

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Informações de Apoio](#)

[Comando bandwidth \(gatekeeper\)](#)

[Configurar](#)

[Diagrama de Rede](#)

[Configurações](#)

[Verificar](#)

[Troubleshooting](#)

[Comandos para Troubleshooting](#)

[Exemplo de show e debug](#)

[Informações Relacionadas](#)

[Introdução](#)

Este documento fornece uma configuração de exemplo para o controle básico de admissão de chamada de gatekeeper.

[Pré-requisitos](#)

[Requisitos](#)

Há diversas circunstâncias a ser encontradas antes que o gateway possa obter o address resolution correto do porteiro. Há diversos pontos importantes a ser verificados para cada solução voip quando os enlaces de velocidade baixa são envolvidos.

Antes de tentar esta configuração, verifique se estes requisitos são atendidos:

- Todos os gateways devem ser registrados em gatekeepers correspondentes
- Todos os porteiros devem ter o dial plan correto assim que podem decidir na rota para os atendimentos.
- O controle de admissão pode ser configurado para restringir o número de chamadas entre algumas zonas.

Porque os primeiros dois pontos são considerados na seção [configurar](#), nós centrar-nos-emos sobre o controle de admissão na seção de [informações de fundo](#).

[Componentes Utilizados](#)

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

- Três Cisco 2600 Router.
- EMPRESA PLUS/H323 MCM de 12.2.8.5 da liberação do Cisco IOS ® Software.

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 você estiver trabalhando em uma rede ativa, certifique-se de que entende o impacto potencial de qualquer comando antes de utilizá-lo.

Convenções

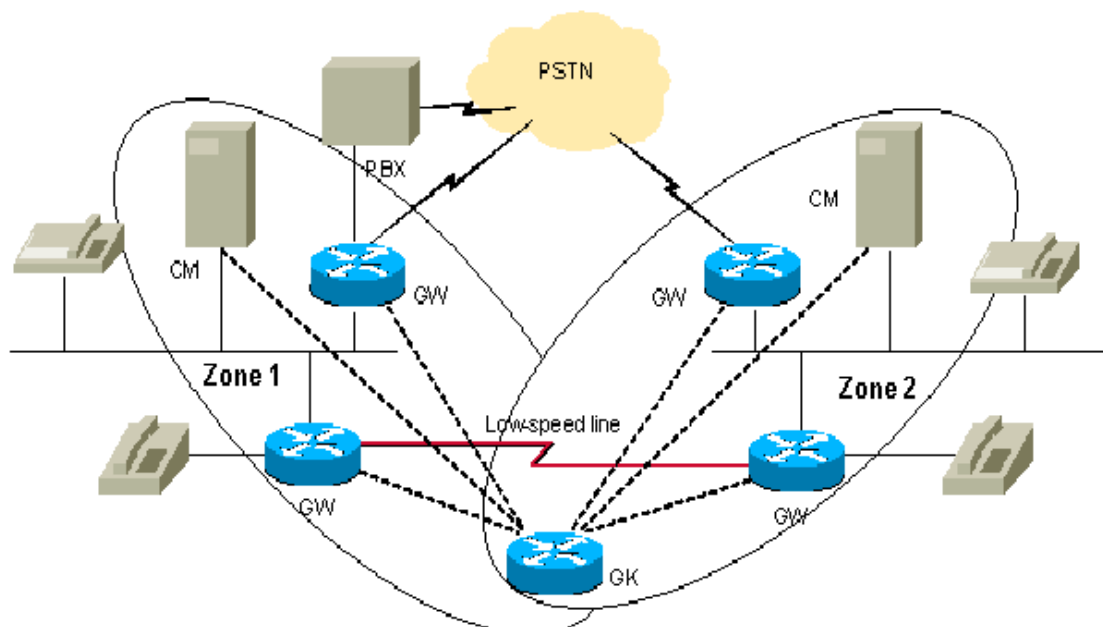
Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

Informações de Apoio

Esta configuração de exemplo estuda uma rede voip com uma topologia da dois-zona, que seja controlada por um porteiro com os três gateways em ambas as zonas. O alvo deste documento é fornecer um exemplo simples de uma configuração de controle de admissão que aplique uma política ao número de atendimentos entre zonas e dentro delas. Este documento inclui a informações de fundo técnica nos recursos configurados, as diretrizes do projeto, e a verificação básica e as estratégias de Troubleshooting.

Nota: Nesta configuração, os quatro Roteadores são ficados situados no mesmo LAN. No entanto, em sua real topologia, todos os dispositivos podem estar em partes diferentes de sua rede.

Muito frequentemente, há diversas fontes de tráfego de alta prioridade nas redes reais. É umas tarefas complexas distinguir todas estas circunstâncias porque são numerosas, e fácil negligenciar. Contudo, há diversas situações comum que acontecem muito frequentemente na vida real isso valem a pena considerar. O controle de admissão transforma-se uma edição quando o Roteadores que fornece a prioridade do tráfego é ele mesmo não as fontes de tal tráfego. A topologia típica envolve diversos gateways de voz em duas estações conectadas pelo enlace fornecido por um par de roteadores. Uma outra topologia envolve CallManagers de Cisco com os Telefones IP em dois locais, junto com os gateways ao PSTN ou ao PBX. Em ambas as situações temos várias origens de tráfego de voz de ambos os lados do enlace.



Às vezes, poderia haver um problema com a Qualidade de voz, se a quantidade de tráfego de voz excede a largura de banda configurada para a fila de prioridade. Isto é porque os Roteadores e os Telefones IP de Cisco CallManager/que originam o tráfego não têm um gerenciamento centralizado para a admissão de chamada no projeto dado acima. Neste caso, os pacotes que excedem a largura de banda serão deixados cair.

Existem várias maneiras de se evitar esse cenário. A solução a mais simples é configurar a largura de banda de voz na fila de latência baixa (LLQ) para aceitar o número máximo de atendimentos de todas as fontes. Na ausência de tráfego de voz, a largura de banda não usada será concedida aos fluxos de dados. Isto pode ser feito quando a largura de banda total do link é mais alta do que a largura de banda exigida para o número máximo de atendimentos.

Mais abordagem sensível é aplicar limitações em cada fonte de tráfego de voz do ambas as extremidades do link. Quando você faz assim, a largura de banda de sumário de todo não excederá a 75% recomendado da largura de banda real do link entre locais. Para aplicar aquelas limitações, use o **comando max-conn** sob a configuração do VoIP dial-peer. Se nós supomos que há um CallManager da Cisco somente em uma instalação central, nós podemos usar suas capacidades de restringir o número de atendimentos à instalação de filial sem um CallManager. Essa abordagem permite gerenciar uma situação em que as origens do tráfego de voz são capazes de subscrever em excesso o enlace. O inconveniente desta aproximação é o uso inflexível de largura de banda concedido às fontes. Esta aproximação não permite que alguns dos gateways coloquem um atendimento extra mesmo se há uma largura de banda livre disponível nesse momento.

A abordagem mais flexível é utilizar uma entidade separada para o controle de admissão de chamada centralizada, o porteiro. As ajudas do porteiro para ligar dois locais com os dois CallManagers de Cisco (ou os Cluster do CallManager).

