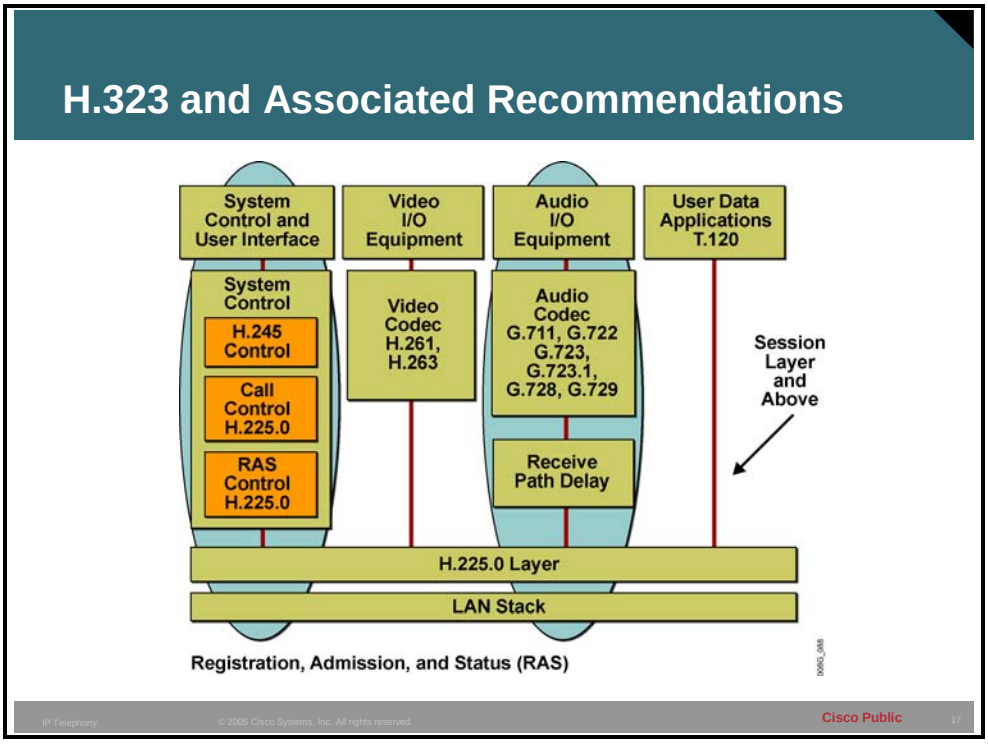
Configuring H.323

H.323 and Associated Recommendations

This topic describes H.323 and its protocols and explains how H.323 is used in the IP internetwork environment.



Recommendation H.323 describes an infrastructure of terminals, common control components, services, and protocols that are used for multimedia (voice, video, and data) communications. The figure illustrates the elements of an H.323 terminal and highlights the protocol infrastructure of an H.323 endpoint.

H.323 was originally created to provide a mechanism for transporting multimedia applications over LANs. Although numerous vendors still use H.323 for videoconferencing applications, it has rapidly evolved to address the growing needs of VoIP networks. H.323 is currently the most widely used VoIP signaling and call control protocol, with international and domestic carriers relying on it to handle billions of minutes of use each year.

H.323 is considered an “umbrella protocol” because it defines all aspects of call transmission, from call establishment to capabilities exchange to network resource availability. H.323 defines the following protocols:

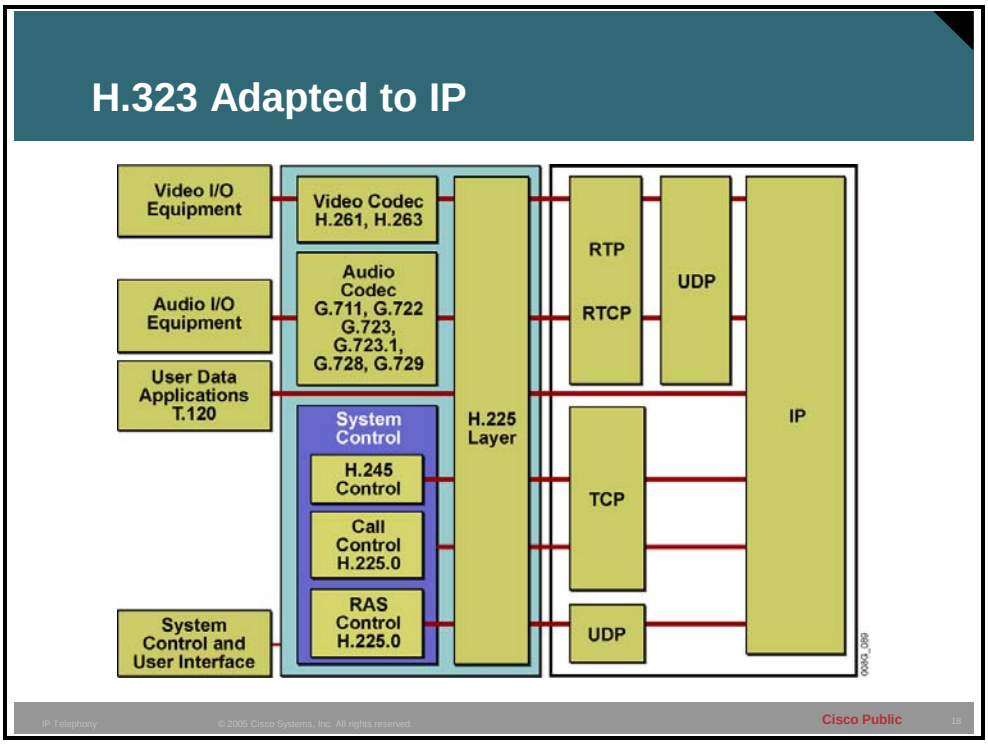
- H.245 for capabilities exchange
- H.225.0 for call setup
- H.225.0 for registration, admission, and status (RAS) control for call routing

H.323 is based on the ISDN Q.931 protocol, which allows H.323 to easily interoperate with legacy voice networks, such as the public switched telephone network (PSTN) or SS7. In addition to providing support for call setup, H.225.0 provides a message transport mechanism for the H.245 control function and the RAS signaling function. Following is a description of these functions:

- **Call-signaling function:** The call-signaling function uses a call-signaling channel that allows an endpoint to create connections with other endpoints. The call-signaling function defines call setup procedures, based on the call setup procedures for ISDN (Recommendation Q.931). The call-signaling function uses messages formatted according to H.225.0.
- **H.245 control function:** The H.245 control function uses a control channel to transport control messages between endpoints or between an endpoint and a common control component, such as a gatekeeper or multipoint controller (MC). The control channel used by the H.245 control function is separate from the call-signaling channel.

  The H.245 control function is responsible for the following:

  - **Logical channel signaling:** Opens and closes the channel that carries the media stream
  - **Capabilities exchange:** Negotiates audio, video, and codec capability between the endpoints
  - **Master or responder determination:** Determines which endpoint is master and which is responder; used to resolve conflicts during the call
  - **Mode request:** Requests a change in mode, or capability, of the media stream
  - **Timer and counter values:** Establishes values for timers and counters and agreement of those values by the endpoints
- **RAS signaling function:** The RAS signaling function uses a separate signaling channel (RAS channel) to perform registration, admissions, bandwidth changes, status, and disengage procedures between endpoints and a gatekeeper. The RAS signaling function uses messages formatted according to H.225.0.

