

Configuration du contrôle d'admission d'appel de contrôleur d'accès de base

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Informations générales](#)

[bandwidth \(gatekeeper\), commande](#)

[Configuration](#)

[Diagramme du réseau](#)

[Configurations](#)

[Vérification](#)

[Dépannage](#)

[Dépannage des commandes](#)

[Exemple de résultat show and debug](#)

[Informations connexes](#)

[Introduction](#)

Ce document fournit un exemple de configuration pour le contrôle d'admission d'appel de base du contrôleur d'accès.

[Conditions préalables](#)

[Conditions requises](#)

Plusieurs conditions doivent être remplies avant que la passerelle puisse obtenir la résolution d'adresse correcte auprès du contrôleur d'accès. Il y a plusieurs points importants à vérifier pour chaque solution VoIP lorsque des liaisons à faible débit sont impliquées.

Avant d'essayer cette configuration, assurez-vous de respecter les conditions suivantes :

- Toutes les passerelles doivent être enregistrées auprès des contrôleurs d'accès correspondants
- Tous les contrôleurs d'accès doivent disposer d'un plan de numérotation correct pour pouvoir décider de la route des appels.
- Le contrôle d'admission peut être configuré pour restreindre le numéro d'appel entre certaines

zones.

Comme les deux premiers points sont pris en compte dans la section [Configurer](#), nous nous concentrerons sur le contrôle d'admission dans la section [Informations générales](#).

[Components Used](#)

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Trois routeurs Cisco 2600.
- Logiciel Cisco IOS[®] Version 12.2.8.5 ENTERPRISE PLUS/H323 MCM.

Les informations présentées dans ce document ont été créées à partir de périphériques dans un environnement de laboratoire spécifique. All of the devices used in this document started with a cleared (default) configuration. Si vous travaillez dans un réseau opérationnel, assurez-vous de bien comprendre l'impact potentiel de toute commande avant de l'utiliser.

[Conventions](#)

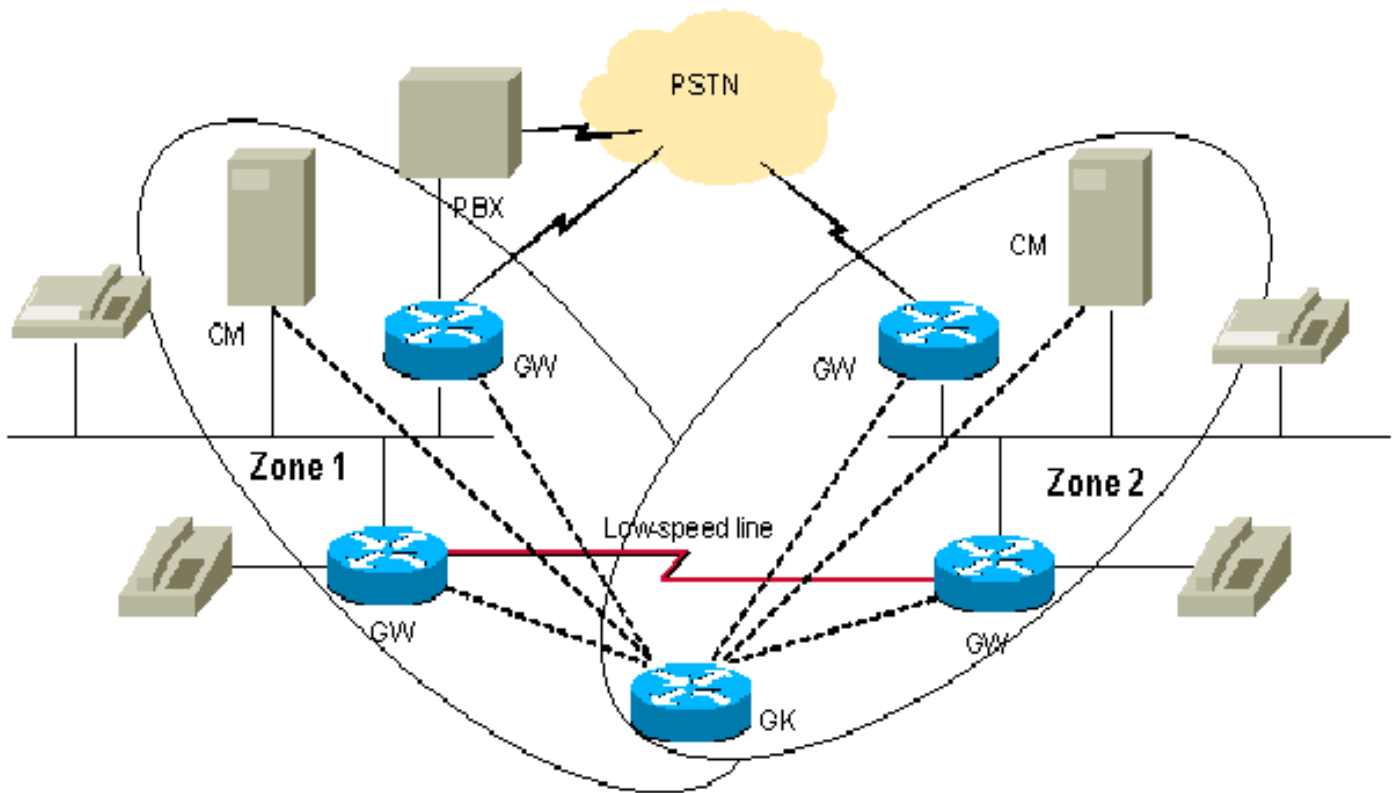
Pour plus d'informations sur les conventions des documents, référez-vous aux [Conventions utilisées pour les conseils techniques de Cisco](#).

[Informations générales](#)

Cet exemple de configuration étudie un réseau VoIP avec une topologie à deux zones, qui est géré par un contrôleur d'accès avec trois passerelles dans les deux zones. L'objectif de ce document est de fournir un exemple simple de configuration d'un contrôle d'admission qui applique une politique au nombre d'appels entre les zones et à l'intérieur de celles-ci. Ce document inclut des informations techniques générales sur les fonctionnalités configurées, des directives de conception et des stratégies de vérification et de dépannage de base.

Remarque : Dans cette configuration, les quatre routeurs sont situés sur le même réseau local. Cependant, dans votre topologie réelle, tous les périphériques peuvent se trouver dans différentes parties de votre réseau.

Très souvent, il existe plusieurs sources de trafic hautement prioritaire dans les réseaux réels. Il est difficile de distinguer toutes ces conditions car elles sont nombreuses et faciles à ignorer. Cependant, il y a plusieurs situations courantes qui se produisent très souvent dans la vie réelle qui méritent d'être prises en compte. Le contrôle d'admission devient un problème lorsque les routeurs qui fournissent la hiérarchisation du trafic ne sont pas eux-mêmes les sources de ce trafic. La topologie type implique plusieurs passerelles vocales sur deux sites connectés via la liaison fournie par une paire de routeurs. Une autre topologie concerne Cisco CallManager avec des téléphones IP sur deux sites, ainsi que les passerelles vers le RTPC ou le PBX. Dans les deux cas, nous avons plusieurs sources de trafic vocal des deux côtés de la liaison.



Parfois, il peut y avoir un problème de qualité de la voix, si la quantité de trafic vocal dépasse la bande passante configurée pour la file d'attente prioritaire. Cela est dû au fait que les routeurs et les téléphones IP Cisco CallManager qui sont à l'origine du trafic ne disposent pas d'une gestion centralisée pour l'admission des appels dans la conception indiquée ci-dessus. Dans ce cas, les paquets dépassant la bande passante seront abandonnés.

Il existe plusieurs façons d'éviter ce scénario. La solution la plus simple consiste à configurer la bande passante vocale dans la file d'attente de faible latence (LLQ) pour accepter le nombre maximal d'appels de toutes les sources. En l'absence de trafic vocal, la bande passante inutilisée sera allouée aux flux de données. Cela peut être fait lorsque la bande passante totale de la liaison est supérieure à la bande passante requise pour le nombre maximal d'appels.

