

Konfigurieren der grundlegenden Anrufzugangskontrolle für Gatekeeper

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Hintergrundinformationen](#)

[Befehl bandwidth \(Gatekeeper\)](#)

[Konfigurieren](#)

[Netzwerkdiagramm](#)

[Konfigurationen](#)

[Überprüfen](#)

[Fehlerbehebung](#)

[Befehle zur Fehlerbehebung](#)

[Beispielausgabe und Debugausgabe](#)

[Zugehörige Informationen](#)

Einführung

Dieses Dokument enthält eine Beispielkonfiguration für die grundlegende Call Admission Control für Gatekeeper.

Voraussetzungen

Anforderungen

Es müssen mehrere Bedingungen erfüllt werden, bevor das Gateway die richtige Adressauflösung vom Gatekeeper erhalten kann. Für jede VoIP-Lösung müssen bei Verbindungen mit niedriger Geschwindigkeit mehrere wichtige Punkte überprüft werden.

Stellen Sie vor dem Versuch dieser Konfiguration sicher, dass Sie die folgenden Anforderungen erfüllen:

- Alle Gateways müssen bei den entsprechenden Gatekeeper registriert werden.
- Alle Gatekeeper sollten über einen korrekten Wählplan verfügen, damit sie die Route für die Anrufe festlegen können.
- Die Zugangskontrolle kann so konfiguriert werden, dass die Rufnummer zwischen bestimmten Zonen eingeschränkt wird.

Da die ersten beiden Punkte im Abschnitt [Konfigurieren](#) berücksichtigt werden, konzentrieren wir uns auf die Zugangskontrolle im Abschnitt [Hintergrundinformationen](#).

[Verwendete Komponenten](#)

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Drei Cisco 2600-Router.
- Cisco IOS[®] Softwareversion 12.2.8.5 ENTERPRISE PLUS/H323 MCM.

Die in diesem Dokument enthaltenen Informationen wurden aus Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Sie in einem Live-Netzwerk arbeiten, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen, bevor Sie es verwenden.

[Konventionen](#)

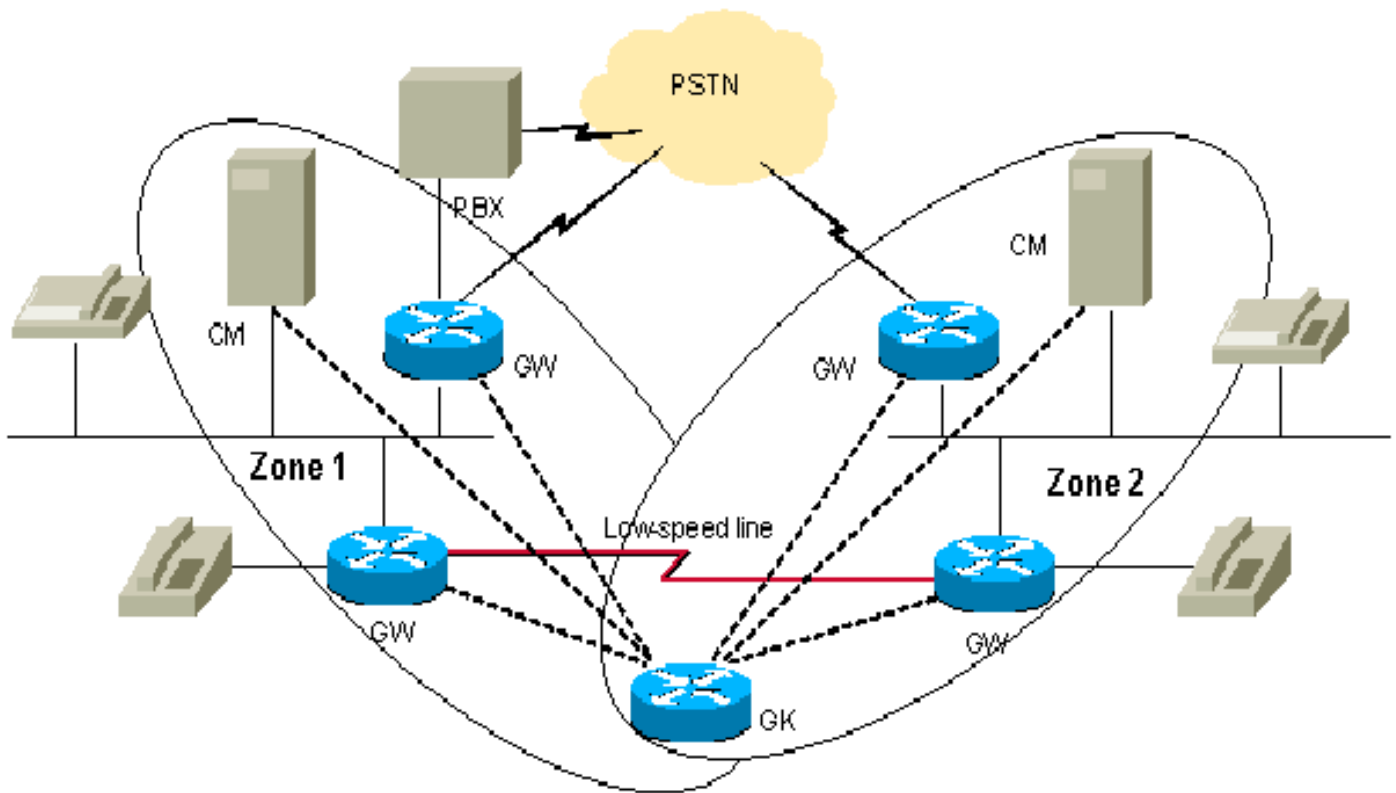
Weitere Informationen zu Dokumentkonventionen finden Sie in den [Cisco Technical Tips Conventions](#).

