

Dépannage de GUP, de point de terminaison alternatif et d'équilibrage de charge

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Définitions](#)

[Topologie de travaux pratiques et configurations](#)

[Topologie de travaux pratiques et configurations](#)

[Garde-portes alternatifs](#)

[Mise à jour Protocol de garde-porte](#)

[mettez au point et des commandes show pour des batteries de garde-porte](#)

[Debugs de "gkb-1" quand c'était premier pour joindre la batterie](#)

[Debugs de "gkb-2" quand il a joint la batterie après "gkb-1"](#)

[Debugs quand un point final s'inscrit à un des garde-portes dans la batterie](#)

[Debugs pour quand un point final avec l'appel actif se déplace pour alterner le garde-porte](#)

[La passerelle Cisco basculent pour alterner le garde-porte](#)

[Dépannez les questions avec des points finaux alternatifs](#)

[Vérifiez le garde-porte a les points finaux alternatifs corrects](#)

[Vérifiez si le garde-porte inclut des points finaux alternatifs dans ses messages LCF ou ACF RAS](#)

[Vérifiez si les essais OGW contacter des remplaçants au cas où le point final principal de destination échouerait](#)

[Dépannez équilibrent la charge](#)

[Informations connexes](#)

[Introduction](#)

Ce document est conçu pour vous aider à dépanner et comprendre ces caractéristiques de Cisco Gatekeeper :

- Batteries de garde-porte et mise à jour Protocol (GUP) de garde-porte
- Points finaux alternatifs
- Équilibrage de charge

Référez-vous au [garde-porte performant de Cisco](#) pour toutes les informations nécessaires sur ces caractéristiques comprenant la vue d'ensemble des fonctionnalités, les Plateformes prises en charge, des versions logicielles de Cisco IOS® requises et comment les configurer, surveiller, et mettre à jour.

Conditions préalables

Conditions requises

Les lecteurs de ce document doivent avoir une bonne connaissance de ce qui suit :

- Connaissance de base de la fonctionnalité de garde-porte.
- Connaissance de base de VoIP, signalisation H.323 et d'enregistrement, d'admission, et d'état (RAS).

Composants utilisés

Les informations dans ce document sont basées sur les versions de logiciel et de matériel ci-dessous.

- Version du logiciel Cisco IOS 12.3(4)T1
- Passerelles Cisco : Cisco AS5300, Cisco AS5400, et Cisco 3725
- Garde-portes de Cisco : Cisco 3725 et Cisco 2611

Les informations présentées dans ce document ont été créées à partir de périphériques dans un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si vous travaillez dans un réseau opérationnel, assurez-vous de bien comprendre l'impact potentiel de toute commande avant de l'utiliser.

Conventions

Pour plus d'informations sur les conventions des documents, référez-vous aux [Conventions utilisées pour les conseils techniques de Cisco](#).

Définitions

Terme	Définition
ARQ	La demande d'admission (ARQ) est un message RAS envoyé d'un point final de Cisco H.323 à un garde-porte qui invite une admission pour établir un appel.
ACF	L'admission confirmant (ACF) est un message RAS envoyé du garde-porte au point final qui confirme l'acceptation d'un appel.
ARJ	Le rejet d'admission (ARJ) est un message RAS du garde-porte au point final qui rejette la demande d'admission.
GCF	Le garde-porte confirmant (GCF) est un message RAS envoyé d'un garde-porte au point final de Cisco H.323 qui confirme la détection du garde-porte.
GRQ	La demande de garde-porte (GRQ) est un message

R Q	RAS envoyé d'un point final de Cisco H.323 pour découvrir le garde-porte.
G U P	La mise à jour Protocol de garde-porte est utilisée pour partager les informations entre les garde-portes dans une batterie au sujet de leurs points finaux et les appels actifs.
L C F	L'emplacement confirmant (LCF) est un message RAS envoyé d'un garde-porte à l'autre qui confirme la demande d'emplacement (LRQ) et inclut l'adresse IP de terminer le point final.
L R J	L'anomalie d'emplacement (LRJ) est un message RAS envoyé d'un garde-porte à l'autre qui rejette le LRQ.
L R Q	La demande d'emplacement est un message RAS envoyé d'un garde-porte à l'autre qui demande l'adresse IP d'un point final de terminaison distant.
R A S	Le protocole RAS permet à un garde-porte pour exécuter le contrôle d'enregistrement, d'admission, et d'état du point final.
R C F	L'enregistrement confirmant (RCF) est un message RAS envoyé du garde-porte au point final qui confirme l'enregistrement.
R R J	L'anomalie d'enregistrement (RRJ) est un message RAS envoyé du garde-porte qui rejette la demande d'enregistrement.
R R Q	La demande d'enregistrement (RRQ) est un message RAS envoyé du point final au garde-porte qui demande de s'inscrire à lui.
U R Q	La demande d'Unregistration (URQ) est message RAS envoyé du point final au garde-porte qui demande l'ONU-registre avec lui.