Une approche plus judicieuse consiste à appliquer des restrictions sur chaque source de trafic vocal des deux extrémités de la liaison. Dans ce cas, la bande passante récapitulative de tous les sites ne dépassera pas les 75 % recommandés de bande passante réelle de la liaison entre les sites. Pour appliquer ces restrictions, utilisez la commande **max-conn** sous la configuration de terminal de numérotation dial-peer VoIP. Si nous supposons qu'il n'y a Cisco CallManager que sur un site central, nous pouvons utiliser ses capacités pour limiter le nombre d'appels vers le site de la filiale sans CallManager. Cette approche nous permet de gérer la situation dans laquelle les sources de trafic vocal sont capables de sursouscription de la liaison. L'inconvénient de cette approche est l'utilisation inflexible de la bande passante accordée aux sources. Cette approche ne permet pas à certaines passerelles de passer un appel supplémentaire même si la bande passante disponible est libre à ce moment-là.

L'approche la plus flexible consiste à utiliser une entité distincte pour le contrôle centralisé d'admission des appels : le contrôleur d'accès. Le contrôleur d'accès permet de lier deux sites à deux Cisco CallManager (ou clusters CallManager).

Remarque : l'utilisation du contrôleur d'accès ne signifie pas toujours l'achat d'un nouveau routeur distinct. En fonction du nombre d'appels et de la charge des routeurs, vous pouvez configurer un contrôleur d'accès sur l'un des routeurs existants avec le jeu de fonctions Cisco IOS approprié

comme Enterprise/PLUS/H323. Cela peut aider à gérer les petites succursales et autoriser un contrôleur d'accès dédié dans le site central uniquement.

L'approche du contrôleur d'accès doit être envisagée avec soin, afin de ne pas submerger le routeur d'une charge supplémentaire. En outre, vous devez vérifier si la topologie permettra de placer le contrôleur d'accès de cette manière pour éviter le trafic supplémentaire sur la liaison critique.

Il est généralement recommandé d'utiliser des routeurs Cisco distincts comme contrôleurs d'accès dédiés dans votre réseau dans un numéro approprié à votre topologie.

Considérez la topologie ci-dessus. Ici, vous pouvez placer tous les périphériques dans les deux zones locales gérées par un seul contrôleur d'accès. Cela vous permet d'avoir un nombre élevé d'appels dans chaque zone, mais limite le nombre d'appels entre eux. Dans notre exemple de test, nous limiterons la bande passante entre les deux zones à un seul appel et autoriserons jusqu'à deux appels (plus nombreux) dans l'un d'eux.

Pour plus d'informations à ce sujet, consultez [Contrôle d'admission des appels VoIP](#).

Pour terminer la tâche, utilisez la commande **bandwidth** (gatekeeper) décrite dans [Cisco High-Performance Gatekeeper](#)

[bandwidth \(gatekeeper\), commande](#)

Pour spécifier la bande passante totale maximale pour le trafic H.323, utilisez la commande de configuration du contrôleur d'accès **bande passante**. Pour désactiver la fonction, utilisez la forme **no** de cette commande.

Remarque : cette commande vous permet de restreindre la bande passante via une liaison unique à partir de la zone. Si la topologie vous permet de passer un appel via plusieurs chemins d'une zone à l'autre, les liaisons peuvent facilement devenir surabonnées. Considérez cette topologie : deux zones sont connectées via deux chemins, ne permettant qu'un appel par chemin. Si la bande passante est limitée par un appel, le deuxième chemin ne sera jamais utilisé. Mais si la bande passante est limitée par deux appels, l'une des liaisons peut être surabonnée. Cette commande peut donc être appliquée aux zones qui n'ont qu'un seul chemin vers toutes les autres zones. La topologie en étoile est une exception. Bien que le concentrateur dispose de plusieurs chemins, il ne surabonnera pas les liaisons, car le nombre d'appels sera limité aux rayons de chaque liaison.

bande passante {interzone | total | session} {valeur par défaut | zone *zone-name*} *bandwidth size*

aucune bande passante {interzone | total | session} {valeur par défaut | zone *nom-zone*} *taille-bande*

[Description de la syntaxe](#)

Le tableau suivant décrit la syntaxe :

Syntaxe	Description
interzone	Spécifie la quantité totale de bande passante pour le trafic H.323 de la zone vers toute autre zone.

total	Spécifie la quantité totale de bande passante pour le trafic H.323 autorisé dans la zone.
session	Spécifie la bande passante maximale autorisée pour une session dans la zone.
par défaut	Spécifie la valeur par défaut pour toutes les zones.
zone nom_zone	Spécifie une zone particulière. Nomme la zone particulière.
taille de bande passant e	Bande passante maximale. Pour l' interzone et le total , la plage est comprise entre 1 et 10 000 000 kbits/s. Pour le session , la plage est comprise entre 1 et 5 000 kbits/s.

[Valeurs par défaut](#)

Aucune

[Modes de commande](#)

Configuration du contrôleur d'accès

[Historique des commandes](#)

Le tableau suivant décrit l'historique des commandes :

Libérer	–
12.1(3)XI	La commande suivante a été ajoutée.
12.1(5)XM	La commande bandwidth a été rendue reconnaissable sans utiliser la commande zone gatekeeper .
12.2(2)T	Cette commande a été intégrée au logiciel Cisco IOS Version 12.2(2)T.
12.2(2)XB1	Cette commande a été implémentée sur la passerelle universelle Cisco AS5850.

[Directives d'utilisation](#)

Dans les versions précédentes du logiciel Cisco IOS, la fonctionnalité de la commande **bandwidth** a été obtenue à l'aide de la commande **zone gatekeeper**.

[Exemples](#)

L'exemple suivant configure la bande passante maximale de la zone à 5 000 kbits/s :

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

Commandes associées

bandwidth remote : spécifie la bande passante totale pour le trafic H.323 entre ce contrôleur d'accès et tout autre contrôleur d'accès.

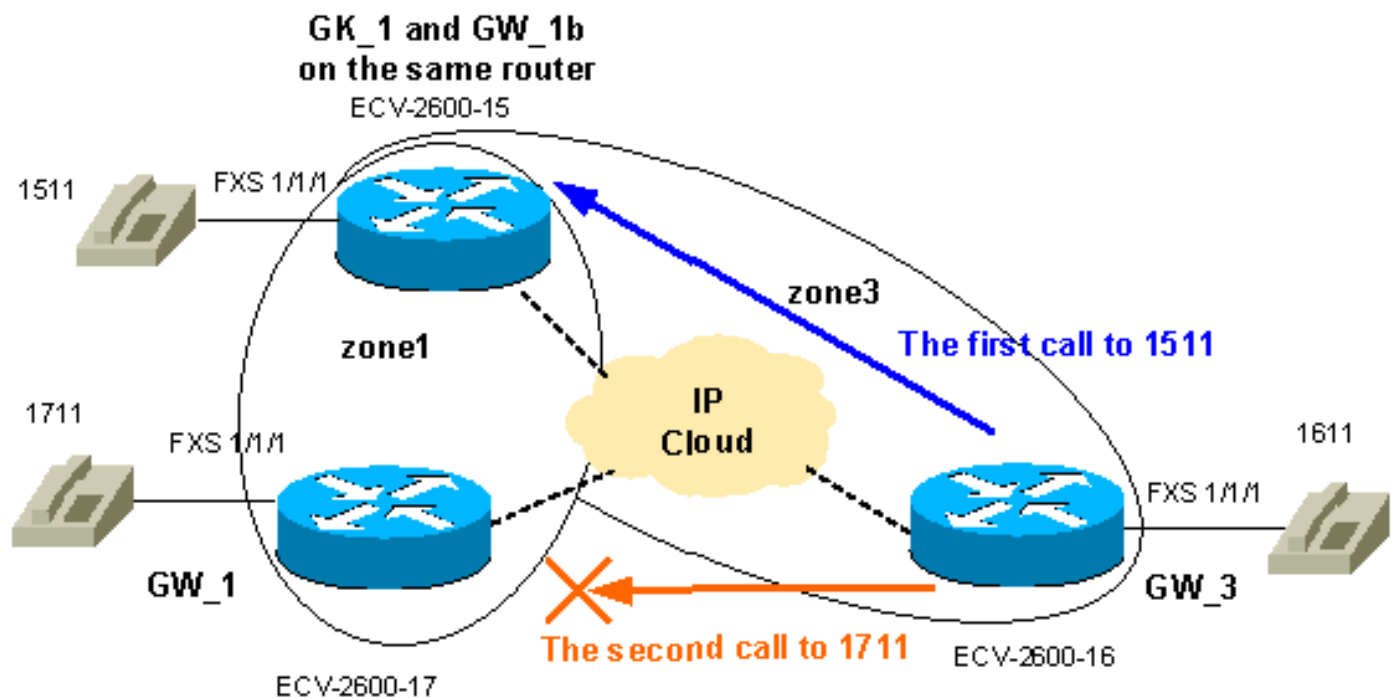
Configuration

Cette section vous fournit des informations pour configurer les fonctionnalités décrites dans ce document.