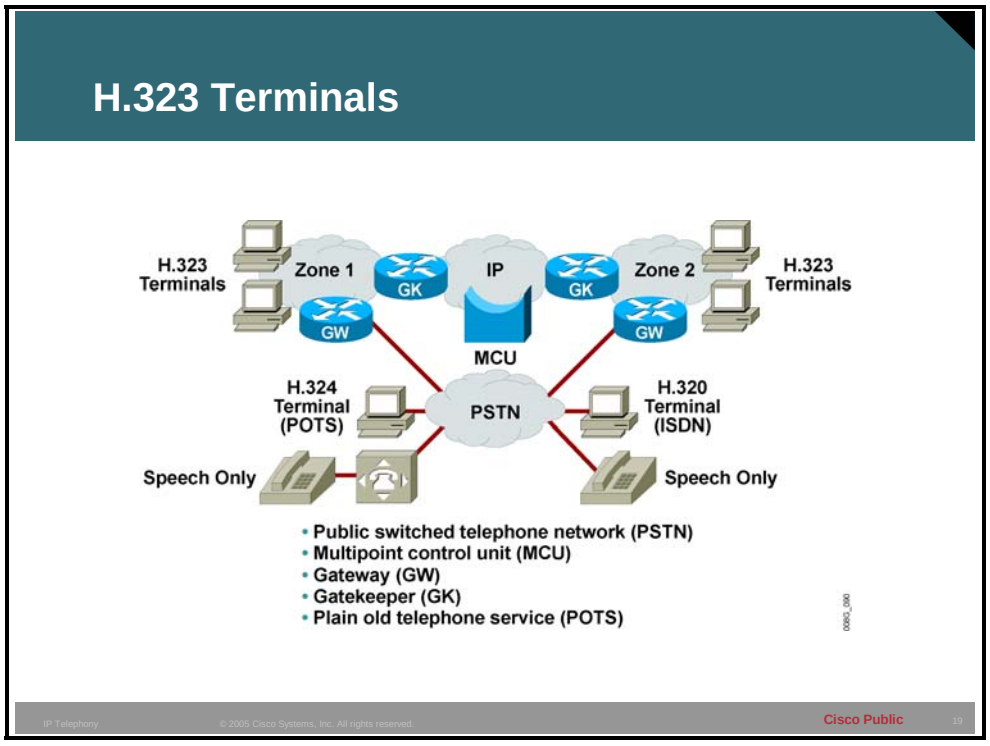


Example: H.323 Adapted to IP

A typical implementation of H.323 goes beyond the original LAN context of H.323. The figure illustrates a specific application of H.323 on an IP internetwork. Notice that real-time aspects of H.323 rely on UDP. Both the session-oriented control procedures and the data media type of H.323 use TCP.

Functional Components of H.323

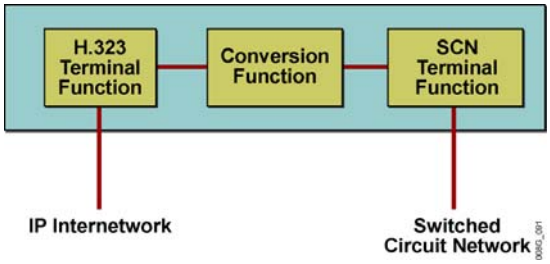
This topic describes the functional components that make up an H.323 environment.



An H.323 terminal is an endpoint that provides real-time voice (and optionally, video and data) communications with another endpoint, such as an H.323 terminal, gateway, or multipoint control unit (MCU).

An H.323 terminal must be capable of transmitting and receiving G.711 (a-law and µ-law) 64-kbps pulse code modulation (PCM)-encoded voice, and may support other encoded voice formats, such as G.729 and G.723.1.

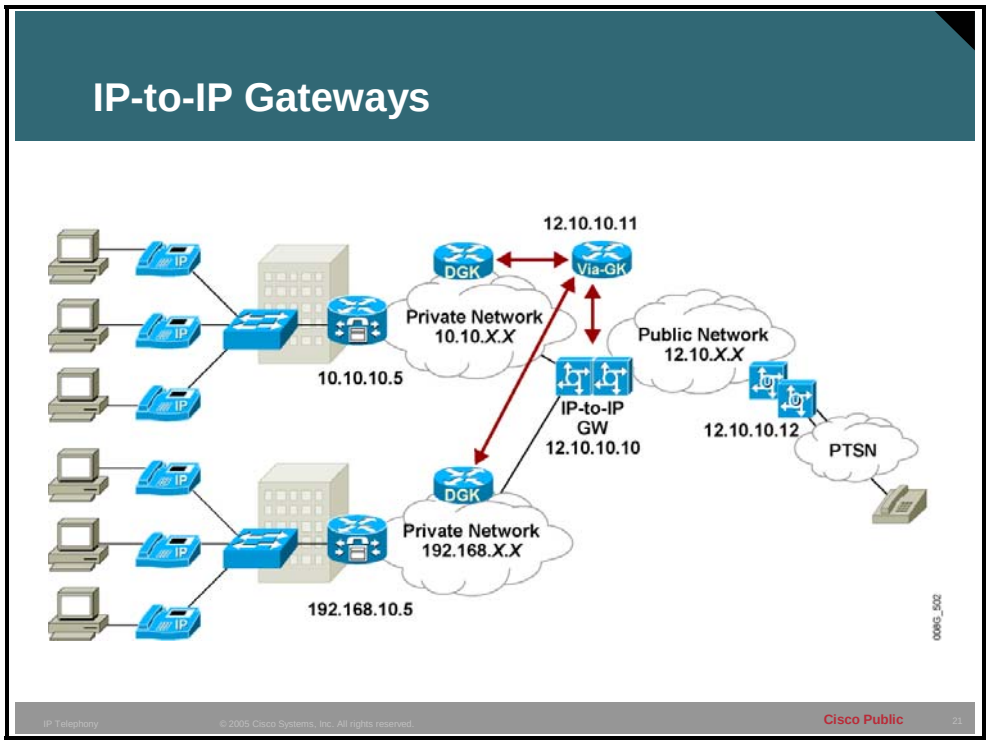
H.323 Gateways



An H.323 gateway is an optional type of endpoint that provides interoperability between H.323 endpoints and endpoints located on a switched-circuit network (SCN), such as the PSTN or an enterprise voice network. Ideally, the gateway is transparent to both the H.323 endpoint and the SCN-based endpoint.

An H.323 gateway performs the following services:

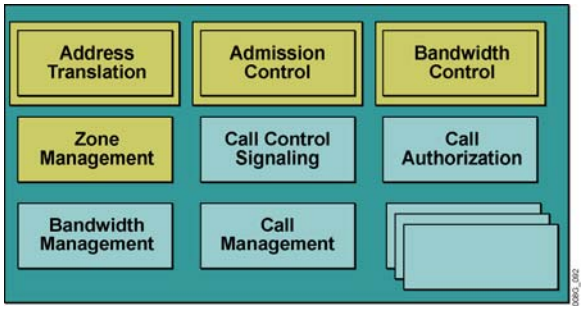
- Translation between audio, video, and data formats
- Conversion between call setup signals and procedures
- Conversion between communication control signals and procedures



The IP-to-IP gateway facilitates easy and cost-effective connectivity between independent VoIP service provider networks. Some in the industry call IP-to-IP gateways “border elements” or “session border controllers.” The IP-to-IP gateway provides a network-to-network interface point for billing, security, Cisco CallManager interconnectivity, call admission control, and signaling interworking. It will perform most of the same functions of a PSTN-to-IP gateway, but will join two VoIP call legs. Media packets can either flow through the gateway and hide the networks from each other, or flow around the IP-to-IP gateway if network security is not of primary importance.

The figure illustrates a basic IP-to-IP gateway network. From the perspective of the private, or customer, networks, the IP-to-IP gateway will appear as a single public address that must be routable on their private networks (in this case a 12.*x.x.x* address routable on the 10.10.*x.x* and 192.168.*x.x* networks). Care must be taken at the IP-to-IP gateway to ensure that proper routing restrictions are in place to prevent communication directly between the private networks attached to it. Also note that this model works only if no overlapping address schemes are used on the customers’ networks. Finally, to the hop-off gateways on the public network, all calls will appear to originate from the 12.*x.x.x* address of the IP-to-IP gateway and not the private addresses on the customer networks. Also note that the gatekeepers shown in the diagram control each zone independently, with the 12.10.10.11 gatekeeper acting as the control point for the public network, and therefore the IP-to-IP gateway.

H.323 Gatekeepers



An H.323 gatekeeper is an optional component that provides call control support and services to H.323 endpoints. Although a gatekeeper is considered a distinct and optional component, it can be colocated with any other H.323 component.

The scope of endpoints over which a gatekeeper exercises its authority is called a “zone.” H.323 defines a one-to-one relationship between a zone and a gatekeeper.

