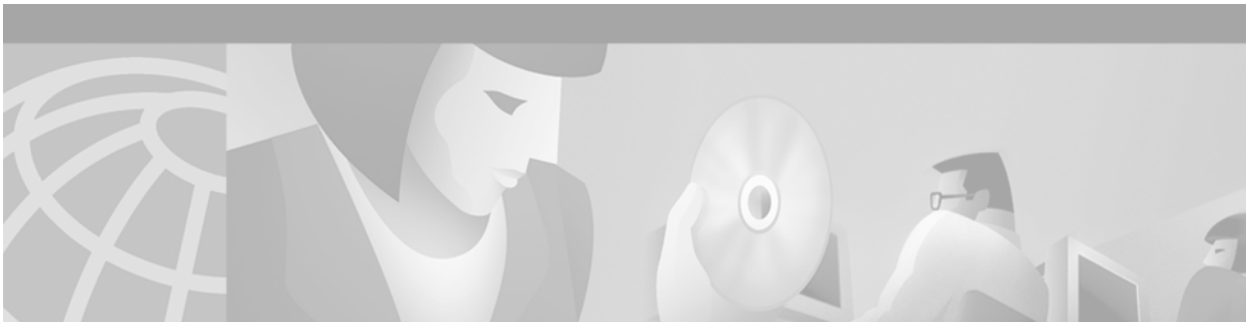


Configuring H.323 Gatekeepers and Proxies

This chapter describes how to configure the Cisco Multimedia Conference Manager. The Multimedia Conference Manager provides gatekeeper and proxy capabilities required for service provisioning and management of H.323-compliant networks.

This chapter includes the following sections:

For a complete description of the H.323 gatekeeper commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation for other commands that appear in this chapter, use the command reference master index or search online.

For more information regarding Resource Reservation Protocol (RSVP), synchronous reservation timers, and slow connect, refer to the Cisco IOS Release 12.1(5)T *VoIP Call Admission Control Using RSVP* or the *Cisco IOS Quality of Service Solutions Configuration Guide.*

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Multimedia Conference Manager Overview

Deploying H.323 applications and services requires careful design and planning for the network infrastructure and for the H.323 devices. The Cisco H.323-compliant Multimedia Conference Manager provides gatekeeper and proxy capabilities, which are required for service provisioning and management of H.323 networks. With the Cisco Multimedia Conference Manager, your current internetwork can be configured to route bit-intensive data, such as audio, telephony, video and audio telephony, and data conferencing using existing telephone and ISDN links without degrading the current level of service of the network. In addition, H.323-compliant applications can be implemented on existing networks in an incremental fashion without upgrades.

Multimedia Conference Manager provides a rich list of networking capabilities, including the following:

- A means to implement quality of service (QoS), which is required for the successful deployment of H.323 applications.
- Interzone routing in the E.164 address space. When using H.323-identification (H.323-ID) format addresses, interzone routing is accomplished by using domain names.

Multimedia Conference Manager allows you to do the following:

- Identify H.323 traffic and apply appropriate policies.
- Limit H.323 traffic on the LAN and WAN.
- Provide user accounting for records based on service use.
- Insert QoS for the H.323 traffic generated by applications such as Voice over IP (VoIP), data conferencing, and video conferencing.
- Implement security for H.323 communications.

Principal Multimedia Conference Manager Functions

The H.323-compliant Multimedia Conference Manager has two principal functions: gatekeeper and proxy. Gatekeeper subsystems provide the following features:

- User authorization in which authentication, authorization, and accounting (AAA) account holders are permitted to register and use the services of the gatekeeper application.
- Accounting using AAA call detail records.
- Zone bandwidth management to limit the number of active sessions.
- H.323 call routing.
- Address resolution.

Cisco Multimedia Conference Managers can be configured to use the Cisco Hot Standby Router Protocol (HSRP) so that when one gatekeeper fails, the standby gatekeeper assumes its role.

Proxy subsystems provide the following features:

- H.323 traffic consolidation.
- Tight bandwidth controls.
- QoS mechanisms such as IP Precedence and RSVP.
- Secure communication over extranets.

H.323 Gatekeeper Features

The following sections describe the main features of a gatekeeper in an H.323 network:

Zone and Subnet Configuration

A zone is defined as the set of H.323 nodes controlled by a single gatekeeper. Gatekeepers that coexist on a network may be configured so that they register endpoints from different subnets.

Endpoints attempt to discover a gatekeeper and consequently the zone of which they are members by using the Registration, Admission, and Status (RAS) message protocol. The protocol supports a discovery message that may be sent multicast or unicast.

If the message is sent multicast, the endpoint registers nondeterministically with the first gatekeeper that responds to the message. To enforce predictable behavior, where endpoints on certain subnets are assigned to specific gatekeepers, the **zone subnet** command can be used to define the subnets that constitute a given gatekeeper zone. Any endpoint on a subnet that is not enabled for the gatekeeper will not be accepted as a member of that gatekeeper zone. If the gatekeeper receives a discovery message from such an endpoint, it will send an explicit reject message.

Redundant H.323 Zone Support

Redundant H.323 zone support allows for the following:

Gatekeeper Multiple Zone Support

Redundant H.323 zone support allows users to configure multiple remote zones to service the same *zone* or *technology prefix*. A user is able to configure more than one remote gatekeeper to which the local gatekeeper can send location requests (LRQs). This allows for more reliable call completion.

Redundant H.323 zone support is supported on all gatekeeper-enabled IOS images.

Gateway Support for Alternate Gatekeepers

Redundant H.323 zone support in the gateway allows a user to configure two gatekeepers in the gateway (one as the primary and the other as the alternate). All gatekeepers are active. The gateway can choose to register with any one (but not both) at a given time. If that gatekeeper becomes unavailable, the gateway registers with the other.

Redundant H.323 zone support is supported on all gateway-enabled images.

Zone Prefixes

The zone prefixes (typically area codes) serve the same purpose as the domain names in the H.323-ID address space.

For example, the local gatekeeper can be configured with the knowledge that zone prefix "212....." (that is, any address beginning "212" and followed by 7 arbitrary digits) is handled by the gatekeeper gatekeeper_2. Then, when the local gatekeeper is asked to admit a call to destination address 2125551111, it knows to send the LRQ to gatekeeper_2.

When gatekeeper_2 receives the request, the gatekeeper must resolve the address so that the call can be sent to its final destination. There may be an H.323 endpoint with that E.164 address that has registered with gatekeeper_2, in which case gatekeeper_2 returns the IP address for that endpoint. However, it is possible that the E.164 address belongs to a non-H.323 device (for example, a telephone or an H.320 terminal). Because non-H.323 devices do not register with gatekeepers, gatekeeper_2 cannot resolve the address. The gatekeeper must be able to select a gateway that can be used to reach the non-H.323 device. This is where the technology prefixes (or "gateway-type") become useful.

Technology Prefixes

The network administrator selects technology prefixes (tech-prefixes) to denote different types or classes of gateways. The gateways are then configured to register with their gatekeepers with these prefixes. For example, voice gateways can register with tech-prefix 1#, H.320 gateways with tech-prefix 2#, and voicemail gateways with tech-prefix 3#. More than one gateway can register with the same type prefix. When this happens, the gatekeeper makes a random selection among gateways of the same type.

If the callers know the type of device that they are trying to reach, they can include the technology prefix in the destination address to indicate the type of gateway to use to get to the destination. For example, if a caller knows that address 2125551111 belongs to a regular telephone, the destination address of 1#2125551111 can be used, where 1# indicates that the address should be resolved by a voice gateway. When the voice gateway receives the call for 1#2125551111, it strips off the technology prefix and bridges the next leg of the call to the telephone at 2125551111.

Gatekeeper-to-Gatekeeper Redundancy and Load-Sharing Mechanism

The gatekeeper-to-gatekeeper redundancy and load-sharing mechanism expands the capability that is provided by the redundant H.323 zone support feature. Redundant H.323 zone support, which was introduced in Cisco IOS Release 12.1(1)T, allows you to configure multiple gatekeepers to service the same zone or technology prefix by sending LRQs to two or more gatekeepers.

With the redundant H.323 zone support feature, the LRQs are sent simultaneously (in a "blast" fashion) to all of the gatekeepers in the list. The gateway registers with the gatekeeper that responds first. Then, if that gatekeeper becomes unavailable, the gateway registers with another gatekeeper from the list.

The gatekeeper-to-gatekeeper redundancy and load-sharing mechanism allows you to configure gatekeeper support and to give preference to specific gatekeepers. You may choose whether the LRQs are sent simultaneously or sequentially (one at a time) to the remote gatekeepers in the list. If the LRQs are sent sequentially, a *delay* is inserted after the first LRQ and before the next LRQ is sent. This delay allows the first gatekeeper to respond before the LRQ is sent to the next gatekeeper. The order in which LRQs are sent to the gatekeepers is based on the order in which the gatekeepers are listed (using either the **zone prefix** command or the **gw-type-prefix** command).

Once the local gatekeeper has sent LRQs to all the remote gatekeepers in the list (either simultaneously or sequentially), if it has not yet received a location confirmation (LCF), it opens a "window." During this window, the local gatekeeper waits to see whether a LCF is subsequently received from any of the remote gatekeepers. If no LCF is received from any of the remote gatekeepers while the window is open, the call is rejected.

Terminal Name Registration

Gatekeepers recognize one of two types of terminal aliases, or terminal names:

- H.323 IDs, which are arbitrary, case-sensitive text strings
- E.164 addresses, which are telephone numbers

If an H.323 network deploys interzone communication, each terminal should at least have a fully qualified e-mail name as its H.323 identification (ID), for example, bob@cisco.com. The domain name of the e-mail ID should be the same as the configured domain name for the gatekeeper of which it is to be a member. As in the previous example, the domain name would be cisco.com.

Interzone Communication

To allow endpoints to communicate between zones, gatekeepers must be able to determine which zone an endpoint is in and be able to locate the gatekeeper responsible for that zone. If the Domain Name System (DNS) mechanism is available, a DNS domain name can be associated with each gatekeeper. See the DNS configuration task in the "Configuring Intergatekeeper Communication" section to understand how to configure DNS.

RADIUS and TACACS+

Version 1 of the H.323 specification does not provide a mechanism for authenticating registered endpoints. Credential information is not passed between gateways and gatekeepers. However, by enabling AAA on the gatekeeper and configuring for RADIUS and TACACS+, a rudimentary form of identification can be achieved.

If the AAA feature is enabled, the gatekeeper attempts to use the registered aliases along with a password and completes an authentication transaction to a RADIUS and TACACS+ server. The registration will be accepted only if RADIUS and TACACS+ successfully authenticates the name.

The gatekeeper can be configured so that a default password can be used for all users. The gatekeeper can also be configured so that it recognizes a password separator character that allows users to piggyback their passwords onto H.323-ID registrations. In this case, the separator character separates the ID and password fields.

Note The names loaded into RADIUS and TACACS+ are probably not the same names provided for dial access because they may all have the same password.

Accounting via RADIUS and TACACS+

If AAA is enabled on the gatekeeper, the gatekeeper will emit an accounting record each time a call is admitted or disconnected.

Interzone Routing Using E.164 Addresses

Interzone routing may be configured using E.164 addresses.

Two types of address destinations are used in H.323 calls. The destination can be specified using either an H.323-ID address (a character string) or an E.164 address (a string that contains telephone keypad characters). The way interzone calls are routed depends on the type of address being used.

When using H.323-ID addresses, interzone routing is handled through the use of domain names. For example, to resolve the domain name bob@cisco.com, the source endpoint gatekeeper finds the gatekeeper for cisco.com and sends it the location request for the target address bob@cisco.com. The destination gatekeeper looks in its registration database, sees bob registered, and returns the appropriate IP address to get to bob.

When using E.164 addresses, call routing is handled through zone prefixes and gateway-type prefixes, also referred to as technology prefixes. The zone prefixes, which are typically area codes, serve the same purpose as domain names in H.323-ID address routing. Unlike domain names, however, more than one zone prefix can be assigned to one gatekeeper, but the same prefix cannot be shared by more than one gatekeeper.

Use the **zone prefix** command to define gatekeeper responsibilities for area codes. The command can also be used to tell the gatekeeper which prefixes are in its own zones and which remote gatekeepers are responsible for other prefixes.

Note Area codes are used as an example in this section, but a zone prefix need not be an area code. It can be a country code, an area code plus local exchange (NPA-NXX), or any other logical hierarchical partition.

The following sample command shows how to configure a gatekeeper with the knowledge that zone prefix 212..... (that is, any address beginning with area code 212 and followed by seven arbitrary digits) is handled by gatekeeper gk-ny:

```
my-gatekeeper(config-gk)# zone prefix gk-ny 212.......
```

When my-gatekeeper is asked to admit a call to destination address 2125551111, it knows to send the location request to gk-ny.

However, once the query gets to gk-ny, gk-ny still needs to resolve the address so that the call can be sent to its final destination. There could be an H.323 endpoint that has registered with gk-ny with that E.164 address, in which case gk-ny would return the IP address for that endpoint. However, it is more likely that the E.164 address belongs to a non-H.323 device, such as a telephone or an H.320 terminal.

Because non-H.323 devices do not register with gatekeepers, gk-ny has no knowledge of which device the address belongs to or which type of device it is, so the gatekeeper cannot decide which gateway should be used for the *hop off* to the non-H.323 device. (The term *hop off* refers to the point at which the call leaves the H.323 network and is destined for a non-H.323 device.)

Note The number of zone prefixes defined for a directory gatekeeper that is dedicated to forwarding LRQs, and not for handling local registrations and calls, should not exceed 10,000; 4 MB of memory must be dedicated to describing zones and zone prefixes to support this maximum number of zone prefixes. The number of zone prefixes defined for a gatekeeper that handles local registrations and calls should not exceed 2000.

To enable the gatekeeper to select the appropriate hop-off gateway, use the **gw-type-prefix** command to configure technology or gateway-type prefixes. Select technology prefixes to denote different types or classes of gateways. The gateways are then configured to register with their gatekeepers using these technology prefixes.

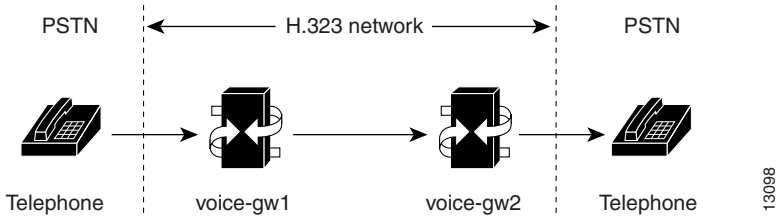
For example, voice gateways might register with technology prefix 1#, and H.320 gateways might register with technology prefix 2#. If there are several gateways of the same type, configure them to register with the same prefix type. By having them register with the same prefix type, the gatekeeper treats the gateways as a pool out of which a random selection is made whenever a call for that prefix type arrives. If a gateway can serve more than one type of hop-off technology, it can register more than one prefix type with the gatekeeper.

Callers will need to know the technology prefixes that are defined. The callers will need to know the type of device they are trying to reach and will need to prepend the appropriate technology prefix to the destination address to indicate the type of gateway needed to reach the destination.

For example, callers might request 1#2125551111 if they know that address 2125551111 is for a telephone and that the technology prefix for voice gateways is 1#. The voice gateway is configured with a dial peer (using the **dial-peer** command) so that when the gateway receives the call for 1#2125551111, it strips off the technology prefix 1# and bridges the next leg of the call to the telephone at 2125551111.

In cases in which the call scenario is as shown in Figure 57, voice-gw1 can be configured to prepend the voice technology prefix 1# so that the use of technology prefixes is completely transparent to the caller.

Figure 57 Call Scenario



Additionally, in using the **gw-type-prefix** command, a particular gateway-type prefix can be defined as the default gateway type to be used for addresses that cannot be resolved. It also forces a technology prefix to always hop off in a particular zone.

If the majority of calls hop off on a particular type of gateway, the gatekeeper can be configured to use that type of gateway as the default type so that callers no longer have to prepend a technology prefix on the address. For example, if voice gateways are mostly used in a network, and all voice gateways have been configured to register with technology prefix 1#, the gatekeeper can be configured to use 1# gateways as the default technology if the following command is entered:

```
my-gatekeeper(config-gk)# gw-type-prefix 1# default-technology
```

Now a caller no longer needs to prepend 1# to use a voice gateway. Any address that does not contain an explicit technology prefix will be routed to one of the voice gateways that registered with 1#.

With this default technology definition, a caller could ask the gatekeeper for admission to 2125551111. If the local gatekeeper does not recognize the zone prefix as belonging to any remote zone, it will route the call to one of its local (1#) voice gateways so that the call hops off locally. However, if it knows that gk-ny handles the 212 area code, it can send a location request for 2125551111 to gk-ny. This requires that gk-ny also be configured with some default gateway type prefix and that its voice gateways be registered with that prefix type.

Note For ease of maintenance, the same prefix type should be used to denote the same gateway type in all zones under your administration. No more than 50 different technology prefixes should be registered per zone.