Remarque : Pour en savoir plus sur les commandes utilisées dans le présent document, utilisez [l'outil de recherche de commandes](#) (clients [inscrits](#) seulement).

Diagramme du réseau

Ce document utilise la configuration réseau suivante :



Configurations

L'objectif est de limiter la bande passante disponible entre zone1 et zone3 à un seul appel et d'autoriser un nombre plus élevé d'appels (jusqu'à deux dans cet exemple) dans zone1. Nous répondrons donc aux exigences générales pour la tâche d'admission d'appel type. Les messages RAS (Registration, Admission, and Status Protocol) sont transmis avant les messages de configuration d'appel H225. Puis la négociation H4245 suit, qui définit en fait les capacités des côtés. Ainsi, la bande passante réelle de l'appel est définie après l'étape d'admission de l'appel et l'échange des messages RAS. C'est pourquoi le contrôleur d'accès traite chaque appel comme un appel de 64 Ko. Par conséquent, l'augmentation des limitations de bande passante entre les zones pour les appels vocaux doit être effectuée par incréments de 64 Ko.

Remarque : le GW_3 est configuré sur le même routeur que le contrôleur d'accès pour illustrer

cette possibilité pour les filiales de bas de gamme.

Remarque : La vérification des configurations des contrôleurs d'accès et des passerelles est la partie importante du dépannage des problèmes GK-GW. Par conséquent, pour simplifier la compréhension des configurations, toutes les commandes de configuration non liées ont été supprimées.

GW_1 ECV-2600-17

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

GW_2 ECV-2600-16

```
!  
hostname ECV-2610-16  
!  
!  
interface Ethernet0/0  
 ip address 10.52.218.48 255.255.255.0  
 h323-gateway voip interface  
 h323-gateway voip id gk-zone3.test.com ipaddr  
10.52.218.47 1718  
 h323-gateway voip h323-id gw_3  
 h323-gateway voip tech-prefix 1#  
 h323-gateway voip bind srcaddr 10.52.218.48  
!  
!  
voice-port 1/1/0  
!  
!
```

```

voice-port 1/1/1
!
dial-peer voice 1 voip
  destination-pattern ....
  session target ras
!
dial-peer voice 2 pots
  destination-pattern 1611
  port 1/1/1
  no register e164
!
gateway
!
!
end

```

GK_1 ECV-2600-15

```

hostname ECV-2610-15
!
boot system tftp c2600-jsx-mz.122-7a.bin 10.52.218.2
!
interface Ethernet0/0
  ip address 10.52.218.47 255.255.255.0
  half-duplex
  h323-gateway voip interface
  h323-gateway voip id gk-zone1.test.com ipaddr
10.52.218.47 1718
  h323-gateway voip h323-id gw_1b
  h323-gateway voip tech-prefix 1#
  h323-gateway voip bind srcaddr 10.52.218.47
!
!
voice-port 1/1/0
!
voice-port 1/1/1
!
!
dial-peer voice 6 pots
  destination-pattern 1511
  port 1/1/1
  no register e164
!
!
dial-peer voice 5 voip
  destination-pattern ....
  session target ras
!
gateway
!
!
gatekeeper
  zone local gk-zone1.test.com test.com 10.52.218.47
  zone local gk-zone3.test.com test.com
  zone prefix gk-zone1.test.com 15.. gw-priority 10 gw_1b
  zone prefix gk-zone3.test.com 16.. gw-priority 10 gw_3
  zone prefix gk-zone1.test.com 17.. gw-priority 10 gw_1
  gw-type-prefix 1#* default-technology
bandwidth interzone zone gk-zone1.test.com 64
!--- Applies the restriction between gk-zone1, and all
!--- other zones to 64kb. That allows one call only.
bandwidth total zone gk-zone1.test.com 128

```



```
!--- Applies the restriction to the total number of
calls in zone1, !--- and allows two call in the gk-
zone1. no shutdown
!
end

ECV-2610-15#
```

Vérification

Cette section présente des informations que vous pouvez utiliser pour vous assurer que votre configuration fonctionne correctement.

Certaines commandes **show** sont prises en charge par l'[Output Interpreter Tool](#) (clients enregistrés uniquement), qui vous permet de voir une analyse de la sortie de la commande show.

- **show gateway** : affiche l'état d'enregistrement de la passerelle.
- **show gatekeeper endpoints** : répertorie toutes les passerelles enregistrées dans le gatekeeper.
- **show gatekeeper zone prefix** - affiche tous les préfixes de zone configurés sur le gatekeeper.
- **show gatekeeper call** : affiche les appels actifs traités par le gatekeeper.

Dépannage

Cette section fournit des informations que vous pouvez utiliser pour dépanner votre configuration.

Dépannage des commandes

Certaines commandes **show** sont prises en charge par l'[Output Interpreter Tool](#) (clients enregistrés uniquement), qui vous permet de voir une analyse de la sortie de la commande show.

Note : Avant d'émettre des commandes **debug**, consultez [Informations importantes sur les commandes de débogage](#).

- **debug h225 asn1** - affiche les messages H225 (configuration des appels RAS et Q931).
- **debug cch323 h225** - affiche les messages de configuration d'appel H225.

Voici quelques liens utiles :

- [Dépannage et débogage des appels VoIP - Notions de base](#)
- [Commandes de débogage VoIP](#)
- [Référence des commandes de voix, vidéo et télécopie Cisco IOS, version 12.2](#)

Exemple de résultat show and debug

```
!--- First step is to check the gateway registrations. !--- On the first gateway:
```

```
ECV-2610-17#show gateway
Gateway gw_1 is registered to Gatekeeper gk-zone1.test.com
```

Alias list (CLI configured)

H323-ID gw_1

Alias list (last RCF)

H323-ID gw_1

H323 resource thresholding is Disabled

ECV-2610-17#

!--- And on the second Gateway: ECV-2610-16#show gateway
Gateway gw_3 is registered to Gatekeeper gk-zone3.test.com

Alias list (CLI configured)

H323-ID gw_3

Alias list (last RCF)

H323-ID gw_3

H323 resource thresholding is Disabled

ECV-2610-16#

!--- The same on the third Gateway: ECV-2610-15#show gateway
Gateway gw_1b is registered to Gatekeeper gk-zone1.test.com

Alias list (CLI configured)

H323-ID gw_1b

Alias list (last RCF)

H323-ID gw_1b

H323 resource thresholding is Disabled

ECV-2610-15#

!--- And on the corresponding Gatekeeper: ECV-2610-15#show gatekeeper end

GATEKEEPER ENDPOINT REGISTRATION

=====

Table with 6 columns: CallSignalAddr, Port, RASSignalAddr, Port, Zone Name, Type F. It lists three active registrations for gatekeepers gw_1b, gw_3, and gw_1.

Total number of active registrations = 3

ECV-2610-15#

!--- To check the dial plan on the Gatekeeper:

ECV-2610-15#show gatekeeper zone pre

ZONE PREFIX TABLE

=====

GK-NAME E164-PREFIX

- gk-zone1.test.com 15..
gk-zone3.test.com 16..
gk-zone1.test.com 17..

ECV-2610-15#

!--- All configured prefixes should be seen in the zone list.

!--- To check the zone status on the Gatekeeper: !-- The output shows one permitted interzone call.