Nota: O uso do porteiro não significa sempre comprar um roteador separado novo. Baseado no número de atendimentos, e na carga do Roteadores, você pode configurar um porteiro em um dos roteadores existentes com o conjunto de características do IOS Cisco apropriado como Enterprise/PLUS/H323. Isto pode ajudar a controlar ramos pequenos, e permite um porteiro dedicado na instalação central somente.

A aproximação do porteiro deve ser considerada com cuidado, de modo a para não oprimir o roteador com uma carga adicional. Além, você deve verificar se a topologia reserve situar o porteiro desta maneira para evitar o tráfego adicional no link crítico.

A recomendação geral é utilizar Cisco routers separados como gatekeepers dedicados em sua rede em um número apropriado para a sua topologia.

Considere a topologia acima. Aqui, você pode pôr todos os dispositivos nas duas zonas local controladas por um gatekeeper único. Isto permite que você tenha um alto número de atendimentos em cada zona, mas restringe o número de atendimentos entre eles. Em nosso exemplo do teste, nós restringiremos a largura de banda entre as duas zonas a um atendimento, e permitimos até dois (número mais alto de) atendimentos em um deles.

Para informações mais detalhadas sobre disto, veja o [controle de admissão da chamada voip](#).

Para terminar a tarefa, use o **comando bandwidth (gatekeeper)** descrito no [porteiro de capacidade elevada de Cisco](#)

Comando bandwidth (gatekeeper)

Para especificar o máximo de largura de banda agregada para o tráfego H.323, use o comando

de configuração do gatekeeper de largura de banda. Para desabilitar a característica, não use **nenhum** formulário deste comando.

Nota: Este comando permite que você restrinja a largura de banda através de um link único da zona. Se a topologia permite que você coloque um atendimento através de diversos trajetos de uma zona a outra, os links poderiam facilmente tornar-se oversubscribed. Considere esta topologia: duas zonas são conectadas através de dois trajetos, permitindo somente um atendimento através de cada trajeto. Se a largura de banda é restringida por um atendimento, o segundo trajeto estará usado nunca. Mas se a largura de banda é restringida por dois atendimentos, um dos links pode ser oversubscribed. Assim este comando pode ser aplicado nas zonas que tenham apenas um caminho para todas as outras zonas. A topologia do “hub and spoke” é uma exceção. Embora o hub tenha caminhos múltiplos, não oversubscribe os links, porque o número de atendimentos será restrito no spokes para cada link.

largura de banda {interzona | total | sessão} {padrão | largura de banda-tamanho do nome de zona da zona}

nenhuma largura de banda {interzona | total | sessão} {padrão | largura de banda-tamanho do nome de zona da zona}

Descrição da sintaxe

A tabela a seguir descreve a sintaxe:

Sintaxe	Descrição
interzona	Especifica a quantidade total de largura de banda para tráfego H.323 da zona para qualquer outra zona.
total	Especifica a quantidade total de largura de banda para o tráfego de H.323 permitido na zona.
sessão	Especifica a largura de banda máxima permitida para uma sessão na zona.
padrão	Especifica o valor padrão para todas as zonas.
nome de zona da zona	Especifica uma zona específica. Dá nome à determinada zona
largura de banda-tamanho	Largura de banda máxima. Para as opções interzona e total, o intervalo é de 1 a 10.000.000 kbps. Para a sessão , a escala é 1 a 5,000 kbps.

Defaults

Nenhum

Modos de comando

Configuração de gatekeeper

[Histórico de comando](#)

A tabela a seguir descreve o comando history:

Versão	Modificação
12.1(3)XI	Esse comando foi introduzido.
12.1(5)XM	O comando bandwidth foi feito reconhecível sem usar o comando zone gatekeeper .
12.2(2)T	Este comando foi integrado ao Cisco IOS Software Release 12.2(2)T.
12.2(2)XB1	Este comando foi implementado no gateway universal Cisco AS5850.

[Diretrizes de uso](#)

Em Cisco IOS Software Release precedentes, a funcionalidade do comando **bandwidth** foi obtida usando o comando **zone gatekeeper**.

[Exemplos](#)

O exemplo a seguir configura a largura de banda máxima da zona para 5.000 kbps:

```
Router(config)# gatekeeperRouter(config-gk)# bandwidth total default 5000
```

[Comandos relacionados](#)

[largura de banda remota](#)? Especifica a largura de banda total para o tráfego de H.323 entre este porteiro, e todo o outro porteiro.

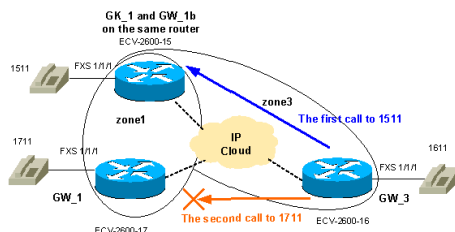
[Configurar](#)

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

Nota: 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

O alvo é restringir a largura de banda disponível entre zone1 e zone3 a um atendimento, e permite um número mais alto dos atendimentos (até dois neste exemplo) em zone1. Desse modo, os requisitos gerais da tarefa de admissão de chamada típica serão atendidos. As mensagens Registration, Admission e Status Protocol (RAS) têm prioridade sobre as mensagens de configuração de chamada H225. Em seguida, segue a negociação H4245, que realmente define os recursos dos lados. A largura de banda real do atendimento é definida assim após a fase da admissão de chamada e a troca dos mensagens RAS. É por isso que o gatekeeper trata cada chamada como uma chamada de 64kb. Conseqüentemente, o aumento nas limitações de largura de banda entre zonas para as chamadas de voz deve ser feito em incrementos de 64kb.

Nota: O GW_3 é configurado no mesmo roteador como o porteiro para ilustrar tal possibilidade para escritórios filiais baixo da gama.

Nota: A verificação das configurações do gatekeeper e gateway é a parte importante de pesquisar defeitos problemas GK-GW. Conseqüentemente, para simplificar a compreensão das configurações, todos os comandos configuration não relacionados foram removidos.

GW_1 ECV-2600-17

```
IOS (tm) C2600 Software (C2600-JSX-M), Version 12.2(7a),  
RELEASE SOFTWARE (fc1)!hostname ECV-2610-17!!interface  
Ethernet0/0 ip address 10.52.218.49 255.255.255.0 h323-  
gateway voip interface h323-gateway voip id gk-  
zone1.test.com ipaddr 10.52.218.47 1718 h323-gateway  
voip h323-id gw_1 h323-gateway voip tech-prefix 1# h323-  
gateway voip bind srcaddr 10.52.218.49  
!voice-port 1/1/0!voice-port 1/1/1!!dial-peer voice 1  
voip destination-pattern .... session target ras!dial-  
peer voice 2 pots destination-pattern 1711 port 1/1/1 no  
register e164!gateway!end
```

GW_2 ECV-2600-16

```
!hostname ECV-2610-16!!interface Ethernet0/0 ip address  
10.52.218.48 255.255.255.0  
h323-gateway voip interface h323-gateway voip id gk-  
zone3.test.com ipaddr 10.52.218.47 1718 h323-gateway  
voip h323-id gw_3 h323-gateway voip tech-prefix 1# h323-  
gateway voip bind srcaddr 10.52.218.48!!voice-port  
1/1/0!voice-port 1/1/1!!dial-peer voice 1 voip  
destination-pattern .... session target ras!dial-peer  
voice 2 pots destination-pattern 1611 port 1/1/1 no  
register e164!gateway !! end
```