Also, with the **gw-type-prefix** command, a hop off can be forced to a particular zone. When an endpoint or gateway makes a call-admission request to its gatekeeper, the gatekeeper determines the destination address by first looking for the technology prefix. When that is matched, the remaining string is compared against known zone prefixes. If the address is determined to be a remote zone, the entire address, including technology and zone prefixes, is sent to the remote gatekeeper in a location request. That remote gatekeeper then uses the technology prefix to decide on which of its gateways to hop off. In other words, the zone prefix (defined using the **zone prefix** command) determines the routing to a zone, and once there, the technology prefix (defined using the **gw-type-prefix** command) determines the gateway to be used in that zone. The zone prefix takes precedence over the technology prefix.

This behavior can be overridden by associating a forced hop-off zone with a particular technology prefix. Associating a forced hop-off zone with a particular technology prefix forces the call to the specified zone, regardless of what the zone prefix in the address is. As an example, you are in the 408 area code and want callers to the 212 area code in New York to use H.323-over-IP and hop off there because it saves on costs. However, the only H.320 gateway is in Denver. In this example, calls to H.320 endpoints must be forced to hop off in Denver, even if the destination H.320 endpoint is in the 212 area code. The forced hop-off zone can be either a local zone (that is, one that is managed by the local gatekeeper) or a remote zone.

HSRP Support

Cisco routers support Hot Standby Router Protocol (HSRP), which allows one router to serve as a backup to another router. Cisco gatekeepers can be configured to use HSRP so that when one gatekeeper fails, the standby gatekeeper assumes its role.

To configure a gatekeeper to use HSRP, perform the following tasks:

- Select one interface on each gatekeeper to serve as the HSRP interface and configure these two interfaces so that they belong to the same HSRP group but have different priorities. The one with the higher priority will be the active gatekeeper; the other assumes the standby role. Make a note of the virtual HSRP IP address shared by both of these interfaces. (For details on HSRP and HSRP configuration, refer to the *Cisco IOS IP Configuration Guide*.)
- Configure the gatekeepers so that the HSRP virtual IP address is the RAS address for all local zones.
- Make sure that the gatekeeper-mode configurations on both routers are identical.
- If the endpoints and gateways are configured so that they use a specific gatekeeper address (rather than multicasting), use the HSRP virtual IP address as the gatekeeper address. You can also let the endpoints and gateways find the gatekeeper by multicasting. As long as it is on standby status, the secondary gatekeeper neither receives nor responds to multicast or unicast requests.

As long as both gatekeepers are up, the one with the higher priority on its HSRP interface will be the active gatekeeper. If this active gatekeeper fails, or if its HSRP interface fails, the standby HSRP interface assumes the virtual HSRP address and, with it, the active gatekeeper role. When the gatekeeper with the higher HSRP priority comes back online, it reclaims the HSRP virtual address and the gatekeeper function, while the secondary gatekeeper goes back to standby status.

Note Gatekeeper failover will not be completely transparent to endpoints and gatekeepers. When the standby gatekeeper takes over, it does not have the state of the failed gatekeeper. If an endpoint that had registered with the failed gatekeeper now makes a request to the new gatekeeper, the gatekeeper responds with a reject, indicating that it does not recognize the endpoint. The endpoint must reregister with the new gatekeeper before it can continue H.323 operations.

For an example of configuring gatekeeper HSRP support, see the "H.323 Gatekeeper and Proxy Configuration Examples" section.

H.323 Proxy Features

Each of the following sections describes how the proxy feature can be used in an H.323 network:

- Security, page 297
- Quality of Service, page 301
- Application-Specific Routing, page 301

Security

When terminals signal each other directly, they must have direct access to each other's addresses. This exposes an attacker to key information about a network. When a proxy is used, the only addressing information that is exposed to the network is the address of the proxy; all other terminal and gateway addresses are hidden.

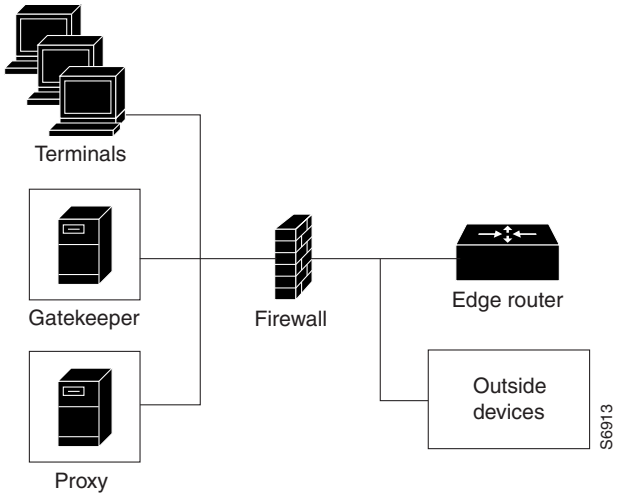
There are several ways to use a proxy with a firewall to enhance network security. The configuration to be used depends on how capable the firewall is of handling the complex H.323 protocol suite. Each of the following sections describes a common configuration for using a proxy with a firewall:

- Proxy Inside the Firewall, page 298
- Proxy in Co-Edge Mode, page 299
- Proxy Outside the Firewall, page 300
- Proxies and NAT, page 300

Proxy Inside the Firewall

H.323 is a complex, dynamic protocol that consists of several interrelated subprotocols. During H.323 call setup, the ports and addresses released with this protocol require a detailed inspection as the setup progresses. If the firewall does not support this dynamic access control based on the inspection, a proxy can be used just inside the firewall. The proxy provides a simple access control scheme, as illustrated in Figure 58.

Figure 58 Proxy Inside the Firewall

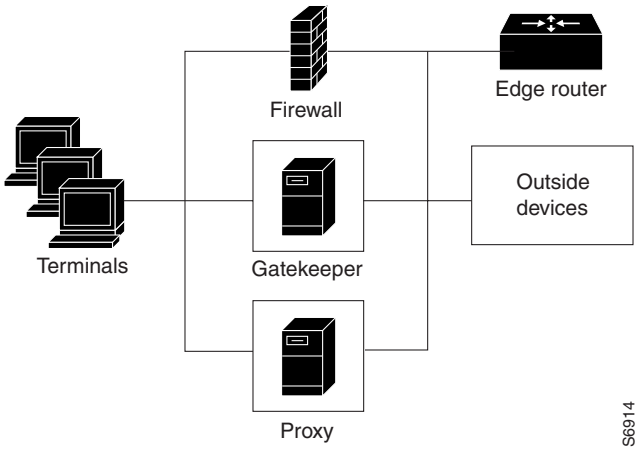


Because the gatekeeper (using RAS) and the proxy (using call setup protocols) are the only endpoints that communicate with other devices outside the firewall, it is simple to set up a tunnel through the firewall to allow traffic destined for either of these two endpoints to pass through.

Proxy in Co-Edge Mode

If H.323 terminals exist in an area with local interior addresses that must be translated to valid exterior addresses, the firewall must be capable of decoding and translating all addresses passed in the various H.323 protocols. If the firewall is not capable of this translation task, a proxy may be placed next to the firewall in a co-edge mode. In this configuration, interfaces lead to both inside and outside networks. (See Figure 59.)

Figure 59 Proxy in Co-Edge Mode



In co-edge mode, the proxy can present a security risk. To avoid exposing a network to unsolicited traffic, configure the proxy to route only proxied traffic. In other words, the proxy routes only H.323 protocol traffic that is terminated on the inside and then repeated to the outside. Traffic that moves in the opposite direction can be configured this way as well.

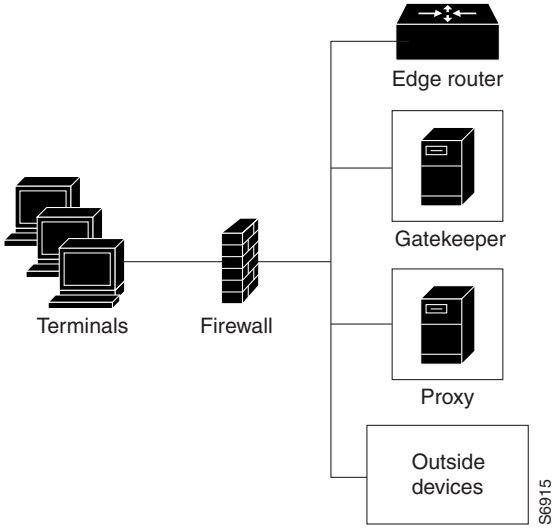
Proxy Outside the Firewall

To place the proxy and gatekeeper outside the firewall, two conditions must exist. First, the firewall must support H.323 dynamic access control. Second, Network Address Translation (NAT) must not be in use.

If NAT is in use, each endpoint must register with the gatekeeper for the duration of the time it is online. This will quickly overwhelm the firewall because a large number of relatively static, internal-to-external address mappings will need to be maintained.

If the firewall does not support H.323 dynamic access control, the firewall can be configured with static access lists that allow traffic from the proxy or gatekeeper through the firewall. This can present a security risk if an attacker can *spoof*, or simulate, the IP addresses of the gatekeeper or proxy and use them to attack the network. Figure 60 illustrates proxy outside the firewall.

Figure 60 Proxy Outside the Firewall



Proxies and NAT

When a firewall is providing NAT between an internal and an external network, proxies may allow H.323 traffic to be handled properly, even in the absence of a firewall that can translate addresses for H.323 traffic. Table 24 and Table 25 provide guidelines for proxy deployment for networks that use NAT.

Table 24 Guidelines for Networks That Use NAT

For Networks Using NAT	Firewall with H.323 NAT	Firewall Without H.323 NAT
Firewall with dynamic access control	Gatekeeper and proxy inside the firewall	Co-edge gatekeeper and proxy
Firewall without dynamic access control	Gatekeeper and proxy inside the firewall, with static access lists on the firewall	Co-edge gatekeeper and proxy

Table 25 Guidelines for Networks That Do Not Use NAT

For Networks Not Using NAT	Firewall with H.323. NAT	Firewall Without H.323 NAT
Firewall with Dynamic Access Control	Gatekeeper and proxy inside the firewall Gatekeeper and proxy outside the firewall	Gatekeeper and proxy inside the firewall Gatekeeper and proxy outside the firewall
Firewall Without Dynamic Access Control	Gatekeeper and proxy inside the firewall, with static access lists on the firewall	Gatekeeper and proxy inside the firewall, with static access lists on the firewall

Quality of Service

Quality of service (QoS) enables complex networks to control and predictably service a variety of applications. QoS expedites the handling of mission-critical applications while sharing network resources with noncritical applications. QoS also ensures available bandwidth and minimum delays required by time-sensitive multimedia and voice applications. In addition, QoS gives network managers control over network applications, improves cost-efficiency of WAN connections, and enables advanced differentiated services. QoS technologies are elemental building blocks for other Cisco IOS-enabling services such as its H.323-compliant gatekeeper. Overall call quality can be improved dramatically in the multimedia network by using pairs of proxies between regions of the network where QoS can be requested.

When two H.323 terminals communicate directly, the resulting call quality can range from good (for high-bandwidth intranets) to poor (for most calls over the public network). As a result, deployment of H.323 is almost always predicated on the availability of some high-bandwidth, low-delay, low-packet-loss network that is separate from the public network or that runs overlaid with the network as a premium service and adequate QoS.

Adequate QoS usually requires terminals that are capable of signaling such premium services. There are two major ways to achieve such signaling:

- RSVP to reserve flows having adequate QoS based on the media codecs of H.323 traffic
- IP precedence bits to signal that the H.323 traffic is special and that it deserves higher priority

Unfortunately, the vast majority of H.323 terminals cannot achieve signaling in either of these ways.

The proxy can be configured to use any combination of RSVP and IP precedence bits.

The proxy is not capable of modifying the QoS between the terminal and itself. To achieve the best overall QoS, ensure that terminals are connected to the proxy using a network that intrinsically has good QoS. In other words, configure a path between a terminal and proxy that provides good bandwidth, delay, and packet-loss characteristics without the terminal needing to request special QoS. A high-bandwidth LAN works well for this.

Application-Specific Routing

To achieve adequate QoS, a separate network may be deployed that is partitioned away from the standard data network. The proxy can take advantage of such a partitioned network using a feature known as application-specific routing (ASR).

Application-specific routing is simple. When the proxy receives outbound traffic, it directs traffic to an interface that is connected directly to the QoS network. The proxy does not send the traffic using an interface that is specified for the regular routing protocol. Similarly, inbound traffic from other proxies is received on the interface that is connected to the QoS network. This is true if all these other proxies around the QoS network use ASR in a consistent fashion. ASR then ensures that ordinary traffic is not routed into the QoS network by mistake.

Implementation of ASR ensures the following:

- Each time a connection is established with another proxy, the proxy automatically installs a host route pointing at the interface designated for ASR.
- The proxy is configured to use a loopback interface address. The proxy address is visible to both the ASR interface and all regular interfaces, but there are no routes established between the loopback interface and the ASR interface. This ensures that no non-H.323 traffic is routed through the ASR interface.

Note ASR is not supported on Frame Relay or ATM interfaces for the Cisco MC3810 platform.

H.323 Prerequisite Tasks and Restrictions

This section contains prerequisite tasks and restrictions for configuring H.323 gatekeepers and proxies.

Redundant H.323 Zone Support

Redundant H.323 zone support has the following restrictions and limitations:

- The gateway can register with only one gatekeeper at any given time.
- Only E.164 address resolution is supported.
- Because the gateway can register with only one gatekeeper at a time, redundant H.323 zone support provides only redundancy and does not provide any load balancing.
- Although redundant H.323 zone support allows you to configure alternate gatekeepers, it will not insert information in the alternate gatekeeper field of some RAS messages.

Gatekeeper-to-Gatekeeper Redundancy and Load-Sharing Mechanism

The gatekeeper-to-gatekeeper redundancy and load-sharing mechanism has the following restrictions and limitations:

- The gatekeeper-to-gatekeeper redundancy and load-sharing mechanism requires the Cisco H.323 VoIP Gatekeeper for Cisco Access Platforms feature.
- The order in which LRQs are sent to the gatekeepers is based on the order in which the gatekeepers are listed. You cannot specify a priority number for a gatekeeper.
- Regardless of the order in which the LRQs are sent, the gateway will still use the first gatekeeper that sends an LCF.
- The settings for delay between LRQs and the LRQ window are global and cannot be set on a per-zone or technology-prefix basis.

- The number of remote gatekeepers multiplied by the delay per LRQ cannot exceed the Routing Information Protocol (RIP) timeout. Therefore, we recommend that you limit your list of remote gatekeepers to two or three.
- If LRQ forwarding is enabled on the directory gatekeeper, the *sequential* setting for LRQs is ignored.
- Only E.164 address resolution is supported.
- Using redundant H.323 zone support in the "directory gatekeeper" can generate extra RAS messages. Therefore, the number of "directory gatekeeper" levels should be kept to a minimum (two or three at the maximum).

H.323 Gatekeeper Configuration Task List

To configure Cisco gatekeepers, perform the tasks in the following sections. The tasks in these two sections are required.

- Configuring the Gatekeeper, page 303 (Required)
- Configuring the Proxy, page 332 (Required)

Configuring the Gatekeeper

To configure gatekeepers, perform the tasks in the following sections. All of the tasks listed are required.

- Starting a Gatekeeper, page 304
  - Configuring Intergatekeeper Communication, page 307
- Configuring Redundant H.323 Zone Support, page 308
- Configuring Local and Remote Gatekeepers, page 309
- Configuring Redundant Gatekeepers for a Zone Prefix, page 310
- Configuring Redundant Gatekeepers for a Technology Prefix, page 311
- Configuring Static Nodes, page 313
- Configuring H.323 Users via RADIUS, page 314
- Configuring a RADIUS/AAA Server, page 318
- Configuring User Accounting Activity for RADIUS, page 320
- Configuring E.164 Interzone Routing, page 321
- Configuring H.323 Version 2 Features, page 322
  - Configuring a Dialing Prefix for Each Gateway, page 323
  - Configuring a Prefix to a Gatekeeper Zone List, page 326
  - Configuring a Gatekeeper for Interaction with External Applications, page 325
  - Configuring Gatekeeper Triggers for Interaction with External Applications, page 327
  - Configuring Redundant H.323 Zone Support, page 308
  - Configuring a Forced Disconnect on a Gatekeeper, page 332

Starting a Gatekeeper

To enter gatekeeper configuration mode and to start the gatekeeper, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. The arguments are as follows: • *gatekeeper-name*—Specifies the gatekeeper name or zone name. This is usually the fully domain-qualified host name of the gatekeeper. For example, if the domain name is cisco.com, the gatekeeper name might be gk1.cisco.com. However, if the gatekeeper is controlling multiple zones, the gatekeeper name for each zone should be some unique string that has a mnemonic value. • *domain-name*—Specifies the domain name served by this gatekeeper. • *ras-IP-address*—(Optional) Specifies the IP address of one of the interfaces on the gatekeeper. When the gatekeeper responds to gatekeeper discovery messages, it signals the endpoint or gateway to use this address in future communications. **Note** Setting this address for one local zone makes it the address used for all local zones.