ECV-2610-15#show gatekeeper zone st

GATEKEEPER ZONES

=====

GK name Domain Name RAS Address PORT FLAGS

!--- The output shows the bandwidth restrictions for this zone. gk-zone1.test.com
10.52.218.47 1719 LS

BANDWIDTH INFORMATION (kbps) :

Maximum total bandwidth : 128
Current total bandwidth : 64

Maximum interzone bandwidth : 64
Current interzone bandwidth : 64

Maximum session bandwidth :

Total number of concurrent calls : 1

SUBNET ATTRIBUTES :

All Other Subnets : (Enabled)

PROXY USAGE CONFIGURATION :

Inbound Calls from all other zones :

to terminals in local zone gk-zone1.test.com : use proxy
to gateways in local zone gk-zone1.test.com : do not use proxy
to MCUs in local zone gk-zone1.test.com : do not use proxy

Outbound Calls to all other zones :

from terminals in local zone gk-zone1.test.com : use proxy
from gateways in local zone gk-zone1.test.com : do not use proxy
from MCUs in local zone gk-zone1.test.com : do not use proxy

!--- There are no bandwidth restrictions for this zone. gk-zone3.test.com 10.52.218.47 1719
LS

BANDWIDTH INFORMATION (kbps) :

Maximum total bandwidth :
Current total bandwidth : 64

Maximum interzone bandwidth :
Current interzone bandwidth : 64

Maximum session bandwidth :

Total number of concurrent calls : 1

SUBNET ATTRIBUTES :

All Other Subnets : (Enabled)

PROXY USAGE CONFIGURATION :

Inbound Calls from all other zones :

to terminals in local zone gk-zone3.test.com : use proxy
to gateways in local zone gk-zone3.test.com : do not use proxy
to MCUs in local zone gk-zone3.test.com : do not use proxy

Outbound Calls to all other zones :

from terminals in local zone gk-zone3.test.com : use proxy
from gateways in local zone gk-zone3.test.com : do not use proxy
from MCUs in local zone gk-zone3.test.com : do not use proxy

ECV-2610-15#

ECV-2610-15#show gatekeeper call
Total number of active calls = 1.

GATEKEEPER CALL INFO

=====

LocalCallID	Age(secs)		BW			
5-0	1		64 (Kbps)			
Endpt(s):	Alias	E.164Addr	CallSignalAddr	Port	RASSignalAddr	Port
src	EP: gw_3	1611	10.52.218.48	1720	10.52.218.48	59067
dst	EP: gw_1b	1511	10.52.218.47	1720	10.52.218.47	58841

ECV-2610-15#

!--- The output shows that we reach maximum number of calls for gk-zone1. ECV-2610-15# ECV-2610-15#show gatekeeper zone st

GATEKEEPER ZONES

```

=====
GK name      Domain Name  RAS Address  PORT  FLAGS
-----
gk-zone1.tes test.com    10.52.218.47  1719  LS

```

BANDWIDTH INFORMATION (kbps) :

Maximum total bandwidth : 128

Current total bandwidth : 128

Maximum interzone bandwidth : 64

Current interzone bandwidth : 64

Maximum session bandwidth :

Total number of concurrent calls : 2

SUBNET ATTRIBUTES :

All Other Subnets : (Enabled)

PROXY USAGE CONFIGURATION :

Inbound Calls from all other zones :

to terminals in local zone gk-zone1.test.com : use proxy

to gateways in local zone gk-zone1.test.com : do not use proxy

to MCUs in local zone gk-zone1.test.com : do not use proxy

Outbound Calls to all other zones :

from terminals in local zone gk-zone1.test.com : use proxy

from gateways in local zone gk-zone1.test.com : do not use proxy

from MCUs in local zone gk-zone1.test.com : do not use proxy

```

gk-zone3.tes test.com    10.52.218.47  1719  LS

```

BANDWIDTH INFORMATION (kbps) :

Maximum total bandwidth :

Current total bandwidth : 64

Maximum interzone bandwidth :

Current interzone bandwidth : 64

Maximum session bandwidth :

Total number of concurrent calls : 1

SUBNET ATTRIBUTES :

All Other Subnets : (Enabled)

PROXY USAGE CONFIGURATION :

Inbound Calls from all other zones :

to terminals in local zone gk-zone3.test.com : use proxy

to gateways in local zone gk-zone3.test.com : do not use proxy

to MCUs in local zone gk-zone3.test.com : do not use proxy

Outbound Calls to all other zones :

from terminals in local zone gk-zone3.test.com : use proxy

from gateways in local zone gk-zone3.test.com : do not use proxy

from MCUs in local zone gk-zone3.test.com : do not use proxy

```

gk-zone2.tes test.com    10.52.218.46  1719  RS

```

ECV-2610-15#

ECV-2610-15#show gatekeeper call

Total number of active calls = 2.

GATEKEEPER CALL INFO

=====

LocalCallID	Age(secs)	BW				
20-33504	49	64 (kbps)				
Endpt(s): Alias	E.164Addr	CallSignalAddr	Port	RASSignalAddr	Port	
src EP: gw_3	1611	10.52.218.48	1720	10.52.218.48	49762	
dst EP: gw_1b	1510	10.52.218.47	1720	10.52.218.47	52344	
LocalCallID	Age(secs)	BW				
21-22720 36		64 (Kbps)				
Endpt(s): Alias	E.164Addr	CallSignalAddr	Port	RASSignalAddr	Port	
src EP: gw_1	1711	10.52.218.49	1720	10.52.218.49	54114	
dst EP: gw_1b	1511	10.52.218.47	1720	10.52.218.47	52344	

ECV-2610-15#

!--- The conversation between the gateway and gatekeeper consists of !--- an exchange of RAS messages. !--- We start call to 1511 from GW_3. ECV-2610-16#deb h225 asn1
H.225 ASN1 Messages debugging is on
ECV-2610-16#

*Mar 1 14:22:20.972: **RAS OUTGOING PDU ::=**

```
value RasMessage ::= admissionRequest :
{
  requestSeqNum 970
  callType pointToPoint : NULL
  callModel direct : NULL
  endpointIdentifier {"8262B76400000019"}
  destinationInfo
{
  e164 : "1511"
}
  srcInfo
{
  h323-ID : {"gw_3"}
}
  bandwidth 640
  callReferenceValue 23
  nonStandardData
{
  nonStandardIdentifier h221NonStandard :
{
  t35CountryCode 181
  t35Extension 0
  manufacturerCode 18
}
  data '000000'H
}
  conferenceID '00000000000000000000000000000000'H
  activeMC FALSE
  answerCall FALSE
  canMapAlias TRUE
  callIdentifier
{
  guid '00000000000000000000000000000000'H
}
  willSupplyUUIEs FALSE
}
```

*Mar 1 14:22:20.992: RAS OUTGOING ENCODE BUFFER::= 27 8803C900 F0003800 32003600
32004200 37003600 34003000 30003000 30003000 30003100 39010180 48440140 03006700
77005F00 33400280 001740B5 00001203 00000000 00000000 00000000 00000000 00000004
E0200180 11000000 00000000 00000000 00000000 00000100

*Mar 1 14:22:21.008:

*Mar 1 14:22:21.073: RAS INCOMING ENCODE BUFFER::= 2B 0003C940 0280000A 34DA2F06
B800EF14 00C00100 020000

*Mar 1 14:22:21.077:

*Mar 1 14:22:21.081: **RAS INCOMING PDU ::=**

!--- The GW_3 gets permission to proceed with that call. value RasMessage ::= admissionConfirm :

```
{
  requestSeqNum 970
  bandwidth 640
  callModel direct : NULL
  destCallSignalAddress ipAddress :
{
  ip '0A34DA2F'H
}
```



```

{
'0000000D4001800A040001000A34DA3041C5'H,
'400000060401004D40018011140001000A34DA30...'H
}
mediaWaitForConnect FALSE
canOverlapSend FALSE
}
h245Tunneling FALSE
}
}

*Mar 1 14:22:21.141: H225.0 OUTGOING ENCODE BUFFER ::= 20
A0060008 914A0002 01400300
67007700 5F003308 80013C05 04010020 40000000 00000000 00000000 00000000 00000045
1C07000A 34DA302B 0A110000 00000000 00000000 00000000 00000032 02120000 000D4001
800A0400 01000A34 DA3041C5 1D400000 06040100 4D400180 11140001 000A34DA 3041C400
0A34DA30 41C50100 01000680 0100
*Mar 1 14:22:21.161:
*Mar 1 14:22:21.417: H225.0 INCOMING ENCODE BUFFER ::= 21
80060008 914A0002 00048811
00000000 00000000 00000000 00000000 00390219 0000000D 40018011 14000100 0A34DA2F
486E000A 34DA2F48 6F1D4000 00060401 004D4001 80111400 01000A34 DA3041C4 000A34DA
2F486F06 800100
*Mar 1 14:22:21.429:
*Mar 1 14:22:21.429: H225.0 INCOMING PDU ::=
!--- The GW_3 gets Call Proceeding from GW_1b. value H323_UserInformation ::= { h323-uu-pdu {
h323-message-body callProceeding :
{
protocolIdentifier { 0 0 8 2250 0 2 }
destinationInfo
{
mc FALSE
undefinedNode FALSE
}
callIdentifier
{
guid '00000000000000000000000000000000'H
}
fastStart
{
'0000000D40018011140001000A34DA2F486E000A...'H,
'400000060401004D40018011140001000A34DA30...'H
}
}
h245Tunneling FALSE
}
}

*Mar 1 14:22:21.617: H225.0 INCOMING ENCODE BUFFER ::= 28
001A0006 0008914A 00020000
00000000 00000000 00000000 00000000 06A00100 120140B5 0000120B 60011000 011E041E
028188
*Mar 1 14:22:21.626:
*Mar 1 14:22:21.626: H225.0 INCOMING PDU ::=
!--- The GW_3 geta Call Progress from GW_1b. value H323_UserInformation ::= { h323-uu-pdu {
h323-message-body progress :
{
protocolIdentifier { 0 0 8 2250 0 2 }
destinationInfo
{
mc FALSE
undefinedNode FALSE
}
}
}

```

```

}
callIdentifier
{
guid '00000000000000000000000000000000'H
}
}
h245Tunneling FALSE
nonStandardControl
{
{
nonStandardIdentifier h221NonStandard :
{
t35CountryCode 181
t35Extension 0
manufacturerCode 18
}
}
data '60011000011E041E028188'

}
}
}
}

```

```

*Mar 1 14:22:21.642: H225 NONSTD INCOMING ENCODE BUFFER ::= 60
01100001 1E041E02 8188
*Mar 1 14:22:21.646:
*Mar 1 14:22:21.646: H225 NONSTD INCOMING PDU ::=
!--- The GW_3 get some facility messages from GW_1b. value H323_UU_NonStdInfo ::= { version 16
protoParam qsigNonStdInfo :
{
iei 30
rawMesg '1E028188'H
}
}

```

```

*Mar 1 14:22:22.831: %SYS-3-MGDTIMER: Running timer, init, timer = 81F1AC08.
-Process= "Virtual Exec", ipl= 0, pid= 61
-Traceback= 803250A4 80325214 80325318 80EB12C0
80EB17DC 802A65F0 802B5080 8033D818
*Mar 1 14:22:22.835: H225 NONSTD OUTGOING PDU ::=
value ARQnonStandardInfo ::=
{
sourceAlias
{
}
sourceExtAlias
{
}
}

```

```

*Mar 1 14:22:22.839: H225 NONSTD OUTGOING ENCODE
BUFFER ::= 00 0000
*Mar 1 14:22:22.839:
*Mar 1 14:22:22.839: RAS OUTGOING PDU ::=
!--- The GW_3 starts the second Call to 1711 now we send RAS message to GK. value RasMessage ::=
admissionRequest :
{

```



```

requestSeqNum 971
callType pointToPoint : NULL
callModel direct : NULL
endpointIdentifier {"8262B76400000019"}
destinationInfo
{
  e164 : "1711"
}
srcInfo
{
  h323-ID : {"gw_3"}
}
bandWidth 640
callReferenceValue 24
nonStandardData
{
  nonStandardIdentifier h221NonStandard :
{
  t35CountryCode 181
  t35Extension 0
  manufacturerCode 18
}
  data '000000'H
}
  conferenceID '00000000000000000000000000000000'H
  activeMC FALSE
  answerCall FALSE
  canMapAlias TRUE
  callIdentifier
{
  guid '00000000000000000000000000000000'H
}
  willSupplyUUIEs FALSE
}

```

```

*Mar 1 14:22:22.860: RAS OUTGOING ENCODE BUFFER ::= 27 8803CA00 F0003800 32003600
32004200 37003600 34003000 30003000 30003000 30003100 39010180 4A440140 03006700
77005F00 33400280 001840B5 00001203 00000000 00000000 00000000 00000000 00000004
E0200180 11000000 00000000 00000000 00000000 00000100

```

```

*Mar 1 14:22:22.876:

```

```

*Mar 1 14:22:22.940: RAS INCOMING ENCODE BUFFER ::= 2B 0003CA40 0280000A 34DA3106
B800EF14 00C00100 020000

```

```

*Mar 1 14:22:22.944:

```

```

*Mar 1 14:22:22.944: RAS INCOMING PDU ::=

```

```

!--- The GW_3 gets permission to proceed as there are no restrictions on zone3. value RasMessage
::= admissionConfirm :

```

```

{
  requestSeqNum 971
  bandWidth 640
  callModel direct : NULL
  destCallSignalAddress ipAddress :
{
  ip '0A34DA31'H
  port 1720
}
  irrFrequency 240
  willRespondToIRR FALSE
  uuiesRequested
{
  setup FALSE
  callProceeding FALSE
  connect FALSE
}

```

```
alerting FALSE
information FALSE
releaseComplete FALSE
facility FALSE
progress FALSE
empty FALSE
}
}
```

*Mar 1 14:22:22.972: **H225.0 OUTGOING PDU** ::=

!--- The GW_3 sends setup message to GW_1. value H323_UserInformation ::= { h323-uu-pdu { h323-
message-body **setup** :

```
{
  protocolIdentifier { 0 0 8 2250 0 2 }
  sourceAddress
  {
    h323-ID : {"gw_3"}
  }
  sourceInfo
  {
    gateway
  {
    protocol
  {
    voice :
  {
    supportedPrefixes
  {
    {
    prefix e164 : "1#"
    }
    }
    }
    }
    }
    mc FALSE
    undefinedNode FALSE
  }
  activeMC FALSE
  conferenceID '00000000000000000000000000000000'H
  conferenceGoal create : NULL
  callType pointToPoint : NULL
  sourceCallSignalAddress ipAddress :
  {
    ip '0A34DA30'H
    port 11019
  }
  callIdentifier
  {
    guid '00000000000000000000000000000000'H
  }
  fastStart
  {
    '0000000D4001800A040001000A34DA30402F'H,
    '400000060401004D40018011140001000A34DA30...'H
  }
  mediaWaitForConnect FALSE
  canOverlapSend FALSE
  }
  h245Tunneling FALSE
  }
}
```

```
*Mar 1 14:22:23.008: H225.0 OUTGOING ENCODE BUFFER::= 20
A0060008 914A0002 01400300
67007700 5F003308 80013C05 04010020 40000000 00000000 00000000 00000000 00000045
1C07000A 34DA302B 0B110000 00000000 00000000 00000000 00000032 02120000 000D4001
800A0400 01000A34 DA30402F 1D400000 06040100 4D400180 11140001 000A34DA 30402E00
0A34DA30 402F0100 01000680 0100
*Mar 1 14:22:23.028:
*Mar 1 14:22:23.220: H225.0 INCOMING ENCODE BUFFER::= 25
80060008 914A0002 01110000
00000000 00000000 00000000 00000006 800100
*Mar 1 14:22:23.224:
*Mar 1 14:22:23.224: H225.0 INCOMING PDU ::=
!--- The GW_1 replies with Release Complete message after asking GK !--- for permission to
accept that call. !--- When the permission is denied, we set bandwidth limit. value
H323_UserInformation ::= { h323-uu-pdu { h323-message-body releaseComplete :
{
  protocolIdentifier { 0 0 8 2250 0 2 }
  callIdentifier
{
  guid '00000000000000000000000000000000'H
  }
  }
  h245Tunneling FALSE
  }
  }
}

*Mar 1 14:22:23.236: RAS OUTGOING PDU ::=
!--- The GW_3 notifies GK that the call does not exist anymore. value RasMessage ::=
disengageRequest :
{
  requestSeqNum 972
  endpointIdentifier {"8262B76400000019"}
  conferenceID '00000000000000000000000000000000'H
  callReferenceValue 24
  disengageReason normalDrop : NULL
  callIdentifier
{
  guid '00000000000000000000000000000000'H
  }
  answeredCall FALSE
  }
}

*Mar 1 14:22:23.248: RAS OUTGOING ENCODE BUFFER::= 3E 03CB1E00 38003200 36003200
42003700 36003400 30003000 30003000 30003000 31003900 00000000 00000000 00000000
00000000 18216111 00000000 00000000 00000000 00000000 000100
*Mar 1 14:22:23.256:
*Mar 1 14:22:23.288: RAS INCOMING ENCODE BUFFER::= 40
03CB
*Mar 1 14:22:23.288:
*Mar 1 14:22:23.288: RAS INCOMING PDU ::=
!--- The GK confirms that message. value RasMessage ::= disengageConfirm :
{
  requestSeqNum 972
  }
}
```