GK_1 ECV-2600-15

```
hostname ECV-2610-15!boot system tftp c2600-jsx-mz.122-  
7a.bin 10.52.218.2!!interface Ethernet0/0 ip address  
10.52.218.47 255.255.255.0 half-duplex h323-gateway voip  
interface h323-gateway voip id gk-zone1.test.com ipaddr  
10.52.218.47 1718 h323-gateway voip h323-id gw_1b h323-  
gateway voip tech-prefix 1# h323-gateway voip bind  
srcaddr 10.52.218.47!!voice-port 1/1/0!voice-port  
1/1/1!!dial-peer voice 6 pots destination-pattern 1511  
port 1/1/1 no register e164!!dial-peer voice 5 voip  
destination-pattern .... session target ras!gateway  
!!gatekeeper zone local gk-zone1.test.com test.com  
10.52.218.47 zone local gk-zone3.test.com test.com zone  
prefix gk-zone1.test.com 15.. gw-priority 10 gw_1b zone
```

```
prefix gk-zone3.test.com 16.. gw-priority 10 gw_3 zone
prefix gk-zone1.test.com 17.. gw-priority 10 gw_1 gw-
type-prefix 1#* default-technologybandwidth interzone
zone gk-zone1.test.com 64!--- Applies the restriction
between gk-zone1, and all !--- other zones to 64bk. That
allows one call only. bandwidth total zone gk-
zone1.test.com 128!--- Applies the restriction to the
total number of calls in zone1, !--- and allows two call
in the gk-zone1.no shutdown!endECV-2610-15#
```

Verificar

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

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 o gateway?** indica o status de registro de gateway.
- **mostre valores-limite do porteiro?** alista todos os gateways registrados ao porteiro.
- **mostre o prefixo da zona de gatekeeper?** indica todos os prefixos de zona configurados no porteiro.
- **show gatekeeper call?** mostra as chamadas ativa processadas pelo porteiro.

Troubleshooting

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

Nota: Antes de emitir **comandos debug**, consulte [Informações importantes sobre comandos debug](#).

- **debugar mensagens h225 asn1?displays H225** (configuração de chamada RAS e Q931).
- **debugar mensagens de configuração de chamada cch323 h225?displays H225.**

Estão aqui alguns links úteis:

- [Identificação e solução de problemas e depuração das chamadas VoIP - Conceitos básicos](#)
- [Comandos de debug VoIP](#)
- [Referência aos Comandos de Fax, Vídeo e Voz do Cisco IOS, Versão 12.2](#)

Exemplo de show e debug

```
!--- First step is to check the gateway registrations. !--- On the first gateway:ECV-2610-
17#show gatewayGateway gw_1 is registered to Gatekeeper gk-zone1.test.comAlias list (CLI
configured) H323-ID gw_1Alias list (last RCF) H323-ID gw_1 H323 resource thresholding is
DisabledECV-2610-17#-----!--- And on
```



```

the second Gateway: ECV-2610-16#show gatewayGateway gw_3 is registered to Gatekeeper gk-
zone3.test.comAlias list (CLI configured) H323-ID gw_3Alias list (last RCF) H323-ID gw_3H323
resource thresholding is DisabledECV-2610-16#-----
-----!--- The same on the third Gateway: ECV-2610-15#show gatewayGateway gw_1b is registered
to Gatekeeper gk-zone1.test.comAlias list (CLI configured) H323-ID gw_1bAlias list (last RCF)
H323-ID gw_1bH323 resource thresholding is DisabledECV-2610-15#-----
-----!--- And on the corresponding Gatekeeper: ECV-2610-15#show
gatekeeper end
GATEKEEPER ENDPOINT REGISTRATION
=====CallSignalAddr Port RASignalAddr Port Zone Name
Type F-----
10.52.218.47 58841 gk-zone1.test.com VOIP-GWH323-ID: gw_1b10.52.218.48 1720 10.52.218.48
59067 gk-zone3.test.com VOIP-GWH323-ID: gw_310.52.218.49 1720 10.52.218.49 52887 gk-
zone1.test.com VOIP-GWH323-ID: gw_1Total number of active registrations = 3ECV-2610-15#-----
-----!--- To check the dial plan on the
Gatekeeper: ECV-2610-15#show gatekeeper zone pre ZONE PREFIX TABLE =====GK-NAME
E164-PREFIX-----gk-zone1.test.com 15..gk-zone3.test.com 16..gk-zone1.test.com
17..ECV-2610-15#!--- All configured prefixes should be seen in the zone list.-----
-----!--- To check the zone status on the
Gatekeeper: !-- The output shows one permitted interzone call.ECV-2610-15#show gatekeeper zone
st GATEKEEPER ZONES =====GK name Domain Name RAS Address PORT FLAGS-----
-----!--- The output shows the bandwidth restrictions for
this zone.gk-zone1.test.com 10.52.218.47 1719 LSBANDWIDTH INFORMATION (kbps) : Maximum total
bandwidth : 128 Current total bandwidth : 64 Maximum interzone bandwidth : 64 Current interzone
bandwidth : 64 Maximum session bandwidth : Total number of concurrent calls : 1 SUBNET
ATTRIBUTES : All Other Subnets : (Enabled) PROXY USAGE CONFIGURATION : Inbound Calls from all
other zones :to terminals in local zone gk-zone1.test.com : use proxy to gateways in local zone
gk-zone1.test.com : do not use proxy to MCUs in local zone gk-zone1.test.com : do not use proxy
Outbound Calls to all other zones : from terminals in local zone gk-zone1.test.com : use proxy
from gateways in local zone gk-zone1.test.com : do not use proxy from MCUs in local zone gk-
zone1.test.com : do not use proxy!--- There are no bandwidth restrictions for this zone.gk-
zone3.test.com 10.52.218.47 1719 LSBANDWIDTH INFORMATION (kbps) : Maximum total bandwidth :
Current total bandwidth : 64 Maximum interzone bandwidth : Current interzone bandwidth : 64
Maximum session bandwidth : Total number of concurrent calls : 1 SUBNET ATTRIBUTES : All Other
Subnets : (Enabled) PROXY USAGE CONFIGURATION : Inbound Calls from all other zones :to terminals
in local zone gk-zone3.test.com : use proxy to gateways in local zone gk-zone3.test.com : do not
use proxy to MCUs in local zone gk-zone3.test.com : do not use proxy Outbound Calls to all other
zones : from terminals in local zone gk-zone3.test.com : use proxy from gateways in local zone
gk-zone3.test.com : do not use proxy from MCUs in local zone gk-zone3.test.com : do not use
proxyECV-2610-15#-----
-----ECV-2610-
15#show gatekeeper callTotal number of active calls = 1. GATEKEEPER CALL
INFO =====LocalCallID Age(secs)
BW5-0 1 64(Kbps) Endpt(s): Alias
E.164Addr CallSignalAddr Port RASignalAddr Port src EP: gw_3 1611
10.52.218.48 1720 10.52.218.48 59067 dst EP: gw_1b 1511 10.52.218.47
1720 10.52.218.47 58841ECV-2610-15#-----
-----!--- The output shows that we reach maximum number of calls for gk-zone1.ECV-2610-
15#ECV-2610-15#show gatekeeper zone st GATEKEEPER ZONES
=====GK name Domain Name RAS Address PORT FLAGS-----
-----gk-zone1.test.com 10.52.218.47 1719
LSBANDWIDTH INFORMATION (kbps) : Maximum total bandwidth : 128 Current total bandwidth : 128
Maximum interzone bandwidth : 64 Current interzone bandwidth : 64 Maximum session bandwidth :
Total number of concurrent calls : 2 SUBNET ATTRIBUTES : All Other Subnets : (Enabled) PROXY
USAGE CONFIGURATION : Inbound Calls from all other zones :to terminals in local zone gk-
zone1.test.com : use proxy to gateways in local zone gk-zone1.test.com : do not use proxy to
MCUs in local zone gk-zone1.test.com : do not use proxy Outbound Calls to all other zones : from
terminals in local zone gk-zone1.test.com : use proxy from gateways in local zone gk-
zone1.test.com : do not use proxy from MCUs in local zone gk-zone1.test.com : do not use
proxygk-zone3.test.com 10.52.218.47 1719 LSBANDWIDTH INFORMATION (kbps) :
Maximum total bandwidth : Current total bandwidth : 64 Maximum interzone bandwidth : Current
interzone bandwidth : 64 Maximum session bandwidth : Total number of concurrent calls : 1 SUBNET
ATTRIBUTES : All Other Subnets : (Enabled) PROXY USAGE CONFIGURATION : Inbound Calls from all
other zones :to terminals in local zone gk-zone3.test.com : use proxy to gateways in local zone
gk-zone3.test.com : do not use proxy to MCUs in local zone gk-zone3.test.com : do not use proxy
Outbound Calls to all other zones : from terminals in local zone gk-zone3.test.com : use proxy

