

Contenu

[Introduction](#)

[Aperçu d'exécution de gestion de la bande passante](#)

[Comment configurer la fonction de gestion de la bande passante sur Cisco Gatekeeper](#)

[Commandes show de garde-porte utilisées pour afficher les informations de bande passante](#)

[Messages relatifs de la bande passante RAS \(BRQ/BCF/BRJ\)](#)

[Messages RAS utilisés pour signaler l'état de bande passante](#)

[Comment BRQ est déclenché de la passerelle pour informer le garde-porte de réduire la bande passante d'appel](#)

[Exemples](#)

[Informations connexes](#)

Introduction

Ce document suppose que le lecteur est au courant d'une compréhension de base des garde-portes et du garde-porte de logiciel de Cisco IOS® à l'enregistrement de la passerelle H.225, à l'admission, et aux Messageries de l'état (RAS). Référez-vous [compréhension derrière le](#) pour en savoir plus de [Contrôleurs d'accès H.323](#).

Conformément H.323 à la recommandation, les garde-portes de Cisco IOS devraient prendre en charge ces messages de gestion de la bande passante H.225 RAS :

- Demande de bande passante (BRQ)
- Rejet de bande passante (BRJ)
- Messages de la confirmation de bande passante (BCF)

Ce concept peut être basé sur la gestion de la bande passante. Ce peut également être une fonction nulle qui reçoit toutes les demandes des modifications de bande passante. En d'autres termes, le garde-porte peut l'un ou l'autre d'utilisation ces messages de gérer la bande passante si elle laisse ou les demandes d'anomalies ou les ignore juste.

Aperçu d'exécution de gestion de la bande passante

Cisco Gatekeeper peut rejeter des appels d'un terminal dû aux limites de bande passante. Ceci peut se produire si le garde-porte détermine qu'il n'y a pas bande passante suffisante disponible sur le réseau afin de prendre en charge l'appel. Cette fonction fonctionne également pendant un appel actif quand un terminal demande la bande passante supplémentaire ou signale un changement de la bande passante utilisée pour l'appel.

Cisco Gatekeeper met à jour un enregistrement de tous les appels actifs de sorte qu'il puisse gérer les ressources en bande passante dans sa zone. En configuration du cluster, le message d'indication d'annonce de Protocol de mise à jour de garde-porte (GUP) est permuté chaque intervalle de positionnement de temps et diffuse des informations sur l'utilisation de bande passante pour la zone. Cet échange de message GUP permet aux garde-portes alternatifs afin de gérer correctement la bande passante pour une zone simple, quoique les garde-portes soient dans des périphériques physiques distincts.

Quand vous décidez s'il y a assez de bande passante afin de recevoir une demande d'admission d'appel (ARQ), Cisco Gatekeeper calcule la bande passante disponible avec cette formule :

$$\text{Available_bandwidth} = (\text{total_allocated_bandwidth}) - (\text{bandwidth_used_locally}) - (\text{bandwidth_used_by_all_alternates}).$$

Si la bande passante disponible est suffisante pour l'appel, une confirmation d'admission (ACF) est retournée, autrement un rejet d'admission (ARJ) est retournée.

Les Passerelles voix devraient considérer des codecs, posent l'encapsulation 2, et les caractéristiques de compactage telles que le RTP comprimé [cRTP] quand ils demandent la bande passante de Cisco Gatekeeper. Parfois ces caractéristiques ne sont pas définies au moment de l'établissement d'appel, dans ce cas une demande de modification de bande passante peut être fournie au garde-porte après établissement d'appel afin d'ajuster la quantité de bande passante que l'appel l'utilise.

Remarque: En date du Logiciel Cisco IOS version 12.2(2)XA, Cisco a mis en application seulement la fonctionnalité de l'état de toutes les modifications de bande passante quand les codecs changent. Voyez la section : [Comment BRQ est déclenché de la passerelle pour informer le garde-porte de réduire le](#) pour en savoir plus de [bande passante d'appel](#).

