

Troubleshooting and Understanding Cisco Gatekeeper Bandwidth Management

Document ID: 18731

Contents

Introduction

Bandwidth Management Operation Overview

How to Configure the Bandwidth Management Feature on the Cisco Gatekeeper

Gatekeeper show Commands Used to Display Bandwidth Information

Bandwidth Related RAS Messages (BRQ/BCF/BRJ)

RAS Messages Used to Report Bandwidth Status

How BRQ is Triggered from the Gateway to Notify the Gatekeeper to Reduce Call Bandwidth

Examples

Related Information

Introduction

This document assumes the reader is familiar with a basic comprehension of Cisco IOS® Software Gatekeepers and Gatekeeper to Gateway H.225 Registration, Admission, and Status (RAS) Messagings. Refer to Understanding H.323 Gatekeepers for more information.

According to the H.323 recommendation, Cisco IOS Gatekeepers should support these H.225 RAS bandwidth management messages:

- Bandwidth Request (BRQ)
- Bandwidth Rejection (BRJ)
- Bandwidth Confirmation (BCF) messages

This concept can be based on bandwidth management. It can also be a null function that accepts all requests for bandwidth changes. In other words, the Gatekeeper can either use these messages to manage bandwidth if it allows or rejects requests or just ignore them.

Bandwidth Management Operation Overview

The Cisco Gatekeeper can reject calls from a terminal due to bandwidth limitations. This can occur if the Gatekeeper determines that there is not sufficient bandwidth available on the network in order to support the call. This function also operates during an active call when a terminal requests additional bandwidth or reports a change in bandwidth used for the call.

The Cisco Gatekeeper maintains a record of all active calls so that it can manage the bandwidth resources in its zone. In a cluster configuration, the Gatekeeper Update Protocol (GUP) announcement indication message is exchanged every set interval of time and carries information about the bandwidth utilization for the zone. This GUP message exchange allows the alternate Gatekeepers in order to properly manage the bandwidth for a single zone, even though the Gatekeepers are in separate physical devices.

When you decide whether there is enough bandwidth in order to accept a call Admission Request (ARQ), the Cisco Gatekeeper calculates the available bandwidth with this formula:

Available_bandwidth = (total_allocated_bandwidth) – (bandwidth_used_locally) – (bandwidth_used_by_all_alternates).

If the available bandwidth is sufficient for the call, an Admission Confirmation (ACF) is returned, otherwise an Admission Rejection (ARJ) is returned.

Voice gateways should consider codec, Layer 2 encapsulation, and compression features such as compressed RTP [cRTP] when they request bandwidth from the Cisco Gatekeeper. Sometimes these features are not defined at the time of call setup, in which case a bandwidth change request can be issued to the gatekeeper after call setup in order to adjust the amount of bandwidth the call uses.

Note: As of Cisco IOS Software Release 12.2(2)XA, Cisco has implemented only the functionality of the report of any Bandwidth changes when Codecs change. See the section: How BRQ is Triggered from the Gateway to Notify the Gatekeeper to Reduce Call Bandwidth for more information.

How to Configure the Bandwidth Management Feature on the Cisco Gatekeeper

As of Cisco IOS Software Release 12.3(1), these types of zone bandwidth limitations can be configured on the Cisco Gatekeeper:

- The maximum bandwidth for all H.323 traffic between the local zone and a specified remote zone. If desired, this configuration can be repeated individually for each remote zone.
- The maximum bandwidth allowed for a single session in the local zone, typically used for video applications, not for voice
- The maximum bandwidth for all H.323 traffic allowed collectively to all remote zones
- The new command **bandwidth check–destination** checks the destination endpoint bandwidth before it responds to ARQ. This command was introduced in Cisco IOS Software Release 12.3(1).

Use these commands in order to configure Cisco Gatekeeper zone bandwidth:

- **bandwidth {interzone | total | session} {default | zone zone-name} max-bandwidth**
- **bandwidth remote max-bandwidth**
- **bandwidth check–destination**

Refer to the **bandwidth** command for more details.

These configured values are used in order to process ARQs and BRQs.

For an ARQ, the Cisco Gatekeeper deducts the bandwidth specified in the message from the appropriate zone counters and/or remote counters. If this causes any counter to go negative, then the call is denied and an ARJ response is sent with the reason ARJ_REQ_DENIED. If the call request exceeds this bandwidth, then the Cisco Gatekeeper returns an Admission Rejection (ARJ).

When a BRQ requests a bandwidth increase, the Cisco Gatekeeper validates the request against the zone and/or remote. If the validation fails, then a BRJ response is sent with a reason of BRJ_INSUFFICIENT_RSC and the maximum amount of bandwidth allowed.

Gatekeeper show Commands Used to Display Bandwidth Information

Enter the **show gatekeeper zone status** command in order to display the bandwidth information for all 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 proxy

gka-1       domainA.com    172.16.13.35   1719  RS

```

Enter the command **show gatekeeper zone cluster** in order to display the bandwidth information, in case the gatekeeper is part of a cluster.