Command	Purpose
Step 3 `Router(config-gk)# zone prefix gatekeeper-name e164-prefix [blast | seq] [gw-priority priority gw-alias [gw-alias, ...]]`	Adds a prefix to the gatekeeper zone list. The keywords and arguments are as follows: • *gatekeeper-name*—Specifies the name of a local or remote gatekeeper, which must have been defined by using the **zone local** or **zone remote** command. • *e164-prefix*—Specifies an E.164 prefix in standard form followed by dots (.). Each dot represents a number in the E.164 address. For example, 212..... is matched by 212 and any 7 numbers. **Note** Although a dot to represent each digit in an E.164 address is the preferred configuration method, you can also enter an asterisk (*) to match any number of digits. • **blast**—(Optional) If you list multiple hopoffs, indicates that the location requests (LRQs) should be sent simultaneously to the gatekeepers based on the order in which they were listed. The default is seq. • **seq**—(Optional) If you list multiple hopoffs, indicates that the LRQs should be sent sequentially to the gatekeepers based on the order in which they were listed. The default is seq. • **gw-priority** *priority gw-alias*—(Optional) Use the **gw-priority** option to define how the gatekeeper selects gateways in its local zone for calls to numbers that begin with prefix e164-prefix. Do not use this option to set priority levels for a prefix assigned to a remote gatekeeper. Use values from 0 to 10. A 0 value prevents the gatekeeper from using the gateway *gw-alias* for that prefix. Value 10 places the highest priority on gateway *gw-alias*. If you do not specify a priority value for a gateway, the value 5 is assigned. To assign the same priority value for one prefix to multiple gateways, list all the gateway names after the pri-0-to-10 value. The *gw-alias* name is the H.323 ID of a gateway that is registered or will register with the gatekeeper. This name is set on the gateway with the **h323-gateway voip h.323-id** command.

	Command	Purpose
Step 4	`Router(config-gk)# zone subnet local-gatekeeper-name [default | subnet-address {/bits-in-mask | mask-address} enable]`	Defines a set of subnets that constitute the gatekeeper zone. Enables the gatekeeper for each of these subnets and disables it for all other subnets. (Repeat for all subnets.) The keywords and arguments are as follows: • *local-gatekeeper-name*—Specifies the name of the local gatekeeper. • **default**—(Optional) Applies to all other subnets that are not specifically defined by the **zone subnet** command. • *subnet-address*—(Optional) Specifies the address of the subnet that is being defined. • *bits-in-mask*—(Optional) Specifies the number of bits of the mask to be applied to the subnet address. **Note** The slash must be entered before this argument. • *mask-address*—(Optional) Specifies the mask (in dotted string format) to be applied to the subnet address. • **enable**—(Optional) Specifies that the gatekeeper accepts discovery and registration from the specified subnets. **Note** To define the zone as being all but one set of subnets by disabling that set and enabling all other subnets, use the **no** form of the command as follows: Configure **no zone subnet** *local-gatekeeper-name subnet-address* {*/bits-in-mask* | *mask-address*} **enable**. **Note** To accept the default behavior, which is that all subnets are enabled, use the **no** form of the command as follows: **no zone subnet** *local-gatekeeper-name* **default enable**.
Step 5	`Router(config-gk)# no shutdown`	Brings the gatekeeper online.

The *local-gatekeeper-name* argument should be a Domain Name System (DNS) host name if DNS is to be used to locate remote zones.

The **zone subnet** command may be used more than once to create a list of subnets controlled by a gatekeeper. The subnet masks need not match actual subnets in use at your site. For example, to specify a particular endpoint, show its address as a 32-bit netmask.

If a local gatekeeper name is contained in the message, it must match the *local-gatekeeper-name* argument.

Note To explicitly enable or disable a particular endpoint, specify its host address using a 32-bit subnet mask.

Configuring Intergatekeeper Communication

This section describes two ways to configure intergatekeeper communication:

- Via DNS, page 307
- Manual Configuration, page 308

Via DNS

To configure intergatekeeper communication using DNS, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip name-server dns-server-name [server-address2...server-address6]`	Specifies the DNS server address. The arguments are as follows: • *dns-server-name*— Specifies the IP address of the name server. • *server-address2...server-address6*—(Optional) IP addresses of additional name servers (a maximum of six name servers).
Step 2	`Router(config)# ip domain-name name`	Defines a default domain name that the Cisco IOS software uses to complete unqualified host names (names without a dotted-decimal domain name). The *name* argument specifies the default domain name used to complete unqualified host names. Do not include the initial period that separates an unqualified name from the domain name.

For all gatekeepers in the system, enter a text record of the form into DNS:

ras [*gk-id@*] *host* [*:port*] [*priority*]

The *gk-id* argument is an optional gatekeeper ID. If the optional gatekeeper ID is not specified, *host* is used as the gatekeeper ID.

The *host* argument is either an IP address or the actual host name of the gatekeeper in the form *host.some_domain.com*.

The *port* argument, if specified, should be some port number other than RAS port 1719.

The *priority* argument specifies the order in which the listed gatekeepers should be searched for endpoints. Gatekeepers with lower priorities are searched before those with higher numbers.

How you enter the text record for a particular domain depends on the DNS implementation. The following examples are for the Berkeley Internet Name Domain (BIND). These records are typically entered into the "hosts" database:

zone1.comintxt"ras gk.zone1.com"
zone2.comintxt"ras gk2@gk.zone2.com"

zone3.comintxt“ras gk.3@gk.zone3.com:1725”
zone4.comintxt“ras gk4@gk.zone4.com:1725 123”
zone5.comintxt“ras gk5@101.0.0.1:1725”

Manual Configuration

If you choose not to use DNS or if DNS is not available, configure intergatekeeper communication manually. To configure intergatekeeper manual communication, use the following command in gatekeeper configuration mode for every other gatekeeper in the network:

Command	Purpose
`Router(config-gk)# zone remote other-gatekeeper-name other-domain-name other-gatekeeper-ip-address [port-number]`	Statically specifies a remote zone if Domain Name System (DNS) is unavailable or undesirable. Enter this command for each gatekeeper. The arguments are as follows: • *other-gatekeeper-name*—Specifies the name of the remote gatekeeper. • *other-domain-name*—Specifies the domain name of the remote gatekeeper. • *other-gatekeeper-ip-address*—Specifies the IP address of the remote gatekeeper. • *port-number*—(Optional) Specifies the RAS signaling port number for the remote zone. Value ranges are from 1 to 65,535. If this option is not set, the default is the well-known RAS port number 1719.

Configuring Redundant H.323 Zone Support

Regardless of whether you specify sequential or blast, there is an order to how the LRQs are sent. With sequential, the LRQs are sent one at a time with a delay between each. With blast, the LRQs are sent back-to-back in a rapid sequence without any delay between them. The order in which zone and technology prefixes are configured determines the order in which the LRQs are sent to the remote gatekeepers. Using zone prefixes as an example, the local gatekeeper routes the call to the first zone that responds with an LCF. If the local gatekeeper is configured for a zone prefix that already has remote gatekeepers configured, the local gatekeeper will automatically put that zone prefix at the top of the list.

For example:

```
gatekeeper
 zone local gnet-2503-2-gk cisco.com
 zone remote gnet-2600-1-gk cisco.com 172.18.194.131 1719
 zone remote gnet-2503-3-gk cisco.com 172.18.194.134 1719
 zone prefix gnet-2600-1-gk 919.......
 zone prefix gnet-2503-6-gk 919.......
```

With this configuration, LRQs are first sent to gnet-2600-1-gk (which is the first zone prefix because it has a remote gatekeeper configured for it) and then to gnet-2503-6-gk (which is the second zone prefix). If you add the local gatekeeper to that zone prefix, it automatically goes to the top of the list, as shown below:

```
gatekeeper
 zone local gnet-2503-2-gk cisco.com
 zone remote gnet-2600-1-gk cisco.com 172.18.194.131 1719
```

```
zone remote gnet-2503-3-gk cisco.com 172.18.194.134 1719
zone prefix gnet-2503-2-gk 919.......
zone prefix gnet-2600-1-gk 919.......
zone prefix gnet-2503-6-gk 919.......
```

As you can see, the zone prefix for the local gatekeeper (gnet-2503-2-gk) has been inserted at the top of the zone prefix list. If the local gatekeeper can resolve the address, it will not send LRQs to the remote zones.

If you are configuring technology prefixes, the zone prefix for the local gatekeeper should be inserted at the top of the zone prefix list. If the local gatekeeper can resolve the address, it will not send LRQs to the remote zones.

Configuring Local and Remote Gatekeepers

To configure local and remote gatekeepers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. The arguments are as follows: • *gatekeeper-name*—Specifies the gatekeeper name or zone name. This is usually the fully domain-qualified host name of the gatekeeper. For example, if the domain name is cisco.com, the gatekeeper name might be gk1.cisco.com. However, if the gatekeeper is controlling multiple zones, the gatekeeper name for each zone should be some unique string that has a mnemonic value. • *domain-name*—Specifies the domain name served by this gatekeeper. • *ras-IP-address*—(Optional) Specifies the IP address of one of the interfaces on the gatekeeper. When the gatekeeper responds to gatekeeper discovery messages, it signals the endpoint or gateway to use this address in future communications. **Note** Setting this address for one local zone makes it the address used for all local zones.

	Command	Purpose
Step 3	`Router(config-gk)# zone remote` *other-gatekeeper-name other-domain-name other-gatekeeper-ip-address* [*port-number*]	Configures the remote gatekeeper. The arguments are as follows: • *other-gatekeeper-name*—Name of the remote gatekeeper. • *other-domain-name*—Domain name of the remote gatekeeper. • *other-gatekeeper-ip-address*—IP address of the remote gatekeeper. • *port-number*—(Optional) RAS signaling port number for the remote zone. Value ranges from 1 to 65,535. If this option is not set, the default is the well-known RAS port number 1719.

Configuring Redundant Gatekeepers for a Zone Prefix

To configure redundant gatekeepers for a zone prefix, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone prefix` *gatekeeper-name e164-prefix* [**blast** | **seq**] [**gw-priority** *priority gw-alias* [*gw-alias, ...*]]	Adds a prefix to the gatekeeper zone list. For an explanation of the keywords and arguments, see Step 3 of the configuration task table in the "Starting a Gatekeeper" section on page 304.

You can configure multiple remote gatekeepers for the same prefix, but only one of the gatekeepers defined for any given zone prefix can be local. It is recommended that you limit the number of remote gatekeepers that service the same zone prefix to two.

By default, LRQs are sent sequentially to the remote gatekeepers. If you would like the LRQs to be sent simultaneously (blast), you need only specify the **blast** keyword on one **zone prefix** command per E.164 prefix.

Verifying Zone Prefix Redundancy

To verify the order in which LRQs will be sent to the gatekeepers defined for a zone prefix, enter the **show gatekeeper zone prefix** command. The following output lists all the gatekeepers, in order, and the zone prefixes serviced by each.

```
router# show gatekeeper zone prefix

ZONE PREFIX TABLE
      =================
GK-NAME               E164-PREFIX
-------               -----------
c3620-1-gk            917300....
c2514-2-gk            917300....
c2600-1-gk            919.......
c2514-1-gk            919.......
```

To verify whether the LRQs will be sent sequentially or simultaneously to the gatekeepers, enter the **show running-config** command. If the LRQs will be sent simultaneously, blast will appear beside the first entry for a particular zone (as shown in the following output for zone 919).

```
Router# show running-config

Building configuration...

Current configuration:
!
gatekeeper
 zone remote c3620-1-gk cisco.com 172.18.194.79 1719
 zone remote c2514-2-gk cisco.com 172.18.194.89 1719
 zone remote gk-cisco-paul cisco.com 172.18.193.155 1719
 zone prefix c3620-1-gk 917300....
 zone prefix c2514-2-gk 917300....
 zone prefix c2514-2-gk 919....... blast
 zone prefix c3620-1-gk 919.......
```

Configuring Redundant Gatekeepers for a Technology Prefix

To configure redundant gatekeepers for a technology prefix, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# gw-type-prefix` *type-prefix* `[[hopoff` *gkid1*`] [hopoff` *gkid2*`] [hopoff` *gkidn*`] [seq | blast]] [default-technology] [[gw ipaddr` *ipaddr* `[`*port*`]]...]`	Configures the gatekeepers to service a technology zone and specifies whether LRQs should be sent in blast or sequential fashion. The default is sequential. The keywords and arguments are as follows: • *type-prefix*—Specifies that a technology prefix is recognized and stripped before checking for the zone prefix. It is strongly recommended that you select technology prefixes that do not lead to ambiguity with zone prefixes. Do this by using the # character to terminate technology prefixes, for example, 3#. • **hopoff** *gkid*—(Optional) Specifies the gatekeeper where the call is to hop off, regardless of the zone prefix in the destination address. The *gkid* argument refers to a gatekeeper previously configured using the **zone local** or **zone remote** command. You can enter this keyword and argument multiple times to configure redundant gatekeepers for a given technology prefix.

Command	Purpose
	• **seq | blast**—(Optional) If multiple hopoffs are listed, indicates that the location requests (LRQs) should be sent sequentially or simultaneously (blast) to the gatekeepers based on the order in which they were listed. The default is to send them sequentially. • **default-technology**—(Optional) Specifies that gateways that register with this prefix option are used as the default for routing any addresses that are otherwise unresolved. • **gw ipaddr** *ipaddr* [*port*]—(Optional) Indicates that the gateway is incapable of registering technology prefixes. When it registers, it adds the gateway to the group for this type-prefix, just as if it had sent the technology prefix in its registration. This parameter can be repeated to associate more than one gateway with a technology prefix.

You can enter the **hopoff** keyword and *gkid* argument multiple times in the same command to define a group of gatekeepers that will service a given technology prefix. After you have listed all of the gatekeepers that will service that technology zone, you can specify whether the LRQs should be sent in blast or sequential fashion.

Note Only one of the gatekeepers in the hopoff list can be local. We recommend that you limit the number of remote gatekeepers that service the same technology prefix to two.

Verifying Technology Prefix Redundancy

To verify that multiple gatekeepers are defined for a technology prefix, enter the **show gatekeeper gw-type-prefix** command. The following output displays the gateway technology prefix table.

```
router# show gatekeeper gw-type-prefix

(GATEWAYS-TYPE PREFIX TABLE
==============================
Prefix:3#*    (Hopoff zone c2600-1-gk c2514-1-gk)
```

To verify whether the LRQs will be sent sequentially or simultaneously to the gatekeepers, enter the **show running-config** command. If the LRQs will be sent simultaneously, blast will appear at the end of the gw-type-prefix line (as shown below).

```
Router# show running-config

Building configuration...

Current configuration:
!
gatekeeper
 zone remote c2600-1-gk cisco.com 172.18.194.70 1719
 zone remote c2514-1-gk cisco.com 172.18.194.71 1719
 gw-type-prefix 3#* hopoff c2600-1-gk hopoff c2514-1-gk blast
```

Configuring Static Nodes

In some cases, the registration information is not accessible for a terminal or endpoint from any gatekeeper. This inaccessible registration information may be because the endpoint does not use RAS, is in an area where no gatekeeper exists, or is in a zone where the gatekeeper addressing is unavailable either through DNS or through configuration.

These endpoints can still be accessed via a gatekeeper by entering them as static nodes. To enter the endpoints as static nodes, obtain the address of the endpoint and then use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. For an explanation of the arguments, see Step 2 of the configuration task table in the "Starting a Gatekeeper" section on page 304.
Step 3	`Router(config-gk)# alias static` *ip-signalling-addr* [*port*] **gkid** *gatekeeper-name* [**ras** *ip-ras-addr port*] [**terminal** | **mcu** | **gateway** {**h320** |**h323-proxy** | **voip**}] [**e164** *e164-address*] [**h323id** *h323-id*]	Creates a static entry in the local alias table for each E.164 address. Repeat this step for each E.164 address you want to add for the endpoint. The keywords and arguments are as follows: • *ip-signalling-addr*—Specifies the IP address of the H.323 node, used as the address to signal when establishing a call. • *port*—(Optional) Specifies the port number other than the endpoint call-signaling well-known port number (1720). • **gkid** *gatekeeper-name*—Specifies the name of the local gatekeeper of whose zone this node is a member. • **ras** *ip-ras-addr*—(Optional) Specifies the node remote access server (RAS) signaling address. If omitted, the *ip-signalling-addr* parameter is used in conjunction with the RAS well-known port. • *port*—(Optional) Specifies a port number other than the RAS well-known port number (1719). • **terminal—**(Optional) Indicates that the alias refers to a terminal. • **mcu**—(Optional) Indicates that the alias refers to a multiple control unit (MCU). • **gateway**—(Optional) Indicates that the alias refers to a gateway. • **h320**—(Optional) Indicates that the alias refers to an H.320 node.**h320**—(Optional) Indicates that the alias refers to an H.320 node.