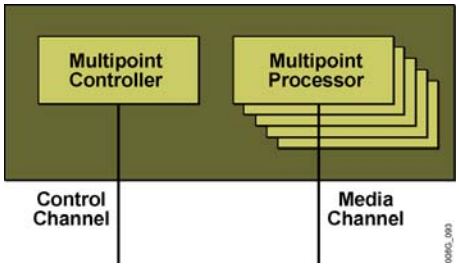
When a gatekeeper is included, it must perform the following functions:

- **Address translation:** Converts an alias address to an IP address
- **Admission control:** Limits access to network resources based on call bandwidth restrictions
- **Bandwidth control:** Responds to bandwidth requests and modifications
- **Zone management:** Provides services to registered endpoints

The gatekeeper may also perform:

- **Call control signaling:** Performs call signaling on behalf of the endpoint (gatekeeper-routed call signaling)
- **Call authorization:** Rejects calls based on authorization failure
- **Bandwidth management:** Limits the number of concurrent accesses to IP internetwork resources (Call Admission Control [CAC])
- **Call management:** Maintains a record of ongoing calls

Multipoint Conference Components



Support for multipoint conferences is provided by the following three functional components:

- **Multipoint controller:** An MC provides the functions that are necessary to support conferences involving three or more endpoints. The MC establishes an H.245 control channel with each of the conference participants. Through the control channel, the MC completes a capability exchange during which the MC indicates the mode of the conference (decentralized or centralized).

  An MC is not modeled as a standalone component; it may be located with an endpoint (terminal or gateway), a gatekeeper, or a multipoint control unit.

- **Multipoint processor:** A multipoint processor (MP) adds functionality to multipoint conferences. An MP can receive multiple streams of multimedia input, process the streams by switching and mixing the streams, and then retransmit the result to all or some of the conference members.

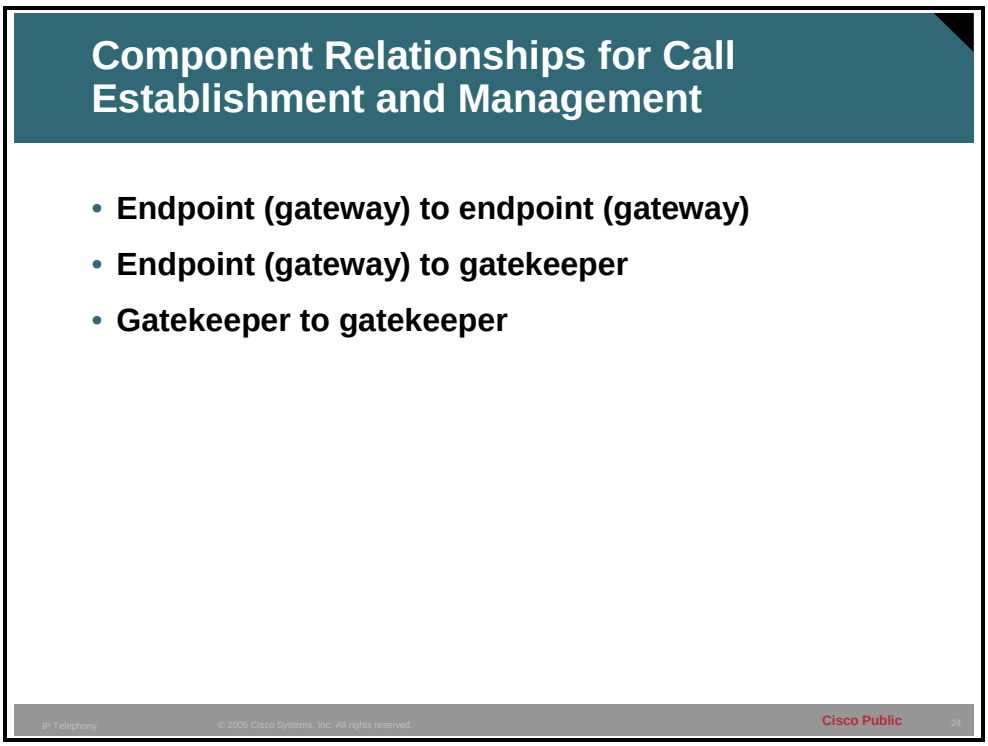
  Similar to an MC, an MP is not modeled as a standalone component; it resides in an MCU.

- **Multipoint control unit:** An MCU is modeled as an endpoint that provides support for multipoint conferences by incorporating one MC and zero or more MPs.

  An MCU is modeled as a standalone component.

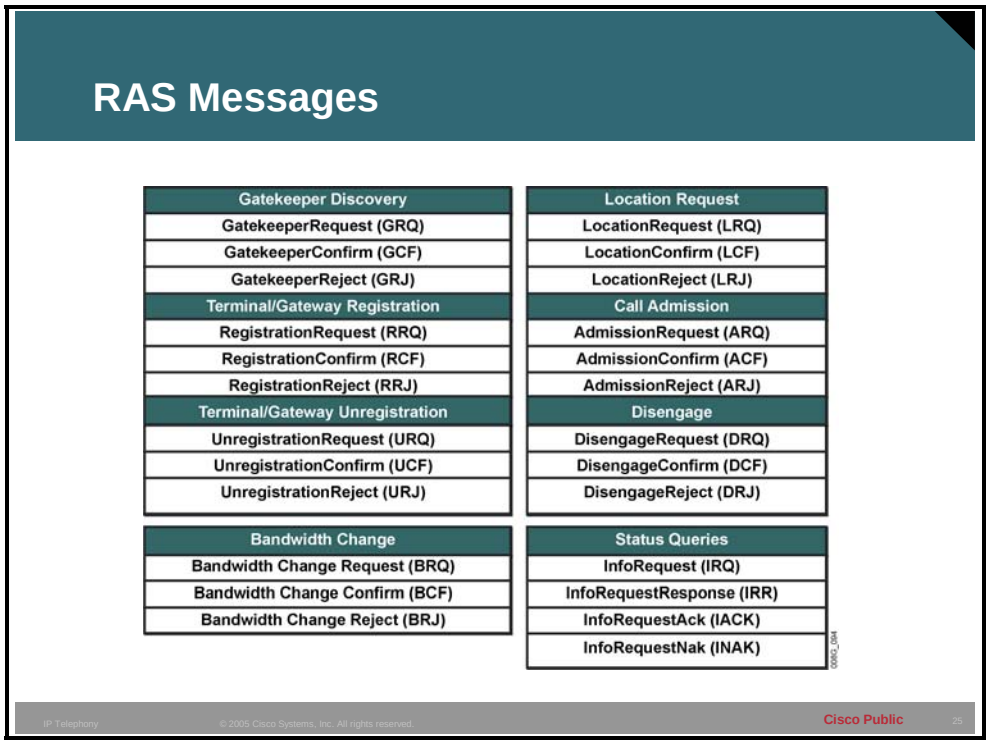
H.323 Call Establishment and Maintenance

This topic describes possible component scenarios required to establish end-to-end connections and the commands used by the components to establish VoIP calls.



Although H.323 is based on the concepts of a distributed call control model, it often embodies centralized call control model concepts. Calls can be established between any of the following components:

- **Endpoint to endpoint:** The intelligence of H.323 endpoints allows them to operate autonomously. In this mode of operation, endpoints locate other endpoints through nonstandard mechanisms and initiate direct communication between the endpoints.
- **Endpoint to gatekeeper:** When a gatekeeper is added to the network, endpoints interoperate with the gatekeeper using the RAS channel.
- **Gatekeeper to gatekeeper:** In the presence of multiple gatekeepers, gatekeepers communicate with each other on the RAS channel.



