

dtmf-relay h245-alphanumeric
session target ipv4:192.168.100.2
ip qos dscp cs5 media
!
dial-peer voice 1 pots
destination-pattern 1000
port 1/0:1
!--- This dial-peer maps to maui-rtr-07's voice-port
1/0:1. ! dial-peer voice 3 pots destination-pattern 1001
port 1/0:2 !--- This dial-peer maps to maui-rtr-07's
voice-port 1/0:2. ! interface Serial0/1 ip address
192.168.100.1 255.255.255.0

```

Digital para digital - maui-rtr-07

```

version 12.2
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname maui-rtr-07
!
voice-card 1
!
controller T1 1/0
framing esf
linecode b8zs
ds0-group 1 timeslots 1 type e & m-wink-start
ds0-group 2 timeslots 2 type e & m-wink-start
clock source line

```

```

!
voice-port 1/0:1
  connection trunk 1000 answer-mode
!--- "slave side" !--- The answer-mode specifies that
the router should not attempt !--- to initiate a trunk
connection, but it should wait for an !--- incoming call
before it establishes the trunk.

!
voice-port 1/0:2
  connection trunk 1001 answer-mode
!
dial-peer voice 1 voip
  destination-pattern 100.
  dtmf-relay h245-alphanumeric
  session target ipv4:192.168.100.1
  ip qos dscp cs5 media
!
dial-peer voice 2 pots
  destination-pattern 2000
  port 1/0:1
!--- This dial-peer terminates the connection !--- from
maui-slt-01 voice-port 1/0:1. !
dial-peer voice 3 pots
  destination-pattern 2001
  port 1/0:2
!--- This dial-peer terminates the connection !--- from
maui-slt-01 voice-port 1/0:2. ! interface Serial10/1 ip
address 192.168.100.2 255.255.255.0 clockrate 128000 !

```

A segunda configuração (digital para analógico) mostra uma configuração típica para um tronco de conexão entre dois roteadores semelhantes, um com interfaces T1 digitais e outro com interfaces analógicas. As interfaces devem ser do mesmo tipo para que isso funcione (por exemplo, piscar E & M para piscar E & M, E & M imediatos para E & M imediatos, FXO para FXS e vice versa). Em nosso exemplo, o loopstart FXO está sinalizando na interface T1 digital e há portas FXS analógicas com sinalização de loopstart FXS no lado correspondente.

Digital para analógico - maui-slt-01

```

version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname maui-slt-01
!
voice vad-time 40000

!
voice-card 1

!
controller T1 1/0
  framing esf
  linecode b8zs
  ds0-group 1 timeslots 1 type fxo-loopstart
  clock source line
!--- The ds0-group command creates the logical voice-
ports: !--- voice-port 1/0:1 and voice-port 1/0:2.

!

```

```

voice-port 1/0:1
  connection trunk 2000
!--- "master side" !--- This starts the trunk connection
using digits 2000 to match !--- a VoIP dial-peer. The
digits are generated internally by the !--- router and
are not received from the voice-port. !!! dial-peer
voice 2 voip
  destination-pattern 200.
!--- Matches connection trunk string 2000 and 2001.

  dtmf-relay h245-alphanumeric
  session target ipv4:192.168.100.2
  ip qos dscp cs5 media
!
dial-peer voice 1 pots
  destination-pattern 1000
  port 1/0:1
!--- This dial-peer maps to maui-rtr-07's voice-port
1/0/0. !!! interface Serial0/1 ip address
192.168.100.1 255.255.255.0 !

```

Digital para analógico - maui-rtr-07

```

version 12.2
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname maui-rtr-07
!
!
voice-port 1/0/0
  connection trunk 1000 answer-mode
!--- "slave side" !--- The answer-mode specifies that
the router should not attempt !--- to initiate a trunk
connection, but it should wait for an !--- incoming call
before it establishes the trunk.