```


from gateways in local zone gk-zone3.test.com : do not use proxy from MCUs in local zone gk-zone3.test.com : do not use proxygk-zone2.tes test.com 10.52.218.46 1719 RS ECV-2610-15#ECV-2610-15#show gatekeeper callTotal number of active calls = 2.

GATEKEEPER CALL INFO =====LocalCallID
Age(secs) BW20-33504 49 64(kbps) Endpt(s):
Alias E.164Addr CallSignalAddr Port RASSignalAddr Port src EP: gw_3 1611
10.52.218.48 1720 10.52.218.48 49762 dst EP: gw_1b 1510 10.52.218.47
1720 10.52.218.47 52344LocalCallID Age(secs) BW21-22720
36 64(Kbps) Endpt(s): Alias E.164Addr CallSignalAddr Port RASSignalAddr Port src EP:
gw_1 1711 10.52.218.49 1720 10.52.218.49 54114 dst EP: gw_1b 1511
10.52.218.47 1720 10.52.218.47 52344ECV-2610-15#-----

-----!--- The conversation between the gateway and gatekeeper consists of
!--- an exchange of RAS messages. !--- We start call to 1511 from GW_3.ECV-2610-16#deb h225
asn1H.225 ASN1 Messages debugging is onECV-2610-16#*Mar 1 14:22:20.972: RAS OUTGOING PDU

::=value RasMessage ::= admissionRequest : { requestSeqNum 970 callType pointToPoint : NULL
callModel direct : NULL endpointIdentifier {"8262B76400000019"} destinationInfo { e164 : "1511"
} srcInfo { h323-ID : {"gw_3"} } bandwidth 640 callReferenceValue 23 nonStandardData {
nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18
} data '000000'H } conferenceID '00000000000000000000000000000000'H activeMC FALSE answerCall
FALSE canMapAlias TRUE callIdentifier { guid '00000000000000000000000000000000'H }
willSupplyUUIEs FALSE } *Mar 1 14:22:20.992: RAS OUTGOING ENCODE BUFFER::= 27 8803C900 F0003800
32003600 32004200 37003600 34003000 30003000 30003000 30003100 39010180 48440140 03006700
77005F00 33400280 001740B5 00001203 00000000 00000000 00000000 00000000 00000004 E0200180
11000000 00000000 00000000 00000000 00000100 *Mar 1 14:22:21.008: *Mar 1 14:22:21.073: RAS
INCOMING ENCODE BUFFER::= 2B 0003C940 0280000A 34DA2F06 B800EF14 00C00100 020000*Mar 1
14:22:21.077: *Mar 1 14:22:21.081: RAS INCOMING PDU ::=!--- The GW_3 gets permission to proceed
with that call.value RasMessage ::= admissionConfirm : { requestSeqNum 970 bandwidth 640
callModel direct : NULL destCallSignalAddress ipAddress : { ip '0A34DA2F'H port 1720 }
irrFrequency 240 willRespondToIRR FALSE uuiEsRequested { setup FALSE callProceeding FALSE
connect FALSE alerting FALSE information FALSE releaseComplete FALSE facility FALSE progress
FALSE empty FALSE } !--- The Call setup message from GW_3 follows.*Mar 1 14:22:21.105: H225.0

OUTGOING PDU ::=value H323_UserInformation ::= { h323-uu-pdu { h323-message-body setup : {
protocolIdentifier { 0 0 8 2250 0 2 } sourceAddress { h323-ID : {"gw_3"} } sourceInfo { gateway
{ protocol { voice : { supportedPrefixes { { prefix e164 : "1#" } } } } } mc FALSE undefinedNode
FALSE } activeMC FALSE conferenceID '00000000000000000000000000000000'H conferenceGoal create :
NULL callType pointToPoint : NULL sourceCallSignalAddress ipAddress : { ip '0A34DA30'H port
11018 } callIdentifier { guid '00000000000000000000000000000000'H } fastStart {
'0000000D4001800A040001000A34DA3041C5'H, '400000060401004D40018011140001000A34DA30...H }
mediaWaitForConnect FALSE canOverlapSend FALSE } h245Tunneling FALSE } *Mar 1 14:22:21.141:
H225.0 OUTGOING ENCODE BUFFER::= 20 A0060008 914A0002 01400300 67007700 5F003308 80013C05
04010020 40000000 00000000 00000000 00000000 00000045 1C07000A 34DA302B 0A110000 00000000
00000000 00000000 00000032 02120000 000D4001 800A0400 01000A34 DA3041C5 1D400000 06040100
4D400180 11140001 000A34DA 3041C400 0A34DA30 41C50100 01000680 0100*Mar 1 14:22:21.161: *Mar 1
14:22:21.417: H225.0 INCOMING ENCODE BUFFER::= 21 80060008 914A0002 00048811 00000000 00000000
00000000 00000000 00390219 0000000D 40018011 14000100 0A34DA2F 486E000A 34DA2F48 6F1D4000
00060401 004D4001 80111400 01000A34 DA3041C4 000A34DA 2F486F06 800100*Mar 1 14:22:21.429: *Mar 1
14:22:21.429: H225.0 INCOMING PDU ::=!--- The GW_3 gets Call Proceeding from GW_1b.value
H323_UserInformation ::= { h323-uu-pdu { h323-message-body callProceeding : { protocolIdentifier
{ 0 0 8 2250 0 2 } destinationInfo { mc FALSE undefinedNode FALSE } callIdentifier { guid
'00000000000000000000000000000000'H } fastStart {
'0000000D40018011140001000A34DA2F486E000A...H, '400000060401004D40018011140001000A34DA30...H }
} h245Tunneling FALSE } } *Mar 1 14:22:21.617: H225.0 INCOMING ENCODE BUFFER::= 28 001A0006
0008914A 00020000 00000000 00000000 00000000 00000000 06A00100 120140B5 0000120B 60011000
011E041E 028188*Mar 1 14:22:21.626: *Mar 1 14:22:21.626: H225.0 INCOMING PDU ::=!--- The GW_3
geta Call Progress from GW_1b.value H323_UserInformation ::= { h323-uu-pdu { h323-message-body
progress : { protocolIdentifier { 0 0 8 2250 0 2 } destinationInfo { mc FALSE undefinedNode
FALSE } callIdentifier { guid '00000000000000000000000000000000'H } } h245Tunneling FALSE
nonStandardControl { { nonStandardIdentifier h221NonStandard : {t35CountryCode 181 t35Extension
0 manufacturerCode 18 } data '60011000011E041E028188' } } } *Mar 1 14:22:21.642: H225 NONSTD
INCOMING ENCODE BUFFER::= 60 01100001 1E041E02 8188*Mar 1 14:22:21.646: *Mar 1 14:22:21.646:

H225 NONSTD INCOMING PDU ::=!--- The GW_3 get some facility messagesfrom GW_1b.value
H323_UU_NonStdInfo ::= { version 16 protoParam qsigNonStdInfo : { iei 30 rawMesg '1E028188'H } }
*Mar 1 14:22:22.831: %SYS-3-MGDTIMER: Running timer, init, timer = 81F1AC08.-Process= "Virtual
Exec", ipl= 0, pid= 61-Traceback= 803250A4 80325214 80325318 80EB12C0 80EB17DC 802A65F0

802B5080 8033D818*Mar 1 14:22:22.835: H225 NONSTD OUTGOING PDU ::=value ARQnonStandardInfo ::= { sourceAlias { } sourceExtAlias { } } *Mar 1 14:22:22.839: H225 NONSTD OUTGOING ENCODE BUFFER::= 00 0000*Mar 1 14:22:22.839: *Mar 1 14:22:22.839: **RAS OUTGOING PDU** ::=!--- *The GW_3 starts the second Call to 1711 now we send RAS message to GK.*value RasMessage ::= **admissionRequest** : { requestSeqNum 971 callType pointToPoint : NULL callModel direct : NULL endpointIdentifier {"8262B76400000019"} destinationInfo { e164 : "1711" } srcInfo { h323-ID : {"gw_3"} } bandwidth 640 callReferenceValue 24 nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '000000'H } conferenceID '00000000000000000000000000000000'H activeMC FALSE answerCall FALSE canMapAlias TRUE callIdentifier { guid '00000000000000000000000000000000'H } willSupplyUUIEs FALSE } *Mar 1 14:22:22.860: RAS OUTGOING ENCODE BUFFER::= 27 8803CA00 F0003800 32003600 32004200 37003600 34003000 30003000 30003000 30003100 39010180 4A440140 03006700 77005F00 33400280 001840B5 00001203 00000000 00000000 00000000 00000000 00000000 00000004 E0200180 11000000 00000000 00000000 00000000 00000100 *Mar 1 14:22:22.876: *Mar 1 14:22:22.940: RAS INCOMING ENCODE BUFFER::= 2B 0003CA40 0280000A 34DA3106 B800EF14 00C00100 020000*Mar 1 14:22:22.944: *Mar 1 14:22:22.944: **RAS INCOMING PDU** ::=!--- *The GW_3 gets permission to proceed as there are no restrictions on zone3.*value RasMessage ::= **admissionConfirm** : { requestSeqNum 971 bandwidth 640 callModel direct : NULL destCallSignalAddress ipAddress : { ip '0A34DA31'H port 1720 } irrFrequency 240 willRespondToIRR FALSE uuiesRequested { setup FALSE callProceeding FALSE connect FALSE alerting FALSE information FALSE releaseComplete FALSE facility FALSE progress FALSE empty FALSE } } *Mar 1 14:22:22.972: **H225.0 OUTGOING PDU** ::=!--- *The GW_3 sends setup message to GW_1.*value H323_UserInformation ::= { h323-uu-pdu { h323-message-body **setup** : { protocolIdentifier { 0 0 8 2250 0 2 } sourceAddress { h323-ID : {"gw_3"} } sourceInfo { gateway { protocol { voice : { supportedPrefixes { { prefix e164 : "1#" } } } } } mc FALSE undefinedNode FALSE } activeMC FALSE conferenceID '00000000000000000000000000000000'H conferenceGoal create : NULL callType pointToPoint : NULL sourceCallSignalAddress ipAddress : { ip '0A34DA30'H port 11019 } callIdentifier { guid '00000000000000000000000000000000'H } fastStart { '0000000D4001800A040001000A34DA30402F'H, '400000060401004D40018011140001000A34DA30...'H } mediaWaitForConnect FALSE canOverlapSend FALSE } h245Tunneling FALSE } *Mar 1 14:22:23.008: H225.0 OUTGOING ENCODE BUFFER::= 20 A0060008 914A0002 01400300 67007700 5F003308 80013C05 04010020 40000000 00000000 00000000 00000000 00000045 1C07000A 34DA302B 0B110000 00000000 00000000 00000000 00000032 02120000 000D4001 800A0400 01000A34 DA30402F 1D400000 06040100 4D400180 11140001 000A34DA 30402E00 0A34DA30 402F0100 01000680 0100*Mar 1 14:22:23.028: *Mar 1 14:22:23.220: H225.0 INCOMING ENCODE BUFFER::= 25 80060008 914A0002 01110000 00000000 00000000 00000000 00000006 800100*Mar 1 14:22:23.224: *Mar 1 14:22:23.224: **H225.0 INCOMING PDU** ::=!--- *The GW_1 replies with Release Complete message after asking GK !--- for permission to accept that call. !--- When the permission is denied, we set bandwidth limit.*value H323_UserInformation ::= { h323-uu-pdu { h323-message-body **releaseComplete** : { protocolIdentifier { 0 0 8 2250 0 2 } callIdentifier { guid '00000000000000000000000000000000'H } } h245Tunneling FALSE } } *Mar 1 14:22:23.236: **RAS OUTGOING PDU** ::=!--- *The GW_3 notifies GK that the call does not exist anymore.*value RasMessage ::= **disengageRequest** : { requestSeqNum 972 endpointIdentifier {"8262B76400000019"} conferenceID '00000000000000000000000000000000'H callReferenceValue 24 disengageReason normalDrop : NULL callIdentifier { guid '00000000000000000000000000000000'H } answeredCall FALSE } *Mar 1 14:22:23.248: RAS OUTGOING ENCODE BUFFER::= 3E 03CB1E00 38003200 36003200 42003700 36003400 30003000 30003000 30003000 31003900 00000000 00000000 00000000 00000000 18216111 00000000 00000000 00000000 00000000 00000000 000100*Mar 1 14:22:23.256: *Mar 1 14:22:23.288: RAS INCOMING ENCODE BUFFER::= 40 03CB*Mar 1 14:22:23.288: *Mar 1 14:22:23.288: **RAS INCOMING PDU** ::=!--- *The GK confirms that message.*value RasMessage ::= **disengageConfirm** : { requestSeqNum 972 } ECV-2610-16#u allAll possible debugging has been turned offECV-2610-16#-----
-----!--- *The incoming RAS message to the GK from GW_3.*ECV-2610-15#debug h225 asn1H.225 ASN1 Messages debugging is onECV-2610-15#*Mar 11 21:54:28.313: **RAS INCOMING PDU** ::=value RasMessage ::= **admissionRequest** : { requestSeqNum 970 callType pointToPoint : NULL callModel direct : NULL endpointIdentifier {"8262B76400000019"} destinationInfo { e164 : "1511" } srcInfo { h323-ID : {"gw_3"} } bandwidth 640 callReferenceValue 23 nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '000000'H } conferenceID '00000000000000000000000000000000'H activeMC FALSE answerCall FALSE canMapAlias TRUE callIdentifier { guid '00000000000000000000000000000000'H } willSupplyUUIEs FALSE } *Mar 11 21:54:28.334: H225 NONSTD INCOMING ENCODE BUFFER::= 00 0000*Mar 11 21:54:28.334: *Mar 11 21:54:28.334: H225 NONSTD INCOMING PDU ::=value ARQnonStandardInfo ::= { sourceAlias { } sourceExtAlias { } }!--- *The outgoing RAS message fro GK to GW_3 with permission to start call.**Mar 11 21:54:28.338: **RAS OUTGOING PDU** ::=value RasMessage ::= **admissionConfirm** : { requestSeqNum 970 bandwidth 640 callModel direct : NULL destCallSignalAddress ipAddress : { ip '0A34DA2F'H port 1720 } irrFrequency 240 willRespondToIRR FALSE uuiesRequested { setup FALSE