```

gkb-1#show gatekeeper zone cluster
LOCAL CLUSTER INFORMATION
=====
LOCAL GK NAME  ALT GK NAME  PRI  TOT BW  INT BW  REM BW  LAST  ALT GK
-----
gkb-1         gkb-2       0    0       0       0       22s   CONNECTED

```

Enter the command **show gatekeeper calls** in order to display the active calls permitted by that gatekeeper and how much bandwidth each one uses.

```

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

```

Bandwidth Related RAS Messages (BRQ/BCF/BRJ)

The BRQ message is used in order to request a change in bandwidth from the Cisco Gatekeeper. This is the procedure:

1. The Cisco Gatekeeper verifies the request by the endpointIdentifier in order to locate the endpoint in the registration database.

2. It locates the call record through the use of the callReferenceValue in order to find a call associated with the endpoint with the same callReferenceValue.
3. If it locates the call record, it then computes the change in bandwidth, then adds or subtracts from the global zone bandwidth, as necessary. It does the same for any proxy or gateway resources in use.
4. A BCF or BRJ message is sent back to the endpoint, which depends on success or failure.

RAS Messages Used to Report Bandwidth Status

The Information Request Response (IRR) "Non-Standard Data" field also carries information about the current used bandwidth on a gateway or proxy.

How BRQ is Triggered from the Gateway to Notify the Gatekeeper to Reduce Call Bandwidth

Prior to Cisco IOS Software Release 12.2(2)XA on the Cisco H.323 Gateway, calls were always reported in order to require a bandwidth of 64 kbps. This is the unidirectional bandwidth for a Cisco G.711 codec. If the endpoints in the call chose to use a more efficient codec, this was not reported to the Cisco Gatekeeper. In the Cisco IOS Software Release 12.2(2)XA version of the Cisco H.323 Gateway or later, which conforms with H.323 version 3, the reported bandwidth is bidirectional. Initially, 128 kb is reserved. If the endpoints in the call select a more efficient codec, the Cisco Gatekeeper is notified of the bandwidth change.

Note: Configure the Cisco H.323 Gateway with this command in global configuration mode in order to use the reported bandwidth behavior used prior to Cisco IOS Software Release 12.2(2)XA for zone bandwidth management:

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

Examples

This section covers these two examples:

- Bandwidth Management in a Cluster Topology
- Use BRQ in Order to Report Bandwidth

Example 1: Bandwidth Management in a Cluster Topology

See the debugs captured from a Cisco Gatekeeper in a cluster. The debug shows the ARQ and ACF messages, which includes the bandwidth required for the call. After you receive these messages, the Cisco Gatekeeper updates the other gatekeepers in the cluster about this bandwidth change.

Note: These commands are used in order to capture this output: **debug h225 asn1, debug ras, debug gatekeeper gup asn1, debug gatekeeper gup events.**

```
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
}
willSupplyUUIEs FALSE
}

```

```

Mar  2 23:59:26.810: ARQ (seq# 5928) rcvd
Mar  2 23:59:26.810: H225 NONSTD INCOMING ENCODE BUFFER ::= 80 00000880
0180
Mar  2 23:59:26.810:
Mar  2 23:59:26.810: H225 NONSTD INCOMING PDU ::=

```

```

value ARQnonStandardInfo ::=
{
  sourceAlias
  {
  }
  sourceExtAlias
  {
  }
  callingOctet3a 128
}

```

```

parse_arq_nonstd: ARQ Nonstd decode succeeded, remlen = 129
Mar  2 23:59:26.814: RAS OUTGOING PDU ::=

```

```

value RasMessage ::= admissionConfirm :

```

!--- ACF is sent back.

```

{
  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

```

Example 2: Use BRQ in Order to Report Bandwidth

Find debugs from a Cisco Gatekeeper in a setup where the bandwidth on the remote gatekeeper is limited to 144 kbps. You see in the debug that the ARQ requested is an initial bandwidth of 128 kbps. When the call is set up, the endpoint reports the change in the bandwidth with a BRQ message and the bandwidth used in 16 kbps, which means the call was set up with Cisco G729 codec. Then another call is requested and is treated the same.

Note that if the second call arrived before the endpoint requested the change in bandwidth for the first call, the Cisco Gatekeeper rejects that call, since $128+128=256$ kbps and that is more than 144 kbps configured.

```

!
!
!
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 l#* default-technology
bandwidth remote 144
no shutdown
endpoint ttl 120
!

```

This output was captured with the commands **debug h225 asn1** and **debug ras**:

```

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

```



```
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 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:
```

Related Information

- **VoIP Call Admission Control**
- **Cisco High-Performance Gatekeeper**
- **Cisco H.323 Scalability and Interoperability Enhancements**
- **VoIP with Gatekeeper**
- **Voice Technology Support**
- **Voice and Unified Communications Product Support**
- **Troubleshooting Cisco IP Telephony**
- **Technical Support & Documentation – Cisco Systems**

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2014 – 2015 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Sep 07, 2006

Document ID: 18731