Command	Purpose
	• **h-323 proxy**—(Optional) Indicates that the alias refers to an H.323 proxy. • **voip**—(Optional) Indicates that the alias refers to VoIP. • **e164** *e164-address*—(Optional) Specifies the node E.164 address. This keyword and argument can be used more than once to specify as many E.164 addresses as needed. Note that there is a maximum number of 128 characters that can be entered for this address. To avoid exceeding this limit, you can enter multiple **alias static** commands with the same call-signaling address and different aliases. • **h323-id** *h323-id*—(Optional) Specifies the node H.323 alias. This keyword and argument can be used more than once to specify as many H.323 identification (ID) aliases as needed. Note that there is a maximum number of 256 characters that can be entered for this address. To avoid exceeding this limit, you can enter multiple commands with the same call signaling address and different aliases.

Configuring H.323 Users via RADIUS

To authenticate H.323 users via RADIUS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) access model.
Step 2	`Router(config)# aaa authentication login {default | list-name} method1 [method2...]`	Sets AAA authentication at login. The keywords and arguments are as follows: • **default**—Uses the listed authentication methods that follow this keyword as the default list of methods when a user logs in. • *list-name*—Specifies the character string used to name the list of authentication methods activated when a user logs in. • *method1* [*method2...*]—Specifies that at least one of the keywords described below be used: – **enable**—Uses the enable password for authentication. – **krb5**—Uses Kerberos 5 for authentication..

Command	Purpose
	– **krb5-telnet**—Uses the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router. – **line**—Uses the line password for authentication. – **local**—Uses the local username database for authentication – **local-case**—Uses case-sensitive local username authentication. – **none**—Uses no authentication. – **group radius**—Uses the list of all RADIUS servers for authentication. – **group tacacs+**—Uses the list of all TACACS+ servers for authentication. – **group group-name**—Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **group server radius** or **aaa group server tacacs+** command.
Step 3 `Router(config)# radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]`	Specifies the RADIUS server host. The keywords and arguments are as follows: • *hostname*—Specifies the Domain Name System (DNS) name of the RADIUS server host. • *ip-address*—Specifies the IP address of the RADIUS server host. • **auth-port**—(Optional) Specifies the User Datagram Protocol (UDP) destination port for authentication requests. • *port-number*—(Optional) Specifies the port number for authentication requests; the host is not used for authentication if set to 0. If unspecified, the port number defaults to 1645. • **acct-port**—(Optional) Specifies the UDP destination port for accounting requests. • *port-number*—(Optional) Specifies the port number for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646. • **acct-port**—(Optional) Specifies the UDP destination port for accounting requests. • *port-number*—(Optional) Specifies the port number for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646.

Command	Purpose
	• **timeout**—(Optional) Specifies the time interval (in seconds) for which the router waits for the RADIUS server to reply before retransmitting. This setting overrides the global value of the **radius-server timeout** command. If no timeout value is specified, the global value is used. Enter a value in the range of from 1 to 1000. • *seconds*—(Optional) Specifies the timeout value. Enter a value in the range of from 1 to 1000. If no timeout value is specified, the global value is used. • **retransmit**—(Optional) Specifies the number of times a RADIUS request is resent to a server if that server is not responding or responding slowly. This setting overrides the global setting of the **radius-server retransmit** command. • *retries*—(Optional) Specifies the retransmit value. Enter a value in the range of from 1 to 100. If no retransmit value is specified, the global value is used. • **key**—(Optional) Specifies the authentication and encryption key used between the router and the RADIUS daemon running on this RADIUS server. This key overrides the global setting of the **radius-server key** command. If no key string is specified, the global value is used. The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax. This is because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key. • *string*—(Optional) Specifies the authentication and encryption key for all RADIUS communications between the router and the RADIUS server. This key must match the encryption used on the RADIUS daemon. All leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

	Command	Purpose
Step 4	`Router(config)# radius-server key {0 string | 7 string | string}`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon. The arguments are as follows: • **0**—Specifies that an unencrypted key will follow. • *string*—Specifies the unencrypted (cleartext) shared key. • **7**—Specifies that a hidden key will follow. • *string*—Specifies the hidden shared key. • *string*—Specifies the unencrypted (cleartext) shared key.
Step 5	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 6	`Router(config-gk)# security {any | h323-id | e164} {password default password | password separator character}`	Enables authentication and authorization on a gatekeeper. The keywords and arguments are as follows: • **any**—Uses the first alias of an incoming Registration, Admission, and Status (RAS) registration, regardless of its type, as the means of identifying the user to RADIUS/TACACS+. • **h323-id**—Uses the first H.323 ID type alias as the means of identifying the user to RADIUS/TACACS+. • **e164**—Uses the first E.164 address type alias as the means of identifying the user to RADIUS/TACACS+. • **password default** *password*—Specifies the default password that the gatekeeper associates with endpoints when authenticating them with an authentication server. The password must be identical to the password on the authentication server.

Command	Purpose
	• **password separator** *character*—Specifies the character that endpoints use to separate the H.323-ID from the piggybacked password in the registration. This allows each endpoint to supply a user-specific password. The separator character and password will be stripped from the string before it is treated as an H.323-ID alias to be registered. Note that passwords may be piggybacked only in the H.323-ID, not the E.164 address. This is because the E.164 address allows a limited set of mostly numeric characters. If the endpoint does not wish to register an H.323-ID, it can still supply an H.323-ID that consists of just the separator character and password. This will be understood to be a password mechanism, and no H.323-ID will be registered.

After the previous steps have been completed, enter each user into the RADIUS database using either the default password if using the **security password default** command or the actual passwords if using the piggybacked password mechanism as the RADIUS authentication for that user. Enter either the user H.323-ID or the E.164 address, depending on how the gatekeeper was configured.

For more information about configuring AAA services or RADIUS, refer to the *Cisco IOS Security Configuration Guide*.

Configuring a RADIUS/AAA Server

To configure the RADIUS/AAA server with information about the gatekeeper for your network installation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) model.
Step 2	`Router(config)# aaa authentication login {default | list-name} method1 [method2...]`	Sets AAA authorization at login. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring H.323 Users via RADIUS" section on page 314.
Step 3	`Router(config)# radius-server deadtime minutes`	Improves the server response time when some servers might be unavailable. The *minutes* argument specifies the length of time, in minutes, for which a RADIUS server is skipped over by transaction requests, up to a maximum of 1440 minutes (24 hours).

	Command	Purpose
Step 4	`Router(config)# radius-server host {host-name | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]`	Specifies the RADIUS server host. For an explanation of the keywords and arguments, see Step 3 in the configuration task table in the "Configuring H.323 Users via RADIUS" section on page 314.
Step 5	`Router(config)# radius-server key {0 string | 7 string | string}`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon. For an explanation of the arguments, see Step 4 in the configuration task table in the "Configuring H.323 Users via RADIUS" section on page 314.

In addition to the above configuration, make sure that the following information is configured in your CiscoSecure AAA server:

- In the /etc/raddb/clients file, ensure that the following information is provided.

```
#Client Name            Key
#-----------       -------------------
gk215.cisco.com         testing123
```

Where:

gk215.cisco.com is resolved to the IP address of the gatekeeper requesting authentication.

- In the /etc/raddb/users file, ensure that the following information is provided:

```
taeduk@cisco.com  Password = "thiswouldbethepassword"
User-Service-Type = Framed-User,
Login-Service = Telnet
```

Where:

taeduk@cisco.com is the h323-id of the gateway authenticating to gatekeeper gk215.cisco.com.

Configuring User Accounting Activity for RADIUS

After AAA has been enabled and the gateway has been configured to recognize RADIUS as the remote security server providing authentication services, the next step is to configure the gateway to report user activity to the RADIUS server in the form of connection accounting records. To send connection accounting records to the RADIUS server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa accounting connection h323 {stop-only | start-stop | wait-start | none} [broadcast] group group-name`	Defines the accounting method list H.323 with RADIUS as a method. The keywords and arguments are as follows: • **stop-only**—Sends a "stop" accounting notice at the end of the requested user process. • **start-stop**—Sends a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process. The "start" accounting record is sent in the background. The requested user process begins regardless of whether the "start" accounting notice was received by the accounting server. • **wait-start**—Sends a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process. The "start" accounting record is sent in the background. The requested user process does not begin until the "start" accounting notice is received by the server. • **none**—Disables accounting services on this line or interface. • **broadcast**—(Optional) Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group. • **group** *group-name*—Specifies the server group to be used for accounting services. The following are valid server group names: – *string*—Specifies the character string used to name a server group. – **radius**—Uses list of all RADIUS hosts. – **tacacs+**—Uses list of all TACACS+ hosts.

	Command	Purpose
Step 2	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 3	`Router(config-gk)# aaa accounting`	Enables authentication, authorization, and accounting (AAA) of requested services for billing or security purposes when you use RADIUS or TACACS+.

For more information about AAA connection accounting services, refer to the *Cisco IOS Security Configuration Guide*.

Configuring E.164 Interzone Routing

With Cisco IOS Release 12.0(3)T and later releases, interzone routing may be configured using E.164 addresses. To configure interzone routing in the E.164 address space, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. For an explanation of the arguments, see Step 2 of the configuration task table in the "Starting a Gatekeeper" section on page 304.
Step 3	`Router(config-gk)# zone remote` *other-gatekeeper-name other-domain-name other-gatekeeper-ip-address* [*port-number*]	Statically specifies a remote zone if Domain Name System (DNS) is unavailable or undesirable. Enter this command for each gatekeeper. The arguments are as follows: • *other-gatekeeper-name*—Specifies the name of the remote gatekeeper. • *other-domain-name*—Specifies the domain name of the remote gatekeeper. • *other-gatekeeper-ip-address*—Specifies the IP address of the remote gatekeeper. • *port-number*—(Optional) Specifies the Registration, Admission, and Status (RAS) signaling port number for the remote zone. Value ranges are from 1 to 65,535. If this option is not set, the default is the well-known RAS port number 1719.

	Command	Purpose
Step 4	`Router(config-gk)# zone prefix gatekeeper-name e164-prefix [blast | seq] [gw-priority priority gw-alias [gw-alias, ...]]`	Adds a prefix to the gatekeeper zone list. For an explanation of the keywords and arguments, see Step 3 of the configuration task table in the "Starting a Gatekeeper" section on page 304.
Step 5	`Router(config-gk)# gw-type-prefix type-prefix [[hopoff gkid1] [hopoff gkid2] [hopoff gkidn] [seq | blast]] [default-technology] [[gw ipaddr ipaddr [port]]...]`	Configures the gatekeepers to service a technology zone and specifies whether location requests (LRQs) should be sent in blast or sequential fashion. The default is sequential. For an explanation of the keywords and arguments, see Step 2 of the configuration task table in the "Configuring Redundant Gatekeepers for a Technology Prefix" section on page 311.

Configuring H.323 Version 2 Features

To configure H.323 Version 2 features using the Cisco gatekeeper, perform the following configuration tasks. The first two tasks are required; the others are optional. Make sure that you include a priority value for selecting between multiple gateways when you configure the gatekeeper.

- Configuring a Dialing Prefix for Each Gateway, page 323 (Required)
- Configuring a Gatekeeper for Interaction with External Applications, page 325 (Required)
- Configuring a Prefix to a Gatekeeper Zone List, page 326 (Optional)
- Configuring Gatekeeper Triggers for Interaction with External Applications, page 327 (Optional)
- Configuring Inbound or Outbound Gatekeeper Proxied Access, page 330 (Optional)
- Configuring a Forced Disconnect on a Gatekeeper, page 332 (Optional)

See the "H.323 Applications" chapter for further information on H.323 Version 2 features supported by Cisco IOS software.

Configuring a Dialing Prefix for Each Gateway

To configure a dialing prefix for each gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. The arguments are as follows: • *gatekeeper-name*—Specifies the gatekeeper name or zone name. This is usually the fully domain-qualified host name of the gatekeeper. For example, if the *domain-name* is cisco.com, the *gatekeeper-name* might be gk1.cisco.com. However, if the gatekeeper is controlling multiple zones, the *gatekeeper-name* for each zone should be some unique string that has a mnemonic value. • *domain-name*—Specifies the domain name served by this gatekeeper. • *ras-IP-address*—(Optional) The IP address of one of the interfaces on the gatekeeper. When the gatekeeper responds to gatekeeper discovery messages, it signals the endpoint or gateway to use this address in future communications. **Note** Setting this address for one local zone makes it the address used for all local zones.

	Command	Purpose
Step 3	`Router(config-gk)# zone prefix` *gatekeeper-name e164-prefix* [**gw-priority** *pri-0-to-10 gw-alias* [*gw-alias, ...*]]	Adds a prefix to the gatekeeper zone list. To remove knowledge of a zone prefix, use the **no** form of this command with the gatekeeper name and prefix. To remove the priority assignment for a specific gateway, use the **no** form of this command with the **gw-priority** option. The keywords and arguments are as follows: • *gatekeeper-name*—Specifies the name of a local or remote gatekeeper, which must have been defined by using the **zone local** or **zone remote** command. • *e164-prefix*—Specifies an E.164 prefix in standard form followed by dots (.). Each dot represents a number in the E.164 address. For example, 212..... is matched by 212 and any seven numbers. **Note** Although a dot representing each digit in an E.164 address is the preferred configuration method, you can also enter an asterisk (*) to match any number of digits. • **gw-priority** *pri-0-to-10 gw-alias*—(Optional) Use the **gw-priority** option to define how the gatekeeper selects gateways in its local zone for calls to numbers that begin with prefix *e164-prefix*. Do not use this option to set priority levels for a prefix assigned to a remote gatekeeper. Use values from 0 to 10. A 0 value prevents the gatekeeper from using the gateway *gw-alias* for that prefix. Value 10 places the highest priority on gateway *gw-alias*. If you do not specify a priority value for a gateway, the value 5 is assigned. To assign the same priority value for one prefix to multiple gateways, list all the gateway names after the *pri-0-to-10* value. The *gw-alias* name is the H.323 identification (ID) of a gateway that is registered or that will register with the gatekeeper. This name is set on the gateway with the **h323-gateway voip h.323-id** command.

To put all your gateways in the same zone, use the **gw-priority** option and specify which gateways are used for calling different area codes. For example:

```
zone local localgk xyz.com
zone prefix localgk 408.......
zone prefix localgk 415....... gw-priority 10 gw1 gw2
zone prefix localgk 650....... gw-priority 0 gw1
```

The above commands accomplish the following:

- Domain xyz.com is assigned to gatekeeper localgk.
- Prefix 408 is assigned to gatekeeper localgk, and no gateway priorities are defined for it; therefore, all gateways registering to localgk can be used equally for calls to the 408 area code. No special gateway lists are built for the 408 prefix; a selection is made from the master list for the zone.
- The prefix 415 is added to gatekeeper localgk, and priority 10 is assigned to gateways gw1 and gw2.

- Prefix 650 is added to gatekeeper localgk, and priority 0 is assigned to gateway gw1.
- A priority 0 is assigned to gateway gw1 to exclude it from the gateway pool for prefix 650. When gw2 registers with gatekeeper localgk, it is added to the gateway pool for each prefix as follows:
  - For gateway pool for 415, gateway gw2 is set to priority 10.
  - For gateway pool for 650, gateway gw2 is set to priority 5.

Configuring a Gatekeeper for Interaction with External Applications

There are two ways of configuring the gatekeeper for interaction with an external application. You can configure a port number where the gatekeeper listens for dynamic registrations from applications. Using this method, the application connects to the gatekeeper and specifies the trigger conditions in which it is interested.

The second method involves using the command-line interface to statically configure the information about the application and its trigger conditions, in which case the gatekeeper initiates a connection to the external application.

To configure a gatekeeper (sj.xyz.com) that uses port 20000 for a specific connection with an external server (Server-123), use the following commands beginning in global configuration mode. Server-123 has a number of triggers that are used to maintain a database of active gateways, which are used for active call resolution.

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config)# server registration-port` *port-number*	Establishes the server registration port that is used for communication between the server and the gatekeeper. The *port-number* argument specifies a single range of values from 1 through 65,535 for the port number on which the gatekeeper listens for external server connections.

Server-123 establishes a connection with gatekeeper sj.xyz.com on port 20000 and sends a REGISTER RRQ message to gatekeeper sj.xyz.com to express interest in all RRQs from voice gateways that support a technology prefix of 1# or 2#.

The following is an example of a registration message:

```
REGISTER RRQ
Version-id:1
From:Server-123
To:sj.xyz.com
Priority:2
Notification-Only:
Content-Length:29

t=voice-gateway
p=1#
p=2#
```

When gatekeeper sj.xyz.com receives this message, the information supplied in the message is added to the trigger list. Then, when an endpoint registers with this gatekeeper by using an RRQ that matches the specified trigger condition in the message, the gatekeeper sends a notification to Server-123.