!
!
dial-peer voice 1 voip
  destination-pattern 100.
  dtmf-relay h245-alphanumeric
  session target ipv4:192.168.100.1
  ip qos dscp cs5 media
!
dial-peer voice 2 pots
  destination-pattern 2000
  port 1/0/0
!--- This dial-peer terminates the connection !--- from
maui-slt-01 voice-port 1/0:1. !!! interface Serial0/1
ip address 192.168.100.2 255.255.255.0 clockrate 128000
!

```

Verificar

Esta seção fornece informações que você pode usar para confirmar se sua configuração está funcionando corretamente.

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

- [show voice call summary](#) — Usado para verificar se todos os troncos estão ativos e no estado S_CONNECT.

Quando os troncos se tornarem ativos, o console exibirá a mensagem %HTSP-5-UPDOWN: A porta de tronco (canal) [1/0:1(1)] está habilitada.

Este é um exemplo de saída do comando **show voice call summary**:

```

PORT          CODEC      VAD VTSP STATE          VPM STATE
=====
3/0:0.1      g729r8    n  S_CONNECT          S_TRUNKED
3/0:1.2      g729r8    n  S_CONNECT          S_TRUNKED
3/0:2.3      g729r8    n  S_CONNECT          S_TRUNKED

```

Um tronco que não está ativo aparecerá como S_TRUNK_PEND:

```

PORT          CODEC      VAD VTSP STATE          VPM STATE
=====
3/0:0.1      -          -   -                S_TRUNK_PEND
3/0:1.2      g729r8    n  S_CONNECT          S_TRUNKED
3/0:2.3      g729r8    n  S_CONNECT          S_TRUNKED

```

[Troubleshoot](#)

Esta seção fornece informações que você pode utilizar para fazer troubleshooting de configuração.

[Comandos para Troubleshooting](#)

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

Observação: antes de emitir comandos debug, consulte [Informações importantes sobre comandos debug](#).

- **show call history voice | Incluir DisconnectText** — Mostra o motivo da desconexão para as últimas chamadas com falha.
- **show voice call summary** — Mostra a chamada ativa em ambos os segmentos de chamada.
- **show voice dsp**—Mostra que os DSPs (Digital Signal Processors - Processadores de Sinal Digital) estão em uso e estão processando pacotes.

Para obter mais informações sobre como solucionar problemas de chamadas VoIP, consulte [Solucionando Problemas e Depurando Conceitos Básicos de Chamadas VoIP](#) e [Comandos de Depuração VoIP](#).

As portas de voz associadas em ambos os roteadores devem ser **desligadas/sem desligamento** após você configurar o tronco de conexão. Isso também limpa as portas de voz se você vir o usuário ocupado como uma causa de desconexão.

Este é um exemplo de saída do comando **show voice dsp**:

```

BOOT          PAK
TYPE DSP CH CODEC  VERS STATE STATE  RST AI PORT    TS ABORT  TX/RX-PAK-CNT

```



```

=====
C549 000 01 g729r8 3.4 busy idle 0 0 3/0:12 13 0 3522765/3578769
      00 g729r8 .41 busy idle 0 0 3/0:0 1 0 3505023/3560759
C549 001 01 g729r8 3.4 busy idle 0 0 3/0:13 14 0 3522761/3578601
      00 g729r8 .41 busy idle 0 0 3/0:1 2 0 3522794/3578579

```

A próxima saída de exemplo é a saída de depuração mais comum para o comando **debug voip ccapi inout**. Essa depuração foi realizada sob o erro comum de um peer POTS ausente no lado chamado. No exemplo, o roteador do lado analógico não tem um peer POTS para terminar o tronco; o lado da chamada digital terá estas depurações nesta situação:

```
maui-slt-01#
```

```

*Mar 1 00:11:19.903: cc_api_call_setup_ind (vdbPtr=0x620B2DE8,
callInfo={called=2000,called_oct3=0x81,calling=,calling_oct3=0x0,
calling_oct3a=0x0,calling_xlated=false,subscriber_type_str=RegularLine
,fdest=1,peer_tag=2, prog_ind=3},callID=0x621C45F0)
*Mar 1 00:11:19.903: cc_api_call_setup_ind type 3 , prot 0
*Mar 1 00:11:19.903: cc_process_call_setup_ind (event=0x62332908)
*Mar 1 00:11:19.903: >>>>CCAPI handed cid 3 with tag 2 to app "DEFAULT"
*Mar 1 00:11:19.907: sess_appl: ev(24=CC_EV_CALL_SETUP_IND), cid(3), disp(0)
*Mar 1 00:11:19.907: sess_appl: ev(SSA_EV_CALL_SETUP_IND), cid(3), disp(0)
*Mar 1 00:11:19.907: ssaCallSetupInd
*Mar 1 00:11:19.907: ccCallSetContext (callID=0x3, context=0x621C4E90)
*Mar 1 00:11:19.907: ssaCallSetupInd cid(3), st(SSA_CS_MAPPING),oldst(0),
ev(24)ev->e.evCallSetupInd.nCallInfo.finalDestFlag = 1
*Mar 1 00:11:19.907: ssaCallSetupInd finalDest cllng(1000), cllcd(2000)
*Mar 1 00:11:19.907: ssaCallSetupInd cid(3), st(SSA_CS_CALL_SETTING),
oldst(0), ev(24)dpMatchPeersMoreArg result= 0
*Mar 1 00:11:19.907: ssaSetupPeer cid(3) peer list:
tag(1) called number (2000)
*Mar 1 00:11:19.907: ssaSetupPeer cid(3), destPat(2000), matched(1),
prefix(), peer(61EE565C), peer->encapType (2)
*Mar 1 00:11:19.907: ccCallProceeding (callID=0x3, prog_ind=0x0)
*Mar 1 00:11:19.907: ccCallSetupRequest (Inbound call = 0x3, outbound
peer =1, dest=, params=0x6233BD30 mode=0, *callID=0x6233C098, prog_ind = 3)
*Mar 1 00:11:19.907: ccCallSetupRequest numbering_type 0x81
*Mar 1 00:11:19.907: ccCallSetupRequest encapType 2 clid_restrict_disable 1
null_orig_clg 1 clid_transparent 0 callingNumber 1000
*Mar 1 00:11:19.907: dest pattern 2..., called 2000, digit_strip 0
*Mar 1 00:11:19.907: callingNumber=1000, calledNumber=2000, redirectNumber=
display_info= calling_oct3a=0
*Mar 1 00:11:19.907: accountNumber=, finalDestFlag=1,
guid=1d0d.9a0f.14f0.11cc.8008.b3df.433e.6402
*Mar 1 00:11:19.911: peer_tag=1
*Mar 1 00:11:19.911: ccIFCallSetupRequestPrivate: (vdbPtr=0x621D74DC, dest=,
callParams={called=2000,called_oct3=0x81, calling=1000,calling_oct3=0x0,
calling_xlated=false,subscriber_type_str=RegularLine, fdest=1,
voice_peer_tag=1}, mode=0x0) vdbPtr type = 1
*Mar 1 00:11:19.911: ccIFCallSetupRequestPrivate: (vdbPtr=0x621D74DC, dest=,
callParams={called=2000, called_oct3 0x81, calling=1000,calling_oct3 0x0,
calling_xlated=false, fdest=1, voice_peer_tag=1}, mode=0x0, xltrc=-5)
*Mar 1 00:11:19.911: ccSaveDialpeerTag (callID=0x3, dialpeer_tag=0x1)
*Mar 1 00:11:19.911: ccCallSetContext (callID=0x4, context=0x624C3094)
*Mar 1 00:11:19.911: ccCallReportDigits (callID=0x3, enable=0x0)
*Mar 1 00:11:19.911: cc_api_call_report_digits_done (vdbPtr=0x620B2DE8,
callID=0x3, disp=0)
*Mar 1 00:11:19.911: sess_appl: ev(52=CC_EV_CALL_REPORT_DIGITS_DONE),
cid(3), disp(0)
*Mar 1 00:11:19.911: cid(3)st(SSA_CS_CALL_SETTING)ev
(SSA_EV_CALL_REPORT_DIGITS_DONE)oldst(SSA_CS_MAPPING)
cfid(-1)csize(0)in(1)fDest(1)

```