[Comment configurer la fonction de gestion de la bande passante sur Cisco Gatekeeper](#)

En date du Logiciel Cisco IOS version 12.3(1), ces types de limites de bande passante de zone peuvent être configurés sur Cisco Gatekeeper :

- La bande passante maximum pour tous trafiquent H.323 entre la zone locale et une zone distante spécifiée. Si désirée, cette configuration peut être répétée individuellement pour chaque zone distante.
- La bande passante maximum a tenu compte d'une session simple dans la zone locale, typiquement utilisé pour des applications vidéo, pas pour la Voix
- La bande passante maximum pour tous trafiquent H.323 autorisé collectivement à toutes les zones distantes
- Le nouveau **bandwidth check-destination** de commande vérifie la bande passante de point final de destination avant qu'il réponde à ARQ. Cette commande a été introduite dans le Logiciel Cisco IOS version 12.3(1).

Employez ces commandes afin de configurer la bande passante de zone de Cisco Gatekeeper :

- **bande passante {interzone | total | session} {transférez | *max-bandwidth de zone-nom de zone*}**
- *max-bandwidth de bandwidth remote*
- **bandwidth check-destination**Référez-vous à la [commande bandwidth](#) pour plus de détails.

Ces valeurs configurées sont utilisées afin de traiter ARQs et BRQs.

Pour un ARQ, Cisco Gatekeeper déduit la bande passante spécifiée dans le message des compteurs appropriés de zone et/ou des compteurs distants. Si ceci entraîne à l'opposé de vont le négatif, alors l'appel est refusé et une réponse ARJ est envoyée avec la raison ARJ_REQ_DENIED. Si la demande d'appel dépasse cette bande passante, alors Cisco

Gatekeeper renvoie un rejet d'admission (ARJ).

Quand un BRQ demande une augmentation de bande passante, Cisco Gatekeeper valide la demande contre la zone et/ou le distant. Si la validation échoue, alors une réponse BRJ est envoyée avec une raison de BRJ_INSUFFICIENT_RSC et la bande passante maximale laissée.

Commandes show de garde-porte utilisées pour afficher les informations de bande passante

Sélectionnez la commande de **show gatekeeper zone status** afin d'afficher les informations de bande passante pour toutes les zones.

```
gkb-1#show gatekeeper zone status
GATEKEEPER ZONES
=====GK name      Domain Name  RAS Address  PORT  FLAGS-----
-----gkb-1      domainB.com  172.16.13.41 1719  LS    BANDWIDTH
INFORMATION (kbps) :   Maximum total bandwidth : 512           Current total bandwidth : 128
Current total bandwidth (w/ Alt GKs) : 128           Maximum interzone bandwidth : 512
Current interzone bandwidth : 128           Current interzone bandwidth (w/ Alt GKs) : 128
Maximum session bandwidth : 512           SUBNET ATTRIBUTES :   All Other Subnets : (Enabled)
PROXY USAGE CONFIGURATION :   Inbound Calls from all other zones :           to terminals in local
zone gkb-1 : use proxy           to gateways in local zone gkb-1 : do not use proxy           to MCUs in
local zone gkb-1 : do not use proxy           Outbound Calls to all other zones :           from terminals
in local zone gkb-1 : use proxy           from gateways in local zone gkb-1 : do not use proxy
from MCUs in local zone gkb-1 : do not use proxygka-1           domainA.com  172.16.13.35 1719
RS
```

Écrivez le **show gatekeeper zone cluster** de commande afin d'afficher les informations de bande passante, au cas où le garde-porte ferait partie d'une batterie.

```
gkb-1#show gatekeeper zone cluster
LOCAL CLUSTER INFORMATION
=====
ALT GKLOCAL GK NAME ALT GK NAME  PRI (kbps) (kbps) (kbps) ANNOUNCE STATUS-----
-----gkb-1           gkb-2           0 0
0 0 22s CONNECTED
```

Écrivez le **show gatekeeper calls** de commande afin d'afficher les appels actifs permis de ce garde-porte et combien de bande passante chacune utilise.

```
gkb-1#show gatekeeper calls Total number of active calls = 1. GATEKEEPER
CALL INFO =====LocalCallID
Age(secs) BW3-63466 9 128(Kbps) Endpt(s): Alias
E.164Addr src EP: gwa-1 4085272923 Endpt(s): Alias E.164Addr
dst EP: gw-1 3653 CallSignalAddr Port RASignalAddr Port
172.16.13.23 1720 172.16.13.23 54670
```

Messages relatifs de la bande passante RAS (BRQ/BCF/BRJ)

Le message BRQ est utilisé afin de demander un changement de bande passante de Cisco Gatekeeper. C'est la procédure :

1. Cisco Gatekeeper vérifie la demande par le endpointIdentifier afin de localiser le point final dans la base de données d'enregistrement.
2. Il localise l'enregistrement d'appel par l'utilisation du callReferenceValue afin de trouver un appel associé avec le point final avec le même callReferenceValue.
3. S'il localise l'enregistrement d'appel, il alors calcule le changement de la bande passante, alors ajoute ou soustrait de la bande passante globale de zone, selon les besoins. Il fait la même chose pour tout le le proxy ou ressources en passerelle en service.