[Hintergrundinformationen](#)

In dieser Beispielkonfiguration wird ein VoIP-Netzwerk mit einer Topologie mit zwei Zonen untersucht, das von einem Gatekeeper mit drei Gateways in beiden Zonen verwaltet wird. Ziel dieses Dokuments ist es, ein einfaches Beispiel für eine Konfiguration der Zugangskontrolle zu liefern, die eine Richtlinie auf die Anzahl der Anrufe zwischen Zonen und innerhalb dieser anwendet. Dieses Dokument enthält technische Hintergrundinformationen zu den konfigurierten Funktionen, Designrichtlinien und grundlegenden Verifizierungs- und Fehlerbehebungsstrategien.

Hinweis: In dieser Konfiguration befinden sich die vier Router im gleichen LAN. In der realen Topologie können sich jedoch alle Geräte in verschiedenen Teilen Ihres Netzwerks befinden.

Sehr häufig gibt es in realen Netzwerken mehrere Quellen für Datenverkehr mit hoher Priorität. Es ist eine komplexe Aufgabe, all diese Bedingungen zu unterscheiden, weil sie zahlreich sind und leicht zu übersehen. Es gibt jedoch mehrere häufig auftretende Situationen, die sich im realen Leben sehr häufig ereignen und die es wert sind, berücksichtigt zu werden. Die Zugangskontrolle wird zu einem Problem, wenn die Router, die die Priorisierung des Datenverkehrs ermöglichen, selbst nicht die Quellen für diesen Datenverkehr sind. Die typische Topologie besteht aus mehreren Sprach-Gateways an zwei Standorten, die über die von einem Router-Paar bereitgestellte Verbindung miteinander verbunden sind. Eine weitere Topologie umfasst Cisco CallManager mit IP-Telefonen an zwei Standorten sowie die Gateways zum PSTN oder PBX. In beiden Situationen gibt es mehrere Quellen für Sprachdatenverkehr von beiden Seiten der Verbindung.



Manchmal kann es zu einem Problem mit der Sprachqualität kommen, wenn die Menge an Sprachverkehr die konfigurierte Bandbreite für die Prioritätswarteschlange überschreitet. Der Grund hierfür ist, dass Router und Cisco CallManager/IP-Telefone, die den Datenverkehr verursachen, im oben beschriebenen Design keine zentrale Verwaltung für die Anrufzugangsverwaltung aufweisen. In diesem Fall werden Pakete, die die Bandbreite überschreiten, verworfen.