```
*Mar 1 00:11:19.911: -cid2(4)st2(SSA_CS_CALL_SETTING)oldst2(SSA_CS_MAPPING)
*Mar 1 00:11:19.911: ssaReportDigitsDone cid(3) peer list: (empty)
*Mar 1 00:11:19.911: ssaReportDigitsDone callid=3 Reporting disabled.
*Mar 1 00:11:19.947: cc_api_call_disconnected(vdbPtr=0x621D74DC,
callID=0x4, cause=0x1)
*Mar 1 00:11:19.947: sess_appl: ev(11=CC_EV_CALL_DISCONNECTED), cid(4), disp(0)
*Mar 1 00:11:19.947: cid(4)st(SSA_CS_CALL_SETTING)ev(SSA_EV_CALL_DISCONNECTED)
oldst(SSA_CS_MAPPING)cfid(-1)csize(0)in(0)fDest(0)
*Mar 1 00:11:19.947: -cid2(3)st2(SSA_CS_CALL_SETTING)oldst2(SSA_CS_CALL_SETTING)
*Mar 1 00:11:19.951: ssaDiscSetting
*Mar 1 00:11:19.951: ssa: Disconnected cid(4) state(1) cause(0x1)
*Mar 1 00:11:19.951: ccCallDisconnect (callID=0x4, cause=0x1 tag=0x0)
*Mar 1 00:11:19.951: ccCallDisconnect (callID=0x3, cause=0x1 tag=0x0)
*Mar 1 00:11:19.951: cc_api_call_disconnect_done(vdbPtr=0x620B2DE8, callID=0x3,
disp=0, tag=0x0)
*Mar 1 00:11:19.955: sess_appl: ev(12=CC_EV_CALL_DISCONNECT_DONE), cid(3),
disp(0)
*Mar 1 00:11:19.955: cid(3)st(SSA_CS_DISCONNECTING)ev
(SSA_EV_CALL_DISCONNECT_DONE)oldst(SSA_CS_CALL_SETTING)
cfid(-1)csize(0)in(1)fDest(1)
*Mar 1 00:11:19.955: -cid2(4)st2(SSA_CS_DISCONNECTING)oldst2(SSA_CS_CALL_SETTING)
*Mar 1 00:11:19.955: ssaDisconnectDone
*Mar 1 00:11:19.963: cc_api_icpif: expect factor = 0
*Mar 1 00:11:19.963: cc_api_call_disconnect_done(vdbPtr=0x621D74DC,
callID=0x4, disp=0, tag=0x0)
*Mar 1 00:11:19.967: sess_appl: ev(12=CC_EV_CALL_DISCONNECT_DONE),
cid(4), disp(0)
*Mar 1 00:11:19.967: cid(4)st(SSA_CS_DISCONNECTING)ev
(SSA_EV_CALL_DISCONNECT_DONE)oldst(SSA_CS_CALL_SETTING)
cfid(-1)csize(1)in(0)fDest(0)
*Mar 1 00:11:19.967: ssaDisconnectDone
```

[Informações Relacionadas](#)

- [Configurando a conexão PLAR para gateways de VoIP](#)
- [Conceitos Básicos de Troubleshooting e Depuração de Chamadas VoIP](#)
- [Comandos de debug VoIP](#)
- [Suporte à Tecnologia de Voz](#)
- [Suporte aos produtos de Voz e Comunicação por IP](#)
- [Troubleshooting da Telefonia IP Cisco](#)
- [Suporte Técnico - Cisco Systems](#)