Troubleshooting de GPU, Ponto Final Alternado e Balanceamento de Carga

Índice

Introdução

Este documento foi elaborado para ajudá-lo a solucionar problemas e a entender esses recursos do Cisco Gatekeeper.

- Clusters de gatekeeper e GUP
- Pontos finais alternativos
- Balanceamento de carga

Refira o porteiro de capacidade elevada de Cisco para todas as informações necessárias acerca destas características que incluem a visão geral de características, plataformas suportadas, os software release de Cisco IOS® necessários e como configurar-las, monitorar, e manter.

Pré-requisitos

Requisitos

Os leitores deste documento devem estar cientes da seguinte informação:

- Conhecimento básico da funcionalidade de gatekeeper.
- Conhecimento básico de VoIP, H.323 e sinalização RAS.

Componentes Utilizados

As informações neste documento são baseadas nas versões de software e hardware abaixo.

- Cisco IOS Software Release 12.3(4)T1
- Cisco gateway: Cisco AS5300, Cisco AS5400, e Cisco 3725
- Gatekeeperes Cisco: Cisco 3725 e Cisco 2611

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.

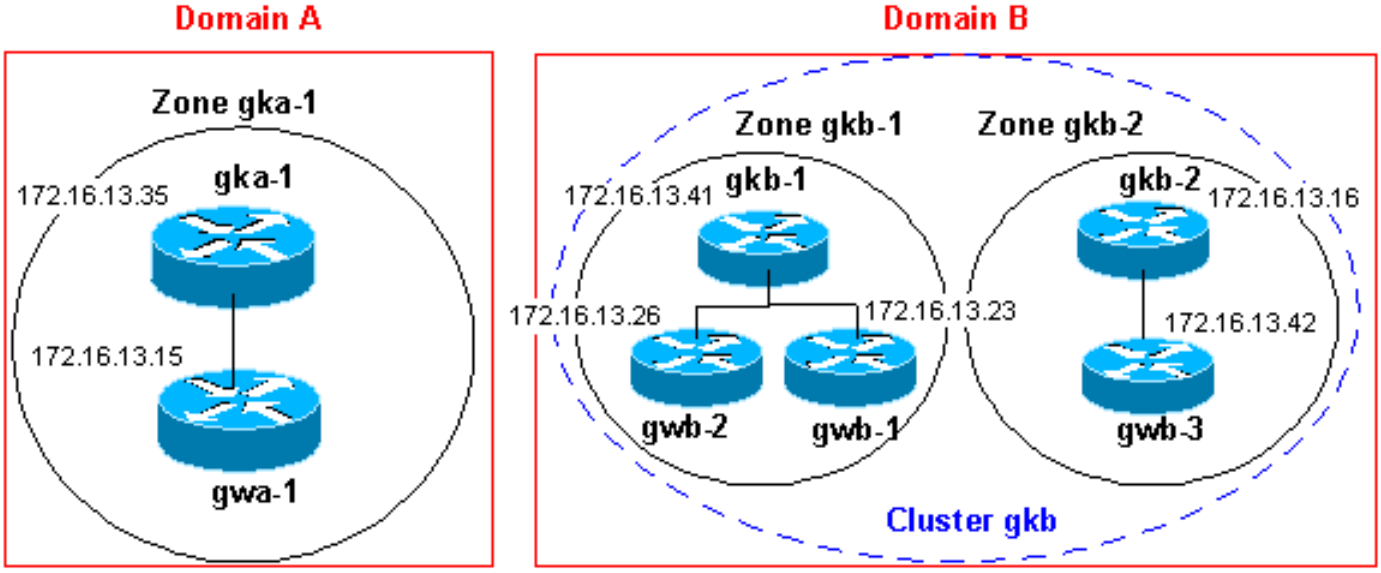
Definições

Termo	Definição
ARQ	Solicitação de Admissão (ARQ) é uma mensagem RAS enviada do ponto final Cisco H.323 para um gatekeeper que solicita um admissão para estabelecer uma chamada.
ACF	Admission Confirm (ACF) é uma mensagem RAS enviada a partir do gatekeeper ao ponto final que confirma a aceitação de uma chamada.
ARJ	Admission Rejection (ARJ) é uma mensagem RAS do gatekeeper para o ponto final que rejeita a requisição de admissão.
GCF	O Gatekeeper Confirm (GCF) é um mensagem RAS enviado de um porteiro ao valor-limite de Cisco H.323 que confirma a descoberta do porteiro.
GRQ	O Gatekeeper Request (GRQ) é um mensagem RAS enviado de um valor-limite de Cisco H.323 para descobrir o porteiro.

GUP	O Protocolo de atualização de gatekeeper é utilizado para compartilhar as informações sobre seus pontos finais e chamadas ativas entre gatekeepers em um cluster.
LCF	LCF (confirmação de localização) é uma mensagem do RAS enviada de um gatekeeper para outro que confirma o LRQ (requisição de localização) e inclui o endereço IP do ponto final de terminação.
LRJ	O Location Reject (LRJ) é um mensagem RAS enviado de um porteiro a outro que rejeita o LRQ.
LRQ	Location Request é uma mensagem do RAS enviada de um gatekeeper a outro que requisita o endereço IP de um ponto final de terminação remoto.
RAS	O protocolo RAS permite ao gatekeeper executar verificação de registro, admissão e status do ponto final.
RCF	O registro confirma (RCF) é um mensagem RAS enviado do porteiro ao valor-limite que confirma o registro.
RRJ	Registration Reject (RRJ) é uma mensagem do RAS enviada a partir do gatekeeper que rejeita uma requisição de registro.
RRQ	A solicitação de registro (RRQ) é uma mensagem de RAS enviada do ponto final para o gatekeeper que solicita o seu registro lá.
URQ	A URQ (Solicitação de descadastro) é uma mensagem de RAS enviada do ponto final para o gatekeeper que solicita o descadastramento com ele.

Topologia de laboratório e configurações

Para explicar como as características trabalham e como pesquisar defeitos, uma instalação de laboratório foi construída com esta topologia:



Topologia de laboratório e configurações

As configurações básicas de todos os gateways e gatekeepers estão na tabela abaixo. Uma certa alteração de configuração foi necessária com casos diferentes. A mudança é indicada quando esta acontece. As configurações abaixo contém apenas as partes que são essenciais às funcionalidades do gateway ou do gatekeeper para esse laboratório.

As configurações de "gwb-1" e de "gwb-2" são quase similares (à exceção do endereço IP de Um ou Mais Servidores Cisco ICM NT e do H.323 ID). Consequentemente, somente `gwb-1` é mostrado abaixo.

gwa-1

```
!
controller E1 3/0
pri-group timeslots 1-2,16
!
interface Ethernet0/0
 ip address 172.16.13.15 255.255.255.224
 half-duplex h323-gateway voip interface
 h323-gateway voip id gka-1 ipaddr 172.16.13.35 1718
 h323-gateway voip
 h323-id gwa-1
 h323-gateway voip tech-prefix 1#
!
voice-port 3/0:15
!
!
dial-peer voice 5336 pots
 incoming called-number
 destination-pattern 5336
 direct-inward-dial
 port 3/0:15
 prefix 21
!
dial-peer voice 3653 voip
 incoming called-number
 destination-pattern 3653
 session target ras
 dtmf-relay h245-alphanumeric
 codec g711ulaw
!
gateway
!
ntp clock-period 17178794
ntp server 172.16.13.35
end
```

gka-1

```
!
gatekeeper
 zone local gka-1 domainA.com 172.16.13.35
 zone remote gkb domainB.com 172.16.13.41 1719
 zone prefix gkb 36*
 zone prefix gka-1 53*
 gw-type-prefix 1#* default-technology
 no shutdown
!
no scheduler
max-task-timentp master
!
```

```
end
```

gwb-1

```
!
controller E1 0
 clock source line primary
 ds0-group 0 timeslots 1-2 type r2-digital r2-compelled
!
interface Ethernet0
 ip address 172.16.13.23 255.255.255.224
 h323-gateway voip interface
 h323-gateway voip id gkb-1 ipaddr 172.16.13.41 1718
 h323-gateway voip h323-id gwb-1
 h323-gateway voip tech-prefix 2#
!
dial-peer voice 3653 pots incoming called-number
 destination-pattern 3653
 port 0:0
 prefix 21
!
dial-peer voice 5336 voip
 incoming called-number
 destination-pattern 5336
 session target ras
 dtmf-relay h245-alphanumeric
 codec g711ulaw
!
gateway
!
ntp clock-period 17179389
ntp server 172.16.13.35
end
```

gwb-3

```
!
interface Ethernet0/0
 ip address 172.16.13.42 255.255.255.224
 half-duplex h323-gateway voip interface
 h323-gateway voip id gkb-1 ipaddr 172.16.13.41 1718
 h323-gateway voip
 h323-id gwb-3
 h323-gateway voip tech-prefix 1#
!
voice-port 3/0/0
!
voice-port 3/0/1
!
dial-peer voice 3653 pots
 destination-pattern 3653
 port 3/0/0
 prefix 21
!
dial-peer voice 5336 voip
 incoming called-number
 destination-pattern 5336
 session target ras
 dtmf-relay h245-alphanumeric

 codec g711ulaw

gateway ! ntp clock-period 17179181 ntp server
172.16.13.35
!
end
```

gkb-1

```
!
gatekeeper
 zone local gkb-1 domainB.com 172.16.13.41
 zone remote gka-1 domainA.com 172.16.13.35 1719
 zone cluster local gkb gkb-1
 element gkb-2 172.16.13.16 1719
 gw-type-prefix 2#* default-technology
 no shutdown
!
ntp clock-period 17179580
ntp server 172.16.13.35
!
end
```

gkb-2

```
!
gatekeeper
 zone local gkb-2 domainB.com 172.16.13.16
 zone cluster local gkb gkb-2
 element gkb-1 172.16.13.41 1719
!
no shutdown
!
ntp clock-period 17179199
ntp server 172.16.13.35
!
end
```

Alternação de gatekeepers

Antes da versão 2 do Cisco H.323, cada zona era controlada somente por um único gatekeeper. O Cisco H.323 versão 2 introduz a idéia de "gatekeeper alternado" para fornecer redundância de gatekeeper. A implementação do recurso de gatekeeper alternado permite que vários gatekeepers controlem uma zona. Quando um ponto final faz o registro com um gatekeeper, uma lista de gatekeepers alternativos é fornecida para a zona em que o ponto final faz o registro e para a qual os alternativos foram especificados via CLI. Se o porteiro falha, o valor-limite pode usar os gatekeeperes alternativos a fim continuar a operação.