Es gibt mehrere Möglichkeiten, dieses Szenario zu vermeiden. Die einfachste Lösung besteht darin, die Sprachbandbreite in der Low Latency Queue (LLQ) so zu konfigurieren, dass sie die maximale Anzahl von Anrufen aus allen Quellen akzeptiert. Ohne Sprachdatenverkehr wird die ungenutzte Bandbreite den Datenflüssen zugewiesen. Dies ist möglich, wenn die Gesamtbandbreite der Verbindung die für die maximale Anzahl von Anrufen erforderliche Bandbreite übersteigt.

Ein vernünftigerer Ansatz besteht darin, für jede Sprachquelle von beiden Enden der Verbindung Beschränkungen anzuwenden. Wenn Sie dies tun, wird die zusammengefasste Bandbreite aller Verbindungen die empfohlenen 75 % der tatsächlichen Bandbreite der Verbindung zwischen den Standorten nicht überschreiten. Um diese Einschränkungen anzuwenden, verwenden Sie den Befehl **max-conn** unter VoIP Dial-Peer-Konfiguration. Wenn wir davon ausgehen, dass sich nur an einem zentralen Standort ein Cisco CallManager befindet, können wir die Anzahl der Anrufe an die Zweigstelle ohne CallManager einschränken. Dieser Ansatz ermöglicht es uns, die Situation zu verwalten, in der die Quellen für Sprachdatenverkehr die Verbindung überzeichnen können. Der Nachteil dieses Ansatzes ist die unflexible Bandbreitennutzung, die den Quellen gewährt wird. Bei diesem Ansatz können einige der Gateways keinen zusätzlichen Anruf tätigen, selbst wenn derzeit freie Bandbreite zur Verfügung steht.

Der flexibelste Ansatz besteht in der Verwendung einer separaten Einheit für die zentrale Call Admission Control: Gatekeeper. Mit dem Gatekeeper können zwei Standorte mit zwei Cisco CallManager-Clustern (oder CallManager-Clustern) verbunden werden.

Hinweis: Die Nutzung des Gatekeeper bedeutet nicht immer den Kauf eines neuen separaten

Routers. Basierend auf der Anzahl der Anrufe und der Last der Router können Sie einen Gatekeeper auf einem der vorhandenen Router konfigurieren, wobei das entsprechende Cisco IOS-Feature-Set Enterprise/PLUS/H323 lautet. Dies kann helfen, kleine Zweigstellen zu verwalten und einen dedizierten Gatekeeper nur in der Zentrale zu ermöglichen.

Der Gatekeeper-Ansatz sollte mit Vorsicht beachtet werden, damit der Router nicht durch eine zusätzliche Last überlastet wird. Darüber hinaus sollten Sie überprüfen, ob die Topologie eine solche Positionierung des Gatekeeper ermöglicht, um den zusätzlichen Datenverkehr auf der kritischen Verbindung zu vermeiden.

Es wird generell empfohlen, separate Cisco Router als dedizierte Gatekeeper in Ihrem Netzwerk zu verwenden, wobei die für Ihre Topologie geeignete Nummer gewählt wird.

Berücksichtigen Sie die oben aufgeführte Topologie. Hier können Sie alle Geräte in die beiden lokalen Zonen einfügen, die von einem einzigen Gatekeeper verwaltet werden. Dadurch können Sie in jeder Zone eine hohe Anzahl von Anrufen haben, während die Anzahl der Anrufe zwischen diesen Bereichen eingeschränkt wird. In unserem Testbeispiel beschränken wir die Bandbreite zwischen den beiden Zonen auf einen Anruf und können bis zu zwei (mehr) Anrufe in einem dieser Bereiche zulassen.

Weitere Informationen hierzu finden Sie unter [VoIP Call Admission Control](#).

