

Fehlerbehebung und Verständnis von Cisco Gatekeeper Bandbreitenmanagement

Inhalt

[Einführung](#)

[Übersicht über die Bandbreitenverwaltung](#)

[Konfigurieren der Bandbreitenmanagementfunktion auf dem Cisco Gatekeeper](#)

[Gatekeeper zeigt Befehle an, die zum Anzeigen von Bandbreiteninformationen verwendet werden.](#)

[Bandbreitenbezogene RAS-Meldungen \(BRQ/BCF/BRJ\)](#)

[RAS-Nachrichten für die Meldung des Bandbreitenstatus](#)

[Wie BRQ vom Gateway ausgelöst wird, um den Gatekeeper zu benachrichtigen, um die Anrufbandbreite zu reduzieren](#)

[Beispiele](#)

[Zugehörige Informationen](#)

Einführung

In diesem Dokument wird davon ausgegangen, dass der Leser mit einem grundlegenden Verständnis von Cisco IOS® Software Gatekeepers und Gatekeeper to Gateway H.225 Registration, Admission and Status (RAS)-Meldungen vertraut ist. Weitere Informationen finden Sie unter [Understanding H.323 Gatekeepers](#).

Gemäß der H.323-Empfehlung sollten Cisco IOS-Gatekeepers diese H.225 RAS-Bandbreitenmanagementmeldungen unterstützen:

- Bandbreitenanforderung (BRQ)
- Bandbreitenverweigerung (BRJ)
- BCF-Nachrichten (Bandwidth Confirmation)

Dieses Konzept kann auf dem Bandbreitenmanagement basieren. Es kann auch eine NULL-Funktion sein, die alle Anforderungen für Bandbreitenänderungen akzeptiert. Anders ausgedrückt: Der Gatekeeper kann diese Nachrichten entweder zum Verwalten der Bandbreite verwenden, wenn er Anfragen zulässt oder ablehnt, oder sie einfach ignorieren.

Übersicht über die Bandbreitenverwaltung

Der Cisco Gatekeeper kann Anrufe von einem Terminal aufgrund von Bandbreitenbeschränkungen ablehnen. Dies kann auftreten, wenn der Gatekeeper feststellt, dass im Netzwerk keine ausreichende Bandbreite für die Unterstützung des Anrufs verfügbar ist. Diese Funktion wird auch während eines aktiven Anrufs ausgeführt, wenn ein Terminal zusätzliche Bandbreite anfordert oder eine Änderung der Bandbreite meldet, die für den Anruf verwendet wird.

Der Cisco Gatekeeper protokolliert alle aktiven Anrufe, sodass er die Bandbreitenressourcen in

seiner Zone verwalten kann. In einer Clusterkonfiguration wird die Ankündigungsmeldung "Gatekeeper Update Protocol (GUP)" in jedem festgelegten Intervall ausgetauscht und enthält Informationen zur Bandbreitennutzung für die Zone. Dieser GUP-Nachrichtenaustausch ermöglicht es den alternativen Gatekeepern, die Bandbreite für eine einzelne Zone ordnungsgemäß zu verwalten, obwohl sich die Gatekeepers in separaten physischen Geräten befinden.

Wenn Sie entscheiden, ob genügend Bandbreite vorhanden ist, um eine Anrufzugangsanfrage (Call Admission Request, ARQ) zu akzeptieren, berechnet Cisco Gatekeeper die verfügbare Bandbreite mit der folgenden Formel:

$$\text{Available_bandwidth} = (\text{total_assigned_bandwidth}) - (\text{bandwidth_used_local}) - (\text{bandwidth_used_by_all_alternates}).$$

Wenn die verfügbare Bandbreite für den Anruf ausreicht, wird eine Zugangsbestätigung (Admission Confirmation, ACF) zurückgegeben. Andernfalls wird eine Admission Rejection (ARJ) zurückgegeben.

Sprach-Gateways sollten Codec, Layer-2-Kapselung und Komprimierungsfunktionen wie komprimiertes RTP [cRTP] berücksichtigen, wenn sie Bandbreite vom Cisco Gatekeeper anfordern. Manchmal sind diese Funktionen zum Zeitpunkt der Anruferichtung nicht definiert. In diesem Fall kann dem Gatekeeper nach der Anruferichtung eine Anforderung zur Bandbreitenänderung gesendet werden, um die Bandbreite anzupassen, die der Anruf verwendet.