A lista de gatekeepers alternativos é fornecida ao Cisco Gatekeeper por meio da CLI para cada zona e é transmitida aos pontos finais por meio do RCF (incluindo lightweight) e mensagens GRQ. Esta lista pode igualmente ser transmitida em outras mensagens, tais como o ARJ ou o URQ, para facilitar uma parada programada controlada do porteiro.

Os gatekeeperes alternativos aprendem sobre chamadas existentes com uma troca da resposta do pedido da solicitação de interrupção (IRQ) /Information (IRR) entre os gateways e os porteiros e mantêm-se a par destes atendimentos.

Um valor-limite que detecte a falha de seu porteiro pode com segurança recuperar dessa falha utilizando um gatekeeper alternativo para os pedidos futuros, incluindo pede para chamadas existentes. Gatekeepers alternativos devem ser configurados em um agrupamento. Compartilham da informação sobre os valores-limite e as chamadas ativa usando o GUP que é executado no TCP.

Protocolo de atualização de gatekeeper

Estão aqui algumas etapas principal e advertências do GUP. Isso também deve ajudá-lo a solucionar o problema.

- Quando um gatekeeper configurado para fazer parte de um cluster fica on-line, ele abre uma porta de TCP para escutar as conexões de entrada para o protocolo GUP.
- Então anuncia sua presença enviando um mensagem de GRQ em uma base periódica. O período padrão é 30 segundos e é configurável usando o temporizador CLI de gatekeeper que o conjunto-elemento anuncia o comando. Essa mensagem GRQ contém dados não padronizados para cada gatekeeper alternado. Estes dados não padronizados são um indicador às substituições que o GRQ não é realmente um GRQ de todo, mas são um pouco apenas um mensagem " anúncio ". Dentro do mensagem de GRQ, o porteiro indica o número de porta que tem aberto para escutar o protocolo GUP.
- Quando você recebe um GRQ do novo gatekeeper, outros gatekeepers no cluster abrem canais TCP para essa porta.
- Os mensagens GUP GRQ podem ser um dos seguintes mensagens: announcementIndication, announcementReject , registrationIndication , unregistrationIndication e resourceIndication.
- A indicação de anúncio também transporta informações sobre a utilização de largura de banda da zona. Isso permite que os gatekeepers alternados gerenciem corretamente a largura de banda para uma única zona, mesmo que os gatekeepers estejam em dispositivos físicos separados.
- Para verificar se os gatekeeperes alternativos se estão comunicando corretamente ou não, use o comando show gatekeeper zone cluster. Esse comando também relata informações de largura de banda para os gatekeepers alternados.
- O porteiro supõe que o gatekeeper alternativo falhou (e supõe que toda a largura de banda previamente atribuída está agora disponível), se o porteiro não recebe um mensagem de anúncio dentro de seis períodos de anúncio, ou se a conexão de TCP com o porteiro está detectada para se quebrar. Com seis períodos de anúncio a cada 30 segundos, o tempo é de três minutos, o que se equipara ao que se supõe ser a duração média de uma chamada. Desse modo, deve ser seguro admitir que a largura de banda foi liberada. Após três minutos, este porteiro declara sua substituição como para baixo e manda uma atualização para notificar todos seus valores-limite registrados que não há nenhum gatekeeper alternativo.
- Quando um ponto final se registra/cancela o registro em um gatekeeper de um cluster, esse gatekeeper usa a mensagem registrationIndication/ unregistrationIndication para atualizar todos os outros gatekeepers desse cluster com as alterações.
- Se um ponto final reportou uma alteração de recurso usando o indicador de disponibilidade de recursos (RAI) para um gatekeeper em um cluster, esse gatekeeper reportará a alteração para todos os gatekeepers alternativos no cluster usando a mensagem GUP resourceIndication.
- As mensagens de GUP são necessárias para que o gatekeeper de um cluster tenha conhecimento suficiente sobre cada ponto final da zona (registro, largura de banda, chamadas ativas, recursos), para poder resolver todas as consultas localmente.
- Quando um valor-limite for comutado de um porteiro a uma substituição, as necessidades alternativas de aprender sobre os atendimentos que são ativos no valor-limite. Quando um gatekeeper envia um RCF para novo registro, ele também envia uma IRQ para obter uma lista de todas as chamadas no ponto final. É importante se certificar de que o IRQ não atinja o ponto final antes da RCF.
- Os gatekeepers em um cluster permitem um fechamento, embora haja chamadas ativas,

desde que tenha um gatekeeper alternativo definido para todas as zonas nas quais haja chamadas ativas. Se qualquer zona não tem uma chamada ativa e nenhum gatekeeper alternativo definida, o porteiro recusa a parada programada.

- Os gatekeeperes alternativos aceitam todas as solicitações de desconexão (DRQ) para atendimentos que não estavam cientes de e passam a informação apropriada aos server do Authentication, Authorization, and Accounting (AAA) e do protocolo cisco gatekeeper transaction message (GKTMP). Isto acontece quando esse valor-limite se move para o gatekeeper alternativo quando houver umas chamadas ativa. Além disso, é possível que sejam enviadas mensagens do IRR que contenham informações de chamada das chamadas anteriormente desconhecidas. Para esses IRRs, os registros de chamada são construídos e a largura de banda é alocada de acordo.
- O gatekeeper cria uma mensagem exclusiva de Indicação de anúncio para cada gatekeeper alternativo. Se um gatekeeper alternativo receber uma mensagem que contenha um identificador de gatekeeper que ele não reconheça (o que pode acontecer se ele for o gatekeeper alternativo de uma zona), mas nenhuma outra, essas informações serão ignoradas. Contudo, o gatekeeper alternativo detecta erros na configuração das substituições examinando aquelas mensagens e relata aqueles erros ao usuário.
- A potência verdadeira do GUP é realizada quando os endereços são resolved para uma zona remota. Em vez da necessidade para que a zona remota envie LRQ (em ordem ou explosão) a todos os porteiros, assim aumentando a carga adicional de transmissão de mensagem em links de área ampla, precisa agora de enviar esta pergunta a apenas um dos porteiros no conjunto. Acoplado com o conjunto novo CLI remoto da zona, pode arredondamento robin entre os porteiros no conjunto e para não tentar enviar o LRQ a um outro porteiro no conjunto se recebe uma rejeição de qualquer.
- Se um gateway for movido para um gatekeeper alternativo, ele tentará sempre se registrar nesse gatekeeper, a não ser que você não emita nenhum gateway e, sim, um comando de gateway. Quando o gatekeeper principal do ponto final estiver de novo online, o ponto final não se registrará nele novamente a menos que o ponto final perca a comunicação com o gatekeeper alternativo. Continua a utilizar o gatekeeper alternativo para as informações de roteamento de chamada.

Comandos debug e show para clusters de gatekeeper

Debuga demonstram abaixo como os porteiros podem se juntar a um conjunto e como compartilham da informação sobre seus valores-limite. Os comandos show são usados para mostrar como monitorar o cluster. Debuga usado são debugam o asn1 do gup do porteiro e debugam o asn1 h225. Tem como base a topologia e a configuração mencionadas anteriormente.

Depurações de "gkb-1" quando foi o primeiro a se unir ao cluster

```
Mar  1 08:15:08.348: gk_gup_listen(): listening port = 11007 !--- Opens a TCP port (here it is
11007) to listen to GUP messages. Mar 1 08:15:08.348: gk_gup_listen(): listening fd = 0 Mar 1
08:15:38.351: H225 NONSTD OUTGOING PDU ::= value GRQnonStandardInfo ::= !--- The non-standard
data that is in the GRQ. { gupAddress { ip 'AC100D29'H !--- Listening IP address 172.16.13.41.
port 11007 !--- Listening TCP port 11007. } } Mar 1 08:15:38.351: H225 NONSTD OUTGOING ENCODE
BUFFER::= 40 AC100D29 2AFF Mar 1 08:15:38.351: Mar 1 08:15:38.351: RAS OUTGOING PDU ::= value
RasMessage ::= gatekeeperRequest : !--- GRQ with the non-standard is sent out. { requestSeqNum
59 protocolIdentifier { 0 0 8 2250 0 3 } nonStandardData { nonStandardIdentifier h221NonStandard
: { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '40AC100D292AFF'H } rasAddress
ipAddress : { ip 'AC100D29'H port 1719 } endpointType { vendor { vendor { t35CountryCode 181
```

t35Extension 0 manufacturerCode 18 } } mc FALSE undefinedNode FALSE } } Mar 1 08:15:38.359: RAS OUTGOING ENCODE BUFFER::= 01 00003A06 0008914A 000340B5 00001207 40AC100D 292AFF00 AC100D29 06B72000 B5000012 00 Mar 1 08:15:38.359:

Depurações de “gkb-2” quando ele se uniu ao cluster depois de “gkb-1”