callProceeding FALSE connect FALSE alerting FALSE information FALSE releaseComplete FALSE
facility FALSE progress FALSE empty FALSE } } *Mar 11 21:54:28.350: RAS OUTGOING ENCODE
BUFFER::= 2B 0003C940 0280000A 34DA2F06 B800EF14 00C00100 020000*Mar 11 21:54:28.354: *Mar 11
21:54:28.446: H225.0 INCOMING ENCODE BUFFER::= 20 A0060008 914A0002 01400300 67007700 5F003308
80013C05 04010020 40000000 00000000 00000000 00000000 00000045 1C07000A 34DA302B 0A110000
00000000 00000000 00000000 00000032 02120000 000D4001 800A0400 01000A34 DA3041C5 1D400000
06040100 4D400180 11140001 000A34DA 3041C400 0A34DA30 41C50100 01000680 0100*Mar 11
21:54:28.466: *Mar 11 21:54:28.470: **H225.0 INCOMING PDU** ::=!--- *The incoming H323(Q931) message
from GW_3 to GW_1b on the same router as GK.*value H323_UserInformation ::= { h323-uu-pdu { h323-
message-body **setup** : { protocolIdentifier { 0 0 8 2250 0 2 } sourceAddress { h323-ID : {"gw_3"}
} sourceInfo { gateway { protocol { voice : { supportedPrefixes { { prefix e164 : "1#" } } } }
} mc FALSE undefinedNode FALSE } activeMC FALSE conferenceID '00000000000000000000000000000000'
conferenceGoal create : NULL callType pointToPoint : NULL sourceCallSignalAddress ipAddress : {
ip '0A34DA30'H port 11018 } callIdentifier { guid '00000000000000000000000000000000'H }
fastStart { '0000000D4001800A040001000A34DA3041C5'H,
'400000060401004D40018011140001000A34DA30...'H } mediaWaitForConnect FALSE canOverlapSend FALSE
} h245Tunneling FALSE } } *Mar 11 21:54:28.514: H225 NONSTD OUTGOING PDU ::=value
ARQnonStandardInfo ::= { sourceAlias { } sourceExtAlias { } } *Mar 11 21:54:28.518: H225 NONSTD
OUTGOING ENCODE BUFFER::= 00 0000*Mar 11 21:54:28.518: *Mar 11 21:54:28.518: **RAS OUTGOING PDU**
::=!--- *The GW_1b asks GK if it can accept call from GW_3.*value RasMessage ::= **admissionRequest**
: { requestSeqNum **1347** callType pointToPoint : NULL callModel direct : NULL endpointIdentifier
{"82717F5C0000001B"} destinationInfo { e164 : "**1511**" } srcInfo { h323-ID : {"gw_3"} }
srcCallSignalAddress ipAddress : { ip '0A34DA30'H port 11018 } bandwidth 640 callReferenceValue
29 nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0
manufacturerCode 18 } data '000000'H } conferenceID '00000000000000000000000000000000'H activeMC
FALSE answerCall TRUE canMapAlias TRUE callIdentifier { guid '00000000000000000000000000000000'H
} willSupplyUUIES FALSE } *Mar 11 21:54:28.542: RAS OUTGOING ENCODE BUFFER::= 27 98054200
F0003800 32003700 31003700 46003500 43003000 30003000 30003000 30003100 42010180 48440140
03006700 77005F00 33000A34 DA302B0A 40028000 1D40B500 00120300 00000000 00000000 00000000
00000000 000044E0 20018011 00000000 00000000 00000000 00000000 000100*Mar 11 21:54:28.558: *Mar
11 21:54:28.562: RAS INCOMING ENCODE BUFFER::= 27 98054200 F0003800 32003700 31003700 46003500
43003000 30003000 30003000 30003100 42010180 48440140 03006700 77005F00 33000A34 DA302B0A
40028000 1D40B500 00120300 00000000 00000000 00000000 00000000 000044E0 20018011 00000000
00000000 00000000 00000000 000100*Mar 11 21:54:28.578: *Mar 11 21:54:28.582: **RAS INCOMING PDU**
::=!--- *That is the same RAS message. The GK gets it, and sees the sequence number. !--- The GK
is on the same router as GW_1b, so all messages can be seen twice.*value RasMessage ::=
admissionRequest : { requestSeqNum **1347** callType pointToPoint : NULL callModel direct :
NULLendpointIdentifier {"82717F5C0000001B"} destinationInfo { e164 : "1511" } srcInfo { h323-ID
: {"gw_3"} } srcCallSignalAddress ipAddress : { ip '0A34DA30'H port 11018 } bandwidth 640
callReferenceValue 29 nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode
181 t35Extension 0 manufacturerCode 18 } data '000000' } conferenceID
'00000000000000000000000000000000'H activeMC FALSE answerCall TRUE canMapAlias TRUE
callIdentifier { guid '00000000000000000000000000000000'H } willSupplyUUIES FALSE } *Mar 11
21:54:28.606: H225 NONSTD INCOMING ENCODE BUFFER::= 00 0000*Mar 11 21:54:28.606: *Mar 11
21:54:28.606: H225 NONSTD INCOMING PDU ::=value ARQnonStandardInfo ::= { sourceAlias { }
sourceExtAlias { } } *Mar 11 21:54:28.610: **RAS OUTGOING PDU** ::=!--- *The GK grants the permission
to GW_1b. !--- This is a message in the GK debug outgoing*value RasMessage ::= **admissionConfirm** :
{ requestSeqNum 1347 bandwidth 640 callModel direct : NULL destCallSignalAddress ipAddress : {
ip '0A34DA2F'H port 1720 } irrFrequency 240 willRespondToIRR FALSE uuiiesRequested { setup FALSE
callProceeding FALSE connect FALSE alerting FALSE information FALSE releaseComplete FALSE
facility FALSE progress FALSE empty FALSE } } *Mar 11 21:54:28.622: RAS OUTGOING ENCODE
BUFFER::= 2B 00054240 0280000A 34DA2F06 B800EF14 00C00100 020000*Mar 11 21:54:28.626: *Mar 11
21:54:28.630: RAS INCOMING ENCODE BUFFER::= 2B 00054240 0280000A 34DA2F06 B800EF14 00C00100
020000*Mar 11 21:54:28.634: *Mar 11 21:54:28.634: **RAS INCOMING PDU** ::=!--- *The GK grants the
permission to GW_1b. !--- This is a message in the GW_1b debug incoming.*value RasMessage ::=
admissionConfirm : { requestSeqNum 1347 bandwidth 640 callModel direct : NULL