Hinweis: Ab der Cisco IOS Software-Version 12.2(2)XA hat Cisco nur noch die Funktionen des Berichts über Bandbreitenänderungen implementiert, wenn Codecs geändert werden. Siehe Abschnitt: [Wie BRQ vom Gateway ausgelöst wird, um den Gatekeeper zu benachrichtigen, um die Anrufbandbreite für weitere Informationen zu reduzieren.](#)

Konfigurieren der Bandbreitenmanagementfunktion auf dem Cisco Gatekeeper

Ab Cisco IOS Software Release 12.3(1) können diese Bandbreitenbeschränkungen für Zonen auf dem Cisco Gatekeeper konfiguriert werden:

- Die maximale Bandbreite für den gesamten H.323-Datenverkehr zwischen der lokalen Zone und einer angegebenen Remote-Zone. Auf Wunsch kann diese Konfiguration für jede Remote-Zone einzeln wiederholt werden.
- Die maximal zulässige Bandbreite für eine einzelne Sitzung in der lokalen Zone, die in der Regel für Videoanwendungen, nicht für Sprache, verwendet wird.
- Die maximale Bandbreite für den gesamten H.323-Datenverkehr, der kollektiv für alle Remote-Zonen zulässig ist
- Der neue Befehl **Bandbreitenprüfungs-Ziel** überprüft die Bandbreite des Zielendpunkts, bevor er auf ARQ reagiert. Dieser Befehl wurde in Version 12.3(1) der Cisco IOS-Software eingeführt.

Verwenden Sie die folgenden Befehle, um die Bandbreite der Cisco Gatekeeper-Zone zu konfigurieren:

- **Bandbreite {Interzone | Insgesamt | session} {default} | Zone Zone-Name} max. Bandbreite**
- **Bandbreite Remote max. Bandbreite**

- **Bandbreitenprüfziel** Weitere Informationen finden Sie im [Befehl bandwidth \(Bandbreite\)](#).

Diese konfigurierten Werte werden zur Verarbeitung von ARQs und BRQs verwendet.

Bei einem ARQ zieht der Cisco Gatekeeper die in der Nachricht angegebene Bandbreite von den entsprechenden Zonenzählern und/oder Remote-Zählern ab. Wenn ein Zähler negativ wird, wird der Anruf abgelehnt und eine ARJ-Antwort mit dem Grund ARJ_REQ_DENIED gesendet. Wenn die Anrufanfrage diese Bandbreite überschreitet, gibt der Cisco Gatekeeper eine "Admission Rejection" (ARJ) zurück.

Wenn ein BRQ eine Bandbreitenerhöhung anfordert, validiert der Cisco Gatekeeper die Anfrage für die Zone und/oder Remote. Wenn die Validierung fehlschlägt, wird eine BRJ-Antwort mit einem Grund von BRJ_INSUFFICIENT_RSC und der maximal zulässigen Bandbreite gesendet.

Gatekeeper zeigt Befehle an, die zum Anzeigen von Bandbreiteninformationen verwendet werden.

Geben Sie den Befehl **show gatekeeper zone status** ein, um die Bandbreiteninformationen für alle Zonen anzuzeigen.

```
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 proxy
```

```
gka-1       domainA.com  172.16.13.35  1719  RS
```

Geben Sie den Befehl **show gatekeeper zone cluster** ein, um die Bandbreiteninformationen anzuzeigen, falls der Gatekeeper Teil eines Clusters ist.

```
gkb-1#show gatekeeper zone cluster
                        LOCAL CLUSTER INFORMATION
                        =====
                        TOT BW  INT BW  REM BW  LAST      ALT GK
LOCAL GK NAME ALT GK NAME  PRI (kbps) (kbps) (kbps) ANNOUNCE STATUS
-----
```

gkb-1 gkb-2 0 0 0 0 22s CONNECTED

Geben Sie den Befehl **show gatekeeper calls** ein, um die aktiven Anrufe anzuzeigen, die von diesem Gatekeeper zulässig sind, und wie viel Bandbreite jeder einzelne benötigt.

```
gkb-1#show gatekeeper calls
```

```
Total number of active calls = 1.
```

```
GATEKEEPER CALL INFO
```

```
=====
```