Mar 1 08:16:38.878: **gk_gup_listen(): listening port = 11006** *!--- Opens a TCP port (here it is 11006) to listen to GUP messages.* Mar 1 08:16:38.878: gk_gup_listen(): listening fd = 0 Mar 1 08:17:08.385: RAS INCOMING ENCODE BUFFER::= 01 00003D06 0008914A 000340B5 00001207 40AC100D 292AFF00 AC100D29 06B72000 B5000012 00 Mar 1 08:17:08.385: Mar 1 08:17:08.385: RAS INCOMING PDU ::= value RasMessage ::= **gatekeeperRequest** : *!--- GRQ message is received from gkb-1 gatekeeper with non-standard information.* { requestSeqNum 62 protocolIdentifier { 0 0 8 2250 0 3 } nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '40AC100D292AFF'H } rasAddress ipAddress : { ip 'AC100D29'H *!--- RAS IP address 172.16.13.41 used gkb-1.* port 1719 *!--- RAS TCP port used by gkb-1.* } endpointType { vendor { vendor { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } } mc FALSE undefinedNode FALSE } } Mar 1 08:17:08.393: H225 NONSTD INCOMING ENCODE BUFFER::= 40 AC100D29 2AFF Mar 1 08:17:08.393: Mar 1 08:17:08.393: H225 NONSTD INCOMING PDU ::= value **GRQnonStandardInfo** ::= *!--- gkb-2 extracts the non-standard data from the GRQ.* **{ gupAddress { ip 'AC100D29'H** *!--- GUP IP address 172.16.13.41 used by gkb-1.* **port 11007** *!--- GUP TCP port 11007 used by gkb-1.* **} }** Mar 1 08:17:08.393: **check_connection: checking connection to 172.16.13.41:11007** Mar 1 08:17:08.393: **gk_gup_connect(): initiating connection** Mar 1 08:17:08.393: **gup_connect: connecting to 172.16.13.41:11007** *!--- A GUP connection is established, and updates follow.* Mar 1 08:17:08.393: gup_connect, fd = 1 Mar 1 08:17:08.401: **GUP OUTGOING PDU** ::= value **GUP_Information** ::= *!--- GUP announcement is sent to alternate GK gkb-1.* { protocolIdentifier { 1 2 840 113548 10 0 0 2 } message **announcementIndication** : { announcementInterval 30 endpointCapacity 100000 callCapacity 100000 hostName '676B622D32'H **percentMemory 8** *!--- Below is information about the status of gkb-2.* **percentCPU 0 currentCalls 0 currentEndpoints 0 zoneInformation { { gatekeeperIdentifier {"gkb-2"} altGKIdentifier {"gkb-1"} totalBandwidth 0 interzoneBandwidth 0 remoteBandwidth 0 } }** } } Mar 1 08:17:08.405: GUP OUTGOING ENCODE BUFFER::= 00 0A2A8648 86F70C0A 00000220 001E8001 86A08001 86A00467 6B622D32 10000000 00014200 0067006B 0062002D 00320800 67006B00 62002D00 31000000 000000 Mar 1 08:17:08.409: Mar 1 08:17:08.409: **Sending GUP ANNOUNCEMENT INDICATION to 172.16.13.41** Mar 1 08:17:08.413: GUP INCOMING ENCODE BUFFER::= 00 0A2A8648 86F70C0A 00000220 001E8001 86A08001 86A00467 6B622D31 32000000 00014200 0067006B 0062002D 00310800 67006B00 62002D00 32000000 000000 Mar 1 08:17:08.413: Mar 1 08:17:08.413: **GUP INCOMING PDU ::=** value **GUP_Information** ::= *!--- GUP announcement is received from alternate GK gkb-1.* { protocolIdentifier { 1 2 840 113548 10 0 0 2 } message announcementIndication : { announcementInterval 30 endpointCapacity 100000 callCapacity 100000 hostName '676B622D31'H **percentMemory 25** *!--- Below is information about the status of gkb-1.* **percentCPU 0 currentCalls 0 currentEndpoints 0 zoneInformation { { gatekeeperIdentifier {"gkb-1"} altGKIdentifier {"gkb-2"} totalBandwidth 0 interzoneBandwidth 0 remoteBandwidth 0** } } } } Mar 1 08:17:08.421: **Received GUP ANNOUNCEMENT INDICATION from 172.16.13.41**

Com o comando show gatekeeper endpoint, não há nenhum valor-limite registrado. A saída é a seguinte:

```
gkb-1#show gatekeeper endpoints GATEKEEPER ENDPOINT REGISTRATION
================================ CallSignalAddr Port RASSignalAddr Port Zone Name Type Flags ---
------------ ----- -------------- ----- --------- ---- ----- Total number of active
registrations = 0 gkb-2# show gatekeeper endpoints GATEKEEPER ENDPOINT REGISTRATION
================================ CallSignalAddr Port RASSignalAddr Port Zone Name Type Flags ---
------------ ----- -------------- ----- --------- ---- ----- Total number of active
registrations = 0 gkb-2#
```

Depura quando um ponto final registra com um dos gatekeepers no cluster

Esta depuração foi obtida do gatekeeper "gkb-1". O ponto final é "gwb-1" registrando no gatekeeper "gkb-1" com debug h225 ans1 e debug ras ativados (ON).

Mar 1 08:22:47.396: RAS INCOMING ENCODE BUFFER::= 00 A00AAD06
0008914A 000300AC 100D17E0 7D088001 3C050401 00205002 00006700 6B006200
2D003101 40040067 00770062 002D0031

```
Mar  1 08:22:47.396:
Mar  1 08:22:47.396: RAS INCOMING PDU ::= value RasMessage ::= gatekeeperRequest : !--- GRQ is
received from "gwb-1" gateway. { requestSeqNum 2734 protocolIdentifier { 0 0 8 2250 0 3 }
rasAddress ipAddress : { ip 'AC100D17'H !--- gwb-1 IP address (172.16.13.23). port 57469 !---
gwb-1 TCP port 57469. } endpointType { gateway { protocol { voice : { supportedPrefixes { {
prefix e164 : "2#" } } } } } } mc FALSE undefinedNode FALSE } gatekeeperIdentifier {"gkb-1"}
endpointAlias { h323-ID : {"gwb-1"} } } Mar 1 08:22:47.404: RAS OUTGOING PDU ::= value
RasMessage ::= gatekeeperConfirm : !--- GCF is sent back with alternate gatekeepers included. {
requestSeqNum 2734 protocolIdentifier { 0 0 8 2250 0 3 } gatekeeperIdentifier {"gkb-1"}
rasAddress ipAddress : { ip 'AC100D29'H !--- Gatekeeper gkb-1 IP address (172.16.13.41). port
1719 } alternateGatekeeper !--- List of alternate gatekeepers, here is "gkb-2" only. { {
rasAddress ipAddress : { ip 'AC100D10'H !--- Alternate gatekeeper gkb-2 IP address
(172.16.13.16) port 1719 } gatekeeperIdentifier {"gkb-2"} needToRegister TRUE priority 0 } } }
Mar 1 08:22:47.412: RAS OUTGOING ENCODE BUFFER::= 06 800AAD06 0008914A 00030800 67006B00
62002D00 3100AC10 0D2906B7 0D001401 40AC100D 1006B708 0067006B 0062002D 003280 Mar 1
08:22:47.412: Mar 1 08:22:47.432: RAS INCOMING ENCODE BUFFER::= 0E C00AAE06 0008914A 00038001
00AC100D 1706B801 00AC100D 17E07D08 80013C05 04010020 50000140 04006700 77006200 2D003108
0067006B 0062002D 003100B5 00001212 8B000200 3B010001 000180 Mar 1 08:22:47.432: Mar 1
08:22:47.436: RAS INCOMING PDU ::= value RasMessage ::= registrationRequest : !--- RRQ is
received from "gwb-1" gateway. { requestSeqNum 2735 protocolIdentifier { 0 0 8 2250 0 3 }
discoveryComplete TRUE callSignalAddress { ipAddress : { ip 'AC100D17'H !--- Gateway gwb-1 IP
address (172.16.13.23). port 1720 } } rasAddress { ipAddress : { ip 'AC100D17'H port 57469 } }
terminalType { gateway { protocol { voice : { supportedPrefixes { { prefix e164 : "2#" } } } } }
mc FALSE undefinedNode FALSE } terminalAlias { h323-ID : {"gwb-1"} } gatekeeperIdentifier {"gkb-
1"} endpointVendor { vendor { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } }
timeToLive 60 keepAlive FALSE willSupplyUUIEs FALSE maintainConnection TRUE } Mar 1
08:22:47.448: GUP OUTGOING PDU ::= value GUP_Information ::= !--- A GUP registration indicates a
message is sent to "gkb-2" to inform it !--- about the new registered endpoint. {
protocolIdentifier { 1 2 840 113548 10 0 0 2 } message registrationIndication : { version 3
callSignalAddress { ipAddress : { ip 'AC100D17'H !--- Gateway gwb-1 IP address (172.16.13.23).
port 1720 } } rasAddress { ipAddress : { ip 'AC100D17'H !--- Gateway gwb-1 IP address
(172.16.13.23). port 57469 } } terminalType { vendor { vendor { t35CountryCode 181 t35Extension
0 manufacturerCode 18 } } gateway { protocol { voice : { supportedPrefixes { { prefix e164 :
"2#*" } } } } } mc FALSE undefinedNode FALSE } terminalAlias { h323-ID : {"gwb-1"} !--- Name/ID
of the new endpoint which has just registered. } gatekeeperIdentifier {"gkb-1"} !--- Name/ID of
the gatekeeper which the new endpoint(gwb-1) has registered to. resourceIndicator {
almostOutOfResources FALSE } } } Mar 1 08:22:47.460: GUP OUTGOING ENCODE BUFFER::= 00 0A2A8648
86F70C0A 00000232 020100AC 100D1706 B80100AC 100D17E0 7D2800B5 00001240 013C0505 01004050
10000140 04006700 77006200 2D003108 0067006B 0062002D 003100 Mar 1 08:22:47.464: Mar 1
08:22:47.464: Sending GUP REGISTRATION INDICATION to 172.16.13.16 Mar 1 08:22:47.464: RAS
OUTGOING PDU ::= value RasMessage ::= registrationConfirm : !--- RCF is sent back to "gwb-1"
gateway. { requestSeqNum 2735 protocolIdentifier { 0 0 8 2250 0 3 } callSignalAddress { }
terminalAlias { h323-ID : {"gwb-1"} } gatekeeperIdentifier {"gkb-1"} endpointIdentifier
{"61809DB800000001"} alternateGatekeeper { { rasAddress ipAddress : { ip 'AC100D10'H port 1719 }
gatekeeperIdentifier {"gkb-2"} needToRegister TRUE priority 0 } } timeToLive 60 willRespondToIRR
FALSE maintainConnection TRUE } Mar 1 08:22:47.472: RAS OUTGOING ENCODE BUFFER::= 12 C00AAE06
0008914A 00030001 40040067 00770062 002D0031 08006700 6B006200 2D00311E 00360031 00380030
00390044 00420038 00300030 00300030 00300030 00300031 0F8A1401 40AC100D 1006B708 0067006B
0062002D 00328002 003B0100 0180 Mar 1 08:22:47.472:
```