4. Un message BCF ou BRJ est renvoyé au point final, qui dépend du succès ou échec.

Messages RAS utilisés pour signaler l'état de bande passante

Le champ « de données non standard » de la réponse de demande de l'information (IRR) diffuse également des informations sur la bande passante utilisée par courant sur une passerelle ou un proxy.

Comment BRQ est déclenché de la passerelle pour informer le garde-porte de réduire la bande passante d'appel

Avant le Logiciel Cisco IOS version 12.2(2)XA sur la passerelle H.323 de Cisco, des appels ont été toujours signalés afin d'exiger une bande passante de 64 Kbits/s. C'est la bande passante unidirectionnelle pour un codec de Cisco G.711. Si les points finaux dans l'appel choisissaient d'utiliser un codec plus efficace, ceci n'a pas été signalé à Cisco Gatekeeper. Dans la version de Logiciel Cisco IOS version 12.2(2)XA de la passerelle H.323 de Cisco ou de plus tard, qui se conforme H.323 à la version 3, la bande passante signalée est bidirectionnelle. Au commencement, le kb 128 est réservé. Si les points finaux dans l'appel sélectionnent un codec plus efficace, Cisco Gatekeeper est avisé de la modification de bande passante.

Remarque: Configurez la passerelle H.323 de Cisco avec cette commande en mode de configuration globale afin d'utiliser le comportement signalé de bande passante utilisé avant le Logiciel Cisco IOS version 12.2(2)XA pour la Gestion de bande passante de zone :

```
Router(config-gateway)#emulate cisco h323 bandwidth
```

Exemples

Cette section couvre ces deux exemples :

- [Gestion de la bande passante dans une topologie de batterie](#)
- [Utilisation BRQ afin de signaler la bande passante](#)

Exemple 1 : Gestion de la bande passante dans une topologie de batterie

Voyez que met au point capturé de Cisco Gatekeeper dans une batterie. Le débogage affiche les messages ARQ et ACF, qui inclut la bande passante exigée pour l'appel. Après que vous receviez ces messages, Cisco Gatekeeper met à jour les autres garde-portes dans la batterie au sujet de cette modification de bande passante.

Remarque: Ces commandes sont utilisées afin de saisir cette sortie : **le debug h225 asn1, debug ras, mettent au point le gup asn1 de garde-porte, mettent au point des événements de gup de garde-porte.**

```
Mar  2 23:59:26.802: Mar  2 23:59:26.802: RAS INCOMING PDU ::=value RasMessage ::=
admissionRequest :  !--- ARQ is received. { requestSeqNum 5928 callType pointToPoint : NULL
callModel direct : NULL endpointIdentifier {"6196296800000001"} destinationInfo { e164 : "3653"
} srcInfo { e164 : "4085272923", h323-ID : {"gwa-1"} } srcCallSignalAddress ipAddress : { ip
'AC100D0F'H port 11002 } bandwidth 1280!--- Intial bandwidth of 128k is requested.
callReferenceValue 14 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) rcvdMar 2 23:59:26.810: H225 NONSTD INCOMING ENCODE BUFFER::= 80
00000880 0180Mar 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 = 129Mar 2 23:59:26.814: RAS OUTGOING PDU
::=value RasMessage ::= admissionConfirm : !--- ACF is sent back. { requestSeqNum 5928
bandwidth 1280!--- BW value is included. callModel direct : NULL destCallSignalAddress ipAddress
: { ip 'AC100D17'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 2 23:59:26.818: RAS OUTGOING ENCODE BUFFER::=
2B 00172740 050000AC 100D1706 B800EF1A 00C00100 020000Mar 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: 51874Mar 2
23:59:26.822: RASLib::RASSendACF: ACF (seq# 5928) sent to 172.16.13.23Mar 2 23:59:36.046: GUP
OUTGOING PDU ::=value GUP_Information ::= !--- GUP update message is sent to all gatekeepers in
the cluster. { 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!--- BW info is included. interzoneBandwidth 1280 remoteBandwidth
1280 } } } } }Mar 2 23:59:36.050: GUP OUTGOING ENCODE BUFFER::= 00
0A2A8648 86F70C0A 00000220 001E40B4 3E80010C B904676B 622D3132 00010002 01420000 67006B00
62002D00 31080067 006B0062 002D0032 40050040 05004005 00Mar 2 23:59:36.054: Mar 2
23:59:36.054: Sending GUP ANNOUNCEMENT INDICATION to 172.16.13.16