Verwenden Sie zum Durchführen dieser Aufgabe den Befehl **bandwidth** (Gatekeeper), der in [Cisco High-Performance Gatekeeper](#) beschrieben wird.

Befehl bandwidth (Gatekeeper)

Um die maximale aggregierte Bandbreite für H.323-Datenverkehr festzulegen, verwenden Sie den Konfigurationsbefehl **bandwidth** Gatekeeper. Um die Funktion zu deaktivieren, verwenden Sie die **no**-Form dieses Befehls.

Hinweis: Mit diesem Befehl können Sie die Bandbreite über eine einzige Verbindung aus der Zone einschränken. Wenn die Topologie es ermöglicht, einen Anruf über mehrere Pfade von einer Zone zur anderen zu tätigen, können die Links leicht überbelegt werden. Betrachten Sie die folgende Topologie: Zwei Zonen sind über zwei Pfade miteinander verbunden, sodass nur ein Anruf über jeden Pfad möglich ist. Wenn die Bandbreite durch einen Anruf begrenzt wird, wird der zweite Pfad niemals verwendet. Wenn die Bandbreite jedoch durch zwei Anrufe begrenzt wird, kann eine der Verbindungen überbelegt sein. Dieser Befehl kann also auf die Zonen angewendet werden, die nur einen Pfad zu allen anderen Zonen haben. Die "Hub and Spoke"-Topologie ist eine Ausnahme. Obwohl der Hub über mehrere Pfade verfügt, werden die Verbindungen nicht überbelegt, da die Anzahl der Anrufe an den Spokes für jede Verbindung beschränkt wird.

Bandbreite {Interzone | Insgesamt | session} {default} | Zone *Zone-Name*} *Bandbreitengröße*

keine Bandbreite {Interzone | Insgesamt | session} {default} | Zone *Zone-Name*} *Bandbreitengröße*

Syntaxbeschreibung

Die folgende Tabelle beschreibt die Syntax:

Syntax	Beschreibung
--------	--------------

Interzone	Gibt die gesamte Bandbreite für H.323-Datenverkehr von der Zone zu einer anderen Zone an.
gesamt	Gibt die in der Zone zulässige Gesamtbandbreite für H.323-Datenverkehr an.
Sitzung	Gibt die maximal zulässige Bandbreite für eine Sitzung in der Zone an.
Standard	Gibt den Standardwert für alle Zonen an.
Zone Zonenna me	Gibt eine bestimmte Zone an. Benennt die bestimmte Zone.
Bandbre itengröß e	Maximale Bandbreite. Für Interzone und insgesamt liegt der Bereich zwischen 1 und 10.000.000 Kbit/s. Für Sitzungen liegt der Bereich zwischen 1 und 5.000 Kbit/s.

Standardwerte

Keine

Befehlsmodi

Gatekeeper-Konfiguration

Befehlsverlauf

Die folgende Tabelle beschreibt den Befehlsverlauf:

Version	Änderung
12.1(3)XI	Dieser Befehl wurde eingeführt.
12.1(5)XM	Der Befehl bandwidth wurde ohne den Befehl zone gatekeeper erkannt.
12,2(2)T	Dieser Befehl wurde in Version 12.2(2)T der Cisco IOS-Software integriert.
12.2(2)XB1	Dieser Befehl wurde auf dem Universal-Gateway Cisco AS5850 implementiert.

Nutzungsrichtlinien

In früheren Versionen der Cisco IOS-Software wurde die Funktionalität des Befehls **Bandbreite** mithilfe des Befehls **zone gatekeeper** ermittelt.

Beispiele

Im folgenden Beispiel wird die maximale Bandbreite für die Zone auf 5.000 Kbit/s konfiguriert:

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

Zugehörige Befehle

bandwidth remote - Gibt die gesamte Bandbreite für H.323-Datenverkehr zwischen diesem Gatekeeper und einem anderen Gatekeeper an.