A saída acima contém afinal os gateways output [comando show gatekeeper endpoint](#) registra-se no conjunto. O acima debuga acontece para cada registro de ponto final. Afinal três gateways no conjunto registram-se, o **comando show gatekeeper endpoint em** ambos os porteiros são como segue:

```
gkb-1#show gatekeeper endpoints GATEKEEPER ENDPOINT REGISTRATION
================================ CallSignalAddr Port RASSignalAddr Port Zone Name Type Flags ---
------------ ----- -------------- ----- --------- ---- ----- 172.16.13.23 1720 172.16.13.23
57469 gkb-1 VOIP-GW H323-ID: gwb-1 172.16.13.26 1720 172.16.13.26 49801 gkb-1 VOIP-GW H323-ID:
gwb-2 172.16.13.42 1720 172.16.13.42 57216 gkb-1 VOIP-GW A !--- A flag set. H323-ID: gwb-3 Total
number of active registrations = 3 gkb-2# show gatekeeper endpoints GATEKEEPER ENDPOINT
REGISTRATION ================================ CallSignalAddr Port RASSignalAddr Port Zone Name
Type Flags -------------- ----- -------------- ----- --------- ---- ----- 172.16.13.23 1720
172.16.13.23 57469 gkb-2 VOIP-GW A !--- A flag set. H323-ID: gwb-1 172.16.13.26 1720
```

172.16.13.26 49801 gkb-2 VOIP-GW A *!--- A flag set.* H323-ID: gwb-2 172.16.13.42 1720 172.16.13.42 57216 gkb-2 VOIP-GW H323-ID: gwb-3 Total number of active registrations = 3 *!--- The "A" under the flag field means that the gatekeeper is an alternate one !--- for this endpoint.*

Depurações para quando um ponto final com chamada ativa se mover para o gatekeeper alternativo

Estes são debugam de um porteiro que comece quando o atendimento é pedido e vá até que esteja desligado. Alguma do desnecessário debuga mensagens foi omitida. Estes debugam são do porteiro de "gkb-1". O atendimento foi colocado por gwa-1 registrado a "gka-1" a um outro gateway (gwb-1) no cluster de zona remota. Debuga demonstram como um fluxo da chamada ativa está seguido do gatekeeper principal ao gatekeeper alternativo enquanto o preliminar vai para baixo.

```
Mar  2 23:59:26.714:  RecvUDP_IPSockData  successfully rcvd message of length 84
from 172.16.13.35:1719
Mar  2 23:59:26.714: RAS INCOMING ENCODE BUFFER::= 4A 80080801 01806986
40B50000 122C8286 B01100C8 C66C7D1
6 8011CC80 0D882828 5B8DF601 80140204 8073B85A 5C564004 00670077
0061002D 003100AC 100D2306 B70B800D 01400
400 67006B00 61002D00 310180
Mar  2 23:59:26.714:
Mar  2 23:59:26.714: RAS INCOMING PDU ::=

value RasMessage ::= locationRequest : !--- LRQ is received from "gka-1" gatekeeper from domain
A. { requestSeqNum 2057 destinationInfo { e164 : "3653" !--- E164 number to be resolved by the
this gatekeeper. } nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode
181 t35Extension 0 manufacturerCode 18 } data '8286B01100C8C66C7D168011CC800D8828285B8D...'H }
replyAddress ipAddress : { ip 'AC100D23'H port 1719 } sourceInfo { h323-ID : {"gka-1"} }
canMapAlias TRUE } Mar 2 23:59:26.722: LRQ (seq# 2057) rcvd Mar 2 23:59:26.722: H225 NONSTD
INCOMING ENCODE BUFFER::= 82 86B01100 C8C66C7D 168011CC 800D8828 285B8DF6 01801402 048073B8
5A5C5640 04006700 77006100 2D0031 Mar 2 23:59:26.722: Mar 2 23:59:26.722: H225 NONSTD INCOMING
PDU ::= !--- LRQ nonStandardInfo decoded output. value LRQnonStandardInfo ::= { ttl 6 nonstd-
callIdentifier { guid 'C8C66C7D168011CC800D8828285B8DF6'H } callingOctet3a 128 gatewaySrcInfo {
e164 : "4085272923", h323-ID : {"gwa-1"} } } parse_lrq_nonstd: LRQ Nonstd decode succeeded,
remlen = 84 Mar 2 23:59:26.726: H225 NONSTD OUTGOING PDU ::= !--- LCF nonStandardInfo reply back
to the LRQ nonStandardInfor. value LCFnonStandardInfo ::= { termAlias { h323-ID : {"gwb-1"} }
gkID {"gkb-1"} gateways { { gwType voip : NULL gwAlias { h323-ID : {"gwb-1"} !--- Gateway gwb-1
is the resolved terminating gateway sent back for the request. } sigAddress { ip 'AC100D17'H !--
- Gateway gwb-1 IP address (172.16.13.23). port 1720 } resources { maxDSPs 0 inUseDSPs 0
maxBChannels 0 inUseBChannels 0 activeCalls 0 bandwidth 0 inuseBandwidth 0 } } } } Mar 2
23:59:26.734: H225 NONSTD OUTGOING ENCODE BUFFER::= 00 01400400 67007700 62002D00 31080067
006B0062 002D0031 01100140 04006700 77006200 2D003100 AC100D17 06B80000 00000000 00000000 Mar 2
23:59:26.734: Mar 2 23:59:26.734: RAS OUTGOING PDU ::= value RasMessage ::= locationConfirm : !-
-- LCF is sent back with "gwb-1" as the resolved terminating gateway. { requestSeqNum 2057
callSignalAddress ipAddress : { ip 'AC100D17'H !--- Resolved terminating gateway gwb-1 IP
address (172.16.13.23). port 1720 } rasAddress ipAddress : { ip 'AC100D17'H !--- Resolved
terminating gateway gwb-1 IP address (172.16.13.23). port 51874 } nonStandardData {
nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18
} data '0001400400670077006200 2D0031080067006B00...'H } destinationType { gateway { protocol {
voice : { supportedPrefixes { } } } } } mc FALSE undefinedNode FALSE } } Mar 2 23:59:26.742: RAS
OUTGOING ENCODE BUFFER::= 4F 080800AC 100D1706 B800AC10 0D17CAA2 40B50000 1239000 1 40040067
00770062 002D0031 08006700 6B006200 2D003101 10014004 00670077 0062002D 003100AC 100D1706 B8000
000 00000000 00000010 40080880 013C0501 0000 Mar 2 23:59:26.746: Mar 2 23:59:26.746:
IPSOCK_RAS_sendto: msg length 91 from 172.16.13.41:1719 to 172.16.13.35: 1719 Mar 2
23:59:26.746: RASLib::RASSendLCF: LCF (seq# 2057) sent to 172.16.13.35 Mar 2 23:59:26.798:
RecvUDP_IPSockData successfully rcvd message of length 129 from 172.16.13.23:51874 Mar 2
23:59:26.798: RAS INCOMING ENCODE BUFFER::= 27 98172700 F0003600 31003900 36003200 39003600
3800300 0 30003000 30003000 30003000 31010180 69860204 8073B85A 5C564004 00670077 0061002D
003100AC 100D0F2A FA400 500 000E40B5 00001207 80000008 800180C8 C66C7D16 8011CC80 0C882828
5B8DF645 60200180 1100C8C6 6C7D1680 11C C800D 8828285B 8DF60100 Mar 2 23:59:26.802: Mar 2
```

```
23:59:26.802: RAS INCOMING PDU ::= value RasMessage ::= admissionRequest : !--- "gwb-1" sent
answerCall ARQ. { requestSeqNum 5928 callType pointToPoint : NULL callModel direct : NULL
endpointIdentifier {"6196296800000001"} destinationInfo { e164 : "3653" !--- E164 number the
caller is trying to reach. } srcInfo { e164 : "4085272923", !--- Caller information. h323-ID :
{"gwa-1"} } srcCallSignalAddress ipAddress : { ip 'AC100D0F'H !--- Originating gateway (gwa-1)
IP address and port. port 11002 } bandWidth 1280 callReferenceValue 14 !--- Remember call
reference, since it is used when the call !--- is disconnected when sending the DRQ.
nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0
manufacturerCode 18 } data '80000008800180'H } conferenceID 'C8C66C7D168011CC800C8828285B8DF6'H
activeMC FALSE answerCall TRUE canMapAlias TRUE callIdentifier { guid
'C8C66C7D168011CC800D8828285B8DF6'H } willSupplyUUIEs FALSE } Mar 2 23:59:26.810: ARQ (seq#
5928) rcvd Mar 2 23:59:26.810: H225 NONSTD INCOMING ENCODE BUFFER::= 80 00000880 0180 Mar 2
23:59:26.810: Mar 2 23:59:26.810: H225 NONSTD INCOMING PDU ::= value ARQnonStandardInfo ::= {
sourceAlias { } sourceExtAlias { } callingOctet3a 128 } parse_arq_nonstd: ARQ Nonstd decode
succeeded, remlen = 129 Mar 2 23:59:26.814: RAS OUTGOING PDU ::= value RasMessage ::=
admissionConfirm : !--- ACF is sent back to "gwb-1". { requestSeqNum 5928 bandWidth 1280
callModel direct : NULL destCallSignalAddress ipAddress : { ip 'AC100D17'H !--- gwb-1 IP address
(172.16.13.23). 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 2 23:59:26.818: RAS OUTGOING ENCODE BUFFER::=
2B 00172740 050000AC 100D1706 B800EF1A 00C00100 020000 Mar 2 23:59:26.818: Mar 2 23:59:26.818:
IPSOCK_RAS_sendto: msg length 24 from 172.16.13.41:1719 to 172.16.13.23: 51874 Mar 2
23:59:26.822: RASLib::RASSendACF: ACF (seq# 5928) sent to 172.16.13.23 Mar 2 23:59:36.046: GUP
OUTGOING PDU ::= value GUP_Information ::= !--- GUP update is sent out and it contains the
inforomation !--- about the last call that is still active. { protocolIdentifier { 1 2 840
113548 10 0 0 2 } message announcementIndication : { announcementInterval 30 endpointCapacity
46142 callCapacity 68793 hostName '676B622D31'H percentMemory 25 percentCPU 0 currentCalls 1
currentEndpoints 2 zoneInformation { { gatekeeperIdentifier {"gkb-1"} altGKIdentifier {"gkb-2"}
totalBandwidth 1280 !--- 1280 is 128 Kbps of total bandwidth used for the zone.
interzoneBandwidth 1280 remoteBandwidth 1280 } } } } Mar 2 23:59:36.050: GUP OUTGOING ENCODE
BUFFER::= 00 0A2A8648 86F70C0A 00000220 001E40B4 3E80010C B904676 B 622D3132 00010002 01420000
67006B00 62002D00 31080067 006B0062 002D0032 40050040 05004005 00 Mar 2 23:59:36.054: Mar 2
23:59:36.054: Sending GUP ANNOUNCEMENT INDICATION to 172.16.13.16
```

Nota: Neste momento "gkb-1" é parada programada. Isto é permitido (mesmo se tem uma chamada ativa) porque há um gatekeeper alternativo para essa zona.