```

Exemple 2 : Utilisation BRQ afin de signaler la bande passante

La découverte met au point de Cisco Gatekeeper dans une installation où la bande passante sur le garde-porte distant est limitée à 144 Kbps. Vous voyez dans le débogage que l'ARQ demandé est une première bande passante de 128 kpbs. Quand l'appel est installé, le point final signale le changement de la bande passante avec un message BRQ et la bande passante utilisée dans 16 kpbs, qui signifie l'appel a été installée avec des codecs de Cisco G729. Alors un autre appel est demandé et est traité les mêmes.

Notez que si le deuxième appel arrivait avant que le point final ait demandé le changement de la bande passante pour le premier appel, les anomalies de Cisco Gatekeeper qui appellent, puisque les kpbs 128+128=256 et qui sont plus de 144 kpbs configurés.

```

!!!gatekeeper zone local gka-1 domainA.com 172.16.13.35 zone remote gkb-1 domainB.com
172.16.13.41 1719 zone prefix gkb-1 36* zone prefix gka-1 53* gw-type-prefix 1#* default-
technology bandwidth remote 144 no shutdown endpoint ttl 120!

```

Cette sortie a été capturée avec le **debug h225 asn1** et le **debug ras de commandes** :

```

gka-1#show loggingSyslog logging: enabled (0 messages dropped, 0 messages rate-limited, 0
flushes, 0 overruns) Console logging: disabled Monitor logging: level debugging, 1076
messages logged Buffer logging: level debugging, 203860 messages logged Logging Exception
size (4096 bytes) Trap logging: level informational, 66 message lines logged Log
Buffer (9999999 bytes):Mar 14 20:18:06.385: RAS INCOMING ENCODE BUFFER::= 27 88039700 F0003800
31004600 36004100 38003900 38003000 30003000 30003000 30003000 31010180 69860140 04006700
77006100 2D003140 0500000B 40B50000 12138000 0008A001 800B1249 53444E2D 564F4943 45DA4A9C
E21FCF11 CC802093 7822E08B 6308E020 00018011 00DA4A9C E21FCF11 CC802193 7822E08B 630100Mar 14
20:18:06.401: Mar 14 20:18:06.405: RAS INCOMING PDU ::=value RasMessage ::= admissionRequest :
!--- ARQ is received. { requestSeqNum 920 callType pointToPoint : NULL callModel direct : NULL
endpointIdentifier {"81F6A89800000001"} destinationInfo { e164 : "3653" } srcInfo { h323-ID :
{"gwa-1"} } bandwidth 1280!--- Intial BW of 128 kpbs is requested. callReferenceValue 11
nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0
manufacturerCode 18 } data '80000008A001800B124953444E2D564F494345'H } conferenceID
'DA4A9CE21FCF11CC8020937822E08B63'H activeMC FALSE answerCall FALSE canMapAlias TRUE
callIdentifier { guid 'DA4A9CE21FCF11CC8021937822E08B63'H } willSupplyUUIES FALSE } Mar 14
20:18:06.425: H225 NONSTD INCOMING ENCODE BUFFER::= 80 000008A0 01800B12 4953444E 2D564F49
4345Mar 14 20:18:06.429: Mar 14 20:18:06.429: H225 NONSTD INCOMING PDU ::=value
ARQnonStandardInfo ::= { sourceAlias { } sourceExtAlias { } callingOctet3a 128
interfaceSpecificBillingId "ISDN-VOICE" }Mar 14 20:18:06.433: H225 NONSTD OUTGOING PDU ::=value