ECV-2610-16#u all
All possible debugging has been turned off
ECV-2610-16#

!--- The incoming RAS message to the GK from GW_3. ECV-2610-15#debug h225 asn1
H.225 ASN1 Messages debugging is on
ECV-2610-15#
*Mar 11 21:54:28.313: **RAS INCOMING PDU** ::=

```
value RasMessage ::= admissionRequest :
{
  requestSeqNum 970
  callType pointToPoint : NULL
  callModel direct : NULL
  endpointIdentifier {"8262B76400000019"}
  destinationInfo
  {
    e164 : "1511"
  }
  srcInfo
  {
    h323-ID : {"gw_3"}
  }
  bandwidth 640
  callReferenceValue 23
  nonStandardData
  {
    nonStandardIdentifier h221NonStandard :
  {
    t35CountryCode 181
    t35Extension 0
    manufacturerCode 18
  }
    data '000000'H
  }
    conferenceID '00000000000000000000000000000000'H
    activeMC FALSE
    answerCall FALSE
    canMapAlias TRUE
    callIdentifier
  {
    guid '00000000000000000000000000000000'H
  }
    willSupplyUUIEs FALSE
  }
}
```

*Mar 11 21:54:28.334: H225 NONSTD INCOMING ENCODE BUFFER::= 00 0000
*Mar 11 21:54:28.334:
*Mar 11 21:54:28.334: H225 NONSTD INCOMING PDU ::=

```
value ARQnonStandardInfo ::=
{
  sourceAlias
  {
  }
  sourceExtAlias
  {
  }
}
```

!--- The outgoing RAS message fro GK to GW_3 with permission to start call. *Mar 11
21:54:28.338: **RAS OUTGOING PDU** ::=

```
value RasMessage ::= admissionConfirm :
{
  requestSeqNum 970
  bandWidth 640
  callModel direct : NULL
  destCallSignalAddress ipAddress :
  {
    ip '0A34DA2F'H
    port 1720
  }
  irrFrequency 240
  willRespondToIRR FALSE
  uuiesRequested
  {
    setup FALSE
    callProceeding FALSE
    connect FALSE
    alerting FALSE
    information FALSE
    releaseComplete FALSE
    facility FALSE
    progress FALSE
    empty FALSE
  }
}
```

*Mar 11 21:54:28.350: RAS OUTGOING ENCODE BUFFER::= 2B 0003C940 0280000A 34DA2F06
B800EF14 00C00100 020000

*Mar 11 21:54:28.354:

*Mar 11 21:54:28.446: H225.0 INCOMING ENCODE BUFFER::= 20

```
A0060008 914A0002 01400300
67007700 5F003308 80013C05 04010020 40000000 00000000 00000000 00000000 00000045
1C07000A 34DA302B 0A110000 00000000 00000000 00000000 00000032 02120000 000D4001
800A0400 01000A34 DA3041C5 1D400000 06040100 4D400180 11140001 000A34DA 3041C400
0A34DA30 41C50100 01000680 0100
```

*Mar 11 21:54:28.466:

*Mar 11 21:54:28.470: **H225.0 INCOMING PDU** ::=

!--- The incoming H323(Q931) message from GW_3 to GW_1b on the same router as GK. value
H323_UserInformation ::= { h323-uu-pdu { h323-message-body **setup** :

```
{
  protocolIdentifier { 0 0 8 2250 0 2 }
  sourceAddress
  {
    h323-ID : {"gw_3"}
  }
  sourceInfo
  {
    gateway
  {
    protocol
  {
    voice :
  {
    supportedPrefixes
  {
    {
    prefix e164 : "1#"
    }
  }
  }
}
```

```

}
}
}
mc FALSE
undefinedNode FALSE
}
activeMC FALSE
conferenceID '00000000000000000000000000000000'H
conferenceGoal create : NULL
callType pointToPoint : NULL
sourceCallSignalAddress ipAddress :
{
ip '0A34DA30'H
port 11018
}
callIdentifier
{
guid '00000000000000000000000000000000'H
}
fastStart
{
'0000000D4001800A040001000A34DA3041C5'H,
'400000060401004D40018011140001000A34DA30...'H
}
mediaWaitForConnect FALSE
canOverlapSend FALSE
}
h245Tunneling FALSE
}
}

```

```

*Mar 11 21:54:28.514: H225 NONSTD OUTGOING PDU ::=
value ARQnonStandardInfo ::=

```

```

{
sourceAlias
{
}
sourceExtAlias
{
}
}

```

```

*Mar 11 21:54:28.518: H225 NONSTD OUTGOING ENCODE BUFFER ::= 00 0000

```

```

*Mar 11 21:54:28.518:

```

```

*Mar 11 21:54:28.518: RAS OUTGOING PDU ::=

```

```

!--- The GW_1b asks GK if it can accept call from GW_3. value RasMessage ::= admissionRequest :

```

```

{
requestSeqNum 1347
callType pointToPoint : NULL
callModel direct : NULL
endpointIdentifier {"82717F5C0000001B"}
destinationInfo
{
e164 : "1511"
}
srcInfo
{
h323-ID : {"gw_3"}
}
srcCallSignalAddress ipAddress :
{

```

```

ip '0A34DA30'H
port 11018
}
bandWidth 640
callReferenceValue 29
nonStandardData
{
nonStandardIdentifier h221NonStandard :
{
t35CountryCode 181
t35Extension 0
manufacturerCode 18
}
data '000000'H
}
conferenceID '00000000000000000000000000000000'H
activeMC FALSE
answerCall TRUE
canMapAlias TRUE
callIdentifier
{
guid '00000000000000000000000000000000'H
}
willSupplyUUIEs FALSE
}

```

```

*Mar 11 21:54:28.542: RAS OUTGOING ENCODE BUFFER ::= 27 98054200 F0003800 32003700
31003700 46003500 43003000 30003000 30003000 30003100 42010180 48440140 03006700
77005F00 33000A34 DA302B0A 40028000 1D40B500 00120300 00000000 00000000 00000000
00000000 000044E0 20018011 00000000 00000000 00000000 00000000 000100

```

```

*Mar 11 21:54:28.558:

```

```

*Mar 11 21:54:28.562: RAS INCOMING ENCODE BUFFER ::= 27
98054200 F0003800 32003700 31003700 46003500 43003000 30003000 30003000 30003100
42010180 48440140 03006700 77005F00 33000A34 DA302B0A 40028000 1D40B500 00120300
00000000 00000000 00000000 00000000 000044E0 20018011 00000000 00000000 00000000
00000000 000100

```

```

*Mar 11 21:54:28.578:

```

```

*Mar 11 21:54:28.582: RAS INCOMING PDU ::=

```

```

!--- That is the same RAS message. The GK gets it, and sees the sequence number. !--- The GK is
on the same router as GW_1b, so all messages can be seen twice. value RasMessage ::=

```

```

admissionRequest :

```

```

{
requestSeqNum 1347
callType pointToPoint : NULL
callModel direct : NULL
endpointIdentifier {"82717F5C0000001B"}
destinationInfo
{
e164 : "1511"
}
srcInfo
{
h323-ID : {"gw_3"}
}
srcCallSignalAddress ipAddress :
{
ip '0A34DA30'H
port 11018
}
bandWidth 640
callReferenceValue 29
nonStandardData

```

```
{
  nonStandardIdentifier h221NonStandard :
  {
    t35CountryCode 181
    t35Extension 0
    manufacturerCode 18
  }
  data '000000'
  }
  conferenceID '00000000000000000000000000000000'H
  activeMC FALSE
  answerCall TRUE
  canMapAlias TRUE
  callIdentifier
  {
    guid '00000000000000000000000000000000'H
  }
  willSupplyUUIEs FALSE
  }
}
```