Os mensagens URQ são enviados a todos os valores-limite registrados com o "gkb-1". Estes valores-limite são gateways de "gwb-1" e de "gwb-2". Estes gateways confirmam o URQ enviando para trás UCF. Também, gkb-1 envia um mensagem de indicação sem registro GUP ao gatekeeper alternativo do conjunto e então fecha a conexão GUP.

```
Mar  2 23:59:55.914: RAS OUTGOING PDU ::= value RasMessage ::= unregistrationRequest : {
requestSeqNum 79 callSignalAddress { ipAddress : { ip 'AC100D17'H !--- UnregistrationRequest
(URQ) sent to gwb-1 (172.16.13.23). port 1720 } } } Mar 2 23:59:55.914: RAS OUTGOING ENCODE
BUFFER::= 18 00004E01 00AC100D 1706B8 Mar 2 23:59:55.914: Mar 2 23:59:55.914: IPSOCK_RAS_sendto:
msg length 12 from 172.16.13.41:1719 to 172.16.13.23: 51874 Mar 2 23:59:55.914:
RASLib::RASSendURQ: URQ (seq# 79) sent to 172.16.13.23 Mar 2 23:59:55.918: RAS OUTGOING PDU ::=
value RasMessage ::= unregistrationRequest : { requestSeqNum 80 callSignalAddress { ipAddress :
{ ip 'AC100D1A'H !--- URQ sent to gwb-2 (172.16.13.26). port 1720 } } } Mar 2 23:59:55.918: RAS
OUTGOING ENCODE BUFFER::= 18 00004F01 00AC100D 1A06B8 Mar 2 23:59:55.918: Mar 2 23:59:55.918:
IPSOCK_RAS_sendto: msg length 12 from 172.16.13.41:1719 to 172.16.13.26: 50041 Mar 2
23:59:55.918: RASLib::RASSendURQ: URQ (seq# 80) sent to 172.16.13.26 Mar 2 23:59:55.922:
RecvUDP_IPSockData successfully rcvd message of length 3 from 172.16.13.23:51874 Mar 2
23:59:55.922: RAS INCOMING ENCODE BUFFER::= 1C 004E Mar 2 23:59:55.922: Mar 2 23:59:55.922: RAS
INCOMING PDU ::= value RasMessage ::= unregistrationConfirm : { requestSeqNum 79 } Mar 2
23:59:55.922: UCF (seq# 79) rcvd Mar 2 23:59:55.926: RecvUDP_IPSockData successfully rcvd
message of length 3 from 172.16.13.26:50041 Mar 2 23:59:55.926: RAS INCOMING ENCODE BUFFER::= 1C
004F Mar 2 23:59:55.926: Mar 2 23:59:55.926: RAS INCOMING PDU ::= value RasMessage ::=
unregistrationConfirm : { requestSeqNum 80 } Mar 2 23:59:55.926: UCF (seq# 80) rcvd Mar 3
00:00:01.922: GUP OUTGOING PDU ::= value GUP_Information ::= { protocolIdentifier { 1 2 840
113548 10 0 0 2 } message unregistrationIndication : { reason explicitUnregister : NULL
callSignalAddress { ipAddress : { ip 'AC100D17'H !--- GUP UnregistrationIndication sent to
```

alternate gatekeeper !--- gkb-2 (172.16.13.16) in the cluster. port 1720 } } } } Mar 3 00:00:01.922: GUP OUTGOING ENCODE BUFFER::= 00 0A2A8648 86F70C0A 00000238 000100AC 100D1706 B8 Mar 3 00:00:01.926: Mar 3 00:00:01.926: **Sending GUP UNREGISTRATION INDICATION to 172.16.13.16** Mar 3 00:00:01.934: **gk_gup_close_connection(): closing connection to 172.16.13.16** Mar 3 00:00:01.934: **gk_gup_close_listen(): closing listen**

Está aqui debugar de "gkb-2". Debuga que mostra que o registro do valor-limite movido "gwb-1" e "gwb-2" está omitido, desde que olham como o registro normal. A finalidade aqui é mostrar a aceitação do DRQ da chamada ativa em "gwb-1" quando é movida para "gkb-2".

```
Mar  3 00:00:24.307:  RecvUDP_IPSockData  successfully rcvd message of length 77
from 172.16.13.23:51874
Mar  3 00:00:24.307: RAS INCOMING ENCODE BUFFER::= 3E 172C1E00 36003100
38003400 44004300 34004300 30003000 30003000 3
0003000 300033C8 C66C7D16 8011CC80 0C882828 5B8DF600 0E21A100 1100C8C6
6C7D1680 11CC800D 8828285B 8DF60180
Mar  3 00:00:24.311:
Mar  3 00:00:24.311: RAS INCOMING PDU ::=
```

value RasMessage ::= **disengageRequest** : *!--- DRQ is received with call reference 14 and normal clearing !--- disconnect cause code. !--- This information is passed to the accounting server and the GKTMP !--- server if configured.* { **requestSeqNum 5933** endpointIdentifier {"6184DC4C00000003"} conferenceID 'C8C66C7D168011CC800C8828285B8DF6'H **callReferenceValue 14 disengageReason normalDrop : NULL** callIdentifier { guid 'C8C66C7D168011CC800D8828285B8DF6'H } answeredCall TRUE } Mar 3 00:00:24.311: **DRQ (seq# 5933) rcvd** Mar 3 00:00:24.315: RAS OUTGOING PDU ::= value RasMessage ::= **disengageConfirm** : *!--- DCF is sent to "gwb-1".* { **requestSeqNum 5933** } Mar 3 00:00:24.315: RAS OUTGOING ENCODE BUFFER::= 40 172C Mar 3 00:00:24.315: Mar 3 00:00:24.315: IPSOCK_RAS_sendto: msg length 3 from 172.16.13.16:1719 to 172.16.13.23: 51874 Mar 3 00:00:24.315: RASLib::RASSendDCF: **DCF (seq# 5933) sent to 172.16.13.23** gkb-2#

Failover de gateway Cisco para gatekeeper alternativo

Àrevelia, os Cisco gateway enviam a um RRQ de pouco peso cada 45 segundos. Caso que o porteiro não enviou nenhum URQ ao gateway (devido a uma questão de roteamento quebrada, por exemplo), as tentativas do gateway (em cima de não ouvir um RCF ou um RRJ para seu RRQ de pouco peso) duas vezes com cinco segundos entre cada um. Se a terceira tentativa falha, considera imediatamente o porteiro como mortos e registra-se com o gatekeeper alternativo que usa o RRQ. Em uma encenação onde o gateway comece o processo de registro inicial com o porteiro, manda o GRQ para encontrar o endereço IP de Um ou Mais Servidores Cisco ICM NT do porteiro. Se há uma resposta GCF para trás, o gateway envia o RRQ ao gatekeeper principal especificado. Se por qualquer razão o porteiro rejeita a requisição de registro, o gateway não tenta contactar seu gatekeeper alternativo. Começa este processo (GRQ, GCF e RRQ) sobre outra vez com o gatekeeper principal.

O gateway contacta somente o gatekeeper alternativo quando a Conectividade ao gatekeeper principal é perdida e não há nenhuma resposta para trás. Se o gatekeeper principal não responde de volta ao mensagem de GRQ quando o gateway mandar primeiramente para descobrir o porteiro, a seguir depois que três falhas de tentativa (aproximadamente cinco minutos pela tentativa), o gateway contactam o gatekeeper alternativo. Em uma situação aonde o gatekeeper principal vá para baixo depois que o gateway se registrou com ele, o gateway perde mensagens do Keepalives do gatekeeper principal. Após ter faltado três mensagens de keepalive consecutivos, o gateway declara o gatekeeper principal como para baixo, e começa o processo de registro outra vez.

Soluções de problemas com pontos finais alternativos

Um ponto final de chamada pode se recuperar de uma falha de configuração de chamada

enviando uma mensagem de configuração para um dos pontos finais alternativos. A chamada pode falhar por várias razões: o gateway está inoperante e o gatekeeper não está ciente disso no momento de enviar o ACF ou o LCF; não há nenhum recurso no gateway e ele não relatou isso ao gatekeeper; a chamada falha em razão de uma configuração incorreta no ponto final principal, entre outros fatores.

Nota: O ponto final de origem tenta somente contactar os gatekeeperes alternativos se o atendimento falha antes da fase alerta (alerta ou progresso). Se os atendimentos falham devido ao usuário ocupado ou à sem resposta, o ponto final de origem não tenta nenhuma outra substituições.

O porteiro aprende sobre aquela a substituição para um determinado valor-limite pela configuração manual usando o ponto final do comando CLI do gatekeeper ALT-EP ou de todos os mensagens RAS recebidos. Cisco apoia um máximo de 20 substituições para cada valor-limite, não importa como o porteiro as aprende.

Os problemas que você deve analisar são:

- Se o porteiro tem o ponto final alternativo correto como desejado.
- Se o porteiro inclui os pontos finais alternados em seus LCF ou mensagens de ACF RAS.
- Se o OGW tenta contactar as substituições caso que o ponto final de destino principal falha.