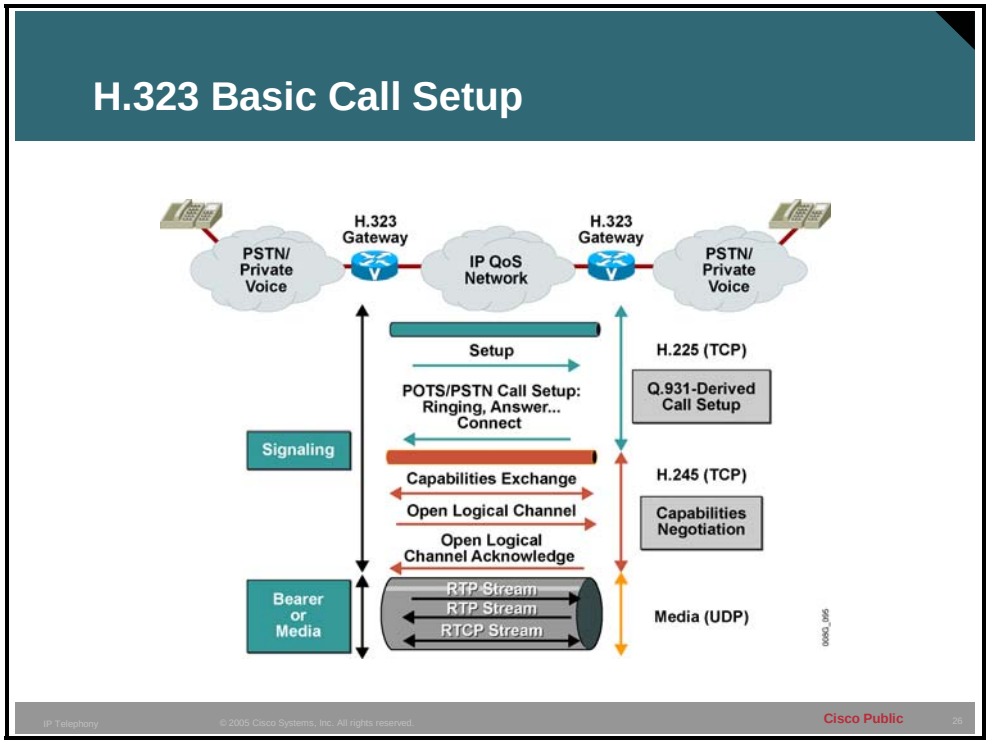
Gatekeepers communicate through the RAS channel using different types of RAS messages. These message types include the following:

- **Gatekeeper discovery:** An endpoint multicasts a gatekeeper discovery request (GRQ). A gatekeeper may confirm (gatekeeper confirmation [GCF]) or reject (gatekeeper rejection [GRJ]) an endpoint.
- **Terminal/gateway registration:** An endpoint sends a registration request (RRQ) to its gatekeeper to register and provide reachable prefixes. A gatekeeper confirms (registration confirmation [RCF]) or rejects (registration rejection [RRJ]) the registration.
- **Terminal/gateway unregistration:** An endpoint or gatekeeper sends an unregistration request (URQ) to cancel a registration. The responding device confirms (unregistration request [UCF]) or rejects (unregistration rejection [URJ]) the request.
- **Location request:** An endpoint or gatekeeper sends a location request (LRQ) to a gatekeeper. An LRQ is sent directly to a gatekeeper if one is known, or it is multicast to the gatekeeper discovery multicast address. An LRQ requests address translation of an E.164 address and solicits information about the responsible endpoint. The responding gatekeeper confirms (location confirmation [LCF]) with the IP address of the endpoint or rejects the request (location rejection [LRJ]) if the address is unknown.
- **Call admission:** An endpoint sends an admission request (ARQ) to its gatekeeper. The request identifies the terminating endpoint and the bandwidth required. The gatekeeper confirms (admission confirmation [ACF]) with the IP address of the terminating endpoint or rejects (admission rejection [ARJ]) if the endpoint is unknown or inadequate bandwidth is available.
- **Bandwidth change:** An endpoint sends a bandwidth change request (BRQ) to its gatekeeper to request an adjustment in call bandwidth. A gatekeeper confirms (bandwidth confirmation [BCF]) or rejects (bandwidth rejection [BRJ]) the request.

- **Disengage:** When a call is disconnected, the endpoint sends a disengage request (DRQ) to the gatekeeper. The gatekeeper confirms (disengage confirmation [DCF]) or rejects (disengage rejection [DRJ]) the request.
- **Status queries:** A gatekeeper uses status request (IRQ) to determine the status of an endpoint. In its response (IRR), the endpoint indicates whether it is online or offline. The endpoint may also reply that it understands the information request (information request acknowledged [IACK]) or that it does not understand the request (information request not acknowledged [INAK]).

Call Flows Without a Gatekeeper

This topic describes call setup scenarios without a gatekeeper and provides examples of actual call-flow procedures.



The figure shows an H.323 basic call setup exchange between two gateways. The optional gatekeeper is not present in this example. Although gateways are shown, the same procedure is used when one or both endpoints are H.323 terminals.

The flow procedure without a gatekeeper includes these steps:

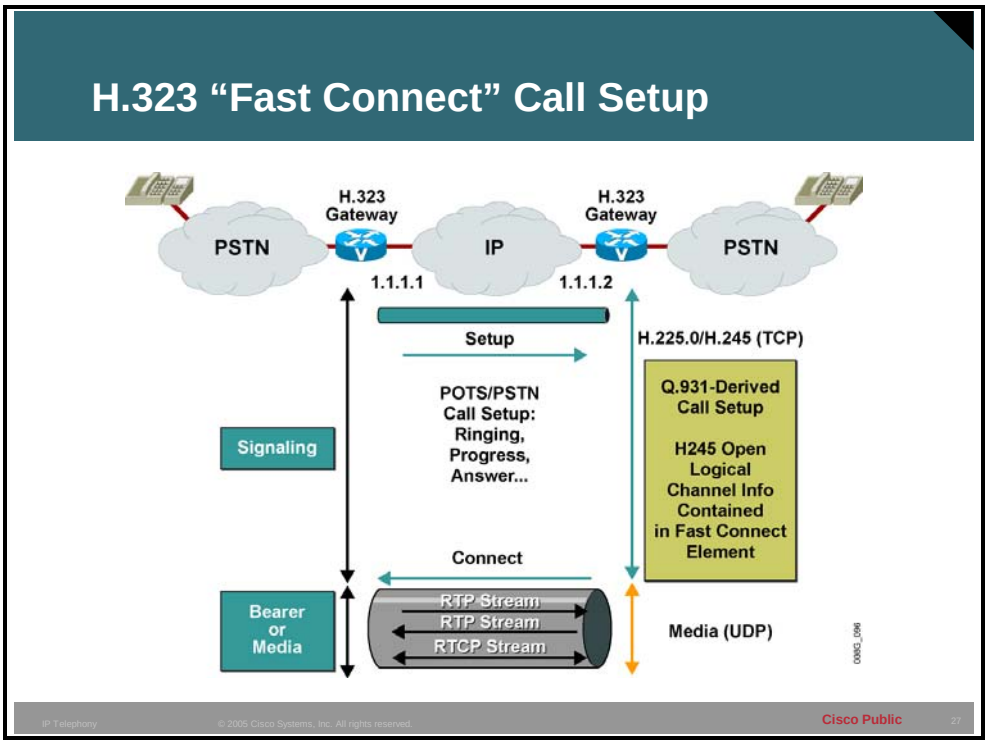
1. The originating gateway initiates an H.225.0 session with the destination gateway on registered TCP port 1720. The gateway determines the IP address of the destination gateway internally. The gateway has the IP address of the destination endpoint in its configuration or it knows a Domain Name System (DNS) resolvable domain name for the destination.

2. Call setup procedures based on Q.931 create a call-signaling channel between the endpoints.

3. The endpoints open another channel for the H.245 control function. The H.245 control function negotiates capabilities and exchanges logical channel descriptions.