```


LRQnonStandardInfo ::= { ttl 6 nonstd-callIdentifier { guid 'DA4A9CE21FCF11CC8021937822E08B63'H } callingOctet3a 128 gatewaySrcInfo { h323-ID : {"gwa-1"} } }Mar 14 20:18:06.437: H225 NONSTD OUTGOING ENCODE BUFFER::= 82 86B01100 DA4A9CE2 1FCF11CC 80219378 22E08B63 01800D01 40040067 00770061 002D0031 Mar 14 20:18:06.445: Mar 14 20:18:06.445: RAS OUTGOING PDU ::= value RasMessage ::= locationRequest : { requestSeqNum 2061 destinationInfo { e164 : "3653" } nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '8286B01100DA4A9CE21FCF11CC8021937822E08B...'H } replyAddress ipAddress : { ip 'AC100D23'H port 1719 } sourceInfo { h323-ID : {"gka-1"} } canMapAlias TRUE }Mar 14 20:18:06.461: RAS OUTGOING ENCODE BUFFER::= 4A 80080C01 01806986 40B50000 12258286 B01100DA 4A9CE21F CF11CC80 21937822 E08B6301 800D0140 04006700 77006100 2D003100 AC100D23 06B70B80 0D014004 0067006B 0061002D 00310180 Mar 14 20:18:06.469: Mar 14 20:18:06.473: RAS OUTGOING PDU ::=value RasMessage ::= requestInProgress : { requestSeqNum 920 delay 9000 } Mar 14 20:18:06.473: RAS OUTGOING ENCODE BUFFER::= 80 05000397 2327Mar 14 20:18:06.473: Mar 14 20:18:06.477: RAS INCOMING ENCODE BUFFER::= 4F 080C00AC 100D1706 B800AC10 0D17DC0E 40B50000 12390001 40040067 00770062 002D0031 08006700 6B006200 2D003101 10014004 00670077 0062002D 003100AC 100D1706 B8000000 00000000 00000010 40080880 013C0501 0000Mar 14 20:18:06.489: Mar 14 20:18:06.489: RAS INCOMING PDU ::=value RasMessage ::= locationConfirm : { requestSeqNum 2061 callSignalAddress ipAddress : { ip 'AC100D17'H port 1720 } rasAddress ipAddress : { ip 'AC100D17'H port 56334 } nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '00014004006700770062002D0031080067006B00...'H } destinationType { gateway { protocol { voice : { supportedPrefixes { } } } } mc FALSE undefinedNode FALSE } }Mar 14 20:18:06.509: H225 NONSTD INCOMING ENCODE BUFFER::= 00 01400400 67007700 62002D00 31080067 006B0062 002D0031 01100140 04006700 77006200 2D003100 AC100D17 06B80000 00000000 00000000 Mar 14 20:18:06.517: Mar 14 20:18:06.521: H225 NONSTD INCOMING PDU ::=value LCFnonStandardInfo ::= { termAlias { h323-ID : {"gwb-1"} } gkID {"gkb-1"} gateways { { gwType voip : NULL gwAlias { h323-ID : {"gwb-1"} } sigAddress { ip 'AC100D17'H port 1720 } resources { maxDSPs 0 inUseDSPs 0 maxBChannels 0 inUseBChannels 0 activeCalls 0 bandwidth 0 inuseBandwidth 0 } } } }Mar 14 20:18:06.537: RAS OUTGOING PDU ::=value RasMessage ::= **admissionConfirm** : *!--- ACF is sent back.* { requestSeqNum 920 **bandwidth 1280!***--- BW is included.* callModel direct : NULL destCallSignalAddress ipAddress : { ip 'AC100D17'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 14 20:18:06.549: RAS OUTGOING ENCODE BUFFER::= 2B 00039740 050000AC 100D1706 B800EF1A 00C00100 020000Mar 14 20:18:06.553: Mar 14 20:18:06.677: RAS INCOMING ENCODE BUFFER::= 32 0003981E 00380031 00460036 00410038 00390038 00300030 00300030 00300030 00300031 DA4A9CE2 1FCF11CC 80209378 22E08B63 000B00A0 15080011 00DA4A9C E21FCF11 CC802193 7822E08B 630100Mar 14 20:18:06.685: Mar 14 20:18:06.689: RAS INCOMING PDU ::=value RasMessage ::= **bandwidthRequest** : *!--- BRQ message to request bandwidth to be changed to 16 kpbs.* { requestSeqNum 921 endpointIdentifier {"81F6A89800000001"} conferenceID 'DA4A9CE21FCF11CC8020937822E08B63'H callReferenceValue 11 **bandwidth 160!