[Topologie de travaux pratiques et configurations](#)

Pour expliquer comment les caractéristiques fonctionnent et comment dépanner, une installation de laboratoire a été établie avec cette topologie :

[Topologie de travaux pratiques et configurations](#)

Les configurations de base de tous les passerelles et garde-portes sont dans la table ci-dessous. Avec différents cas, un certain changement de configuration était nécessaire. La modification est indiquée quand ceci se produit. Les configurations ci-dessous contiennent seulement les pièces qui sont essentielles à la fonctionnalité de passerelle ou de garde-porte pour ce laboratoire.

Les configurations de "gwb-1" et de "gwb-2" sont presque semblables (excepté l'adresse IP et H.323 l'ID). Par conséquent, seulement `gwb-1` est affiché ci-dessous.

<code>gwa-1</code>

```
!  
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
```

Garde-portes alternatifs

Avant de Cisco la version 2 H.323, chaque zone a été seulement contrôlée par un garde-porte simple. La version 2 de Cisco H.323 introduit l'idée « de garde-porte alternatif » de fournir la Redondance de garde-porte. La mise en oeuvre de la caractéristique alternative de garde-porte permet à de plusieurs garde-portes pour contrôler une zone. Quand un point final s'inscrit à un garde-porte, on lui donne en liste de garde-portes alternatifs pour la zone dans laquelle le point final s'enregistre, et pour quels remplaçants ont été spécifiés utilisant le CLI. Si le garde-porte échoue, le point final peut utiliser les garde-portes alternatifs afin de continuer l'exécution.

La liste alternative de garde-porte est donnée à Cisco Gatekeeper par le CLI pour chaque zone et est transmise aux points finaux par les messages RCF (poids léger y compris) et GRQ. Cette liste peut également être transmise dans d'autres messages, tels qu'ARJ ou URQ, pour faciliter un arrêt commandé de garde-porte.

Les garde-portes alternatifs se renseignent sur des appels existants par un échange de la réponse de demande de la demande d'interruption (IRQ) /Information (IRR) entre les passerelles et les garde-portes et maintiennent ces appels.

Un point final qui détecte la panne de son garde-porte peut sans risque récupérer de cette panne en utilisant un garde-porte alternatif pour de futures demandes, y compris des demandes des appels existants. Des garde-portes alternatifs doivent être configurés dans une batterie. Ils partagent les informations sur les points finaux et les appels actifs utilisant le GUP qui fonctionne sur le TCP.

Mise à jour Protocol de garde-porte

Voici quelques étapes principales et mises en garde du GUP. Ceci devrait également vous aider à dépanner.

- Une fois qu'un garde-porte qui est configuré pour faire partie d'une batterie est livré sur la ligne, elle ouvre un port TCP pour écouter les connexions entrantes pour le protocole GUP.
- Alors il annonce sa présence en envoyant un message GRQ sur une base périodique. La période par défaut est de 30 secondes et est configurable utilisant le garde-porte que le [timer cluster-element](#) CLI [annoncent la](#) commande. Ce message GRQ contient des données non standard à chaque garde-porte alternatif. Ces données non standard sont un indicateur aux remplaçants que le GRQ n'est vraiment pas un GRQ du tout, mais sont plutôt juste un message de « annonce ». À l'intérieur du message GRQ, le garde-porte indique le numéro de

port qu'il a ouvert pour écouter le protocole GUP.