Para mostrar como pesquisar defeitos estas edições, a mesma topologia que é usado acima com esta mudança na configuração de gatekeeper de "gkb-1" para incluir dois gateways alternativos: "gwb-3" e "gwb-1" para o gateway "gwb-2". Está aqui a configuração do porteiro de "gkb-1":

```
!
gatekeeper
 zone local gkb-1 domainB.com 172.16.13.41
 zone remote gka-1 domainA.com 172.16.13.35 1719
 zone cluster local gkb gkb-1
  element gkb-2 172.16.13.16 1719
 !
 gw-type-prefix 2#* default-technology
 bandwidth total zone gkb-1 512
 bandwidth session zone gkb-1 512
 no shutdown
endpoint alt-ep h323id gwb-2 172.16.13.42 !--- 172.16.13.42 is gwb-3. endpoint alt-ep h323id
gwb-2 172.16.13.23 !--- 172.16.13.23 is gwb-1. !
```

Verifique que o porteiro tem os pontos finais alternativo correto

Para verificar se o gatekeeper tem os pontos finais alternativos corretos, use o comando show gatekeeper endpoints alternates.

```
gkb-1#show gatekeeper endpoints alternates GATEKEEPER ENDPOINT REGISTRATION
================================= CallSignalAddr Port RASSignalAddr Port Zone Name Type Flags -
------------------ ------ -------------------- ------- --------------- --------- --------
172.16.13.23 1720 172.16.13.23 54670 gkb-1 VOIP-GW H323-ID: gwb-1 172.16.13.26 1720 172.16.13.26
57233 gkb-1 VOIP-GW H323-ID: gwb-2 ALT_EP: 172.16.13.42 <1720> 172.16.13.23 <1720> !--- This
shows the information about all collected endpoints. 172.16.13.42 1720 172.16.13.42 58430 gkb-1
VOIP-GW A H323-ID: gwb-3 Total number of active registrations = 3 ALL CONFIGURED ALTERNATE
ENDPOINTS !--- Only manually configured. ===================================== Endpoint H323
Id RASSignalAddr Port ---------------------- -------------------- ------- gwb-2 172.16.13.42
1720 gwb-2 172.16.13.23 1720 gkb-1#
```

Verifique se o gatekeeper inclui pontos finais alternativos em suas mensagens de RAS de LCF ou ACF

Para ver se o porteiro envia o endereço IP de Um ou Mais Servidores Cisco ICM NT para pontos finais alternados, você pode girar sobre **debuga o asn1 h225** e olha o mensagem de ACF ou o LCF. Este é um exemplo de depuração tirado de gkb-1.

```
Mar  3 04:12:47.676: H225 NONSTD OUTGOING ENCODE BUFFER::= 00 01400400
67007700 62002D00 32080067 00
6B0062 002D0031 01100140 04006700 77006200 2D003200 AC100D1A 06B80000
00000000 00000000
Mar  3 04:12:47.676:
Mar  3 04:12:47.676: RAS OUTGOING PDU ::=

value RasMessage ::= locationConfirm : { requestSeqNum 2070 callSignalAddress ipAddress : { ip
'AC100D1A'H !--- This is IP address of main destination. port 1720 } rasAddress ipAddress : { ip
'AC100D1A'H port 50041 } nonStandardData { nonStandardIdentifier h221NonStandard : {
t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data
'0001400400670077006200 2D0032080067006B00...'H } destinationType { gateway { protocol { voice :
{ supportedPrefixes { } } } } mc FALSE undefinedNode FALSE } alternateEndpoints !--- Alternate
endpoints. { { callSignalAddress { ipAddress : { ip 'AC100D2A'H !--- This is the first alternate
IP address (172.16.13.42 gwb-3). port 1720 }, ipAddress : { ip 'AC100D17'H !--- This is the
second alternate IP address (172.16.13.23 gwb-1). port 1720 } } } } }
```

Verifique se o OGW tenta entrar em contato com alternativas, caso o principal ponto final de destino falhar

Esta seção mostra como o OGW reage quando recebe pontos finais alternativos na sua mensagem de ACF Neste exemplo, a chamada é feita para falhar quando ela tenta entrar em contato com o principal ponto final de terminação (gw). As depurações a ativar aqui são debug voip ccapi inout e debug h225 asn1.

A primeira coisa que você verá na depuração é a mensagem ccapi que mostra o trecho de telefonia de origem.

```
Mar  3 04:12:47.616: cc_api_call_setup_ind (vdbPtr=0x6264A60C,
callInfo={called=3653,called_oct3=0x8
0,calling=4085272923,calling_oct3=0x21,calling_oct3a=0x80,calling_xlated=false,subsc
riber_type_str=R egularLine,fdest=1,peer_tag=5336, prog_ind=0},callID=0x62155454) Mar 3
04:12:47.616: cc_api_call_setup_ind type 13 , prot 0 Mar 3 04:12:47.620:
cc_process_call_setup_ind (event=0x6231C454) Mar 3 04:12:47.620: >>>>CCAPI handed cid 51 with
tag 5336 to app "DEFAULT" Mar 3 04:12:47.620: sess_appl: ev(24=CC_EV_CALL_SETUP_IND), cid(51),
disp(0) Mar 3 04:12:47.620: sess_appl: ev(SSA_EV_CALL_SETUP_IND), cid(51), disp(0) Mar 3
04:12:47.620: ssaCallSetupInd Mar 3 04:12:47.620: ccCallSetContext (callID=0x33,
context=0x626EAC9C) Mar 3 04:12:47.620: ssaCallSetupInd cid(51), st(SSA_CS_MAPPING),oldst(0),
ev(24)ev->e.evCallSetupIn d.nCallInfo.finalDestFlag = 1 Mar 3 04:12:47.620: ssaCallSetupInd
finalDest cllng(4085272923), clled(3653) Mar 3 04:12:47.620: ssaCallSetupInd cid(51),
st(SSA_CS_CALL_SETTING),oldst(0), ev(24)dpMatchPeersMo reArg result= 0 Mar 3 04:12:47.620:
ssaSetupPeer cid(51) peer list: tag(3653) called number (3653) Mar 3 04:12:47.620: ssaSetupPeer
cid(51), destPat(3653), matched(4), prefix(), peer(62663E7C), peer ->encapType (2) Mar 3
04:12:47.620: ccCallProceeding (callID=0x33, prog_ind=0x0) Mar 3 04:12:47.620:
ccCallSetupRequest (Inbound call = 0x33, outbound peer =3653, dest=, params=0x62327730 mode=0,
*callID=0x62327A98, prog_ind = 0) Mar 3 04:12:47.624: ccCallSetupRequest numbering_type 0x80 Mar
3 04:12:47.624: ccCallSetupRequest encapType 2 clid_restrict_disable 1 null_orig_clg 0 clid_tra
nsparent 0 callingNumber 4085272923 Mar 3 04:12:47.624: dest pattern 3653, called 3653,
digit_strip 0 Mar 3 04:12:47.624: callingNumber=4085272923, calledNumber=3653, redirectNumber=
display_info= call ing_oct3a=80 Mar 3 04:12:47.624: accountNumber=, finalDestFlag=1,
guid=2d3a.ac33.16a4.11cc.8068.8828.285b.8df6 Mar 3 04:12:47.624: peer_tag=3653 Mar 3
04:12:47.624: ccIFCallSetupRequestPrivate: (vdbPtr=0x621B2360, dest=, callParams={called=3653
,called_oct3=0x80, calling=4085272923,calling_oct3=0x21, calling_xlated=false,
subscriber_type_str= RegularLine, fdest=1, voice_peer_tag=3653},mode=0x0) vdbPtr type = 1 !---
The OGW establishes the second leg. Mar 3 04:12:47.624: ccIFCallSetupRequestPrivate:
(vdbPtr=0x621B2360, dest=, callParams={called=3653 , called_oct3 0x80,
calling=4085272923,calling_oct3 0x21, calling_xlated=false, fdest=1, voice_pee r_tag=3653},
```