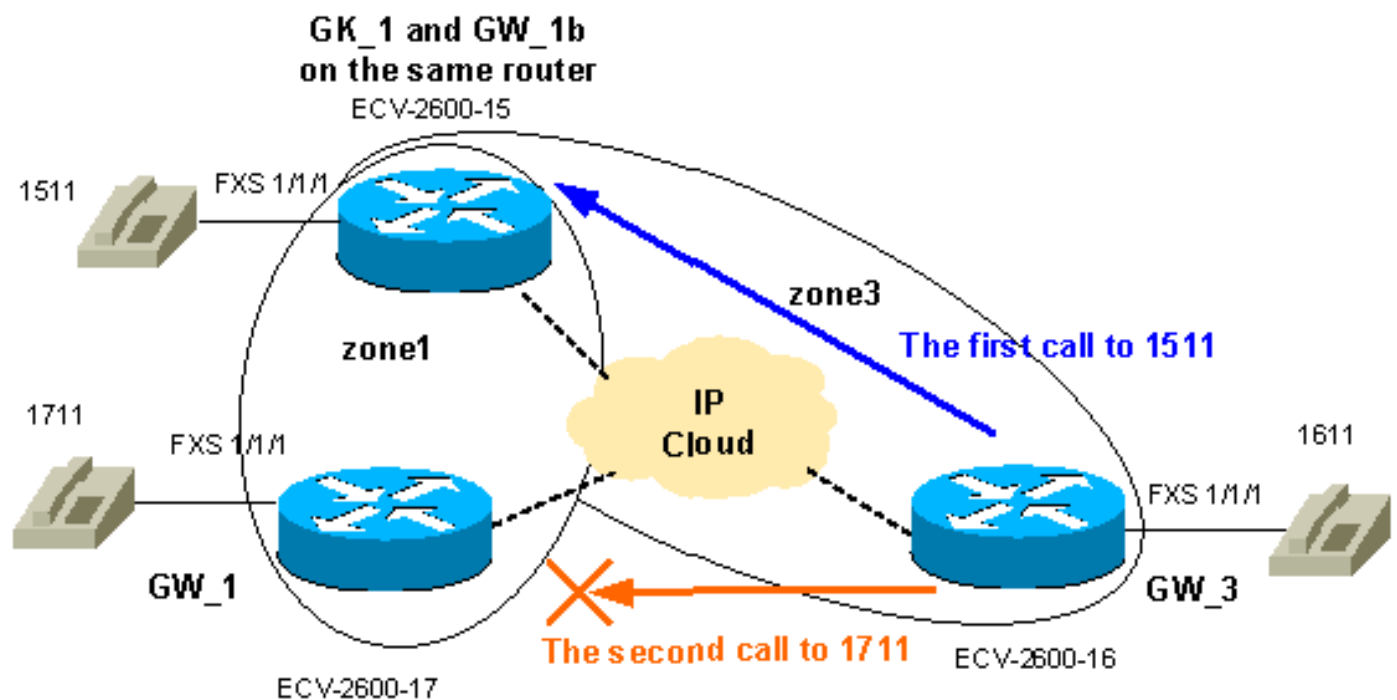
Konfigurieren

In diesem Abschnitt erhalten Sie Informationen zum Konfigurieren der in diesem Dokument beschriebenen Funktionen.

Hinweis: Um weitere Informationen zu den in diesem Dokument verwendeten Befehlen zu erhalten, verwenden Sie das [Command Lookup Tool](#) ([nur registrierte Kunden](#)).

Netzwerkdiagramm

In diesem Dokument wird die folgende Netzwerkeinrichtung verwendet:



Konfigurationen

Ziel ist es, die verfügbare Bandbreite zwischen Zone1 und Zone3 auf einen Anruf zu beschränken und eine höhere Anzahl von Anrufen (in diesem Beispiel bis zu zwei) in Zone1 zuzulassen. So erfüllen wir die allgemeinen Anforderungen für die typische Call Admission Task. Die RAS-Meldungen (Registration, Admission, and Status Protocol) werden vor den H225-Anruferichtungsnachrichten gesendet. Anschließend folgt die H4245-Aushandlung, die die Fähigkeiten der Seiten definiert. Die tatsächliche Bandbreite des Anrufs wird also nach der Call Admission Phase und dem Austausch der RAS-Nachrichten definiert. Aus diesem Grund betrachtet der Gatekeeper jeden Anruf als 64-KB-Anruf. Daher sollten die Bandbreitenbeschränkungen zwischen Zonen für Sprachanrufe in Schritten von 64 KB erhöht

werden.