LocalCallID	Age(secs)	BW	
3-63466	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

Bandbreitenbezogene RAS-Meldungen (BRQ/BCF/BRJ)

Die BRQ-Nachricht dient dazu, eine Bandbreitenänderung beim Cisco Gatekeeper anzufordern. Dies ist das Verfahren:

1. Der Cisco Gatekeeper überprüft die Anforderung des Endpunktidentifizier, um den Endpunkt in der Registrierungsdatenbank zu finden.
2. Er sucht mithilfe von callReferenceValue den Anrufdatensatz, um einen mit demselben callReferenceValue verknüpften Anruf zu finden.
3. Wenn der Anrufdatensatz lokalisiert wird, wird die Änderung der Bandbreite berechnet und die Bandbreite der globalen Zone ggf. hinzugefügt oder von ihr abgezogen. Gleiches gilt für alle Proxy- oder Gateway-Ressourcen, die verwendet werden.
4. Eine BCF- oder BRJ-Nachricht wird an den Endpunkt zurückgesendet, was vom Erfolg oder Misserfolg abhängt.

RAS-Nachrichten für die Meldung des Bandbreitenstatus

Das Feld Information Request Response (IRR) "Non-Standard Data" (Nicht standardmäßige Daten) enthält auch Informationen zur aktuell genutzten Bandbreite auf einem Gateway oder Proxy.

Wie BRQ vom Gateway ausgelöst wird, um den Gatekeeper zu benachrichtigen, um die Anrufbandbreite zu reduzieren

Vor der Cisco IOS Software-Version 12.2(2)XA auf dem Cisco H.323-Gateway wurden immer Anrufe gemeldet, um eine Bandbreite von 64 Kbit/s zu benötigen. Dies ist die unidirektionale Bandbreite für einen Cisco G.711-Codec. Wenn die Endpunkte des Anrufs einen effizienteren Codec verwenden wollten, wurde dies dem Cisco Gatekeeper nicht gemeldet. In der Cisco IOS Software-Version 12.2(2)XA des Cisco H.323-Gateways oder höher, die H.323 Version 3 entspricht, wird eine bidirektionale Bandbreite gemeldet. Zunächst sind 128 KB reserviert. Wenn die Endpunkte des Anrufs einen effizienteren Codec auswählen, wird der Cisco Gatekeeper über die Bandbreitenänderung benachrichtigt.

Hinweis: Konfigurieren Sie das Cisco H.323-Gateway mit diesem Befehl im globalen Konfigurationsmodus, um das vor der Cisco IOS-Softwareversion 12.2(2)XA verwendete

Bandbreitenverhalten für die Bandbreitenverwaltung der Zone zu verwenden:

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

Beispiele

In diesem Abschnitt werden die folgenden beiden Beispiele erläutert:

- [Bandbreitenmanagement in einer Cluster-Topologie](#)
- [BRQ verwenden, um Bandbreite zu melden](#)

Beispiel 1: Bandbreitenmanagement in einer Cluster-Topologie

Sehen Sie sich die von einem Cisco Gatekeeper in einem Cluster erfassten Debugging an. Das Debuggen zeigt die ARQ- und ACF-Meldungen an, die die für den Anruf erforderliche Bandbreite enthalten. Nachdem Sie diese Nachrichten erhalten haben, aktualisiert Cisco Gatekeeper die anderen Gatekeeper im Cluster über diese Bandbreitenänderung.

Hinweis: Diese Befehle werden verwendet, um diese Ausgabe zu erfassen: `debug h225 asn1`, `debug ras`, `debug ggatekeeper group asn1`, `debug gggatekeeper-Gruppenereignisse`.

```
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
```

```
!--- Initial 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 }  
willSupplyUUUIES 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. { 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 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 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 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

```

Beispiel 2: BRQ verwenden, um Bandbreite zu melden

Suchen Sie Debugger von einem Cisco Gatekeeper in einer Konfiguration, in der die Bandbreite auf dem Remote-Gatekeeper auf 144 Kbit/s beschränkt ist. Im Debugging sehen Sie, dass die angeforderte ARQ eine Ausgangsbandbreite von 128 Kbit/s ist. Wenn der Anruf eingerichtet ist, meldet der Endpunkt die Bandbreitenänderung mit einer BRQ-Nachricht und der in 16 Kbit/s verwendeten Bandbreite. Dies bedeutet, dass der Anruf mit dem Cisco G729 Codec eingerichtet wurde. Anschließend wird ein weiterer Anruf angefordert und gleich behandelt.