The following is an example of an RRQ notification sent from the gatekeeper to the server when the above trigger condition matches:

```
REQUEST RRQ
Version-id:1
From:sj.xyz.com
To:Server-123
Notification-Only:
Content-Length:89

c=I:172.18.00.00:1720
r=I:172.20.01.40:16523
a=H:gw3-sj
t=voice-gateway
p=1# 2#
```

Configuring a Prefix to a Gatekeeper Zone List

To add a prefix to a gatekeeper zone list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone prefix` *gatekeeper-name e164-prefix* `[blast | seq] [gw-priority` *priority gw-alias* `[`*gw-alias*`, ...]]`	Adds a prefix to the gatekeeper zone list. For an explanation of the keywords and arguments, see Step 3 of the configuration task table in the "Starting a Gatekeeper" section on page 304.

Note Note that the **zone prefix** command matches a prefix to a gateway. It does not register the gateway. The gateway must register with the gatekeeper before calls can be completed through that gateway.

Verifying an Added Prefix

To view the prefixes added to the gatekeeper zone list, use the **show gatekeeper zone prefix** command. To see gatekeeper zone information, use the **show gatekeeper zone status** command.

Configuring Gatekeeper Triggers for Interaction with External Applications

To establish statically configured triggers on a router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config)# server trigger {arq | lcf | lrj | lrq | rrq | urq} gkid priority server-id server-ip-address server-port`	Configures a static server trigger for external applications. Enter the **all** form of the **no server trigger all** command to remove every static trigger that you configured if you want to delete them all. The keywords and arguments are as follows: • *all*—Deletes all command-line interface- (CLI-) configured triggers. • **arq, lcf, lrj, lrq, rrq, urq**—Specifies Registration, Admission, and Status (RAS) message types. Use these message types to specify a submode in the gatekeeper configuration mode where you configure a trigger for the gatekeeper to act upon. Specify only one message type per server trigger command. There is a different trigger submode for each message type. Each trigger submode has its own set of applicable commands. • *gkid*—Specifies the local gatekeeper identifier. • *priority*—Specifies the priority for each trigger. The range is from 1 through 20, with 1 being the highest priority. • *server-id*—Specifies the identification (ID) number of the external application. • *server-ip-address*—Specifies the IP address of the server. • *server-port*—Specifies the port on which the Cisco IOS gatekeeper listens for messages from the external server connection.
Step 3	`Router(config)# info-only`	Indicates to the Cisco IOS gatekeeper that messages that meet the specified trigger parameters should be sent as notifications only and that the Cisco IOS gatekeeper should not wait for a response from the external application.

	Command	Purpose
Step 4	`Router(config)# destination-info {e164 | email-id | h323-id} value`	Configures a trigger that is based on a particular destination. Repeat this command for more destinations. The keywords and arguments are as follows: • **e164**—Indicates that the destination address is an E.164 address. • **email-id**—Indicates that the destination address is an e-mail ID. • **h323-id**—Indicates that the destination address is an H.323 ID. • *value*—Specifies the value against which to compare the destination address in the RAS messages. For E.164 addresses, the following wildcards can be used: – A trailing series of periods, each of which represents a single character. – A trailing asterisk, which represents one or more characters.
Step 5	`Router(config)# redirect-reason value`	Configures a trigger that is based on a specific redirect reason. Repeat this command for more destinations. The *value* argument specifies the value against which to compare the redirect reason in the RAS messages. Possible values are from 0 to 65,535. Currently used redirect reasons are as follows: • 0—Unknown reason. • 1—Call forwarding is busy or called DTE is busy. • 2—Call forwarded; no reply. • 4—Call deflection. • 9—Called DTE out of order. • 10—Call forwarding by the call DTE 15—Call forwarding unconditionally. • 15—Call forwarding unconditionally.

	Command	Purpose
Step 6	`Router(config)# remote-ext-address [e164] value`	Limits the qualifying messages based on the remote extension address. Repeat this command for more destinations. The keywords and arguments are as follows: • **e164**—(Optional) Indicates that the remote extension address is an E.164 address. • *value*—Specifies the value against which to compare the destination address in the RAS messages. The following wildcards can be used: – A trailing series of periods, each of which represents a single character. – A trailing asterisk, which represents one or more characters.
Step 7	`Router(config)# endpoint-type value`	Configures a trigger that is based on a specific endpoint. Repeat this command for more destinations. The *value* argument specifies the value against which to compare the endpoint type in the RAS messages. The possible values are as follows: • **gatekeeper**—Specifies that the endpoint is an H.323 gatekeeper. • **h320-gateway**—Specifies that the endpoint is an H.320 gateway. • **mcu**—Specifies that the endpoint is a multipoint control unit (MCU). • **other-gateway**—Specifies that the endpoint is a type of gateway not specified on this list. • **proxy**—Specifies that the endpoint is an H.323 proxy. • **terminal**—Specifies that the endpoint is an H.323 terminal. • **voice-gateway**—Specifies that the endpoint is a voice type gateway.
Step 8	`Router(config)# supported-prefix value`	Configures a trigger that is based on a specific supported prefix. Repeat this command for more destinations. The *value* argument specifies the value against which to compare the supported prefix in the RAS messages. The possible values are any E.164 pattern used as a gateway technology prefix. The value string may contain any of the following: 0123456789#*,

Note Repeat Steps 2 through 8 in the above configuration task table for each trigger that you want to define.

Note To remove a trigger, enter the **no server trigger** command. To temporarily suspend a trigger, enter the trigger configuration mode, as described in Step 2, and enter the **shutdown** subcommand.

Configuring Inbound or Outbound Gatekeeper Proxied Access

By default, a gatekeeper will offer the IP address of the local proxy when queried by a remote gatekeeper (synonymous with remote zone). This is considered proxied access. Before Cisco IOS Release 12.0(5)T, the local gatekeeper was configured using the **zone access** command to offer the address of the local endpoint instead of the address of the local proxy (considered direct access).

Note The **use-proxy** command replaces the **zone access** command. The **use-proxy** command, configured on a local gatekeeper, affects only the use of proxies for incoming calls (that is, it does not affect the use of local proxies for outbound calls). When originating a call, a gatekeeper will use a proxy only if the remote gatekeeper offers a proxy at the remote end. A call between two endpoints in the same zone will always be a direct (nonproxied) call.

To configure a proxy for inbound calls from remote zones to gateways in its local zone and to configure a proxy for outbound calls from gateways in its local zone to remote zones, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# use-proxy` *local-zone-name* `{default | remote-zone` *remote-zone-name*`} {inbound-to | outbound-from} {gateway | terminal}`	Enables proxy communications for calls between local and remote zones. The keywords and arguments are as follows: • *local-zone-name*—Specifies the name or zone name of the gatekeeper, which is usually the fully domain-qualified host name of the gatekeeper. For example, if the domain name is cisco.com, the gatekeeper name might be gk1.cisco.com. However, if the gatekeeper is controlling multiple zones, the name of the gatekeeper for each zone should be a unique string that has a mnemonic value. • **default**—Defines the default proxy policy for all calls that are not defined by a **use-proxy** command that includes the **remote-zone** keyword. • **remote-zone** *remote-zone-name*—Defines a proxy policy for calls to or from a specific remote gatekeeper or zone.

Command	Purpose
	• **inbound-to**—Applies the proxy policy to calls that are inbound to the local zone from a remote zone. Each **use-proxy** command defines the policy for only one direction. • **outbound-from**—Applies the proxy policy to calls that are outbound from the local zone to a remote zone. Each **use-proxy** command defines the policy for only one direction. • **gateway**—Defines the type of local device to which the policy applies. The gateway option applies the policy only to local gateways. • **terminal**—Defines the type of local device to which the policy applies. The terminal option applies the policy only to local terminals.

Verifying Gatekeeper Proxied Access Configuration

Use the **show gatekeeper zone status** command to see information about the configured gatekeeper proxies and gatekeeper zone information (as shown in the following output).

Router# **show gatekeeper zone status**

```
                    GATEKEEPER ZONES
                    ================
GK name      Domain Name   RAS Address     PORT  FLAGS MAX-BW   CUR-BW
                                                     (kbps)    (kbps)
-------      -----------   -----------     ----  ----- ------    ------
sj.xyz.com   xyz.com       10.0.0.9 1719  LS             0
  SUBNET ATTRIBUTES :
    All Other Subnets :(Enabled)
  PROXY USAGE CONFIGURATION :
    inbound calls from germany.xyz.com :
      to terminals in local zone sj.xyz.com :use proxy
      to gateways in local zone sj.xyz.com  :do not use proxy
    outbound calls to germany.xyz.com
      from terminals in local zone germany.xyz.com :use proxy
      from gateways in local zone germany.xyz.com  :do not use proxy
    inbound calls from all other zones :
      to terminals in local zone sj.xyz.com :use proxy
      to gateways in local zone sj.xyz.com  :do not use proxy
    outbound calls to all other zones :
      from terminals in local zone sj.xyz.com :do not use proxy
      from gateways in local zone sj.xyz.com  :do not use proxy
tokyo.xyz.co xyz.com       172.21.139.89      1719  RS             0
milan.xyz.co xyz.com       172.16.00.00       1719  RS             0
```

Configuring a Forced Disconnect on a Gatekeeper

To force a disconnect on a gatekeeper, use the following command in privileged EXEC mode:

Command	Purpose
`Router# clear h323 gatekeeper call {all | local-callID local-callID}`	Forces a disconnect on a specific call or on all calls currently active on this gatekeeper. The keywords and arguments are as follows: • **all**—Forces all active calls currently associated with this gatekeeper to be disconnected. • **local-callID**—Forces a single active call associated with this gatekeeper to be disconnected. • *local-callID*—Specifies the local call identification number (CallID) that identifies the call to be disconnected.

To force a particular call to be disconnected (as opposed to all active calls on the H.323 gateway), use the local call identification number (CallID) to identify that specific call. Find the local CallID number for a specific call by using the **show gatekeeper calls** command; the ID number is displayed in the LocalCallID column.

Verifying a Forced Disconnect

To show the status of each ongoing call that a gatekeeper is aware of, use the **show gatekeeper calls** command. If you have forced a disconnect either for a particular call or for all calls associated with a particular H.323 gatekeeper, the system will not display information about those calls.

The following is sample output from the **show gatekeeper calls** command:

```
router# show gatekeeper calls

Total number of active calls =1
                    Gatekeeper Call Info
                    ====================
LocalCallID                     Age (secs)      BW
12-3339                         94              768 (Kbps)
 Endpt(s): Alias    E.164Addr    CallSignalAddr   Port   RASSignalAddr   Port
  src EP: epA                     10.0.0.11        1720   10.0.0.11        1700
  dst EP: epB2zoneB.com
  src PX: pxA                     10.0.0.1         1720   10.0.0.11        24999
  dst PX: pxB                     172.21.139.90    1720   172.21.139.90    24999
```

Configuring the Proxy

This section describes the following configuration tasks for configuring the proxy. Depending on your specific network design, either the first task or the second task is required.

- Configuring a Proxy Without ASR, page 333
- Configuring a Proxy with ASR, page 337

Configuring a Proxy Without ASR

To start the proxy without application-specific routing (ASR), start the proxy and then define the H.323 name, zone, and QoS parameters on the interface whose IP address the proxy will use. To start the proxy without ASR, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# proxy h323`	Starts the proxy feature.

Step 2	`Router(config)# interface type number [name-tag]`
	
	Cisco 4000 Series with Channelized T1 or E1 and the Cisco MC3810
	`Router(config)# interface serial number:channel-group`
	
	To configure a subinterface, use these forms of the interface global configuration command:
	
	Cisco 7200 Series
	`Router(config)# interface type slot/port-adapter/port.subinterface-number [multipoint | point-to-point]`
	
	Cisco 7200 Series and Cisco 7500 Series with a Packet over SONET Interface Processor
	`Router(config)# interface type slot/port`
	
	Cisco 7500 Series
	`Router(config)# interface type slot/port-adapter.subinterface-number [multipoint | point-to-point][ethernet | serial]`
	
	Cisco 7500 Series with Channelized T1 or E1
	`Router(config)# interface serial slot/port:channel-group`
	
	Cisco 7500 Series with Ports on VIP Cards
	`Router(config)# interface type slot/port-adapter/port [ethernet | serial]`
		Configures an interface type and enters interface configuration mode.
		
		The keywords and arguments are as follows:
		
		• *type*—Specifies the type of interface to be configured. (See Table 26 that follows this configuration task table.)
		• *number*—Specifies the port, connector, or interface card number. On a Cisco 4000 series router, specifies the network process monitor (NPM) number. The numbers are assigned at the factory at the time of installation or when added to a system, and they can be displayed with the **show interfaces** command.
		• *name-tag*—(Optional) Specifies the logic name to identify the server configuration so that multiple entries of server configuration can be entered. This optional argument is for use with the Redundant Link Manager (RLM) feature.
		• *slot*—Specifies the number of the slot being configured. Refer to the appropriate hardware manual for slot and port information.
		• *port*—Specifies the number of the port being configured. Refer to the appropriate hardware manual for slot and port information.
		• *port-adapter*—Specifies the number of the port adapter being configured. Refer to the appropriate hardware manual for information about port adapter compatibility.
		• **ethernet**—(Optional) Specifies an Ethernet IEEE 802.3 interface.
		• **serial**—(Optional) Specifies a serial interface.

	Command	Purpose
		• **:***channel-group*—Specifies a T1 channel group number in the range 0 to 23 defined with the **channel-group** controller configuration command. On a dual port card, it is possible to run channelized on one port and primary rate on the other port. Cisco MC3810 specifies the T1/E1 channel group number in the range 0 to 23 defined with the **channel-group** controller configuration command. • .*subinterface-number*—Specifies a subinterface number in the range of 1 to 4,294,967,293. The number that precedes the period (.) must match the number to which this subinterface belongs. • **multipoint** | **point-to-point**—(Optional) Specifies a multipoint or point-to-point subinterface. There is no default.
Step 3	`Router(config-if)# h323 interface [port-number]`	Selects an interface whose IP address will be used by the proxy to register with the gatekeeper. The *port-number* argument specifies the port number on which the proxy will listen for incoming call setup requests: • The *port-number* range is from 1 to 65,356. The default port number for the proxy is 11,720 in -isx- or -jsx- Cisco IOS images. • The default port number for the proxy is 1720 in -ix- Cisco IOS images, which do not contain the Voice over IP (VoIP) gateway.
Step 4	`Router(config-if)# h323 h323-id h323-id`	Configures the proxy name. (More than one name may be configured if necessary.) The *h323-id* argument specifies the name of the proxy. It is recommended that this be a fully qualified e-mail identification (ID), with the domain name being the same as that of its gatekeeper.

	Command	Purpose
Step 5	`Router(config-if)# h323 gatekeeper [id gatekeeper-id] {ipaddr ipaddr [port] | multicast}`	Specifies the gatekeeper associated with a proxy and controls how the gatekeeper is discovered. The keywords and arguments are as follows: • **id** *gatekeeper-id*—(Optional) Specifies the gatekeeper name. Typically, this is a Domain Name System (DNS) name, but it can also be a raw IP address in dotted form. If this parameter is specified, gatekeepers that have either the default or the explicit flags set for the subnet of the proxy will respond. If this parameter is not specified, only those gatekeepers with the default subnet flag will respond. • **ipaddr** *ipaddr* [*port*]—If this parameter is specified, the gatekeeper discovery message will be unicast to this address and, optionally, to the port specified. • **multicast**—If this parameter is specified, the gatekeeper discovery message will be multicast to the well-known Registration, Admission, and Status (RAS) multicast address and port.
Step 6	`Router(config-if)# h323 qos {ip-precedence value | rsvp {controlled-load | guaranteed-qos}}`	Enables quality of service (QoS) on the proxy. The keywords and arguments are as follows: • *ip-precedence value*—Specifies that Realtime Transport Protocol (RTP) streams should set their IP precedence bits to the specified value. • **rsvp** [**controlled-load**]—Specifies controlled load class of service. • **rsvp** [**guaranteed-qos**]—Specifies guaranteed QoS class of service.
Step 7	`Router(config-if)# ip route-cache [cbus] same-interface [flow] distributed`	Controls the use of high-speed switching caches for IP routing. The keywords are as follows: • **cbus**—(Optional) Enables both autonomous switching and fast switching. • **same-interface**—Enables fast-switching packets to back out through the interface on which they arrived.

Command	Purpose
	• **flow**—(Optional) Enables the Route Switch Processor (RSP) to perform flow switching on the interface. • **distributed**—Enables Versatile Interface Processor (VIP) distributed switching on the interface. This feature can be enabled on Cisco 7500 series routers with RSP and VIP controllers. If both the **ip route-cache flow** command and the **ip route-cache distributed** command are configured, the VIP does distributed flow switching. If only the **ip route-cache distributed** command is configured, the VIP does distributed switching.

Table 26 lists interface types that may be used for the *type* argument in Step 2 of the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.