***--- 16 kpbs is requested.* callIdentifier { guid 'DA4A9CE21FCF11CC8021937822E08B63'H } answeredCall FALSE }Mar 14 20:18:06.697: RAS OUTGOING PDU ::=value RasMessage ::= **bandwidthConfirm** : *!--- BCF is sent back approving the bandwidth request change.* { requestSeqNum 921 **bandwidth 160** }Mar 14 20:18:06.697: RAS OUTGOING ENCODE BUFFER::= 34 039800A0 Mar 14 20:18:06.701: Mar 14 20:18:12.066: RAS INCOMING ENCODE BUFFER::= 0E 40039906 0008914A 00030000 0100AC10 0D0FE511 00040067 006B0061 002D0031 00B50000 12288F00 0002003B 0180211E 00380031 00460036 00410038 00390038 00300030 00300030 00300030 00300031 01000180 Mar 14 20:18:12.074: Mar 14 20:18:12.078: RAS INCOMING PDU ::=value RasMessage ::= registrationRequest : { requestSeqNum 922 protocolIdentifier { 0 0 8 2250 0 3 } discoveryComplete FALSE callSignalAddress { } rasAddress { ipAddress : { ip 'AC100D0F'H port 58641 } } terminalType { mc FALSE undefinedNode FALSE } gatekeeperIdentifier {"gka-1"} endpointVendor { vendor { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } } timeToLive 60 keepAlive TRUE endpointIdentifier {"81F6A89800000001"} willSupplyUUies FALSE maintainConnection TRUE }Mar 14 20:18:12.098: RAS OUTGOING PDU ::=value RasMessage ::= registrationConfirm : { requestSeqNum 922 protocolIdentifier { 0 0 8 2250 0 3 } callSignalAddress { } gatekeeperIdentifier {"gka-1"} endpointIdentifier {"81F6A89800000001"} alternateGatekeeper { } timeToLive 60 willRespondToIRR FALSE maintainConnection TRUE } Mar 14 20:18:12.106: RAS OUTGOING ENCODE BUFFER::= 12 40039906 0008914A 00030008 0067006B 0061002D 00311E00 38003100 46003600 41003800 39003800 30003000 30003000 30003000 3000310F 8A010002 003B0100 0180Mar 14 20:18:12.114: Mar 14 20:18:14.586: RAS INCOMING ENCODE BUFFER::= 5A C0039A08 80013C05 04010020 40078000 38003100 46003600 41003800 39003800 30003000 30003000 30003000 30003000 30003100 AC100D0F E5110100 AC100D0F 06B80140 04006700

```

77006100 2D003101 C100B500 00120570 2BA39307 000BDA4A 9CE21FCF 11CC8020 937822E0 8B630000
A003C000 1100DA4A 9CE21FCF 11CC8021 937822E0 8B630E20 0100Mar 14 20:18:14.602: Mar 14
20:18:14.602: RAS INCOMING PDU ::=value RasMessage ::= infoRequestResponse : !--- IRR message is
received and it includes the bandwidth used on the gateway. { requestSeqNum 923 endpointType {
gateway { protocol { voice : { supportedPrefixes { { prefix e164 : "1#" } } } } mc FALSE
undefinedNode FALSE } endpointIdentifier {"81F6A89800000001"} rasAddress ipAddress : { ip
'AC100D0F'H port 58641 } callSignalAddress { ipAddress : { ip 'AC100D0F'H port 1720 } }
endpointAlias { h323-ID : {"gwa-1"} } perCallInfo { { nonStandardData { nonStandardIdentifier
h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '702BA39307'H }
callReferenceValue 11 conferenceID 'DA4A9CE21FCF11CC8020937822E08B63'H h245 { } callSignaling {
} callType pointToPoint : NULL bandwidth 160 callModel direct : NULL
callIdentifier { guid 'DA4A9CE21FCF11CC8021937822E08B63'H }
} } needResponse FALSE } Mar 14 20:18:14.646: H225 NONSTD INCOMING ENCODE
BUFFER::= 70 2BA39307 Mar 14 20:18:14.646: Mar 14 20:18:14.646: H225 NONSTD INCOMING PDU
::=value IRRperCallnonStandardInfo ::= { startTime 732140295 }Mar 14 20:18:28.008:
RAS INCOMING ENCODE BUFFER::= 27 88039B00 F0003800 31004600 36004100 38003900 38003000 30003000