Hinweis: Der GW_3 ist auf demselben Router konfiguriert wie der Gatekeeper, um solche Möglichkeiten für Zweigstellen mit geringem Verwaltungsaufwand zu veranschaulichen.

Hinweis: Die Überprüfung der Gatekeeper- und Gateway-Konfigurationen ist ein wichtiger Teil der Fehlerbehebung für GK-GW-Probleme. Um das Verständnis der Konfigurationen zu vereinfachen, wurden alle nicht verknüpften Konfigurationsbefehle entfernt.

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#
```

Überprüfen

Dieser Abschnitt enthält Informationen, mit denen Sie überprüfen können, ob Ihre Konfiguration ordnungsgemäß funktioniert.

Bestimmte **show**-Befehle werden vom [Output Interpreter Tool](#) unterstützt (nur [registrierte](#) Kunden), mit dem Sie eine Analyse der **show**-Befehlsausgabe anzeigen können.

- **show gateway**: Zeigt den Registrierungsstatus des Gateways an.
- **show gatekeeper endpoints**: listet alle zum Gatekeeper registrierten Gateways auf.
- **show gatekeeper zone prefix** - Zeigt alle auf dem Gatekeeper konfigurierten Zonenpräfixe an.
- **show gatekeeper call**: Zeigt aktive Anrufe an, die vom Gatekeeper verarbeitet werden.

Fehlerbehebung

Dieser Abschnitt enthält Informationen zur Fehlerbehebung in Ihrer Konfiguration.

Befehle zur Fehlerbehebung

Bestimmte **show**-Befehle werden vom [Output Interpreter Tool](#) unterstützt (nur [registrierte](#) Kunden), mit dem Sie eine Analyse der **show**-Befehlsausgabe anzeigen können.

Hinweis: Bevor Sie **Debugbefehle** ausgeben, lesen Sie [Wichtige Informationen über Debug-Befehle](#).

- **debug h225 asn1**: Zeigt H225-Meldungen (RAS und Q931-Anrufeinrichtung) an.
- **debug cch323 h225** - Zeigt H225-Anrufeinrichtungsmeldungen an.

Hier einige nützliche Links:

- [Fehlerbehebung und Debuggen von VoIP-Anrufen - Grundlagen](#)
- [VoIP-Debug-Befehle](#)
- [Befehlsreferenz für Cisco IOS Voice, Video und Fax, Version 12.2](#)

Beispielausgabe und Debugausgabe

```
!--- 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

=====

CallSignalAddr	Port	RASSignalAddr	Port	Zone Name	Type F
10.52.218.47	1720	10.52.218.47	58841	gk-zone1.test.com	VOIP-GW
H323-ID: gw_1b					
10.52.218.48	1720	10.52.218.48	59067	gk-zone3.test.com	VOIP-GW
H323-ID: gw_3					
10.52.218.49	1720	10.52.218.49	52887	gk-zone1.test.com	VOIP-GW
H323-ID: 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.tes 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.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

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 :
{
```



```
fastStart
{
'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
```

[Zugehörige Informationen](#)

- [Fehlerbehebung und Verständnis von Cisco Gatekeeper Bandbreitenmanagement](#)
- [H.323-Gatekeepers](#)
- [Cisco High-Performance Gatekeeper](#)
- [Konfigurieren von H.323-Gateways](#)
- [Konfigurieren von H.323-Gatekeepern](#)
- [Konfigurieren der H.323-Unterstützung für virtuelle Schnittstellen](#)
- [Unterstützung von Sprachtechnologie](#)
- [Produkt-Support für Sprach- und Unified Communications](#)
- [Fehlerbehebung bei Cisco IP-Telefonie](#)
- [Technischer Support - Cisco Systems](#)