4. The logical channel descriptions open RTP sessions.

5. The endpoints exchange multimedia over the RTP sessions.

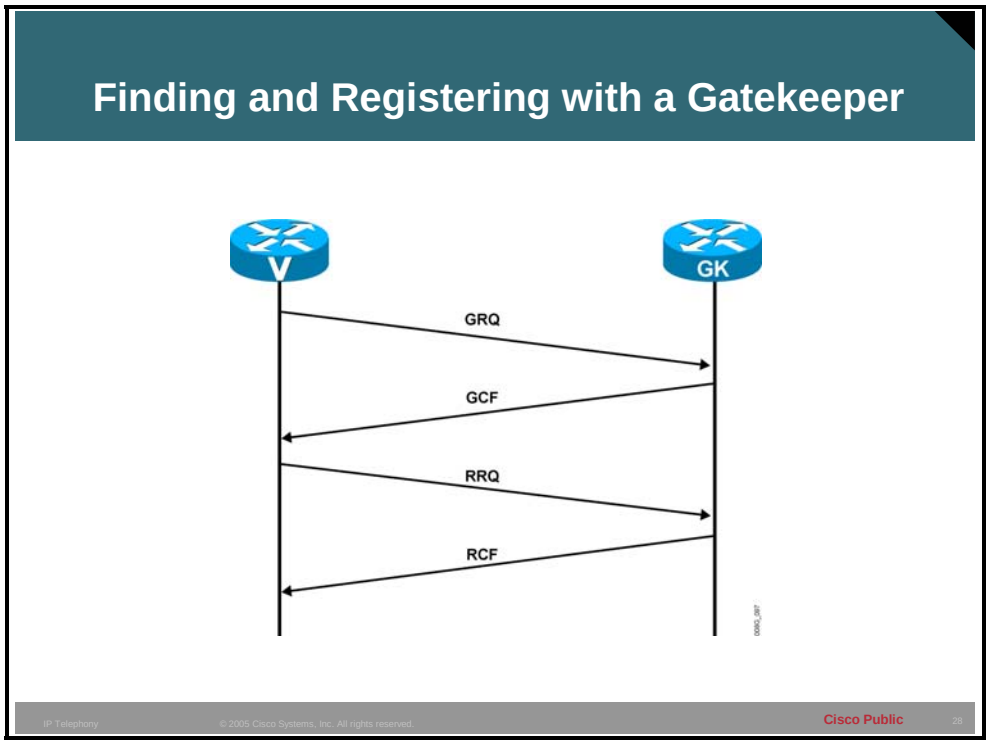


The figure shows an H.323 setup exchange that uses the Fast Connect abbreviated procedure available in version 2 of Recommendation H.323. The Fast Connect procedure reduces the number of round-trip exchanges and achieves the capability exchange and logical channel assignments in one round trip.

The Fast Connect procedure includes these steps:

1. The originating gateway initiates an H.225.0 session with the destination gateway on registered TCP port 1720.
2. Call setup procedures based on Q.931 create a combined call-signaling channel and control channel for H.245. Capabilities and logical channel descriptions are exchanged within the Q.931 call setup procedure.
3. Logical channel descriptions open RTP sessions.
4. The endpoints exchange multimedia over the RTP sessions.

Note	Cisco H.323 voice equipment supports up to version 4 of H.323 and is backward compatible to earlier versions.



The figure illustrates how an endpoint locates and registers with a gatekeeper. A gatekeeper adds scalability to H.323. Without a gatekeeper, an endpoint must recognize or have the ability to resolve the IP address of the destination endpoint.

Before an endpoint can use a gatekeeper, it must register with the gatekeeper. To register, an endpoint must recognize the IP address of the gatekeeper.

One of these two methods are used to determine the address of the gatekeeper:

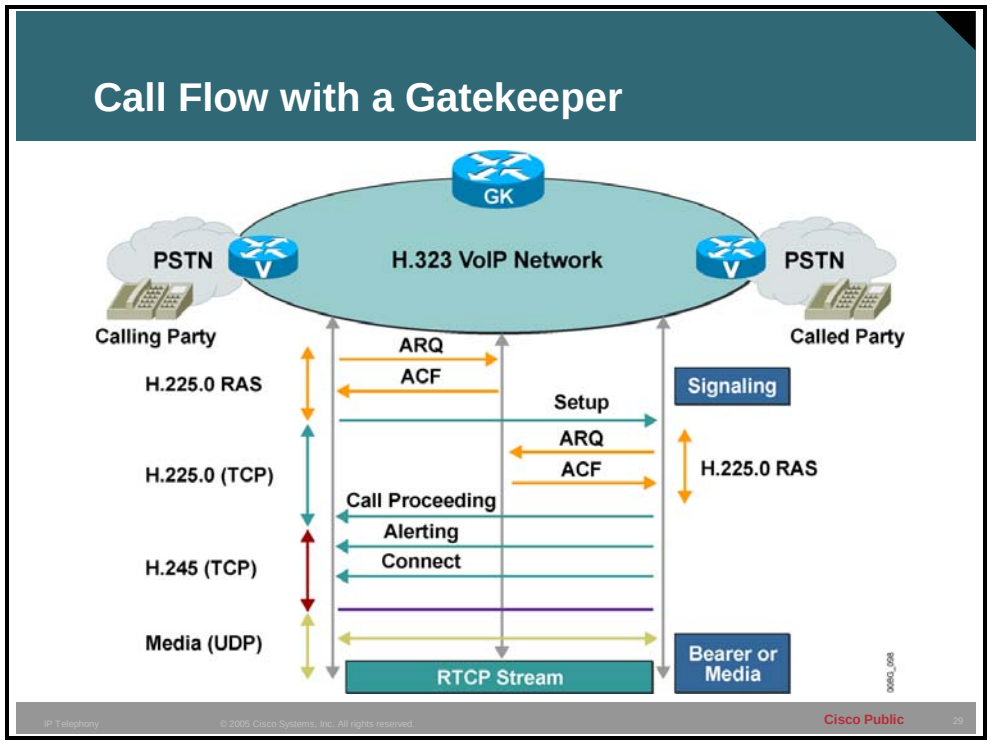
- An endpoint can be preconfigured to recognize the domain name or IP address of its gatekeeper. If configured to recognize the name, an endpoint must have a means to resolve the name to an IP address. A common address resolution technique is the DNS.
- An endpoint can issue a multicast GRQ to the gatekeeper discovery address (224.0.1.41) to discover the IP address of its gatekeeper. If the endpoint receives a GCF to the request, it uses the IP address to proceed with registration.

To initiate registration, an endpoint sends an RRQ to the gatekeeper. In the register request, the endpoint identifies itself with its ID and provides its IP address. Optionally, the endpoint lists the prefixes (for example, telephone numbers) that it supports. These prefixes are gleaned from the plain old telephone service (POTS) dial-peer destination patterns associated with any FXS port.

With this procedure, a gatekeeper determines the location and identity of endpoints and the identities of SCN endpoints from gateway registrations.

Call Flows with a Gatekeeper

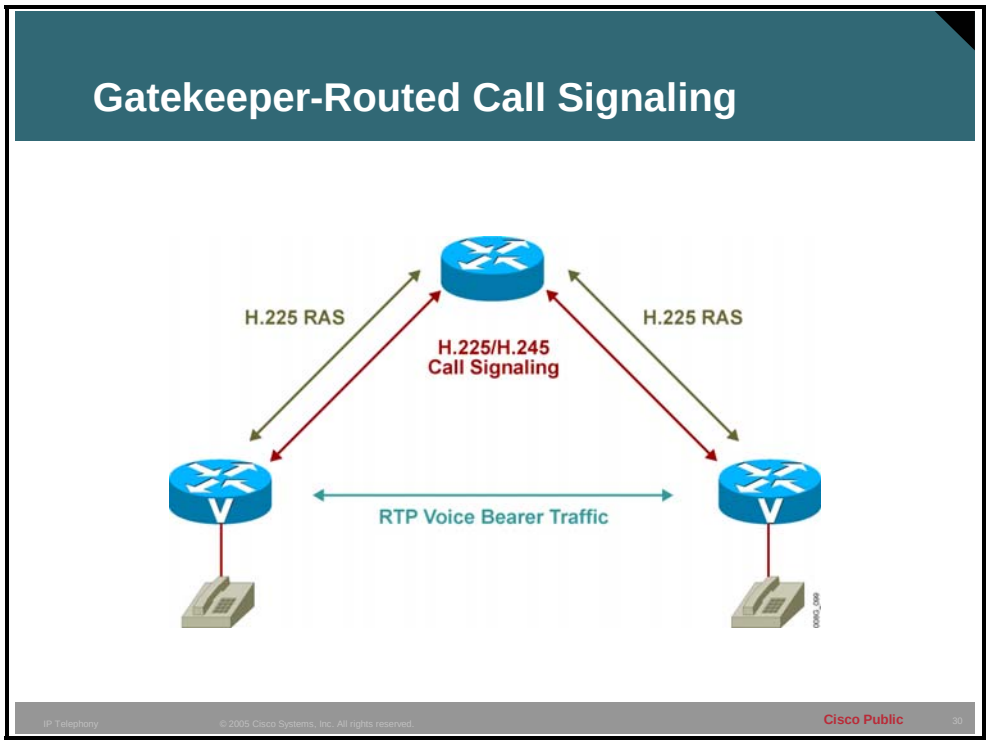
This topic discusses call setup scenarios with a gatekeeper.