mode=0x0, xltrc=-5) Mar 3 04:12:47.624: ccSaveDialpeerTag (callID=0x33, dialpeer_tag=0xE45) Mar 3 04:12:47.624: ccCallSetContext (callID=0x34, context=0x626EB9A4) Mar 3 04:12:47.624: ccCallReportDigits (callID=0x33, enable=0x0) Mar 3 04:12:47.624: cc_api_call_report_digits_done (vdbPtr=0x6264A60C, callID=0x33, disp=0) Mar 3 04:12:47.624: sess_appl: ev(52=CC_EV_CALL_REPORT_DIGITS_DONE), cid(51), disp(0) Mar 3 04:12:47.624: cid(51)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 3 04:12:47.624: -cid2(52)st2(SSA_CS_CALL_SETTING)oldst2(SSA_CS_MAPPING) Mar 3 04:12:47.624: ssaReportDigitsDone cid(51) peer list: (empty) Mar 3 04:12:47.624: ssaReportDigitsDone callid=51 Reporting disabled. Mar 3 04:12:47.628: H225 NONSTD OUTGOING PDU ::= value ARQnonStandardInfo ::= { sourceAlias { } sourceExtAlias { } callingOctet3a 128 interfaceSpecificBillingId "ISDN-VOICE" } Mar 3 04:12:47.628: H225 NONSTD OUTGOING ENCODE BUFFER::= 80 000008A0 01800B12 4953444E 2D564F49 43 45 Mar 3 04:12:47.628: Mar 3 04:12:47.628: RAS OUTGOING PDU ::= value RasMessage ::= **admissionRequest** : *!--- ARQ is sent to the gatekeeper.* requestSeqNum 2210 callType pointToPoint : NULL callModel direct : NULL endpointIdentifier {"81206D2C00000001"} destinationInfo { e164 : "3653" } srcInfo { e164 : "4085272923", h323-ID : {"gwa-1"} } bandWidth 640 callReferenceValue 26 nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '80000008A001800B124953444E2D564F494345'H } conferenceID '2D3AAC3316A411CC80688828285B8DF6'H activeMC FALSE answerCall FALSE canMapAlias TRUE callIdentifier { guid '2D3AAC3316A411CC80698828285B8DF6'H } willSupplyUUIEs FALSE } Mar 3 04:12:47.636: RAS OUTGOING ENCODE BUFFER::= 27 8808A100 F0003800 31003200 30003600 44003200 4 3003000 30003000 30003000 30003000 31010180 69860204 8073B85A 5C564004 00670077 0061002D 00314002 80 001A40 B5000012 13800000 08A00180 0B124953 444E2D56 4F494345 2D3AAC33 16A411CC 80688828 285B8DF6 04E 02001 8011002D 3AAC3316 A411CC80 69882828 5B8DF601 00 Mar 3 04:12:47.640: Mar 3 04:12:47.656: RAS INCOMING ENCODE BUFFER::= 80 050008A1 2327 Mar 3 04:12:47.656: Mar 3 04:12:47.656: RAS INCOMING PDU ::= value RasMessage ::= requestInProgress : { requestSeqNum 2210 delay 9000 } Mar 3 04:12:47.704: RAS INCOMING ENCODE BUFFER::= 2B 0008A140 028000AC 100D1A06 B800EF1A 10C01201 1 0000200 AC100D2A 06B800AC 100D1706 B8010002 0000 Mar 3 04:12:47.704: Mar 3 04:12:47.704: RAS INCOMING PDU ::= value RasMessage ::= **admissionConfirm** : *!--- ACF is received.* { requestSeqNum 2210 bandWidth 640 callModel direct : NULL **destCallSignalAddress ipAddress** : *!--- Primary destination endpoint.* { **ip 'AC100D1A'H** port 1720 } irrFrequency 240 **alternateEndpoints** *!--- List of alternate endpoints.* **{ { callSignalAddress { ipAddress : { ip 'AC100D2A'H** *!--- 172.16.13.42.* **port 1720 }, ipAddress : { ip 'AC100D17'H** *!--- 172.16.13.23.* **port 1720 } } } }** willRespondToIRR FALSE uuiesRequested { setup FALSE callProceeding FALSE connect FALSE alerting FALSE information FALSE releaseComplete FALSE facility FALSE progress FALSE empty FALSE } } Mar 3 04:12:47.720: H225 NONSTD OUTGOING PDU ::= value H323_UU_NonStdInfo ::= { version 2 protoParam qsigNonStdInfo : { iei 4 rawMesg '04038090A31803A983816C0C2180343038353237...'H } } Mar 3 04:12:47.720: H225 NONSTD OUTGOING ENCODE BUFFER::= 60 01020001 041F0403 8090A318 03A98381 6C 0C2180 34303835 32373239 32337005 80333635 33 Mar 3 04:12:47.724: Mar 3 04:12:47.724: H225.0 OUTGOING PDU ::= value H323_UserInformation ::= { h323-uu-pdu { **h323-message-body setup** : *!--- H.225 setup sent to primary endpoint.* { protocolIdentifier { 0 0 8 2250 0 2 } sourceAddress { h323-ID : {"gwa-1"} } sourceInfo { gateway { protocol { voice : { supportedPrefixes { { prefix e164 : "1#" } } } } } } mc FALSE undefinedNode FALSE } activeMC FALSE conferenceID '2D3AAC3316A411CC80688828285B8DF6'H conferenceGoal create : NULL callType pointToPoint : NULL sourceCallSignalAddress ipAddress : { ip 'AC100D0F'H port 11025 } callIdentifier { guid '2D3AAC3316A411CC80698828285B8DF6'H } fastStart { '0000000C6013800A04000100AC100D0F47F1'H, '400000060401004C60138011140000100AC100D0F...'H } mediaWaitForConnect FALSE canOverlapSend FALSE } h245Tunneling TRUE nonStandardControl { { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '6001020001041F04038090A31803A983816C0C21...'H } } } } Mar 3 04:12:47.740: H225.0 OUTGOING ENCODE BUFFER::= 20 A0060008 914A0002 01400400 67007700 61002D0 0 31088001 3C050401 00204000 2D3AAC33 16A411CC 80688828 285B8DF6 00451C07 00AC100D 0F2B1111 002D3AAC 3316A411 CC806988 28285B8D F6320212 0000000C 6013800A 04000100 AC100D0F 47F11D40 00000604 01004C60 13801114 000100AC 100D0F47 F000AC10 0D0F47F1 01000100 06A00180 2D0140B5 00001226 60010200 01041F04 0 38090A3 1803A983 816C0C21 80343038 35323732 39323370 05803336 3533 Mar 3 04:12:47.744: Mar 3 04:12:47.760: H225.0 INCOMING ENCODE BUFFER::= 25 80060008 914A0004 11001100 2D3AAC33 16A411C C 80698828 285B8DF6 10800180 Mar 3 04:12:47.760: Mar 3 04:12:47.760: H225.0 INCOMING PDU ::= value H323_UserInformation ::= { h323-uu-pdu { **h323-message-body releaseComplete** : *!--- First setup message failed.* { protocolIdentifier { 0 0 8 2250 0 4 } callIdentifier { guid '2D3AAC3316A411CC80698828285B8DF6'H } } h245Tunneling TRUE } } Mar 3 04:12:47.776: H225 NONSTD OUTGOING PDU ::= value H323_UU_NonStdInfo ::= { version 2 protoParam qsigNonStdInfo : { iei 4 rawMesg '04038090A31803A983816C0C2180343038353237...'H } } Mar 3 04:12:47.776: H225 NONSTD OUTGOING ENCODE BUFFER::= 60 01020001 041F0403 8090A318 03A98381 6C 0C2180 34303835 32373239 32337005 80333635 33 Mar 3 04:12:47.776: Mar 3 04:12:47.776: H225.0 OUTGOING PDU ::= value

```
H323_UserInformation ::= { h323-uu-pdu { h323-message-body setup : !--- Second setup sent to
alternate endpoint. { protocolIdentifier { 0 0 8 2250 0 2 } sourceAddress { h323-ID : {"gwa-1"}
} sourceInfo { gateway { protocol { voice : { supportedPrefixes { { prefix e164 : "1#" } } } } } }
mc FALSE undefinedNode FALSE } activeMC FALSE conferenceID '2D3AAC3316A411CC80688828285B8DF6'H
conferenceGoal create : NULL callType pointToPoint : NULL sourceCallSignalAddress ipAddress : {
ip 'AC100D0F'H port 11027 } callIdentifier { guid '2D3AAC3316A411CC80698828285B8DF6'H }
fastStart { '0000000C6013800A04000100AC100D0F47F1'H,
'400000060401004C6013801114000100AC100D0F...'H } mediaWaitForConnect FALSE canOverlapSend FALSE
} h245Tunneling TRUE nonStandardControl { { nonStandardIdentifier h221NonStandard : {
t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data
'6001020001041F04038090A31803A983816C0C21...'H } } } } Mar 3 04:12:47.796: H225.0 OUTGOING
ENCODE BUFFER::= 20 A0060008 914A0002 01400400 67007700 61002D0 0 31088001 3C050401 00204000
2D3AAC33 16A411CC 80688828 285B8DF6 00451C07 00AC100D 0F2B1311 002D3AAC 3316A411 CC806988
28285B8D F6320212 0000000C 6013800A 04000100 AC100D0F 47F11D40 00000604 01004C60 13801114
000100AC 100D0F47 F000AC10 0D0F47F1 01000100 06A00180 2D0140B5 00001226 60010200 01041F04 0
38090A3 1803A983 816C0C21 80343038 35323732 39323370 05803336 3533 Mar 3 04:12:47.800: Mar 3
04:12:47.872: H225.0 INCOMING ENCODE BUFFER::= 21 80060008 914A0003 00078E11 002D3AAC 3316A41 1
CC806988 28285B8D F6390219 0000000C 60138011 14000100 AC100D17 479E00AC 100D1747 9F1D4000
00060401 004C6013 80111400 0100AC10 0D0F47F0 00AC100D 17479F01 00010008 800180 Mar 3
04:12:47.872: Mar 3 04:12:47.876: H225.0 INCOMING PDU ::= value H323_UserInformation ::= { h323-
uu-pdu { h323-message-body callProceeding : !--- Call proceeding received. { protocolIdentifier
{ 0 0 8 2250 0 3 } destinationInfo { mc FALSE undefinedNode FALSE } callIdentifier { guid
'2D3AAC3316A411CC80698828285B8DF6'H } fastStart {
'0000000C6013801114000100AC100D17479E00AC...'H, '400000060401004C6013801114000100AC100D0F...'H }
} h245Tunneling TRUE } } Mar 3 04:12:47.884: H225.0 OUTGOING PDU ::= value H323_UserInformation
::= { h323-uu-pdu { h323-message-body empty : NULL h245Tunneling TRUE h245Control {
'0270010600088175000380138000140001000001...'H } } } Mar 3 04:12:47.884: H225.0 OUTGOING ENCODE
BUFFER::= 28 10010006 C0018063 01610270 01060008 8175000 3 80138000 14000100 00010000 0100000C
C0010001 00048000 104810B5 0000120C 52747044 746D6652 656C6179 00008000 16830150 80001583
01408000 12830110 80000020 C0130080 01020000 16020015 00120010 000000 Mar 3 04:12:47.888: Mar 3
04:12:47.888: H225.0 OUTGOING PDU ::= value H323_UserInformation ::= { h323-uu-pdu { h323-
message-body empty : NULL h245Tunneling TRUE h245Control { '01003C4010F3'H } } } Mar 3
04:12:47.892: H225.0 OUTGOING ENCODE BUFFER::= 28 10010006 C0018008 01060100 3C4010F3 Mar 3
04:12:47.892: Mar 3 04:12:47.892: cc_api_call_proceeding(vdbPtr=0x621B2360, callID=0x34,
prog_ind=0x0) Mar 3 04:12:47.896: sess_appl: ev(21=CC_EV_CALL_PROCEEDING), cid(52), disp(0) Mar
3 04:12:47.896: cid(52)st(SSA_CS_CALL_SETTING)ev(SSA_EV_CALL_PROCEEDING)
oldst(SSA_CS_MAPPING)cfid(-1)csize(0)in(0)fDest(0) Mar 3 04:12:47.896: -
cid2(51)st2(SSA_CS_CALL_SETTING)oldst2(SSA_CS_CALL_SETTING) Mar 3 04:12:47.896: ssaCallProc Mar
3 04:12:47.896: ccGetDialpeerTag (callID=0x33) Mar 3 04:12:47.896: ssaIgnore cid(52),
st(SSA_CS_CALL_SETTING),oldst(1), ev(21) Mar 3 04:12:47.900: H225.0 INCOMING ENCODE BUFFER::= 28
10010008 C0018063 01610270 01060008 8175000 6 80138000 14000100 00010000 0100000C C0010001
00048000 104810B5 0000120C 52747044 746D6652 656C6179 00008000 16830150 80001583 01408000
12830110 80000020 C0130080 01020000 16020015 00120010 000000 Mar 3 04:12:47.904: Mar 3
04:12:47.904: H225.0 INCOMING PDU ::= value H323_UserInformation ::= { h323-uu-pdu { h323-
message-body empty : NULL h245Tunneling TRUE h245Control {
'0270010600088175000680138000140001000001...'H } } } !--- Some of the unnecessary H.225 debug
messages are deleted here. Mar 3 04:12:52.116: H225.0 INCOMING ENCODE BUFFER::= 23 80060008
914A0003 000A8600 11002D3A AC3316A 4 11CC8069 8828285B 8DF60100 01000880 0180 Mar 3
04:12:52.120: Mar 3 04:12:52.120: H225.0 INCOMING PDU ::= value H323_UserInformation ::= { h323-
uu-pdu { h323-message-body alerting : !--- Alerting message received. { protocolIdentifier { 0 0
8 2250 0 3 } destinationInfo { mc FALSE undefinedNode FALSE } callIdentifier { guid
'2D3AAC3316A411CC80698828285B8DF6'H } } h245Tunneling TRUE } } Mar 3 04:12:52.124:
cc_api_call_alert(vdbPtr=0x621B2360, callID=0x34, prog_ind=0x8, sig_ind=0x1) Mar 3 04:12:52.124:
sess_appl: ev(7=CC_EV_CALL_ALERT), cid(52), disp(0) Mar 3 04:12:52.124:
cid(52)st(SSA_CS_CALL_SETTING)ev(SSA_EV_CALL_ALERT) oldst(SSA_CS_CALL_SETTING)cfid(-
1)csize(0)in(0)fDest(0)
```