destCallSignalAddress ipAddress : { ip '0A34DA2F'H port 1720 } irrFrequency 240 willRespondToIRR
FALSE uuiiesRequested { setup FALSE callProceeding FALSE connect FALSE alerting FALSE information
FALSE releaseComplete FALSE facility FALSE progress FALSE empty FALSE } } *Mar 11 21:54:28.654:
%SYS-3-MGDTIMER: Timer has parent, timer link, timer = 820AE990.-Process= "CC-API_VCM", ipl= 6,
pid= 93-Traceback= 80325850 8032A720 80E74850 8033D818*Mar 11 21:54:28.666: **H225.0 OUTGOING PDU**
::=!--- *The GW_1b replies to GW_3 setup message.*value H323_UserInformation ::= { h323-uu-pdu {
h323-message-body **callProceeding** : { protocolIdentifier { 0 0 8 2250 0 2 } destinationInfo { mc
FALSE undefinedNode FALSE } callIdentifier { guid '00000000000000000000000000000000'H }
}

```
fastStart { '0000000D40018011140001000A34DA2F486E000A...'H,
'400000060401004D40018011140001000A34DA30...'H } } h245Tunneling FALSE } } *Mar 11 21:54:28.682:
H225.0 OUTGOING ENCODE BUFFER::= 21 80060008 914A0002 00048811 00000000 00000000 00000000
00000000 00390219 0000000D 40018011 14000100 0A34DA2F 486E000A 34DA2F48 6F1D4000 00060401
004D4001 80111400 01000A34 DA3041C4 000A34DA 2F486F06 800100*Mar 11 21:54:28.694: *Mar 11
21:54:28.710: H225 NONSTD OUTGOING PDU ::=value H323_UU_NonStdInfo ::= {version 16 protoParam
qsigNonStdInfo : { iei 30 rawMesg '1E028188'H } } *Mar 11 21:54:28.714: H225 NONSTD OUTGOING
ENCODE BUFFER::= 60 01100001 1E041E02 8188*Mar 11 21:54:28.714: *Mar 11 21:54:28.714: H225.0
OUTGOING PDU ::=!-- The GW_1b replies to GW_3 setup message and sends second message.value
H323_UserInformation ::= { h323-uu-pdu { h323-message-body progress : { protocolIdentifier { 0 0
8 2250 0 2 } destinationInfo { mc FALSE undefinedNode FALSE } callIdentifier { guid
'00000000000000000000000000000000'H } } h245Tunneling FALSE nonStandardControl { {
nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18
} data '60011000011E041E028188'H } } } *Mar 11 21:54:28.734: H225.0 OUTGOING ENCODE BUFFER::= 28
001A0006 0008914A 00020000 00000000 00000000 00000000 00000000 06A00100 120140B5 0000120B
60011000 011E041E 028188*Mar 11 21:54:28.742: *Mar 11 21:54:30.161: RAS INCOMING ENCODE
BUFFER::= 27 8803CA00 F0003800 32003600 32004200 37003600 34003000 30003000 30003000 30003100
39010180 4A440140 03006700 77005F00 33400280 001840B5 00001203 00000000 00000000 00000000
00000000 00000004 E0200180 11000000 00000000 00000000 00000000 00000100 *Mar 11 21:54:30.177:
*Mar 11 21:54:30.181: RAS INCOMING PDU ::=!-- The GK gets ARQ from GW_3 for the second
call.value RasMessage ::= admissionRequest: { requestSeqNum 971 callType pointToPoint : NULL
callModel direct : NULL endpointIdentifier {"8262B76400000019"} destinationInfo { e164 : "1711"
} srcInfo { h323-ID : {"gw_3"} } bandwidth 640 callReferenceValue 24 nonStandardData {
nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18
} data '000000'H } conferenceID '00000000000000000000000000000000'H activeMC FALSE answerCall
FALSE canMapAlias TRUE callIdentifier { guid '00000000000000000000000000000000'H }
willSupplyUUIEs FALSE } *Mar 11 21:54:30.197: H225 NONSTD INCOMING ENCODE BUFFER::= 00 0000*Mar
11 21:54:30.201: *Mar 11 21:54:30.201: H225 NONSTD INCOMING PDU ::=value ARQnonStandardInfo ::=
{ sourceAlias { } sourceExtAlias { } } *Mar 11 21:54:30.205: RAS OUTGOING PDU ::=!-- The GK
grants permission to GW_3, as there are no restrictions for zone3.value RasMessage ::=
admissionConfirm : { requestSeqNum 971 bandwidth 640 callModel direct : NULL
destCallSignalAddress ipAddress : { ip '0A34DA31'H !-- The hexadecimal number is 10.52.218.49,
IP of GW_1. port 1720 } irrFrequency 240 willRespondToIRR FALSE uuiesRequested { setup FALSE
callProceeding FALSE connect FALSE alerting FALSE information FALSE releaseComplete FALSE
facility FALSE progress FALSE empty FALSE } } *Mar 11 21:54:30.217: RAS OUTGOING ENCODE
BUFFER::= 2B 0003CA40 0280000A 34DA3106 B800EF14 00C00100 020000*Mar 11 21:54:30.221: *Mar 11
21:54:30.429: RAS INCOMING ENCODE BUFFER::= 27 98045F00 F0003800 32003300 38003600 30004400
34003000 30003000 30003000 30003100 41010180 4A440140 03006700 77005F00 33000A34 DA302B0B
40028000 2840B500 00120300 00000000 00000000 00000000 00000000 000044E0 20018011 00000000
00000000 00000000 00000000 000100*Mar 11 21:54:30.445: *Mar 11 21:54:30.445: RAS INCOMING PDU
::=!-- The incoming request from GW_1 asks for permission to accept call from GW_3.value
RasMessage ::= admissionRequest : { requestSeqNum 1120 callType pointToPoint : NULL callModel
direct : NULL endpointIdentifier {"823860D40000001A"} destinationInfo { e164 : "1711" } srcInfo
{ h323-ID : {"gw_3"} }srcCallSignalAddress ipAddress : { ip '0A34DA30'H port 11019 } bandwidth
640 callReferenceValue 40 nonStandardData { nonStandardIdentifier h221NonStandard : {
t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '000000'H } conferenceID
'00000000000000000000000000000000'H activeMC FALSE answerCall TRUE canMapAlias TRUE
callIdentifier {guid '00000000000000000000000000000000'H } willSupplyUUIEs FALSE } *Mar 11