The exchanges in the figure illustrate the use of a gatekeeper by both endpoints. In this example, both endpoints have registered with the same gatekeeper. Call flow with a gatekeeper proceeds as follows:

1. The gateway sends an ARQ to the gatekeeper to initiate the procedure. The gateway is configured with the domain or address of the gatekeeper.
2. The gatekeeper responds to the admission request with an ACF. In the confirmation, the gatekeeper provides the IP address of the remote endpoint.
3. When the originating endpoint identifies the terminating endpoint, it initiates a basic call setup.
4. Before the terminating endpoint accepts the incoming call, it sends an ARQ to the gatekeeper to gain permission.
5. The gatekeeper responds affirmatively, and the terminating endpoint proceeds with the call setup procedure.

During this procedure, if the gatekeeper responds to either endpoint with an ARJ to the admission request, the endpoint that receives the rejection terminates the procedure.

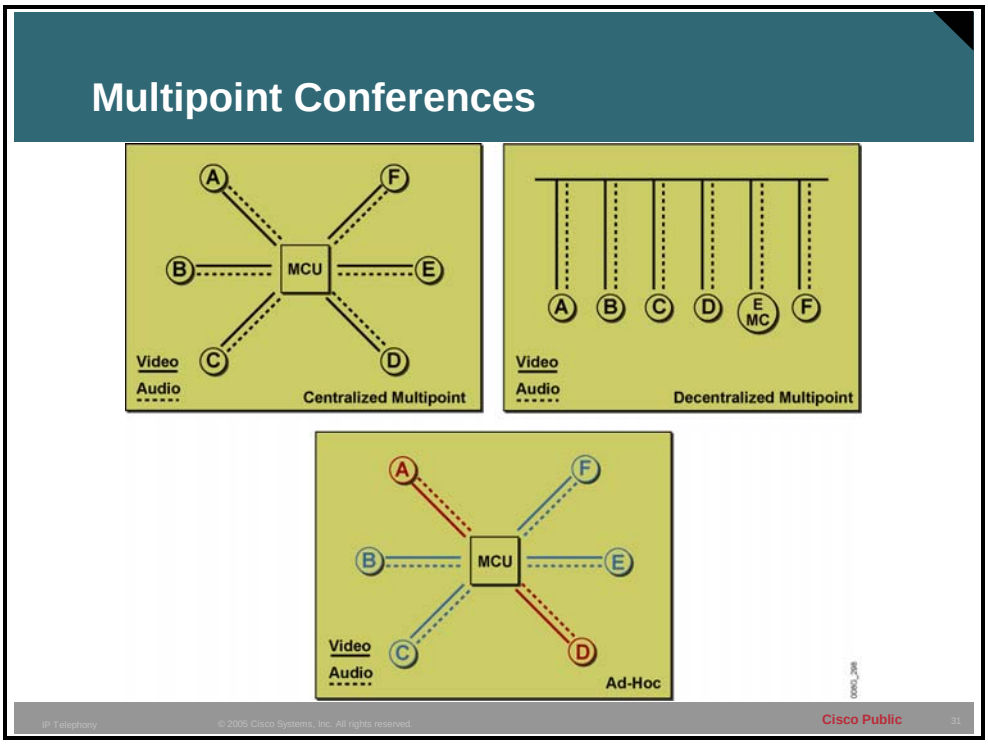


In the previous examples, the call-signaling channel is created from endpoint to endpoint. In some cases, it is desirable to have the gatekeeper represent the other endpoint for signaling purposes. This method is called gatekeeper-routed call signaling. The process for gatekeeper-routed call signaling is as follows:

1. The gatekeeper responds to an admission request and advises the endpoint to perform the call setup procedure with the gatekeeper, not with the terminating endpoint.
2. The endpoint initiates the setup request with the gatekeeper.
3. The gatekeeper sends its own request to the terminating endpoint and incorporates some of the details acquired from the originating request.
4. When a connect message is received from the terminating endpoint, the gatekeeper sends a connect to the originating endpoint.
5. The two endpoints establish an H.245 control channel between them. The call procedure continues normally from this point.

Multipoint Conferences

H.323 defines three types of multipoint conferences: centralized, distributed, and ad-hoc. H.323 also defines a hybrid of the first two. This topic describes the multipoint conference control components used to support these conferences.



All types of multipoint conferences rely on a single MC to coordinate the membership of a conference. Each endpoint has an H.245 control channel connection to the MC. Either the MC or the endpoint initiates the control channel setup. H.323 defines the following three types of conferences:

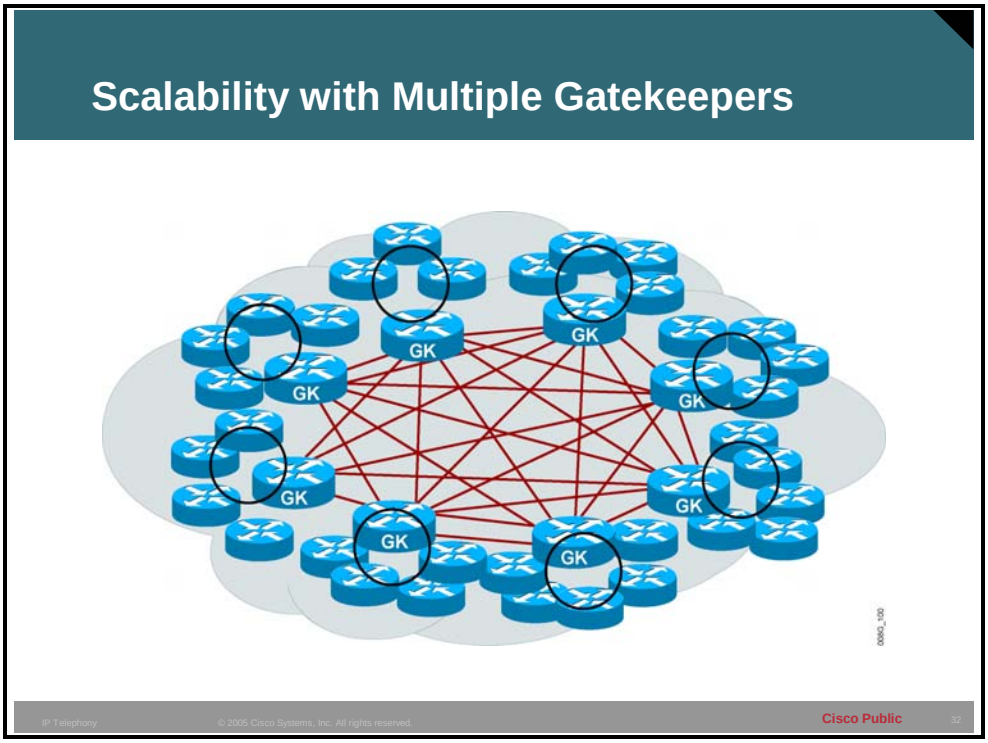
- **Centralized multipoint conference:** The endpoints must have their audio, video, or data channels connected to an MP. The MP performs mixing and switching of the audio, video, and data, and if the MP supports the capability, each endpoint can operate in a different mode.
- **Distributed multipoint conference:** The endpoints do not have a connection to an MP. Instead, endpoints multicast their audio, video, and data streams to all participants in the conference. Because an MP is not available for switching and mixing, any mixing of the conference streams is a function of the endpoint, and all endpoints must use the same communication parameters.