Solucionar problemas de equilíbrio de carga

Com a característica do equilíbrio da carga, você pode ajustar o porteiro com um determinado ponto inicial para o número de atendimentos, de memória, de CPU, e de número de valores-limite registrados. Uma vez que esse ponto inicial é alcançado, o porteiro move valores-limite

registrados de Cisco H.323 para um gatekeeper alternativo ou rejeita atendimentos e registros novos. O balanceamento de carga é habilitada utilizando-se o seguinte comando do CLI de gatekeeper:

```
Router(config-gk)#load-balance [endpoints max-endpoints] [calls max-calls] [cpu max-%cpu][memory max-%mem-used]
```

Quando o limiar é atingido, o gatekeeper usa a mensagem RRJ RAS para informar o ponto final sobre os gatekeepers alternativos e o motivo da rejeição. Quando esta mensagem é recebida, o valor-limite envia um RRQ novo ao gatekeeper alternativo. Depois de registrada com o gatekeeper alternativo, a mensagem GUP é utilizada para informar todos os gatekeepers do conjunto sobre o novo ponto final registrado.

Alguns dos problemas a serem observados ao solucionar problemas incluem verificar a configuração no gatekeeper e certificar-se de que os gatekeepers alternativos e o balanceamento de carga estejam funcionais. A topologia acima é usada para fins de Troubleshooting. A configuração do gatekeeper gkb-1 é alterada para exibir os casos a seguir:

- Como o porteiro pode rejeitar um atendimento quando um ponto inicial for encontrado.
- Como o gatekeeper pode mover o registro de um ponto final para um gatekeeper alternativo quando o limiar é alcançado.

Para debugar a característica do Balanceamento de carga, o uso **debuga a carga de gatekeeper** e **debuga o asn1 h225** para considerar como o porteiro reage quando o ponto inicial é encontrado.

Aqui está a configuração do gatekeeper "gkb-1" que é usado para cobrir os dois casos mencionados acima (limiar de número de chamadas e número de pontos finais registrados):

```
!
gatekeeper
 zone local gkb-1 domainB.com 172.16.13.41
 zone remote gka-1 domainA.com 172.16.13.35 1719
 zone cluster local gkb gkb-1
  element gkb-2 172.16.13.16 1719
 !
 security token required-for all
 gw-type-prefix 2#* default-technology
 bandwidth total zone gkb-1 512
 bandwidth session zone gkb-1 512
 load-balance endpoints 2 calls 1 !--- maximum of 2 endpoints and call threshold is 1 no shutdown ! !
```

Um atendimento é feito através do porteiro gkb-1. Quando esse atendimento estiver acima, um outro atendimento está feito. A depuração capturada mostra a aparência da depuração do balanceamento de carga e como o gatekeeper rejeita a segunda chamada pelo fato de o limitar ter sido atingido. É possível usar o seguinte comando para mostrar quantas chamadas ativas são executadas usando o gatekeeper:

```
gkb-1#show gatekeeper call Total number of active calls = 1. GATEKEEPER CALL INFO
==================== LocalCallID Age(secs) BW 5-29514 9 128(Kbps) Endpt(s): Alias E.164Addr src EP: gwa-1 4085272923 Endpt(s): Alias E.164Addr dst EP: gwb-1 3653 CallSignalAddr Port RASSignalAddr Port 172.16.13.23 1720 172.16.13.23 54670
```

Esta é a depuração do H.225 asn1 e a carga do gatekeeper quando a segunda chamada é solicitada.

```
Mar  3 05:04:55.354: RAS INCOMING ENCODE BUFFER::= 4A 80080501 01806986
40B50000 12298286 B0110075 7
95BF216 AB11CC80 95882828 5B8DF601 81110201 80866940 04006700 77006100
```

```
2D003100 AC100D23 06B70B80 0D
014004 0067006B 0061002D 00310180
Mar  3 05:04:55.358:
Mar  3 05:04:55.358: RAS INCOMING PDU ::=

value RasMessage ::= locationRequest : !--- LRQ is received. { requestSeqNum 2054
destinationInfo { e164 : "3653" } nonStandardData { nonStandardIdentifier h221NonStandard : {
t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data
'8286B0110075795BF216AB11CC80958828285B8D...'H } replyAddress ipAddress : { ip 'AC100D23'H port
1719 } sourceInfo { h323-ID : {"gka-1"} } canMapAlias TRUE } Mar 3 05:04:55.362: H225 NONSTD
INCOMING ENCODE BUFFER::= 82 86B01100 75795BF2 16AB11CC 80958828 28 5B8DF6 01811102 01808669
40040067 00770061 002D0031 Mar 3 05:04:55.366: Mar 3 05:04:55.366: H225 NONSTD INCOMING PDU ::=
value LRQnonStandardInfo ::= { ttl 6 nonstd-callIdentifier { guid
'75795BF216AB11CC80958828285B8DF6'H } callingOctet3a 129 gatewaySrcInfo { e164 : "5336", h323-ID
: {"gwa-1"} } } Mar 3 05:04:55.366: gk_load_overloaded: Overloaded due to reaching specified
call limits !--- Number of calls threshold has met. Mar 3 05:04:55.370: RAS OUTGOING PDU ::=
value RasMessage ::= locationReject : !--- LRJ is sent. { requestSeqNum 2054 rejectReason
undefinedReason : NULL }
```

No segundo exemplo, o gatekeeper "gkb-1" tem dois pontos finais registrados. O registro para um outro valor-limite é tentado. O porteiro moveu o valor-limite, tentando registrar ao gatekeeper alternativo "gkb-2", desde que está no mesmo conjunto. Aqui está uma mensagem de depuração para debug h225 asn1 e debug gatekeeper gup asn1 para esse caso:

```
Mar  3 05:21:05.682: RAS INCOMING PDU ::=

value RasMessage ::= registrationRequest : !--- RRQ message is received. { requestSeqNum 4621
protocolIdentifier { 0 0 8 2250 0 3 } discoveryComplete TRUE callSignalAddress { ipAddress : {
ip 'AC100D2A'H port 1720 } } rasAddress { ipAddress : { ip 'AC100D2A'H port 49998 } }
terminalType { gateway { protocol { voice : { supportedPrefixes { { prefix e164 : "1#" } } } } } }
mc FALSE undefinedNode FALSE } terminalAlias { h323-ID : {"gwb-3"} } gatekeeperIdentifier {"gkb-
1"} endpointVendor { vendor { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } }
timeToLive 60 tokens { { tokenOID { 1 2 840 113548 10 1 2 1 } timeStamp 731136065 challenge
'5A70CA112E6C7A3834792BD64FF7AD2F'H random 58 generalID {"gwb-3"} } } cryptoTokens {
cryptoEPPwdHash : { alias h323-ID : {"gwb-3"} timeStamp 731136065 token { algorithmOID { 1 2 840
113549 2 5 } paramS { } hash "B1C1DAD962BEE42B1E53F368238B1D8" } } } keepAlive FALSE
willSupplyUUIEs FALSE maintainConnection TRUE } Mar 3 05:21:05.698: gk_load_overloaded:
Overloaded due to reaching specified endpoint limits !--- Endpoint threshold is met. Mar 3
05:21:05.702: RAS OUTGONG PDU ::= value RasMessage ::= registrationReject : !--- RRJ is sent. {
requestSeqNum 4621 protocolIdentifier { 0 0 8 2250 0 3 } rejectReason resourceUnavailable : NULL
!--- Reject reason. gatekeeperIdentifier {"gkb-1"} altGKInfo !--- List of alternate gatekeepers.
{ alternateGatekeeper { { rasAddress ipAddress : { ip 'AC100D10'H port 1719 }
gatekeeperIdentifier {"gkb-2"} needToRegister TRUE priority 0 } } altGKisPermanent TRUE !---
Informs the endpoint that the move is permanent. } Mar 3 05:21:05.706: RAS OUTGOING ENCODE
BUFFER::= 16 80120C06 0008914A 00038101 00080067 006B0062 0 02D0031 07001600 0140AC10 0D1006B7
08006700 6B006200 2D003280 80 Mar 3 05:21:05.706: Mar 3 05:21:05.782: Received GUP REGISTRATION
INDICATION from 172.16.13.16 !--- GUP update for the new endpoint. gkb-1#
```

Informações Relacionadas

- Suporte à Tecnologia de Voz
- Suporte de Produtos de Comunicação de Voz e de IP
- Troubleshooting da Telefonia IP Cisco
- Suporte Técnico - Cisco Systems