*Mar 11 21:54:28.606: H225 NONSTD INCOMING ENCODE BUFFER ::= 00 0000

*Mar 11 21:54:28.606:

*Mar 11 21:54:28.606: H225 NONSTD INCOMING PDU ::=

value ARQnonStandardInfo ::=

```
{
  sourceAlias
  {
  }
  sourceExtAlias
  {
  }
  }
}
```

*Mar 11 21:54:28.610: **RAS OUTGOING PDU** ::=

!--- The GK grants the permission to GW_1b. !--- This is a message in the GK debug outgoing

value RasMessage ::= **admissionConfirm** :

```
{
  requestSeqNum 1347
  bandwidth 640
  callModel direct : NULL
  destCallSignalAddress ipAddress :
  {
    ip '0A34DA2F'H
    port 1720
  }
  irrFrequency 240
  willRespondToIRR FALSE
  uuiesRequested
  {
    setup FALSE
    callProceeding FALSE
    connect FALSE
    alerting FALSE
    information FALSE
    releaseComplete FALSE
    facility FALSE
    progress FALSE
    empty FALSE
  }
  }
}
```



```

*Mar 11 21:54:28.622: RAS OUTGOING ENCODE BUFFER::= 2B 00054240 0280000A 34DA2F06
B800EF14 00C00100 020000
*Mar 11 21:54:28.626:
*Mar 11 21:54:28.630: RAS INCOMING ENCODE BUFFER::= 2B 00054240 0280000A 34DA2F06
B800EF14 00C00100 020000
*Mar 11 21:54:28.634:
*Mar 11 21:54:28.634: RAS INCOMING PDU ::=
!--- The GK grants the permission to GW_1b. !--- This is a message in the GW_1b debug incoming.
value RasMessage ::= admissionConfirm :
{
  requestSeqNum 1347
  bandwidth 640
  callModel direct : NULL
  destCallSignalAddress ipAddress :
{
  ip '0A34DA2F'H
  port 1720
}
  irrFrequency 240
  willRespondToIRR FALSE
  uuiesRequested
{
  setup FALSE
  callProceeding FALSE
  connect FALSE
  alerting FALSE
  information FALSE
  releaseComplete FALSE
  facility FALSE
  progress FALSE
  empty FALSE
}
}

*Mar 11 21:54:28.654: %SYS-3-MGDTIMER: Timer has parent, timer link, timer =
820AE990.
-Process= "CC-API_VCM", ipl= 6, pid= 93
-Traceback= 80325850 8032A720 80E74850 8033D818
*Mar 11 21:54:28.666: H225.0 OUTGOING PDU ::=
!--- The GW_1b replies to GW_3 setup message. value H323_UserInformation ::= { h323-uu-pdu {
h323-message-body callProceeding :
{
  protocolIdentifier { 0 0 8 2250 0 2 }
  destinationInfo
{
  mc FALSE
  undefinedNode FALSE
}
  callIdentifier
{
  guid '00000000000000000000000000000000'H
}
  fastStart
{
  '0000000D40018011140001000A34DA2F486E000A...'H,
  '400000060401004D40018011140001000A34DA30...'H
}
}
h245Tunneling FALSE
}

```

}

*Mar 11 21:54:28.682: H225.0 OUTGOING ENCODE BUFFER ::= 21 80060008 914A0002 00048811
00000000 00000000 00000000 00000000 00390219 0000000D 40018011 14000100 0A34DA2F
486E000A 34DA2F48 6F1D4000 00060401 004D4001 80111400 01000A34 DA3041C4 000A34DA
2F486F06 800100

*Mar 11 21:54:28.694:

*Mar 11 21:54:28.710: H225 NONSTD OUTGOING PDU ::=

value H323_UU_NonStdInfo ::=

```
{
version 16
protoParam qsigNonStdInfo :
{
iei 30
rawMesg '1E028188'H
}
}
```

*Mar 11 21:54:28.714: H225 NONSTD OUTGOING ENCODE BUFFER ::= 60 01100001 1E041E02 8188

*Mar 11 21:54:28.714:

*Mar 11 21:54:28.714: **H225.0 OUTGOING PDU** ::=

!--- The GW_1b replies to GW_3 setup message and sends second message. value

H323_UserInformation ::= { h323-uu-pdu { h323-message-body **progress** :

```
{
protocolIdentifier { 0 0 8 2250 0 2 }
destinationInfo
{
mc FALSE
undefinedNode FALSE
}
callIdentifier
{
guid '00000000000000000000000000000000'H
}
}
h245Tunneling FALSE
nonStandardControl
{
{
nonStandardIdentifier h221NonStandard :
{
t35CountryCode 181
t35Extension 0
manufacturerCode 18
}
data '60011000011E041E028188'H
}
}
}
```

*Mar 11 21:54:28.734: H225.0 OUTGOING ENCODE BUFFER ::= 28 001A0006 0008914A 00020000
00000000 00000000 00000000 00000000 06A00100 120140B5 0000120B 60011000 011E041E
028188

*Mar 11 21:54:28.742:

*Mar 11 21:54:30.161: RAS INCOMING ENCODE BUFFER ::= 27 8803CA00 F0003800 32003600
32004200 37003600 34003000 30003000 30003000 30003100 39010180 4A440140 03006700

77005F00 33400280 001840B5 00001203 00000000 00000000 00000000 00000000 00000004
E0200180 11000000 00000000 00000000 00000000 00000100

*Mar 11 21:54:30.177:

*Mar 11 21:54:30.181: **RAS INCOMING PDU** ::=

!--- The GK gets ARQ from GW_3 for the second call. value RasMessage ::= **admissionRequest**:

```
{
  requestSeqNum 971
  callType pointToPoint : NULL
  callModel direct : NULL
  endpointIdentifier {"8262B76400000019"}
  destinationInfo
  {
    e164 : "1711"
  }
  srcInfo
  {
    h323-ID : {"gw_3"}
  }
  bandwidth 640
  callReferenceValue 24
  nonStandardData
  {
    nonStandardIdentifier h221NonStandard :
    {
      t35CountryCode 181
      t35Extension 0
      manufacturerCode 18
    }
    data '000000'H
  }
  conferenceID '00000000000000000000000000000000'H
  activeMC FALSE
  answerCall FALSE
  canMapAlias TRUE
  callIdentifier
  {
    guid '00000000000000000000000000000000'H
  }
  willSupplyUUIEs FALSE
}
```

*Mar 11 21:54:30.197: H225 NONSTD INCOMING ENCODE BUFFER ::= 00 0000

*Mar 11 21:54:30.201:

*Mar 11 21:54:30.201: H225 NONSTD INCOMING PDU ::=

value ARQnonStandardInfo ::=

```
{
  sourceAlias
  {
  }
  sourceExtAlias
  {
  }
}
```

*Mar 11 21:54:30.205: **RAS OUTGOING PDU** ::=

!--- The GK grants permission to GW_3, as there are no restrictions for zone3. value RasMessage