To accommodate situations in which two streams (audio and video) would be handled by the different multipoint conference models, H.323 defines a “hybrid.” A hybrid describes a situation in which the audio and video streams are managed by a single H.245 control channel with the MC, but where one stream relies on multicast (according to the distributed model) and the other uses the MP (as in the centralized model).

- **Ad-hoc multipoint conference:** Any two endpoints in a call can convert their relationship into a point-to-point conference. If neither of the endpoints has a colocated MC, then the services of a gatekeeper are used. When the point-to-point conference is created, other endpoints become part of the conference by accepting an invitation from a current participant, or the endpoint can request to join the conference.

Call Flows with Multiple Gatekeepers

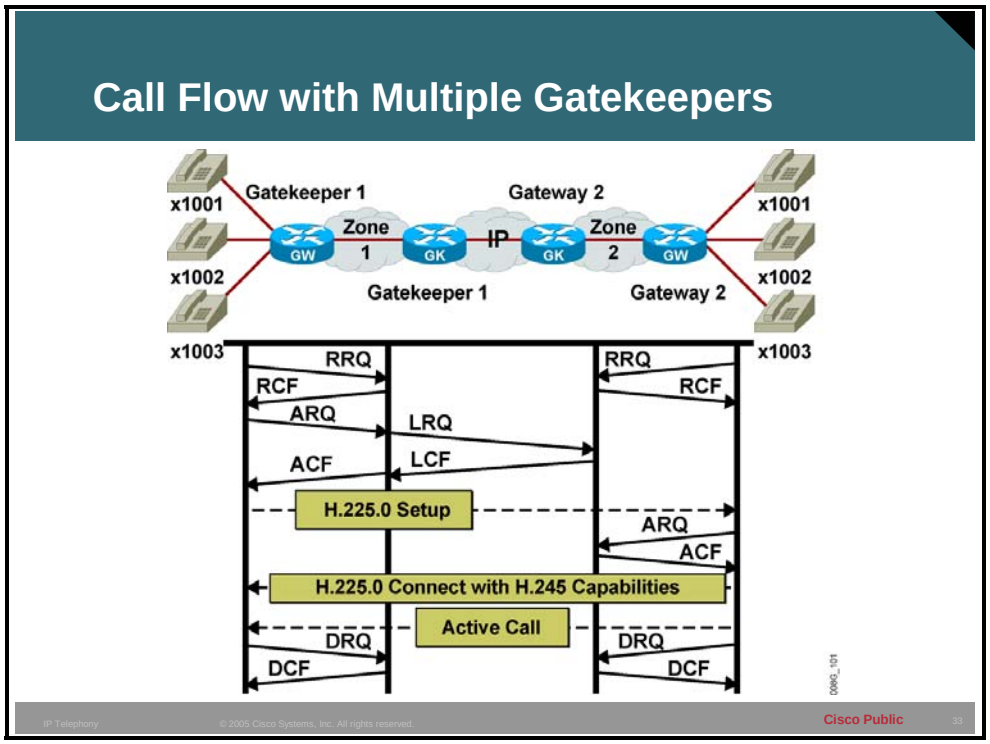
By simplifying configuration of the endpoints, gatekeepers aid in building large-scale VoIP networks. As the VoIP network grows, incorporating additional gatekeepers enhances the network scalability. This topic discusses the use of multiple gatekeepers for scalability and illustrates call flow in a multiple gatekeeper environment.



Without a gatekeeper, endpoints must find each other by any means available. This limits the growth potential of the VoIP network. Through the registration and address resolution services of a gatekeeper, growth potential improves significantly.

A single gatekeeper design may not be appropriate for several reasons. A single gatekeeper can become overloaded, or it can have an inconvenient network location, necessitating a long and expensive round trip to it.

Deploying multiple gatekeepers offers a more scalable and robust environment.

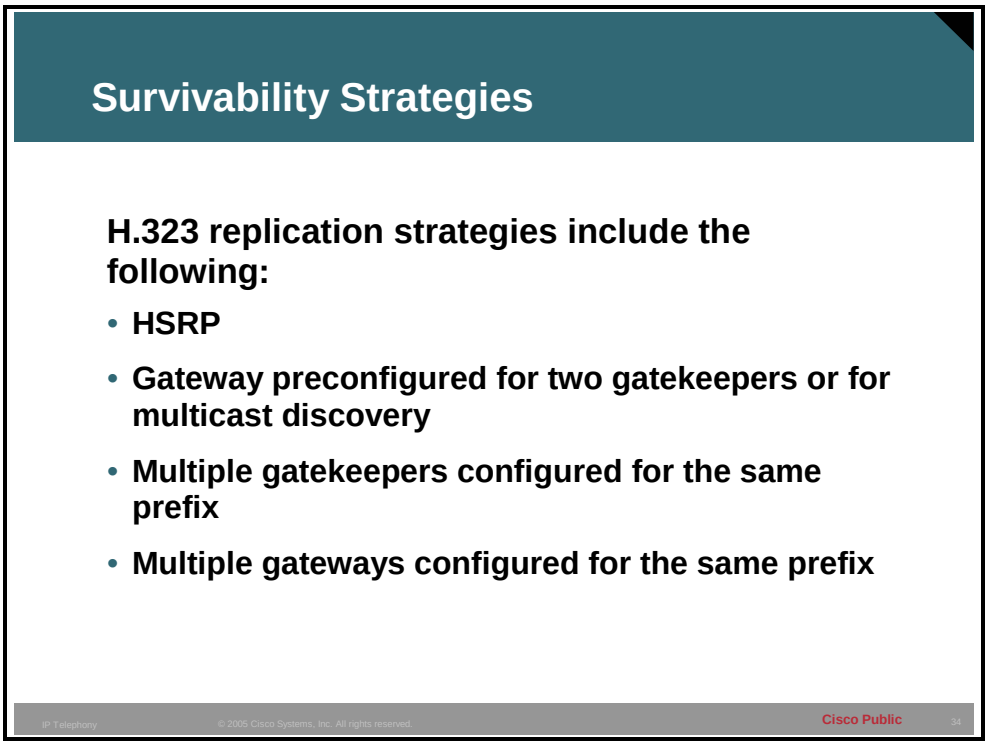


The figure illustrates a call setup involving two gatekeepers. In this example, each endpoint is registered with a different gatekeeper. Notice the changes in the following call setup procedure:

1. The originating endpoint sends an admission request to its gatekeeper requesting permission to proceed and asking for the session parameters for the terminating endpoint.
2. The gatekeeper for the originating endpoint (gatekeeper 1) determines from its configuration or from a directory resource that the terminating endpoint is potentially associated with gatekeeper 2. Gatekeeper 1 sends an LRQ to gatekeeper 2.
3. Gatekeeper 2 recognizes the address and sends back an LCF. In the confirmation, gatekeeper 2 provides the IP address of the terminating endpoint.
4. If gatekeeper 1 considers the call acceptable for security and bandwidth reasons, it maps the LCF to an ARQ and sends the confirmation back to the originating endpoint.
5. The endpoint initiates a call setup to the remote endpoint.
6. Before accepting the incoming call, the remote endpoint sends an ARQ to gatekeeper 2 requesting permission to accept the incoming call.
7. Gatekeeper 2 performs admission control on the request and responds with a confirmation.
8. The endpoint responds to the call setup request.
9. The call setup progresses through the H.225.0 call function and H.245 control function procedures until the RTP sessions are initiated.
10. At the conclusion of the call, each endpoint sends a disconnect request to its gatekeeper to advise the gatekeeper that the call is complete.
11. The gatekeeper responds with a confirmation.

Survivability Strategies

Maintaining high availability in an H.323 environment requires a design that accommodates failure of a critical component. This topic describes strategies for maintaining VoIP service.