30003000 30003000 31010180 69860140 04006700 77006100 2D003140 0500000C 40B50000 12030000
00000000 00000000 00000000 00000000 0008E020 00018011 00000000 00000000 00000000 00000000
000100Mar 14 20:18:28.024: Mar 14 20:18:28.024: RAS INCOMING PDU ::=value RasMessage ::=
admissionRequest : { requestSeqNum 924 callType pointToPoint : NULL callModel
direct : NULL endpointIdentifier {"81F6A89800000001"} destinationInfo {
e164 : "3653" } srcInfo { h323-ID : {"gwa-1"} } bandwidth 1280
callReferenceValue 12 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 } willSupplyUIEs FALSE }Mar 14
20:18:28.044: H225 NONSTD INCOMING ENCODE BUFFER::= 00 0000Mar 14 20:18:28.044: Mar 14
20:18:28.044: H225 NONSTD INCOMING PDU ::=value ARQnonStandardInfo ::= { sourceAlias
{ } sourceExtAlias { } }Mar 14 20:18:28.048: H225 NONSTD OUTGOING PDU
::=value LRQnonStandardInfo ::= { ttl 6 nonstd-callIdentifier { guid
'00000000000000000000000000000000'H } gatewaySrcInfo { h323-ID : {"gwa-
1"} } } Mar 14 20:18:28.056: H225 NONSTD OUTGOING ENCODE BUFFER::= 82 86901100
00000000 00000000 00000000 00000000 0D014004 00670077 0061002D 0031Mar 14 20:18:28.060: Mar 14
20:18:28.060: RAS OUTGOING PDU ::=value RasMessage ::= locationRequest : {
requestSeqNum 2062 destinationInfo { e164 : "3653" } nonStandardData
{ nonStandardIdentifier h221NonStandard : { t35CountryCode 181
t35Extension 0 manufacturerCode 18 } data
'82869011000000000000000000000000000000000000000000... 'H } replyAddress ipAddress : {
ip 'AC100D23'H port 1719 } sourceInfo { h323-ID : {"gka-1"} }
canMapAlias TRUE }Mar 14 20:18:28.076: RAS OUTGOING ENCODE BUFFER::= 4A 80080D01 01806986
40B50000 12238286 90110000 00000000 00000000 00000000 00000000 01400400 67007700 61002D00
3100AC10 0D2306B7 0B800D01 40040067 006B0061 002D0031 0180Mar 14 20:18:28.084: Mar 14
20:18:28.088: RAS OUTGOING PDU ::=value RasMessage ::= requestInProgress : {
requestSeqNum 924 delay 9000 }Mar 14 20:18:28.088: RAS OUTGOING ENCODE BUFFER::= 80
0500039B 2327Mar 14 20:18:28.088: Mar 14 20:18:28.097: RAS INCOMING ENCODE BUFFER::= 4F 080D00AC
100D1706 B800AC10 0D17DC0E 40B50000 12390001 40040067 00770062 002D0031 08006700 6B006200
2D003101 10014004 00670077 0062002D 003100AC 100D1706 B8000000 00000000 00000010 40080880
013C0501 0000Mar 14 20:18:28.105: Mar 14 20:18:28.109: RAS INCOMING PDU ::=value RasMessage ::=
locationConfirm : { requestSeqNum 2062 callSignalAddress ipAddress : {
ip 'AC100D17'H port 1720 } rasAddress ipAddress : { ip 'AC100D17'H
port 56334 } nonStandardData { nonStandardIdentifier h221NonStandard :
{ t35CountryCode 181 t35Extension 0 manufacturerCode 18 }
data '00014004006700770062002D0031080067006B00... 'H } destinationType {
gateway { protocol { voice : {
supportedPrefixes { } } } mc
FALSE undefinedNode FALSE } }Mar 14 20:18:28.129: H225 NONSTD INCOMING ENCODE
BUFFER::= 00 01400400 67007700 62002D00 31080067 006B0062 002D0031 01100140 04006700 77006200
2D003100 AC100D17 06B80000 00000000 00000000 Mar 14 20:18:28.133: Mar 14 20:18:28.137: H225
NONSTD INCOMING PDU ::=value LCFnonStandardInfo ::= { termAlias { h323-ID
: {"gwb-1"} } gkID {"gkb-1"} gateways { { gwType voip :
NULL gwAlias { h323-ID : {"gwb-1"} } sigAddress
{ ip 'AC100D17'H port 1720 } resources {
maxDSPs 0 inUseDSPs 0 maxBChannels 0 inUseBChannels 0

```