- Quand vous recevez un GRQ du nouveau garde-porte, d'autres garde-portes dans la batterie ouvrent des canaux de TCP à ce port.
- Les messages GUP GRQ peuvent être l'un des messages suivants : `announcementIndication`, `announcementReject`, `registrationIndication`, `unregistrationIndication`, et `resourceIndication`.
- L'indication d'annonce diffuse également des informations sur l'utilisation de bande passante pour la zone. Ceci permet aux garde-portes alternatifs pour gérer correctement la bande passante pour une zone simple, quoique les garde-portes soient dans des périphériques physiques distincts.
- Pour vérifier, que les garde-portes alternatifs communiquent correctement ou pas, utilisez le [show gatekeeper zone cluster de](#) commande. Cette commande signale également les informations de bande passante pour les garde-portes alternatifs.
- Le garde-porte suppose que le garde-porte alternatif a manqué (et suppose que précédemment la bande passante allouée est maintenant disponible), si le garde-porte ne reçoit pas un message d'annonce au cours de six périodes d'annonce, ou si la connexion TCP avec le garde-porte est détectée pour être cassée. Avec six périodes d'annonce toutes les 30 secondes, le temps est de trois minutes, qui égale on assume qu'à ce qui est la longueur moyenne d'un appel. Il devrait alors être assez sûr de supposer que la bande passante a été libérée. Après trois minutes, ce garde-porte déclare son remplaçant en tant que vers le bas et envoie une mise à jour pour informer tous ses points finaux enregistrés qu'il n'y a aucun garde-porte alternatif.
- Quand un point final s'inscrit/unregisters à un garde-porte dans une batterie, ce garde-porte emploie le message `unregistrationIndication` `registrationIndication`/pour mettre à jour tous autres garde-portes dans cette batterie au sujet de cette modification.
- Si un point final signalait un changement de ressource utilisant la disponibilité des ressources d'indicateur (RAI) à un garde-porte d'une batterie, ce garde-porte signale le changement à tous les garde-portes alternatifs de cette batterie à l'aide du `resourceIndication` de message GUP.
- Tous les messages GUP sont nécessaires pour le garde-porte dans une batterie pour avoir la connaissance suffisante au sujet de chaque point final dans la zone (enregistrement, bande passante, appels actifs, ressources) à pouvoir résoudre questionne localement.
- Quand un point final est commuté d'un garde-porte à un remplaçant, les besoins alternatifs de se renseigner sur les appels qui sont en activité sur le point final. Quand un garde-porte envoie un RCF pour un nouvel enregistrement, il envoie également un IRQ pour obtenir une liste de tous les faire appel au point final. Il est important de s'assurer que l'IRQ n'atteint pas le point final avant le RCF.
- Les garde-portes dans une batterie permettent un arrêt, quoiqu'il y ait des appels actifs, tant que il y a un garde-porte alternatif défini pour toutes les zones pour lesquelles il y a des appels actifs. Si n'importe quelle zone n'a un appel actif et aucun garde-porte alternatif définis, le garde-porte refuse l'arrêt.
- Les garde-portes alternatifs reçoivent toutes les demandes de débranchement (DRQs) des appels qu'ils ne se rendaient pas compte de et passent l'information correcte à l'Authentification, autorisation et comptabilité (AAA) et aux serveurs du Message Protocol de transaction de Cisco Gatekeeper (GKTMP). Ceci se produit quand ce point final se déplace au garde-porte alternatif tandis qu'il y a des appels actifs. En outre, on peut envoyer des messages IRR qui contiennent les informations d'appel pour les appels qui n'ont pas été précédemment connus. Pour des ces IRRs, des enregistrements d'appel sont construits et la

bande passante est allouée en conséquence.

- Le garde-porte crée un seul message d'indication d'annonce pour chaque garde-porte alternatif. Si un garde-porte alternatif reçoit un message qui contient un identifiant de garde-porte qu'il n'identifie pas (qui peut se produire si le garde-porte alternatif est un remplaçant pour une zone), mais pas un autre, ces informations sont ignorées. Cependant, le garde-porte alternatif détecte des erreurs dans la configuration des remplaçants en examinant ces messages et elle signale ces erreurs à l'utilisateur.
- L'alimentation vraie du GUP est réalisée quand des adresses sont résolues pour une zone distante. Au lieu du besoin de la zone distante d'envoyer LRQs (dans l'ordre ou revenu) à tous les garde-portes, de ce fait augmentant le temps système de Messagerie sur des liens d'étendu, il doit maintenant envoyer cette requête juste à une des garde-portes dans la batterie. Ajouté à la nouvelle [zone cluster remote](#) CLI, lui pouvez circulaire entre les garde-portes dans la batterie et pas la tentative d'envoyer LRQ à un autre garde-porte dans la batterie si elle reçoit une anomalie de n'importe quelle.
- Au cas où une passerelle était déplacée à un garde-porte alternatif, elle essaye toujours de s'enregistrer à ce garde-porte à moins que vous n'émettiez une **aucune passerelle** et puis une commande de **passerelle**. Quand le garde-porte primaire du point final est de retour en ligne, le point final ne fait pas re-registre à lui à moins que le point final ait perdu la transmission avec le garde-porte alternatif. Il continue à utiliser le garde-porte alternatif pour ses informations de routage d'appels.