Wenn der zweite Anruf einging, bevor der Endpunkt die Bandbreitenänderung für den ersten Anruf beantragte, lehnt der Cisco Gatekeeper diesen Anruf ab, da $128+128=256$ Kbit/s konfiguriert wurden und mehr als 144 Kbit/s konfiguriert wurden.

```

!
!
!
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
!

```

Diese Ausgabe wurde mit den Befehlen **debug h225 asn1** und **debug ras** erfasst:

```

gka-1#show logging
Syslog 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 630100

Mar 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 4345

Mar 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 2327 Mar 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 0000 Mar 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 020000 Mar 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 630100 Mar 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 0180

```

```

Mar 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 30003100 AC100D0F E5110100 AC100D0F 06B80140 04006700 77006100
2D003101 C100B500 00120570 2BA39307 000BDA4A 9CE21FCF 11CC8020 937822E0
8B630000 A003C000 1100DA4A 9CE21FCF 11CC8021 937822E0 8B630E20 0100

```

```

Mar 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 000100
Mar 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
  }
  willSupplyUUIEs FALSE
}
```

Mar 14 20:18:28.044: H225 NONSTD INCOMING ENCODE BUFFER ::= 00 0000
Mar 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 0031

Mar 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 '828690110000000000000000000000000000000000000000...'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 0000000D 01400400
67007700 61002D00 3100AC10 0D2306B7 0B800D01 40040067 006B0061 002D0031
0180

Mar 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 2327

```
Mar 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 0000
Mar 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
      activeCalls 0
      bandwidth 0
      inuseBandwidth 0
    }
  }
}
}

```

Mar 14 20:18:28.153: RAS OUTGOING PDU ::=

```

value RasMessage ::= admissionConfirm :
{
  requestSeqNum 924
  bandwidth 1280
  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 20:18:28.169: RAS OUTGOING ENCODE BUFFER ::= 2B 00039B40 050000AC
100D1706 B800EF1A 00C00100 020000
Mar 14 20:18:28.169:

```
Mar 14 20:18:28.289: RAS INCOMING ENCODE BUFFER ::= 32 00039C1E 00380031
00460036 00410038 00390038 00300030 00300030 00300030 00300031 00000000
00000000 00000000 00000000 000C00A0 15080011 00000000 00000000 00000000
00000000 000100
```

```
Mar 14 20:18:28.301:
```

```
Mar 14 20:18:28.301: RAS INCOMING PDU ::=
```

```
value RasMessage ::= bandwidthRequest :
{
  requestSeqNum 925
  endpointIdentifier {"81F6A89800000001"}
  conferenceID '00000000000000000000000000000000'H
  callReferenceValue 12
  bandwidth 160
  callIdentifier
  {
    guid '00000000000000000000000000000000'H
  }
  answeredCall FALSE
}
```

```
Mar 14 20:18:28.309: RAS OUTGOING PDU ::=
```

```
value RasMessage ::= bandwidthConfirm :
{
  requestSeqNum 925
  bandwidth 160
}
```

```
Mar 14 20:18:28.313: RAS OUTGOING ENCODE BUFFER ::= 34 039C00A0
```

```
Mar 14 20:18:28.313:
```

[Zugehörige Informationen](#)

- [VoIP-Anrufzugangskontrolle](#)
- [Cisco High-Performance Gatekeeper](#)
- [Cisco H.323 - Verbesserte Skalierbarkeit und Interoperabilität](#)
- [VoIP mit Gatekeeper](#)
- [Unterstützung von Sprachtechnologie](#)
- [Produkt-Support für Sprach- und Unified Communications](#)
- [Fehlerbehebung bei Cisco IP-Telefonie](#)
- [Technischer Support und Dokumentation - Cisco Systems](#)