21:54:30.469: H225 NONSTD INCOMING ENCODE BUFFER::= 00 0000*Mar 11 21:54:30.469: *Mar 11
21:54:30.469: H225 NONSTD INCOMING PDU ::=value ARQnonStandardInfo ::= { sourceAlias { }
sourceExtAlias { } }!-- The GK does not allow the call to come through, and replies with
ARJ.*Mar 11 21:54:30.473: RAS OUTGOING PDU ::=value RasMessage ::= admissionReject : {
requestSeqNum 1120 rejectReason requestDenied : NULL } *Mar 11 21:54:30.477: RAS OUTGOING ENCODE
BUFFER::= 2C 045F20*Mar 11 21:54:30.477: *Mar 11 21:54:30.541: RAS INCOMING ENCODE BUFFER::= 3E
03CB1E00 38003200 36003200 42003700 36003400 30003000 30003000 30003000 31003900 00000000
00000000 00000000 00000000 18216111 00000000 00000000 00000000 00000000 000100*Mar 11
21:54:30.553: *Mar 11 21:54:30.557: RAS INCOMING PDU ::=!-- The GW_3 notifies GK that call does
not exist anymore.value RasMessage ::= disengageRequest : { requestSeqNum 972 endpointIdentifier
{"8262B76400000019"} conferenceID '00000000000000000000000000000000'H callReferenceValue 24
disengageReason normalDrop : NULL callIdentifier { guid '00000000000000000000000000000000'H }
answeredCall FALSE } *Mar 11 21:54:30.565: RAS OUTGOING PDU ::=!-- The GK confirms the message
from GW_3value RasMessage ::= disengageConfirm : { requestSeqNum 972 }-----
-----!-- The call setup from GW_1 perspective.ECV-2610-
17#deb h225 asn1H.225 ASN1 Messages debugging is onECV-2610-17#*Mar 2 22:55:40: H225.0 INCOMING
```




```

ENCODE BUFFER::= 20 A0060008 914A0002 01400300 67007700 5F003308 80013C05 04010020 40000000
00000000 00000000 00000000 00000045 1C07000A 34DA302B 0B110000 00000000 00000000 00000000
00000032 02120000 000D4001 800A0400 01000A34 DA30402F 1D400000 06040100 4D400180 11140001
000A34DA 30402E00 0A34DA30 402F0100 01000680 0100*Mar 2 22:55:40: *Mar 2 22:55:40: H225.0
INCOMING PDU ::=!--- The GW_1 gets the H323 (Q931) setup message from GW_3.value
H323_UserInformation ::= { h323-uu-pdu { h323-message-body setup : { protocolIdentifier { 0 0 8
2250 0 2 } sourceAddress { h323-ID : {"gw_3"} } sourceInfo { gateway { protocol { voice : {
supportedPrefixes { { prefix e164 : "1#" } } } } } mc FALSE undefinedNode FALSE } activeMC
FALSE conferenceID '00000000000000000000000000000000'H conferenceGoal create : NULL callType
pointToPoint : NULL sourceCallSignalAddress ipAddress : { ip '0A34DA30'H port 11019 }
callIdentifier { guid '00000000000000000000000000000000'H } fastStart {
'0000000D4001800A040001000A34DA30402F'H, '400000060401004D40018011140001000A34DA30...'H }
mediaWaitForConnect FALSE canOverlapSend FALSE } h245Tunneling FALSE } } *Mar 2 22:55:40: H225
NONSTD OUTGOING PDU ::=value ARQnonStandardInfo ::= { sourceAlias { } sourceExtAlias { } } *Mar
2 22:55:40: H225 NONSTD OUTGOING ENCODE BUFFER::= 00 0000*Mar 2 22:55:40: *Mar 2 22:55:40: RAS
OUTGOING PDU ::=!--- The GW_1 asks GK for permission to accept the call.value RasMessage ::=
admissionRequest : { requestSeqNum 1120 callType pointToPoint : NULL callModel direct : NULL
endpointIdentifier {"823860D40000001A"} destinationInfo { e164 : "1711" } srcInfo { h323-ID :
{"gw_3"} } srcCallSignalAddress ipAddress : { ip '0A34DA30'H port 11019 } bandwidth 640
callReferenceValue 40 nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode
181 t35Extension 0 manufacturerCode 18 } data '000000'H } conferenceID
'00000000000000000000000000000000'H activeMC FALSE answerCall TRUE canMapAlias TRUE
callIdentifier { guid '00000000000000000000000000000000'H } willSupplyUUIEs FALSE } *Mar 2
22:55:40: RAS OUTGOING ENCODE BUFFER::= 27 98045F00 F0003800 32003300 38003600 30004400 34003000
30003000 30003000 30003100 41010180 4A440140 03006700 77005F00 33000A34 DA302B0B 40028000
2840B500 00120300 00000000 00000000 00000000 00000000 000044E0 20018011 00000000 00000000
00000000 00000000 000100*Mar 2 22:55:41: *Mar 2 22:55:41: RAS INCOMING ENCODE BUFFER::= 2C
045F20*Mar 2 22:55:41: *Mar 2 22:55:41: RAS INCOMING PDU ::=!--- The GK denies permission to
accept the call from GW_3 due to bandwidth limit.value RasMessage ::= admissionReject : {
requestSeqNum 1120 rejectReason requestDenied : NULL } *Mar 2 22:55:41: H225.0 OUTGOING PDU
::=!--- The GW_1 rejects call setup from GW_3.value H323_UserInformation ::= { h323-uu-pdu {
h323-message-body releaseComplete : { protocolIdentifier { 0 0 8 2250 0 2 } callIdentifier {
guid '00000000000000000000000000000000'H } } h245Tunneling FALSE } } *Mar 2 22:55:41: H225.0
OUTGOING ENCODE BUFFER::= 25 80060008 914A0002 01110000 00000000 00000000 00000000 00000006
800100*Mar 2 22:55:41: ECV-2610-17#ECV-2610-17#ECV-2610-17#u allAll possible debugging has been
turned off-----

```

[Informações Relacionadas](#)

- [Troubleshooting e Entendendo o Gerenciamento de Largura de Banda do Cisco Gatekeeper](#)
- [Entendendo Gatekeepers H.323](#)
- [Gatekeeper de alto desempenho Cisco](#)
- [Configurando gateways H.323](#)
- [Configurando gatekeepers H.323](#)
- [Configurando o suporte H.323 para interfaces virtuais](#)
- [Suporte à Tecnologia de Voz](#)
- [Suporte ao Produto de Voz e Comunicações Unificadas](#)
- [Troubleshooting da Telefonia IP Cisco](#) 
- [Suporte Técnico - Cisco Systems](#)