[mettez au point et des commandes show pour des batteries de garde-porte](#)

Met au point expliquent ci-dessous comment les garde-portes peuvent joindre une batterie et comment ils partagent les informations sur leurs points finaux. **des commandes show** sont utilisées d'afficher comment surveiller la batterie. Met au point utilisé sont [mettent au point le gup asn1 de garde-porte](#) et le [debug h225 asn1](#). Ceci est basé sur la topologie et la configuration mentionnées ci-dessus.

[Debugs de "gkb-1" quand c'était premier pour joindre la batterie](#)

```
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:
```

[Debugs de "gkb-2" quand il a joint la batterie après "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

```

Avec la commande de [point final de garde-porte d'exposition](#), il n'y a aucun point final enregistré. La sortie est comme suit :

```

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

```

[Debugs quand un point final s'inscrit à un des garde-portes dans la batterie](#)

Ceci mettent au point est pris du garde-porte de "gkb-1". Le point final est "gwb-1" s'enregistrant au garde-porte de "gkb-1" avec le [debug h225 ans1](#) et au [debug ras](#) activé.

```

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 willSupplyUIEs 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:

```

La sortie ci-dessus contient le [garde-porte d'exposition que les](#) passerelles de sortie de commande de [point final](#) après tout s'enregistrent dans la batterie. Ce qui précède met au point se produit pour chaque enregistrement de point final. Après tout trois passerelles dans la batterie s'enregistrent, le [garde-porte d'exposition que la](#) commande de [point final](#) sur les deux garde-portes est comme suit :

```

gkb-1#show gatekeeper endpoints GATEKEEPER ENDPOINT REGISTRATION
===== CallSignalAddr Port RASignalAddr Port Zone Name Type Flags ---
----- 172.16.13.23 1720 172.16.13.26 1720 172.16.13.26 49801 gkb-1 VOIP-GW H323-ID:
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 RASignalAddr 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.

```

[Debugs pour quand un point final avec l'appel actif se déplace pour alterner le garde-porte](#)

Ce sont met au point d'un garde-porte qui commence quand l'appel est demandé et va jusqu'à ce qu'il soit déconnecté. Certains des messages de débogage inutiles ont été omis. Ceux-ci met au point sont du garde-porte de "gkb-1". L'appel a été placé par gwa-1 enregistré à "gka-1" à une autre passerelle (gwb-1) dans la batterie distante de zone. Met au point expliquent comment un écoulement d'appel actif est dépisté du garde-porte primaire au garde-porte alternatif pendant que le primaire descend.

```
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 '00014004006700770062002D0031080067006B00...'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 } willSupplyUIEs 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 uiiesRequested { 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
information !--- 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

```

Remarque: En ce moment "gkb-1" est arrêté. Ceci est permis (même si il a un appel actif) parce qu'il y a un garde-porte alternatif pour cette zone.

Des messages URQ sont envoyés à tous les points finaux inscrits au "gkb-1". Ces points finaux sont des passerelles de "gwb-1" et de "gwb-2". Ces passerelles confirment l'URQ en renvoyant UCFs. En outre, gkb-1 envoie un message d'indication d'unregISTRATION GUP au garde-porte de remplaçant de batterie et alors il ferme la connexion 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