::= **admissionConfirm** :

```
{
  requestSeqNum 971
```

```
bandWidth 640
callModel direct : NULL
destCallSignalAddress ipAddress :
{
  ip '0A34DA31'H
  !--- The hexadecimal number is 10.52.218.49, IP of GW_1. port 1720 } irrFrequency 240
willRespondToIRR FALSE uuiesRequested { setup FALSE callProceeding FALSE connect FALSE alerting
FALSE information FALSE releaseComplete FALSE facility FALSE progress FALSE empty FALSE } } *Mar
11 21:54:30.217: RAS OUTGOING ENCODE BUFFER::= 2B 0003CA40 0280000A 34DA3106 B800EF14 00C00100
020000 *Mar 11 21:54:30.221: *Mar 11 21:54:30.429: RAS INCOMING ENCODE BUFFER::= 27 98045F00
F0003800 32003300 38003600 30004400 34003000 30003000 30003000 30003100 41010180 4A440140
03006700 77005F00 33000A34 DA302B0B 40028000 2840B500 00120300 00000000 00000000 00000000
00000000 000044E0 20018011 00000000 00000000 00000000 00000000 000100 *Mar 11 21:54:30.445: *Mar
11 21:54:30.445: RAS INCOMING PDU ::=
!--- The incoming request from GW_1 asks for permission to accept call from GW_3. value
RasMessage ::= admissionRequest :
{
  requestSeqNum 1120
  callType pointToPoint : NULL
  callModel direct : NULL
  endpointIdentifier {"823860D40000001A"}
  destinationInfo
{
  e164 : "1711"
}
  srcInfo
{
  h323-ID : {"gw_3"}
}
srcCallSignalAddress ipAddress :
{
  ip '0A34DA30'H
  port 11019
}
  bandWidth 640
  callReferenceValue 40
  nonStandardData
{
  nonStandardIdentifier h221NonStandard :
{
  t35CountryCode 181
  t35Extension 0
  manufacturerCode 18
}
  data '000000'H
}
  conferenceID '00000000000000000000000000000000'H
  activeMC FALSE
  answerCall TRUE
  canMapAlias TRUE
  callIdentifier
{
  guid '00000000000000000000000000000000'H
}
  willSupplyUUIEs FALSE
}

*Mar 11 21:54:30.469: H225 NONSTD INCOMING ENCODE BUFFER::= 00 0000
*Mar 11 21:54:30.469:
*Mar 11 21:54:30.469: H225 NONSTD INCOMING PDU ::=

value ARQnonStandardInfo ::=
```

```
{
  sourceAlias
{
}
  sourceExtAlias
{
}
}
!--- The GK does not allow the call to come through, and replies with ARJ. *Mar 11
21:54:30.473: RAS OUTGOING PDU ::=
```

```
value RasMessage ::= admissionReject :
{
  requestSeqNum 1120
  rejectReason requestDenied : NULL
}
```

```
*Mar 11 21:54:30.477: RAS OUTGOING ENCODE BUFFER::= 2C 045F20
*Mar 11 21:54:30.477:
*Mar 11 21:54:30.541: RAS INCOMING ENCODE BUFFER::= 3E 03CB1E00 38003200 36003200
42003700 36003400 30003000 30003000 30003000 31003900 00000000 00000000 00000000
00000000 18216111 00000000 00000000 00000000 00000000 000100
*Mar 11 21:54:30.553:
*Mar 11 21:54:30.557: RAS INCOMING PDU ::=
```

```
!--- The GW_3 notifies GK that call does not exist anymore. value RasMessage ::=
disengageRequest :
{
  requestSeqNum 972
  endpointIdentifier {"8262B76400000019"}
  conferenceID '00000000000000000000000000000000'H
  callReferenceValue 24
  disengageReason normalDrop : NULL
  callIdentifier
{
  guid '00000000000000000000000000000000'H
}
  answeredCall FALSE
}
```

```
*Mar 11 21:54:30.565: RAS OUTGOING PDU ::=
!-- The GK confirms the message from GW_3
value RasMessage ::= disengageConfirm :
{
  requestSeqNum 972
}
```

```
-----
!--- The call setup from GW_1 perspective. ECV-2610-17#deb h225 asn1
H.225 ASN1 Messages debugging is on
ECV-2610-17#
```

```
*Mar 2 22:55:40: H225.0 INCOMING ENCODE BUFFER::= 20 A0060008 914A0002 01400300
67007700 5F003308 80013C05 04010020 40000000 00000000 00000000 00000000 00000045
1C07000A 34DA302B 0B110000 00000000 00000000 00000000 00000032 02120000 000D4001
800A0400 01000A34 DA30402F 1D400000 06040100 4D400180 11140001 000A34DA 30402E00
0A34DA30 402F0100 01000680 0100
*Mar 2 22:55:40:
*Mar 2 22:55:40: H225.0 INCOMING PDU ::=
```

```
!--- The GW_1 gets the H323 (Q931) setup message from GW_3. value H323_UserInformation ::= {
h323-uu-pdu { h323-message-body setup :
{
```

```

protocolIdentifier { 0 0 8 2250 0 2 }
sourceAddress
{
h323-ID : {"gw_3"}
}
sourceInfo
{
gateway
{
protocol
{
voice :
{
supportedPrefixes
{

{
prefix e164 : "1#"
}
}
}
}
}
mc FALSE
undefinedNode FALSE
}
activeMC FALSE
conferenceID '00000000000000000000000000000000'H
conferenceGoal create : NULL
callType pointToPoint : NULL
sourceCallSignalAddress ipAddress :
{
ip '0A34DA30'H
port 11019
}
callIdentifier
{
guid '00000000000000000000000000000000'H
}
fastStart
{
'0000000D4001800A040001000A34DA30402F'H,
'400000060401004D40018011140001000A34DA30...'H
}
mediaWaitForConnect FALSE
canOverlapSend FALSE
}
h245Tunneling FALSE
}
}

```

*Mar 2 22:55:40: H225 NONSTD OUTGOING PDU ::=

```

value ARQnonStandardInfo ::=
{
sourceAlias
{
}
sourceExtAlias
{
}
}
}

```

```

*Mar 2 22:55:40: H225 NONSTD OUTGOING ENCODE BUFFER::= 00 0000
*Mar 2 22:55:40:
*Mar 2 22:55:40: RAS OUTGOING PDU ::=
!--- The GW_1 asks GK for permission to accept the call. value RasMessage ::= admissionRequest :
{
  requestSeqNum 1120
  callType pointToPoint : NULL
  callModel direct : NULL
  endpointIdentifier {"823860D40000001A"}
  destinationInfo
{
  e164 : "1711"
}
  srcInfo
{
  h323-ID : {"gw_3"}
}
  srcCallSignalAddress ipAddress :
{
  ip '0A34DA30'H
  port 11019
}
  bandwidth 640
  callReferenceValue 40
  nonStandardData
{
  nonStandardIdentifier h221NonStandard :
{
  t35CountryCode 181
  t35Extension 0
  manufacturerCode 18
}
  data '000000'H
}
  conferenceID '00000000000000000000000000000000'H
  activeMC FALSE
  answerCall TRUE
  canMapAlias TRUE
  callIdentifier
{
  guid '00000000000000000000000000000000'H
}
  willSupplyUUIEs FALSE
}

```

```

*Mar 2 22:55:40: RAS OUTGOING ENCODE BUFFER::= 27 98045F00 F0003800 32003300
38003600 30004400 34003000 30003000 30003000 30003100 41010180 4A440140 03006700
77005F00 33000A34 DA302B0B 40028000 2840B500 00120300 00000000 00000000 00000000
00000000 000044E0 20018011 00000000 00000000 00000000 00000000 000100
*Mar 2 22:55:41:
*Mar 2 22:55:41: RAS INCOMING ENCODE BUFFER::= 2C 045F20
*Mar 2 22:55:41:
*Mar 2 22:55:41: RAS INCOMING PDU ::=
!--- The GK denies permission to accept the call from GW_3 due to bandwidth limit. value
RasMessage ::= admissionReject :
{
  requestSeqNum 1120
  rejectReason requestDenied : NULL
}

```

```
*Mar 2 22:55:41: H225.0 OUTGOING PDU ::=
!--- The GW_1 rejects call setup from GW_3. value H323_UserInformation ::= { h323-uu-pdu { h323-
message-body releaseComplete :
{
  protocolIdentifier { 0 0 8 2250 0 2 }
  callIdentifier
{
  guid '00000000000000000000000000000000'H
  }
  }
  h245Tunneling FALSE
  }
  }
```

```
*Mar 2 22:55:41: H225.0 OUTGOING ENCODE BUFFER::= 25 80060008 914A0002 01110000
00000000 00000000 00000000 00000006 800100
```

```
*Mar 2 22:55:41:
ECV-2610-17#
ECV-2610-17#
ECV-2610-17#u all
All possible debugging has been turned off
-----
```

[Informations connexes](#)

- [Présentation et dépannage de la gestion de la bande passante de Cisco Gatekeeper](#)
- [Présentation des contrôleurs d'accès H.323](#)
- [Contrôleur d'accès hautes performances Cisco](#)
- [Configuration des passerelles H.323](#)
- [Configuration des contrôleurs d'accès H.323](#)
- [Configuration de la prise en charge H.323 pour les interfaces virtuelles](#)
- [Assistance technique concernant la technologie vocale](#)
- [Assistance concernant les produits vocaux et de communications unifiées](#)
- [Dépannage des problèmes de téléphonie IP Cisco](#)
- [Support technique - Cisco Systems](#)