Table 26 Interface Type Keywords

Keyword	Interface Type
async	Port line used as an asynchronous interface.
atm	ATM interface.
bri	ISDN BRI. This interface configuration is propagated to each of the B channels. B channels cannot be individually configured. The interface must be configured with dial-on-demand commands for calls to be placed on that interface.
dialer	Dialer interface.
ethernet	Ethernet IEEE 802.3 interface.
fastethernet	100-Mbps Ethernet interface on the Cisco 4500, Cisco 4700, Cisco 7000, and Cisco 7500 series routers.
fddi	FDDI interface.
group-async	Master asynchronous interface.
hssi	High-Speed Serial Interface (HSSI).
lex	LAN Extender (LEX) interface.
loopback	Software-only loopback interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The interface-number is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create.
null	Null interface.
port-channel	Port channel interface.
pos	Packet OC-3 interface on the Packet over SONET Interface Processor.
serial	Serial interface.
switch	Switch interface.
tokenring	Token Ring interface.

Table 26 Interface Type Keywords (continued)

Keyword	Interface Type
tunnel	Tunnel interface; a virtual interface. The number is the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
vg-anylan	100VG-AnyLAN port adapter.

Configuring a Proxy with ASR

To enable ASR on the proxy, start the proxy and then define the H.323 name, zone, and QoS parameters on the loopback interface. Next, determine which interface will be used to route the H.323 traffic and configure ASR on it. The ASR interface and all other interfaces must be separated so that routing information never travels from one to the other. There are two different ways to separate the ASR interface and all other interfaces:

- Use one type of routing protocol on the ASR interface and another on all the non-ASR interfaces. Include the loopback subnet in both routing domains.
- Set up two different autonomous systems, one that contains the ASR network and the loopback network and another that contains the other non-ASR networks and loopback network.

To ensure that the ASR interface and all other interfaces never route packets between each other, configure an access control list. (The proxy traffic will be routed specially because it is always addressed to the loopback interface first and then translated by the proxy subsystem.)

To start the proxy with ASR enabled on the proxy using one type of routing protocol on the ASR interface and another on all of the non-ASR interfaces, and with the loopback subnet included in both routing domains, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# proxy h323`	Starts the proxy.
Step 2	`Router(config)# interface` *type number* [*name-tag*]	Enters loopback interface configuration mode. For an explanation of the arguments, see Step 2 in the "Configuring a Proxy Without ASR" configuration task table. To configure a proxy with ASR enabled on the proxy using one type of routing protocol, the *type* argument is **loopback**. The **loopback** type specifies the software-only loopback interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The *number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.

	Command	Purpose
Step 3	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for an interface. The keywords and arguments are as follows: • *ip-address*—Specifies the IP address. • *mask*—Specifies the mask for the associated IP subnet. • **secondary**—(Optional) Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 4	`Router(config-if)# h323 interface [port-number]`	Signals the proxy that this interface IP address is the one to use. For an explanation of the arguments, see Step 3 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.
Step 5	`Router(config-if)# h323 h323-id h323-id`	Configures the proxy name. (More than one name can be configured if necessary.) The *h323-id* argument specifies the name of the proxy. It is recommended that this be a fully qualified e-mail identification (ID), with the domain name being the same as that of its gatekeeper.
Step 6	`Router(config-if)# h323 gatekeeper [id gatekeeper-id] {ipaddr ipaddr [port] | multicast}`	Specifies the gatekeeper associated with a proxy and controls how the gatekeeper is discovered. For an explanation of the keywords and arguments, see Step 5 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.
Step 7	`Router(config-if)# h323 qos {ip-precedence value | rsvp {controlled-load | guaranteed-qos}}`	Enables quality of service (QoS) on the proxy. For an explanation of the keywords and arguments, see Step 6 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.
Step 8	`Router(config-if)# interface type number [name-tag]`	If ASR is to be used, enters the interface through which outbound H.323 traffic should be routed. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.

	Command	Purpose
Step 9	`Router(config-if)# h323 asr [bandwidth max-bandwidth]`	Enables ASR and specifies the maximum bandwidth for a proxy. The keywords and arguments are as follows: • **bandwidth** *max-bandwidth*—Specifies the maximum bandwidth on the interface. Value ranges are from 1 to 10,000,000 kbps. If you do not specify a value for the *max-bandwidth* argument, the value defaults to the bandwidth on the interface. If you specify the *max-bandwidth* value as a value greater than the interface bandwidth, the bandwidth will default to the interface bandwidth.
Step 10	`Router(config-if)# ip address ip-address mask [secondary]`	Sets up the ASR interface network number. For an explanation of the keywords and arguments, see Step 3 in this configuration task table.
Step 11	`Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	`Router(config)# interface type number [name-tag]`	Enters interface configuration mode for a non-ASR interface. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.
Step 13	`Router(config-if)# ip address ip-address mask [secondary]`	Sets up a non-ASR interface network number. For an explanation of the keywords and arguments, see Step 3 in this configuration task table.
Step 14	`Router(config-if)# exit`	Exits interface configuration mode.
Step 15	`Router(config)# router rip`	Configures the Routing Information Protocol (RIP) for a non-ASR interface.
Step 16	`Router(config)# network network-number`	Specifies a list of networks for the RIP routing process or a loopback interface in an Interior Gateway Routing Protocol (IGRP) domain. The *network-number* argument specifies the IP address of the directly connected networks.
Step 17	`Router(config)# router igrp autonomous-system`	Configures Interior IGRP for an ASR interface. The *autonomous-system* argument specifies the autonomous system number that identifies the routes to the other IGRP routers. It is also used to tag the routing information.
Step 18	`Router(config)# network network-number`	Specifies a list of networks for the Routing Information Protocol (RIP) routing process. The *network-number* argument should include an ASR interface in an IGRP domain.
Step 19	`Router(config)# network loopback-addr`	Includes a loopback interface in an IGRP domain.

	Command	Purpose
Step 20	`Router(config)# access-list access-list-number {permit | deny} source source-mask [destination destination-mask] {eq | neq} [[source-object] [destination-object] [identification] any]`	Creates an access list. The keywords and arguments are as follows: • *access-list-number*—Specifies the integer that you choose. The number should be between 300 and 399, and it uniquely identifies the access list. • **permit**—Permits access when there is an address match. • **deny**—Denies access when there is an address match. • *source*—Specifies the source address. DECnet addresses are written in the form area.node. For example, 50.4 is node 4 in area 50. All addresses are in decimal. • *source-mask*—Specifies the mask to be applied to the address of the source node. All masks are in decimal. • *destination*—(Optional) Specifies the DECnet address of the destination node in decimal format. DECnet addresses are written in the form area.node. For example, 50.4 is node 4 in area 50. All addresses are in decimal. • *destination-mask*—(Optional) Specifies the destination mask. DECnet addresses are written in the form area.node. For example, 50.4 is node 4 in area 50. All masks are in decimal. • **eq**—Specifies that the item matches the packet if all the specified parts of the source object, destination object, and identification match the data in the packet. • **neq**—Specifies that the item matches the packet if any of the specified parts do not match the corresponding entry in the packet. • *source-object*—(Optional) Contains the mandatory keyword **src** and one of the following optional keywords: – **eq | neq | lt | gt**—Specifies equal to, not equal to, less than, or greater than. These keywords must be followed by the argument *object-number*, a numeric DECnet object number. – **exp**—Stands for expression; followed by a regular-expression that matches a string. See the “Regular Expressions” appendix in the *Cisco IOS Dial Technologies Command Reference* for a description of regular expressions.

Command	Purpose
	• *destination-object*—(Optional) Contains the mandatory keyword **dst** and one of the following optional keywords: – **eq | neq | lt | gt**—Specifies equal to, not equal to, less than, or greater than. These keywords must be followed by the argument *object-number*, a numeric DECnet object number. – **exp**—Stands for expression; followed by a regular expression that matches a string. See the "Regular Expressions" appendix in the *Cisco IOS Dial Technologies Command Reference* for a description of regular expressions. – **uic**—Stands for user identification code; followed by a numeric UID expression. The argument [*group*, *user*] is a numeric UID expression. In this case, the bracket symbols are literal; they must be entered. The *group* and *user* parts can be specified either in decimal, in octal by prefixing the number with a 0, or in hex by prefixing the number with 0x. The **uic** expression displays as an octal number. • *identification*—(Optional) Uses any of the following three keywords: – **id**—Specifies regular expression; refers to the user ID. – **password**—Specifies regular expression; the password to the account. – **account**—Specifies regular expression; the account string. – **any**—(Optional) Specifies that the item matches if *any* of the specified parts *do* match the corresponding entries for *source-object*, *destination-object*, or *identification*.

	Command	Purpose
Step 21	`Router(config)# interface type number [name-tag]`	Enters interface configuration mode on an ASR interface. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.
Step 22	`Router(config-if)# ip access-group {access-list-number | access-list-name}{in | out}`	Controls access to an interface. Use this command to set the outbound access group and then the inbound access group. The keywords and arguments are as follows: • *access-list-number*—Specifies the number of an access list. This is a decimal number from 1 to 199 or from 1300 to 2699. • *access-list-name*—Name of an IP access list as specified by an IP **access-list** command. • **in**—Filters on inbound packets. • **out**—Filters on outbound packets.

Note ASR is not supported on Frame Relay or ATM interfaces for the Cisco MC3810 platform.

To start the proxy with ASR enabled on the proxy using two different autonomous systems (one that contains the ASR network and the loopback network and another that contains the other non-ASR networks and the loopback network), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# proxy h323`	Starts the proxy.
Step 2	`Router(config)# interface type number [name-tag]`	Enters loopback interface configuration mode. For an explanation of the arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333. To start the proxy with ASR enabled on the proxy using two different autonomous systems, the *type* argument is **loopback**. The **loopback** type specifies the software-only loopback interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The *number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create.

	Command	Purpose
Step 3	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for an interface. The keywords and arguments are as follows: • *ip-address*—Specifies the IP address. • *mask*—Specifies the mask for the associated IP subnet. • **secondary**—(Optional) Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 4	`Router(config-if)# h323 interface [port-number]`	Signals the proxy that this interface IP address is the one to use. For an explanation of the arguments, see Step 3 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.
Step 5	`Router(config-if)# h323 h323-id h323-id`	Configures the proxy name. (More than one name can be configured if necessary.) The *h323-id* argument specifies the name of the proxy. It is recommended that this be a fully qualified e-mail identification (ID), with the domain name being the same as that of its gatekeeper.
Step 6	`Router(config-if)# h323 gatekeeper [id gatekeeper-id] {ipaddr ipaddr [port] | multicast}`	Specifies the gatekeeper associated with a proxy and controls how the gatekeeper is discovered. For an explanation of the keywords and arguments, see Step 5 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.
Step 7	`Router(config-if)# h323 qos {ip-precedence value | rsvp {controlled-load | guaranteed-qos}}`	Enables quality of service (QoS) on the proxy. The keywords and arguments are as follows: • **ip-precedence** *value*—Specifies that Real-time Transport Protocol (RTP) streams should set their IP precedence bits to the specified value. • **rsvp** {**controlled-load**}—Specifies controlled load class of service. • **rsvp** {**guaranteed-qos**}—Specifies guaranteed QoS class of service.
Step 8	`Router(config-if)# interface type number [name-tag]`	If application-specific routing (ASR) is to be used, enters the interface through which outbound H.323 traffic should be routed. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.

	Command	Purpose
Step 9	`Router(config-if)# h323 asr [bandwidth max-bandwidth]`	Enables ASR and specifies the maximum bandwidth for a proxy. The optional *max-bandwidth* argument specifies the maximum bandwidth on the interface. Value ranges are from 1 to 10,000,000 kbps. If you do not specify *max-bandwidth*, this value defaults to the bandwidth on the interface. If you specify *max-bandwidth* as a value greater than the interface bandwidth, the bandwidth will default to the interface bandwidth.
Step 10	`Router(config-if)# ip address ip-address mask [secondary]`	Sets up the ASR interface network number. For an explanation of the keywords and arguments, see Step 3 in this configuration task table.
Step 11	`Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	`Router(config)# interface type number [name-tag]`	Enters interface configuration mode on a non-ASR interface. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.
Step 13	`Router(config-if)# ip address ip-address mask [secondary]`	Sets up a non-ASR interface network number. For an explanation of the keywords and arguments, see Step 3 in this configuration task table.
Step 14	`Router(config-if)# exit`	Exits interface configuration mode.
Step 15	`Router(config)# router igrp autonomous-system`	Configures Interior Gateway Routing Protocol (IGRP) for a non-ASR interface. The *autonomous-system* argument specifies the autonomous system number that identifies the routes to the other IGRP routers. It is also used to tag the routing information.
Step 16	`Router(config)# network network-number`	Includes a non-ASR interface in an IGRP domain. The *network-number* argument specifies the IP address of the network of the directly connected networks.
Step 17	`Router(config)# network network-number`	Includes a loopback interface in an IGRP domain. The *network-number* argument specifies the IP address of the network of the directly connected networks.
Step 18	`Router(config)# router igrp autonomous-system`	Configures IGRP for an ASR interface. The *autonomous-system* argument specifies the autonomous system number that identifies the routes to the other IGRP routers. It is also used to tag the routing information.

	Command	Purpose
Step 19	`Router(config)# network network-number`	Specifies a list of networks for the Routing Information Protocol (RIP) routing process. The *network-number* argument should include an ASR interface in an IGRP domain.
Step 20	`Router(config)# network network-number`	Specifies a list of networks for the RIP routing process. The *network-number* argument should include a loopback interface in an IGRP domain.
Step 21	`Router(config)# access-list access-list-number {permit | deny} source source-mask [destination destination-mask] {eq | neq} [[source-object] [destination-object] [identification] any]`	Creates an access list. For an explanation of the keywords and arguments, see Step 20 in the configuration task table in the "Configuring a Proxy with ASR" section on page 337.
Step 22	`Router(config)# interface type number [name-tag]`	Enters interface configuration mode on an ASR interface. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 333.
Step 23	`Router(config-if)# ip access-group {access-list-number | access-list-name} {in | out}`	Controls access to an interface. Use this command to set the outbound access group and then the inbound access group. The keywords and arguments are as follows: • *access-list-number*—Specifies the number of an access list. This is a decimal number from 1 to 199 or from 1300 to 2699. • *access-list-name*—Name of an IP access list as specified by an IP **access-list** command. • **in**—Filters on inbound packets. • **out**—Filters on outbound packets.

H.323 Gatekeeper Configuration Examples

This section includes the following configuration examples:

Configuring a Gatekeeper Example

The following is an annotated example of how to configure a gatekeeper:

```
hostname gk-eng.xyz.com
! This router serves as the gatekeeper for the engineering community.
! at xyz.com.
ip domain-name xyz.com
! Domain name of this company.
interface Ethernet0
 ip address 172.21.127.27 255.255.255.0
! This gatekeeper can be found at address 172.21.127.27.
gatekeeper
! Enter gatekeeper config mode.
 zone local gk-eng.xyz.com xyz.com
! Because a zone is, by definition, the area of control of a gatekeeper,
! we tend to use the terms "zone name" and "gatekeeper name" synonymously.
! Here we use the host name as the name of the gatekeeper and zone.
! This is not necessary, but it does simplify administration.
 zone remote gk-mfg.xyz.com xyz.com 172.12.10.14 1719
 zone remote gk-corp.xyz.com xyz.com 172.12.32.80 1719
! A couple of other zones within xyz.com. We make lots of calls
! between these departments, so we just configure these so we save
! a little time bypassing DNS lookup to find their gatekeepers.
 use-proxy gk-eng.xyz.com remote-zone gk-mfg.xyz.com direct
 use-proxy gk-eng.xyz.com remote-zone gk-corp.xyz.com direct
 use-proxy gk-eng.xyz.com default proxied
! We have good QoS on our local network, so we don't need proxies when
! calling between the xyz.com zones. But for all other zones, we want
! to use proxies.
zone subnet gk-eng.xyz.com 172.21.127.0/24 enable
no zone subnet gk-eng.xyz.com default enable
! We will accept registrations from our local subnet as long as they
! do not specify some other gatekeeper name. We will not accept any
! registrations from any other subnet.
 zone bw gk-eng.xyz.com 2000
```

```
! Preserve our good QoS by not allowing excessive amounts of H.323 traffic
! on the local network. This restricts the traffic within our zone,
! for both intra-zone and interzone calls, to 2 kbps at any given time.
 alias static 172.21.127.49 gkid gk-eng.xyz.com terminal h323id joeblow ras
172.21.127.49 1719
! The "user" has an H.323 terminal, which does not support RAS. So we have
! to configure his alias manually so that callers can find him.
```

Redundant Gatekeepers for a Zone Prefix Example

In the following example, two remote gatekeepers are configured to service the same zone prefix:

```
gatekeeper
 zone remote c2600-1-gk cisco.com 172.18.194.70 1719
 zone remote c2514-1-gk cisco.com 172.18.194.71 1719
 zone prefix c2600-1-gk 919.......
 zone prefix c2514-1-gk 919.......
```

Redundant Gatekeepers for a Technology Prefix Example

In the following example, two remote gatekeepers are configured to service the same technology prefix:

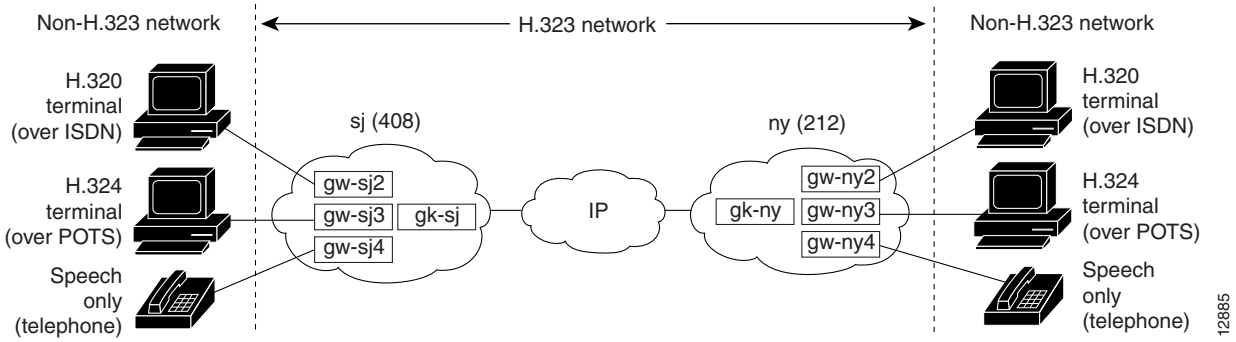
```
gatekeeper
 zone remote c2600-1-gk cisco.com 172.18.194.70 1719
 zone remote c2514-1-gk cisco.com 172.18.194.71 1719
 gw-type-prefix 3#* hopoff c2600-1-gk hopoff c2514-1-gk
```

E.164 Interzone Routing Example

Interzone routing may be configured by using E.164 addresses.