```

Voici le débogage de "gkb-2". Met au point qui affichent que l'enregistrement du point final déplacé "gwb-1" et "gwb-2" sont omis, puisqu'ils ressemblent à l'enregistrement normal. Le but ici est

d'afficher l'acceptation du DRQ de l'appel actif sur "gwb-1" quand il est déplacé à "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#
```

[La passerelle Cisco basculent pour alterner le garde-porte](#)

Par défaut, les passerelles Cisco envoient à un poids léger RRQ toutes les 45 secondes. Au cas où le garde-porte n'envoyait aucun URQ à la passerelle (due à une question cassée de routage, par exemple), les essais de passerelle (en n'entendant pas un RCF ou un RRJ pour son poids léger RRQ) deux fois avec cinq secondes entre chacun. Si la troisième tentative échoue, elle considère immédiatement le garde-porte comme morts et s'inscrit au garde-porte alternatif qui utilise RRQ. Dans un scénario où la passerelle commence la procédure d'enregistrement initiale avec le garde-porte, il envoie le GRQ pour localiser l'adresse IP de garde-porte. S'il y a une réponse GCF de retour, la passerelle envoie le RRQ au garde-porte primaire spécifié. Si pour une raison quelconque, le garde-porte rejette la demande d'enregistrement, la passerelle n'essaye pas de contacter son garde-porte alternatif. Il commence ce processus (GRQ, GCF et RRQ) plus de nouveau par le garde-porte primaire.

La passerelle contacte seulement le garde-porte alternatif quand la Connectivité au garde-porte primaire est perdue et il n'y a aucune réponse de retour. Si le garde-porte primaire ne répond pas de nouveau au message GRQ quand la passerelle envoie d'abord pour découvrir le garde-porte, alors après que trois essais ratés (approximativement cinq minutes par tentative), la passerelle contacte le garde-porte alternatif. Dans une situation où le garde-porte primaire descend après que la passerelle se soit inscrite à elle, la passerelle perd des messages de Keepalives du garde-porte primaire. Après avoir manqué trois messages consécutifs de keepalive, la passerelle déclare le garde-porte primaire en tant que vers le bas, et elle reprend la procédure d'enregistrement.

[Dépannez les questions avec des points finaux alternatifs](#)

Un point final appelant peut récupérer d'une panne d'établissement d'appel en envoyant un message de configuration à un des points finaux alternatifs. L'appel peut échouer pour de nombreuses raisons : la passerelle est vers le bas et le garde-porte ne se rend pas compte de elle au moment d'envoyer l'ACF ou le LCF, il n'y a aucune ressource sur la passerelle et n'a pas signalé cela au garde-porte, l'appel échoue en raison d'une configuration incorrecte sur le point final principal, et davantage.

Remarque: Les essais d'origine de point final seulement pour contacter les garde-portes alternatifs

si l'appel échoue avant l'étape vigilante (alerte ou progression). Si les appels échouent en raison de l'utilisateur occupé ou du pas de réponse, le point final d'origine n'essaye aucun autre remplaçant.

Le garde-porte se renseigne sur celui le remplaçant pour un certain point final par configuration manuelle utilisant l'Alt-[PE de point final de](#) commande CLI de garde-porte ou de tous les messages reçus RAS. Cisco prend en charge un maximum de 20 remplaçants pour chaque point final, n'importe comment le garde-porte les apprend.

Les questions dont vous avez besoin pour jeter un regard incluent :

- Si le garde-porte a le point final alternatif correct comme désiré.
- Si le garde-porte inclut les points finaux alternatifs dans ses messages LCF ou ACF RAS.
- Si les essais OGW pour contacter les remplaçants au cas où le point final principal de destination échouerait.

Pour afficher comment dépanner ces questions, la même topologie comme ci-dessus est utilisée avec cette modification sur la configuration du contrôleur d'accès de "gkb-1" pour inclure deux passerelles alternatives : "gwb-3" et "gwb-1" pour la passerelle "gwb-2". Voici la configuration du garde-porte 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. !
```

[Vérifiez le garde-porte a les points finaux alternatifs corrects](#)

Pour voir si le garde-porte a les bons points finaux alternatifs, utilisez l'ordre de [remplaçants de show gatekeeper endpoints](#).

```
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#
```

[Vérifiez si le garde-porte inclut des points finaux alternatifs dans ses messages LCF ou ACF RAS](#)

Pour voir si le garde-porte envoie l'adresse IP pour des points finaux alternatifs, vous pouvez activer le **debug h225 asn1** et regarder le message ACF ou le LCF. C'est un exemple mettant au point pris 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
'00014004006700770062002D0032080067006B00...'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 } } } }
```

Vérifiez si les essais OGW contacter des remplaçants au cas où le point final principal de destination échouerait

Cette section affiche comment l'OGW réagit quand il reçoit des points finaux alternatifs dans son message ACF. Dans cet exemple l'appel est fait pour échouer quand il tente d'entrer en contact avec le point final de terminaison principal (gw). Les debugs à activer ici sont [debug voip ccapi inout](#) et [debug h225 asn1](#).

La première chose que vous voyez dans le débogage est le message de ccapi qui affiche le tronçon d'origine de téléphonie.

```
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), cllcd(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)csz(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,
'400000060401004C6013801114000100AC100D0F...'H } mediaWaitForConnect FALSE canOverlapSend FALSE
} h245Tunneling TRUE nonStandardControl { { nonStandardIdentifier h221NonStandard : {
t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data
'6001020001041F04038090A31803A983816C0C21...'H } } } } Mar 3 04:12:47.740: 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 16A411 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)csz(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)csz(0)in(0)fDest(0)

```

[Dépannez équilibrent la charge](#)

Avec la configuration d'équilibrer la charge, vous pouvez placer le garde-porte avec un certain seuil pour le nombre d'appels, de mémoire, de CPU, et de nombre de points finaux enregistrés. Une fois que ce seuil est atteint, le garde-porte déplace des points finaux d'enregistré Cisco H.323 à un garde-porte alternatif ou rejette de nouveaux appels et enregistrements. L'Équilibrage de charge est activé utilisant la commande suivante CLI de garde-porte :

```

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

```

Quand le seuil est rencontré, le garde-porte emploie le message RRJ RAS pour informer le point final au sujet des garde-portes alternatifs et de la raison d'anomalie. Quand ce message est reçu, le point final envoie un nouveau RRQ au garde-porte alternatif. Une fois qu'il est inscrit au garde-porte alternatif, il emploie le message GUP pour informer tous les garde-portes dans la batterie au sujet du nouveau point final enregistré.

Certaines des questions à surveiller quand vous dépannez sont de vérifier la configuration sur le garde-porte et veiller les garde-portes et l'Équilibrage de charge alternatifs sont fonctionnelles. La topologie ci-dessus est utilisée pour dépanner des but. La configuration du garde-porte de "gkb-1" est changée pour afficher les cas suivants :

- Comment le garde-porte peut rejeter un appel quand un seuil est rencontré.
- Comment le garde-porte peut déplacer l'enregistrement d'un point final à un garde-porte alternatif quand un seuil est rencontré.

Pour mettre au point la caractéristique d'Équilibrage de charge, le [debug gatekeeper load](#) d'utilisation et le **debug h225 asn1** pour voir comment le garde-porte réagit quand le seuil est rencontré.

Voici la configuration du garde-porte de "gkb-1" qui est utilisé pour couvrir les deux cas mentionnés ci-dessus (nombre d'appels seuil et nombre de points enregistrés d'extrémité) :

```
!
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 ! !
```

Un appel est fait par le garde-porte gkb-1. Tandis que cet appel est en hausse, un autre appel est fait. Capturés mettent au point des expositions ce que l'équilibrer la charge mettent au point ressemble à et comment le garde-porte rejette le deuxième appel parce que le seuil a été rencontré. Vous pouvez utiliser la commande suivante d'afficher combien d'appels actifs fonctionnent utilisant le garde-porte :

```
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
```

Voici le débogage de H.225 asn1 et le chargement de garde-porte quand le deuxième appel est demandé :

```
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 }
```

Pour le deuxième exemple, le garde-porte de "gkb-1" a deux points finaux enregistrés. L'inscription à un autre point final est tentée. Le garde-porte a déplacé le point final, essayant d'enregistrer au garde-porte alternatif "gkb-2", puisqu'ils sont dans la même batterie. Être le message de débogage pour le **debug h225 asn1** et voici **mettent au point le gup asn1 de garde-porte** pour ce cas :

```
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
willSupplyUIEs 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#
```

[Informations connexes](#)

- [Assistance technique concernant la technologie vocale](#)
- [Support produit pour Voix et Communications IP](#)
- [Dépannage des problèmes de téléphonie IP Cisco](#)
- [Support technique - Cisco Systems](#)