In this example, there are two gatekeepers that need to be able to resolve E.164 addresses. One is in San Jose and the other is in New York. (See Figure 61.)

Figure 61 E.164 Interzone Routing



In sj (San Jose in the 408 area code), the gateways are configured to register with gk-sj as follows:

- gw-sj2 configured to register with technology prefix 2#
- gw-sj3 configured to register with technology prefix 3#
- gw-sj4 configured to register with technology prefix 4#

Similarly, in ny (New York in the 212 area code), gateways are configured to register with gk-ny as follows:

- gw-ny2 configured to register with technology prefix 2#
- gw-ny3 configured to register with technology prefix 3#
- gw-ny4 configured to register with technology prefix 4#

For the gatekeeper for San Jose, the configuration commands are as follows:

```
gatekeeper
 zone local gk-sj cisco.com
 zone remote gk-ny cisco.com 172.21.127.27
 use-proxy gk-sj default direct
 zone prefix gk-sj 408.......
 zone prefix gk-ny 212.......
 gw-type-prefix 3# hopoff gk-sj
 gw-type-prefix 4# default-technology
```

For the gatekeeper for New York, the configuration commands are as follows:

```
gatekeeper
 zone local gk-ny cisco.com
 zone remote gk-sj cisco.com 172.21.1.48
 use-proxy gk-ny default direct
 zone prefix gk-sj 408.......
 zone prefix gk-ny 212.......
 gw-type-prefix 3# hopoff gk-ny
 gw-type-prefix 4# default-technology
```

When a call is presented to gatekeeper gk-sj with the following target address in San Jose:

```
2#2125551212
```

Gatekeeper gk-sj recognizes that 2# is a technology prefix. It was not configured as such, but because gw-sj2 registered with it, the gatekeeper now treats 2# as a technology prefix. It strips the prefix, which leaves the telephone number 2125551212. This is matched against the zone prefixes that have been configured. It is a match for 212....., so gk-sj knows that gk-ny handles this call. Gatekeeper gk-sj forwards the entire address 2#2125551212 over to Gatekeeper gk-ny, which also looks at the technology prefix 2# and routes it to gw-ny2.

When a call is presented to gatekeeper gk-sj with the following target address in San Jose:

```
2125551212
```

Gatekeeper gk-sj checks it against known technology prefixes but finds no match. It then checks it against zone prefixes and matches on 212..... for gk-ny, and therefore routes this call to gk-ny. Gatekeeper gk-ny does not have any local registrations for this address, and there is no technology prefix on the address, but the default prefix is 4#, and gw-ny4 is registered with 4#, so the call gets routed to gw-ny4.

Another call is presented to gatekeeper gk-sj with the following target address in San Jose:

```
3#2125551212
```

The call has technology prefix 3#, which is defined as a local hopoff prefix, so gk-sj routes this call to gw-sj3, despite the fact that it has a New York zone prefix.

In this last example, a call is presented to gatekeeper gk-sj with the following target address in San Jose:

```
6505551212
```

Gatekeeper gk-sj checks for a technology prefix match but does not find one. It then searches for a zone prefix match and fails again. But there is a match for default gateway prefix of 4#, and gw-sj4 is registered with 4#, so the call is routed out on gw-sj4.

Configuring HSRP on the Gatekeeper Example

This sample configuration uses Ethernet 0 as the HSRP interface on both gatekeepers.

On the primary gatekeeper, enter these commands:

```
configure terminal
 ! Enter global configuration mode.
interface ethernet 0
 ! enter interface configuration mode for interface ethernet 0.
 standby 1 ip 172.21.127.55
 ! Member of standby group 1, sharing virtual address 172.21.127.55.
 standby 1 preempt
 ! Claim active role when it has higher priority.
 standby 1 timers 5 15
 ! Hello timer is 5 seconds; hold timer is 15 seconds.
 standby 1 priority 110
 ! Priority is 110.
```

On the backup gatekeeper, enter these commands:

```
configure terminal
interface ethernet 0
 standby 1 ip 172.21.127.55
 standby 1 preempt
 standby 1 timers 5 15
```

The configurations are identical except that gk2 has no standby priority configuration, so it assumes the default priority of 100—meaning that gk1 has a higher priority.

On both gk1 and gk2, set up identical gatekeeper mode configurations, as follows:

```
configure terminal
 ! Enter global configuration mode.
gatekeeper
 ! Enter gatekeeper configuration mode.
 zone local gk-sj cisco.com 172.21.127.55
 ! Define local zone using HSRP virtual address as gatekeeper RAS address.
.
.
.
 ! Various other gk-mode configurations.
 no shut
 ! Bring up the gatekeeper.

configure terminal
 ! Enter global configuration mode.
gatekeeper
 ! Enter gatekeeper configuration mode.
 zone local gk-sj cisco.com 172.21.127.55
 ! Define local zone using HSRP virtual address as gatekeeper RAS address.
 ! Note this uses the same gkname and address as on gk1.
.
.
! Various other gk-mode configurations.
 no shut
 ! Bring up the gatekeeper.
```

Note The **no shut** command is issued on both gatekeepers, primary and secondary. If the **show gatekeeper status** command is issued on the two gatekeepers, gk1 will show the following:

```
Gatekeeper State: UP
But gk2 will show the following:
Gatekeeper State: HSRP STANDBY
```

Using ASR for a Separate Multimedia Backbone Example

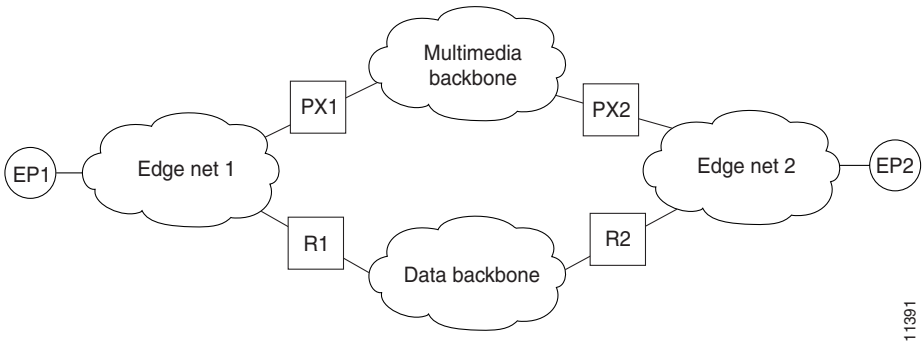
The examples in this section illustrate a separate multimedia backbone network dedicated to transporting only H.323 traffic. The closed functionality of the H.323 proxy is necessary for creating this type of backbone. Place a closed H.323 proxy on each edge of the multimedia backbone to achieve the following goals:

- The proxy directs all inter-proxy H.323 traffic, including Q.931 signaling, H.245, and media stream, to the multimedia backbone.
- The proxy shields the multimedia backbone so that routers on edge networks and other backbone networks are not aware of its existence. In this way, only H.323-compliant packets can access or traverse the multimedia backbone.
- The proxy drops any unintended non-H.323 packets that attempt to access the multimedia backbone.

This section contains the following subsections:

Figure 62 illustrates a network that has a multimedia backbone. A gatekeeper (not shown) in the edge network (zone) directs all out-of-zone H.323 calls to the closed proxy on the edge of that network. The closed proxy forwards this traffic to the remote zone through the multimedia backbone. A closed proxy and the edge router may reside in the same Cisco router, or they may be in separate routers, as shown in Figure 62.

Figure 62 Sample Network with Multimedia Backbone



Enabling the Proxy to Forward H.323 Packets

To enable the proxy to forward H.323 packets received from the edge network to the multimedia backbone, designate the interface that connects the proxy to the multimedia backbone to the ASR interface by entering the **h323 asr** command in interface configuration mode. Enabling the proxy to forward H.323 packets satisfies the first goal identified earlier in this section.

Because the proxy terminates two call legs of an H.323 call and bridges them, any H.323 packet that traverses the proxy will have the proxy address either in its source field or in its destination field.

To prevent problems that can occur in proxies that have multiple IP addresses, designate only one interface to be the proxy interface by entering the **h323 interface** command in interface configuration mode. Then all H.323 packets that originate from the proxy will have the address of this interface in their source fields, and all packets that are destined to the proxy will have the address of this interface in their destination fields.

Figure 62 illustrates that all physical proxy interfaces belong either to the multimedia network or to the edge network. These two networks must be isolated from each other for the proxy to be closed; however, the proxy interface must be addressable from both the edge network and the multimedia network. For this reason, a loopback interface must be created on the proxy and configured to the proxy interface.

It is possible to make the loopback interface addressable from both the edge network and the multimedia network without exposing any physical subnets on one network to routers on the other network. Only packets that originate from the proxy or packets that are destined to the proxy can pass through the proxy interface to the multimedia backbone in either direction. All other packets are considered unintended packets and are dropped. This can be achieved by configuring access control lists (ACLs) so that the closed proxy acts like a firewall that only allows H.323 packets to pass through the ASR interface. This satisfies the second goal identified earlier in this section, which is to ensure that only H.323-compliant packets can access or traverse the multimedia backbone.

Isolating the Multimedia Network

The last step is to configure the network so that non-H.323 traffic never attempts to traverse the multimedia backbone and so that it never risks being dropped by the proxy. This is achieved by completely isolating the multimedia network from all edge networks and from the data backbone and by configuring routing protocols on the various components of the networks.

The example provided in Figure 62 requires availability of six IP address classes, one for each of the four autonomous systems and one for each of the two loopback interfaces. Any Cisco-supported routing protocol can be used on any of the autonomous systems, with one exception: Routing Information Protocol (RIP) cannot be configured on two adjacent autonomous systems because this protocol does not include the concept of an autonomous system. The result would be the merging of the two autonomous systems into one.

If the number of IP addresses are scarce, use subnetting, but the configuration can get complicated. In this case, only the Enhanced IGRP, Open Shortest Path First (OSPF), and RIP Version 2 routing protocols, which allow variable-length subnet masks (VLSMs), can be used.

Assuming these requirements are met, configure the network illustrated in Figure 62 as follows:

- Configure each of the four networks as a separate routing autonomous system and do not redistribute routes between the multimedia backbone and any other autonomous system.
- Create a loopback interface on the proxy and configure it to be the proxy interface. That way no subnets of the multimedia backbone will be exposed to the edge network, or the other way around.
- To ensure that the address of the loopback interface does not travel outside the edge network, configure the appropriate distribution list on the edge router that connects the edge network to the data backbone. Configuring the appropriate distribution list guarantees that any ongoing H.323 call will be interrupted if the multimedia backbone fails. Otherwise, H.323 packets that originate from one proxy and that are destined to another proxy might discover an alternate route using the edge networks and the data backbone.

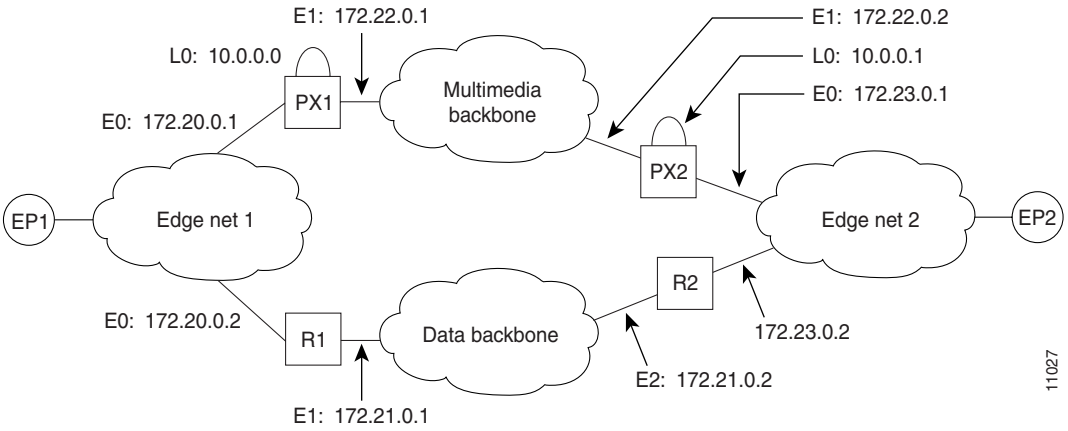
In some topologies, the two edge networks and the data backbone may be configured as a single autonomous system, but it is preferable to separate them as previously described because they are different networks with different characteristics.

The following examples illustrate the router configuration that is relevant to the closed proxy operation.

Configuring a Co-Edge Proxy with ASR Without Subnetting Example

See Figure 63 and the following configuration examples to see how to configure RIP on the two edge networks and how to configure IGRP on the two backbone networks.

Figure 63 Sample Configuration Without Subnetting



PX1 Configuration

The following output is for the PX1 configuration:

```
!
proxy h323
!
interface Loopback0

 ip address 10.0.0.0 255.0.0.0
!Assume PX1 is in Zone 1, and the gatekeeper resides in the same routers as PX1:
 h323 interface
 h323 h323-id PX1@zone1.com
 h323 gatekeeper ipaddr 10.0.0.0
!
```

```
interface Ethernet0
 ip address 172.20.0.1 255.255.0.0
!
interface Ethernet1
 ip address 172.22.0.1 255.255.0.0
 ip access-group 101 in
 ip access-group 101 out
 h323 asr
!
router rip
 network 172.20.0.0
 network 10.0.0.0
!
router igrp 4000
 network 172.22.0.0
 network 10.0.0.0
!
access-list 101 permit ip any host 10.0.0.0
access-list 101 permit ip host 10.0.0.0 any
access-list 101 permit igrp any any
```

R1 Configuration

The following output is for the R1 configuration:

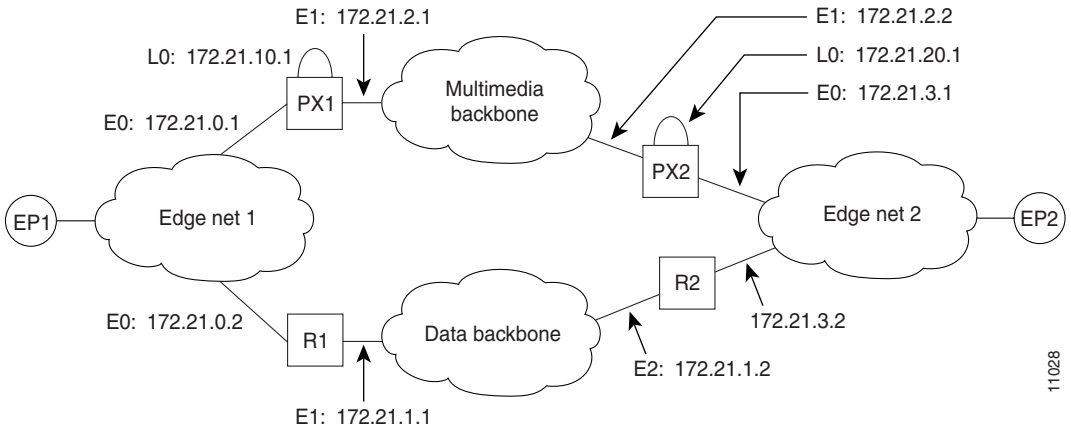
```
!
interface Ethernet0
 ip address 172.20.0.2 255.255.0.0
!
interface Ethernet1
 ip address 172.21.0.1 255.255.0.0
!
router rip
 redistribute igrp 5000 metric 1
 network 172.20.0.0
!
router igrp 5000
 redistribute rip metric 10000 10 255 255 65535
 network 172.21.0.0
 distribute-list 10 out
!
access-list 10 deny ip 10.0.0.0 255.255.255
access-list 10 permit any
```

Note The configuration for PX2 and R2 is the same as that for PX1 and R1.

Co-Edge Proxy with Subnetting Example

Figure 64 and the examples that follow illustrate how to configure Enhanced IGRP on all networks.

Figure 64 Sample Configuration with Subnetting



PX1 Configuration

The following output is for the PX1 configuration:

```
!
proxy h323
!
interface Loopback0
 ip address 172.21.10.1 255.255.255.192
 h323 interface
 h323 h323-id PX1@zone1.com
 h323 gatekeeper ipaddr 172.21.20.1
!
interface Ethernet0
 ip address 172.21.0.1 255.255.255.192
!
interface Ethernet1
 ip address 172.21.2.1 255.255.255.192
 ip access-group 101 in
 ip access-group 101 out
 h323 asr
!
router eigrp 4000
 redistribute connected metric 10000 10 255 255 65535
 passive-interface Ethernet1
 network 172.21.0.0
 distribute-list 10 out
 no auto-summary
!
router eigrp 5000
 redistribute connected metric 10000 10 255 255 65535
 passive-interface Ethernet0
 network 172.21.0.0
 distribute-list 11 out
 no auto-summary
!
access-list 10 deny 172.21.2.0 0.0.0.63
access-list 10 permit any
```

```
access-list 11 deny 172.21.0.0 0.0.0.63
access-list 11 permit any
access-list 101 permit ip any host 172.21.10.1
access-list 101 permit ip host 172.21.10.1 any
access-list 101 permit eigrp any any
```

R1 Configuration

The following output is for the R1 configuration:

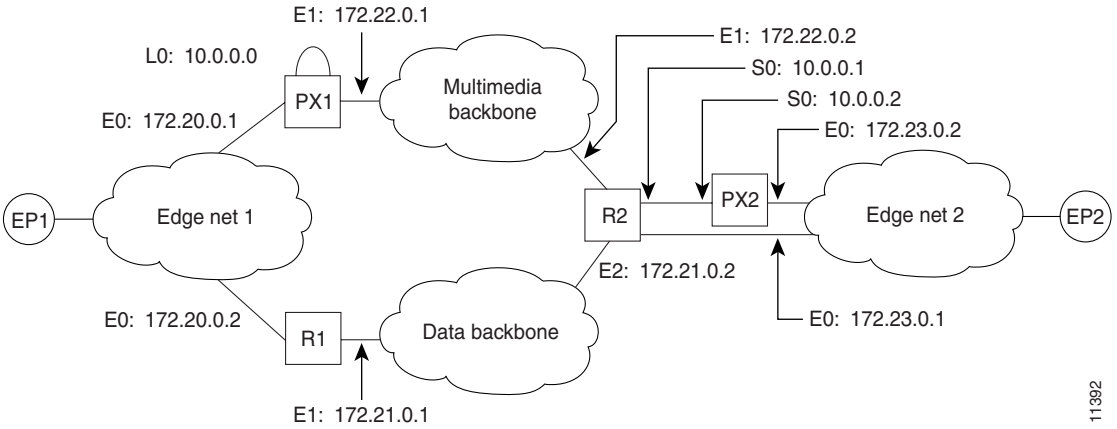
```
!
interface Ethernet0
 ip address 172.21.0.2 255.255.255.192
!
interface Ethernet1
 ip address 172.21.1.1 255.255.255.192
!
router eigrp 4000
 redistribute eigrp 6000 metric 10000 10 255 255 65535
 passive-interface Ethernet1
 network 172.21.0.0
 no auto-summary
!
router eigrp 6000
 redistribute eigrp 4000 metric 10000 10 255 255 65535
 passive-interface Ethernet0
 network 172.21.0.0
 distribute-list 10 out
 no auto-summary
!
access-list 10 deny 172.21.10.0 0.0.0.63
access-list 10 permit any
```

Note The configuration for PX2 and R2 is the same as that for PX1 and R1.

Configuring an Inside-Edge Proxy with ASR Without Subnetting Example

The configuration of the co-edge proxy in Edge net 1 has already been presented above. Figure 65 illustrates the configuration of the inside-edge proxy PX2 and edge router R2 of Edge net 2. RIP is used on the edge networks. IGRP is used on the data backbone and the multimedia backbone.

Figure 65 Edge Net 2 with Inside-Edge Proxy and No Subnetting



PX2 Configuration

The following output is for the PX2 configuration:

```
!
proxy h323
!
interface Ethernet0
 ip address 172.23.0.2 255.255.0.0
!
interface Serial0
 ip address 10.0.0.2 255.0.0.0
 ip access-group 101 in
 ip access-group 101 out
 h323 interface
 h323 asr
 h323 h323-id PX2@zone2.com
 h323 gatekeeper ipaddr 10.0.0.2
!
router rip
 redistribute connected metric 10000 10 255 255 65535
 network 172.23.0.0
!
access-list 101 permit ip any host 10.0.0.2
access-list 101 permit ip host 10.0.0.2 any
```

R2 Configuration

The following output is for the R2 configuration:

```
!
interface Ethernet0
 ip address 172.23.0.1 255.255.0.0
!
interface Ethernet1
 ip address 172.22.0.1 255.255.0.0
```

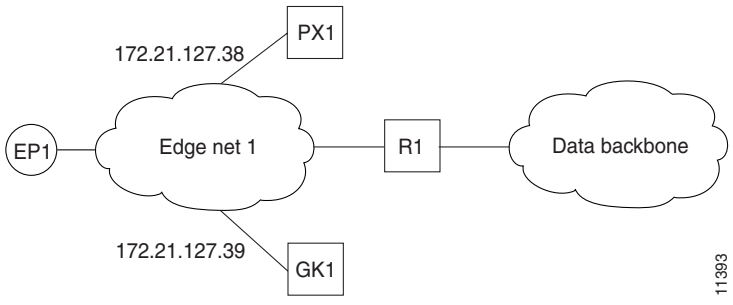
```
 ip access-group 101 in
 ip access-group 101 out
!
interface Ethernet2
 ip address 172.21.0.2 255.255.0.0
!
interface Serial0
 ip address 10.0.0.1 255.0.0.0
!
router rip
 redistribute igrp 5000 metric 1
 network 172.23.0.0
!
router igrp 4000
 network 10.0.0.0
 network 172.22.0.0
!
router igrp 5000
 redistribute rip metric 10000 10 255 255 65535
 network 172.21.0.0
 distribute-list 10 out
!
ip route 10.0.0.2 255.255.255.255 Serial0
access-list 10 deny ip 10.0.0.0 255.255.255
access-list 10 permit any
access-list 101 permit ip any host 10.0.0.2
access-list 101 permit ip host 10.0.0.2 any
```

Note To guarantee that all traffic between the proxy and other proxies is carried over the multimedia backbone, run IGRP 4000 on the 10.0.0.0 network and on the 172.22.0.0 network. Make sure that the H.323 proxy interface address (10.0.0.2) is not advertised over the data network (distribution list 10 in IGRP 5000). Doing this also eliminates the need to configure policy routes or static routes.

Configuring a QoS-Enforced Open Proxy Using RSVP Example

Figure 66 illustrates a proxy configuration that was created on a Cisco 2500 router with one Ethernet interface and two serial interfaces. Only the Ethernet interface is in use.

Figure 66 Configuring a QoS-Enforced Open Proxy Using RSVP



PX1 Configuration

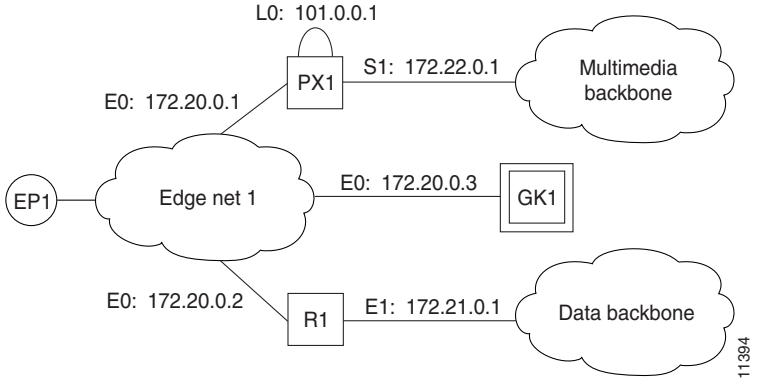
The following output is for the PX1 configuration:

```
!
version 11.3
no service password-encryption
service tcp-small-servers
!
hostname ExampleProxy
!
no ip domain-lookup
!
proxy h323
!
interface Ethernet0
 ip address 172.21.127.38 255.255.255.192
 no ip redirects
 ip rsvp bandwidth 7000 7000
 ip route-cache same-interface
 fair-queue 64 256 1000
 h323 interface
 h323 qos rsvp controlled-load
 h323 h323-id px1@zone1.com
 h323 gatekeeper ipaddr 172.21.127.39
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
router rip
 network 172.21.0.0
!
ip classless
!
line con 0
 exec-timeout 0 0
line aux 0
 transport input all
line vty 0 4
 password lab
 login
!
end
```

Configuring a Closed Co-Edge Proxy with ASR Without Subnetting Example

Figure 67 illustrates how to configure RIP on the edge networks and IGRP on the two backbone networks. A Cisco 2500 router is used for the proxy.

Figure 67 Configuring a Closed Co-Edge Proxy with ASR



PX1 Configuration

The following output is for the PX1 configuration:

```
!
version 11.3
no service password-encryption
service tcp-small-servers
!
hostname ExampleProxy
!
!
no ip domain-lookup
!
!
proxy h323
!
interface Loopback0
 ip address 10.0.0.1 255.0.0.0
 h323 interface
 h323 qos ip-precedence 4
 h323 h323-id px1@zone1.com
 h323 gatekeeper ipaddr 172.20.0.3
!
interface Ethernet0
 ip address 172.20.0.1 255.255.255.192
 no ip redirects
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 ip address 172.22.0.1 255.255.0.0
 ip access-group 101 in
 ip access-group 101 out
 h323 asr
!
router rip
```

```
 network 172.20.0.0
 network 10.0.0.0
!
router igrp 4000
 network 172.22.0.0
 network 101.0.0.0
!
ip classless
access-list 101 permit ip any host 10.0.0.1
access-list 101 permit ip host 10.0.0.1 any
access-list 101 permit igrp any any
!
!
line con 0
 exec-timeout 0 0
line aux 0
 transport input all
line vty 0 4
 password lab
 login
```

Defining Multiple Zones Example

The following example shows how to define multiple local zones for separating gateways:

```
zone local gk408or650 xyz.com
zone local gk415 xyz.com
zone prefix gk408or650 408.......
zone prefix gk408or650 650.......
zone prefix gk415 415.......
```

All the gateways used for area codes 408 or 650 can be configured so that they register with gk408or650, and all gateways used for area code 415 can be configured so that they register with gk415.

Defining One Zone for Multiple Gateways Example

The following example shows how to put all the gateways in the same zone and use the **gw-priority** keyword to determine which gateways will be used for calling different area codes:

```
zone local localgk xyz.com
zone prefix localgk 408.......
zone prefix localgk 415....... gw-priority 10 gw1 gw2
zone prefix localgk 650....... gw-priority 0 gw1
```

The commands shown accomplish the following tasks:

- Domain xyz.com is assigned to gatekeeper localgk.
- Prefix 408..... is assigned to gatekeeper localgk, and no gateway priorities are defined for it; therefore, all gateways that register to localgk can be used equally for calls to the 408 area code. No special gateway lists are built for the 408..... prefix; selection is made from the master list for the zone.
- The prefix 415..... is added to gatekeeper localgk, and priority 10 is assigned to gateways gw1 and gw2.
- Prefix 650..... is added to gatekeeper localgk, and priority 0 is assigned to gateway gw1.

A priority 0 is assigned to gateway gw1 to exclude it from the gateway pool for prefix 650..... When gateway gw2 registers with gatekeeper localgk, it is added to the gateway pool for each prefix as follows:

- For gateway pool for 415....., gateway gw2 is set to priority 10.
- For gateway pool for 650....., gateway gw2 is set to priority 5.

To change gateway gw2 from priority 10 for zone 415..... to the default priority 5, enter the following command:

```
no zone prefix localgk 415..... gw-pri 10 gw2
```

To change both gateways gw1 and gw2 from priority 10 for zone 415..... to the default priority 5, enter the following command:

```
no zone prefix localgk 415..... gw-pri 10 gw1 gw2
```

In the preceding example, the prefix 415..... remains assigned to gatekeeper localgk. All gateways that do not specify a priority level for this prefix are assigned a default priority of 5. To remove the prefix and all associated gateways and priorities from this gatekeeper, enter the following command:

```
no zone prefix localgk 415.....
```

Configuring a Proxy for Inbound Calls Example

In the following example, the local zone sj.xyz.com is configured to use a proxy for inbound calls from remote zones tokyo.xyz.com and milan.xyz.com to gateways in its local zone. The sj.xyz.com zone is also configured to use a proxy for outbound calls from gateways in its local zone to remote zones tokyo.xyz.com and milan.xyz.com.

```
gatekeeper
use-proxy sj.xyz.com remote-zone tokyo.xyz.com inbound-to gateway
use-proxy sj.xyz.com remote-zone tokyo.xyz.com outbound-from gateway
use-proxy sj.xyz.com remote-zone milan.xyz.com inbound-to gateway
use-proxy sj.xyz.com remote-zone milan.xyz.com outbound-from gateway
```

Because the default mode disables proxy communications for all gateway calls, only the gateway call scenarios listed can use the proxy.

Configuring a Proxy for Outbound Calls Example

In the following example, the local zone sj.xyz.com uses a proxy for only those calls that are outbound from H.323 terminals in its local zone to the specified remote zone germany.xyz.com:

```
gatekeeper
no use-proxy sj.xyz.com default outbound-from terminal
use-proxy sj.xyz.com remote-zone germany.xyz.com outbound-from
 terminal
```

Note that any calls inbound to H.323 terminals in the local zone sj.xyz.com from the remote zone germany.xyz.com use the proxy because the default applies.

Removing a Proxy Example

The following example shows how to remove one or more proxy statements for the remote zone germany.xyz.com from the proxy configuration list:

```
no use-proxy sj.xyz.com remote-zone germany.xyz.com
```

The command removes all special proxy configurations for the remote zone germany.xyz.com. After the command is entered like this, all calls between the local zone (sj.xyz.com) and germany.xyz.com are processed according to the defaults defined by any **use-proxy** commands that use the **default** option.

H.235 Security Example

The following example shows output from configuring secure registrations from the gatekeeper and identifying which RAS messages the gatekeeper will check to find authentication tokens:

```
dial-peer voice 10 voip
 destination-pattern 4088000
 session target ras
 dtmf-relay h245-alphanumeric
!
gateway
 security password 09404F0B level endpoint
```

The following example shows output from configuring which RAS messages will contain gateway generated tokens:

```
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
radius-server host 10.0.0.1 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server deadtime 5
radius-server key lab
radius-server vsa send accounting
!
gatekeeper
 zone local GK1 test.com 10.0.0.3
 zone remote GK2 test2.com 10.0.2.2 1719
 accounting
 security token required-for registration
 no use-proxy GK1 remote-zone GK2 inbound-to terminal
 no use-proxy GK1 remote-zone GK2 inbound-to gateway
 no shutdown
```

GKTMP and RAS Messages Example

The following is an example of a gatekeeper that has interaction with external applications. The registration message from Server-123 establishes a connection with gatekeeper sj.xyz.com on port 20000. Server-123 sends a REGISTER RRQ message to gatekeeper sj.xyz.com to express interest in all RRQs from voice gateways that support a technology prefix of 1# or 2#.

```
REGISTER RRQ
Version-id:1
From:Server-123
To:sj.xyz.com
Priority:2
Notification-Only:
Content-Length:29

t=voice-gateway
p=1#
p=2#
```

Prohibiting Proxy Use for Inbound Calls Example

To prohibit proxy use for inbound calls to H.323 terminals in a local zone from a specified remote zone, enter a command similar to the following:

```
no use-proxy sj.xyz.com remote-zone germany.xyz.com inbound-to terminal
```

This command overrides the default and disables proxy use for inbound calls from remote zone germany.xyz.com to all H.323 terminals in the local zone sj.xyz.com.

Disconnecting a Single Call Associated with an H.323 Gateway Example

The following example forces an active call on the H.323 gateway to be disconnected. The local ID number of the active call is 12-3339.

```
Router> enable
Router# clear h323 gatekeeper call local-callID 12-3339
```

Disconnecting All Calls Associated with an H.323 Gateway Example

The following example forces all active calls on the H.323 gateway to be disconnected:

```
Router> enable
Router# clear h323 gatekeeper call all
```