In any environment that depends on common control components, the vulnerability of the environment is directly proportional to the probability of common control component failure. In a classical telephone application, fault tolerance is accommodated by incorporating extra common control technology. One strategy replicates all critical components. This expensive approach is often replaced with the more cost-effective solution of "*n* out of *n* + 1" redundancy; a single spare component is available to step in when any one of the active *n* components fails. The essential part of either strategy is the replication of key components.

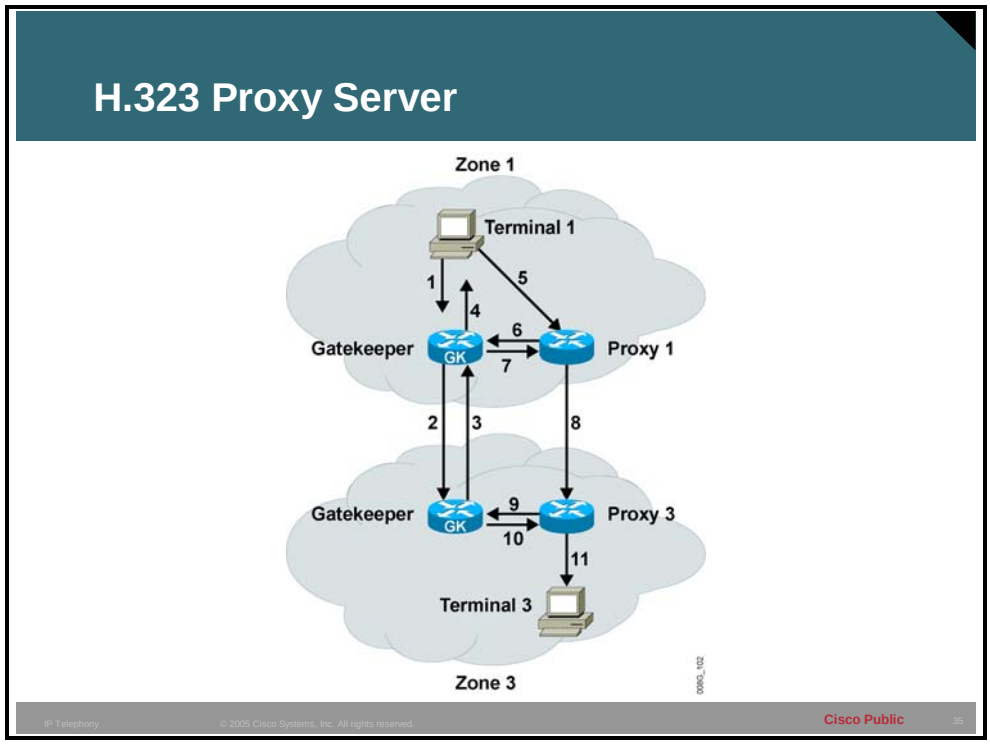
In H.323, the key components are the gateways and the gatekeeper. H.323 can employ any of the following strategies:

- **Hot Standby Router Protocol:** The Hot Standby Router Protocol (HSRP) allows two gatekeepers to share both an IP address and access to a common LAN; however, at any time, only one is active. Endpoints are configured with the name of the gatekeeper, which they can resolve using DNS or the IP address of the gatekeeper.
- **Multiple gatekeepers with gatekeeper discovery:** Deployment of multiple gatekeepers reduces the probability of total loss of gatekeeper access. However, adding new gatekeepers presents a new challenge. Each gatekeeper creates a unique H.323 zone. Because an H.323 endpoint is associated with only one gatekeeper at a time (in only one zone at a time), endpoints are configured to find only one of several working gatekeepers. Fortunately, a gateway can be configured with an ordered list of gatekeepers, or to use IP multicast to locate a gatekeeper.

- **Multiple gatekeepers configured for the same prefix:** Gatekeepers send location request messages to other gatekeepers when locating an endpoint. By supporting the same prefix on multiple gatekeepers, the location request can be resolved by multiple gatekeepers. This strategy makes the loss of one gatekeeper less significant.
- **Multiple gateways configured for the same prefix:** Survivability is enhanced at the gateway with multiple gateways that are configured to reach the same SCN destination. By configuring the same prefix of destinations in multiple gateways, the gatekeeper sees the same prefix more than once as each gateway registers with its gatekeeper.

H.323 Proxy Server

This topic describes a call setup scenario involving a proxy server.



An H.323 proxy server can circumvent the shortcomings of a direct path in cases where the direct path between two H.323 endpoints is not the most appropriate; for example, when the direct path has poor throughput and delay characteristics, is not easily available because of a firewall, or zones are configured as inaccessible on the gatekeepers in order to isolate addressing information in different zones.

When a proxy server is involved, two sessions are typically established as follows:

- Originating endpoint to the proxy server
- Proxy server to the terminating endpoint

However, when a proxy server also represents the terminating endpoint, a third session is required, as follows:

- Originating endpoint to proxy server 1
- Proxy server 1 to proxy server 2
- Proxy server 2 to terminating endpoint

Example: H.323 Proxy

The figure illustrates an example with three sessions. The objective in this scenario is for terminal 1 and terminal 3 to establish an end-to-end relationship for multimedia communications. The following sequence of events occurs:

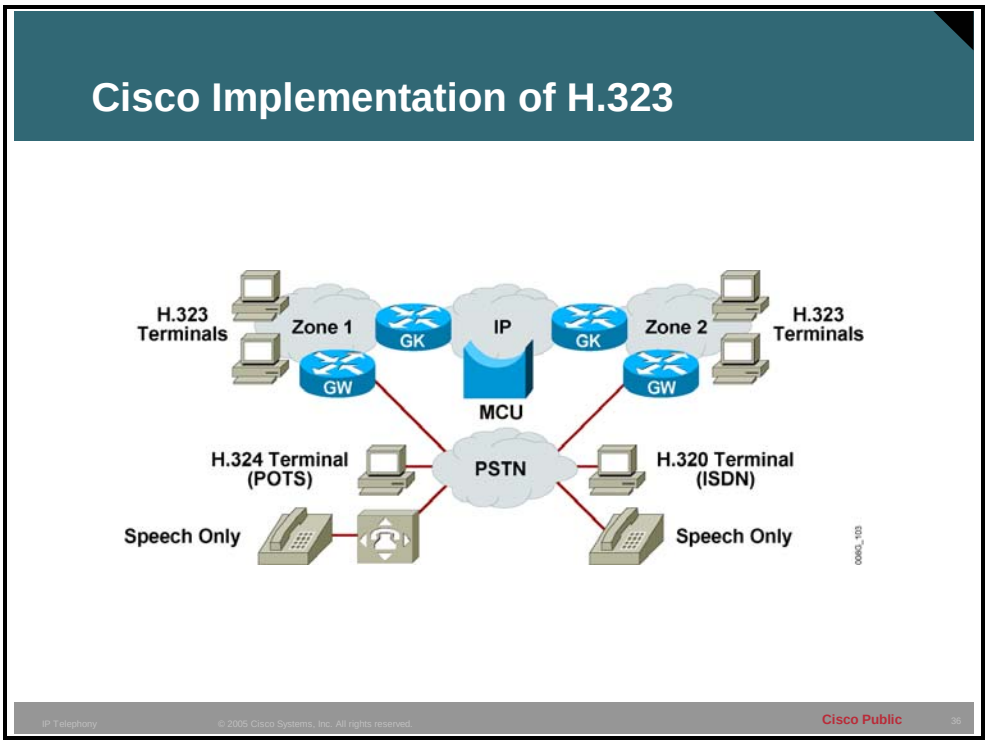
1. Terminal 1 asks gatekeeper 1 for permission to call terminal 3.

2. Gatekeeper 1 locates gatekeeper 3 as the terminal 3 gatekeeper. Gatekeeper 1 asks gatekeeper 3 for the address of terminal 3.
3. Gatekeeper 3 responds with the address of proxy 3 (instead of the address of terminal 3) to hide the identity of terminal 3.
4. Gatekeeper 1 is configured to get to proxy 3 by way of proxy 1, so gatekeeper 1 returns the address of proxy 1 to terminal 1.
5. Terminal 1 calls proxy 1.
6. Proxy 1 consults gatekeeper 1 to discover the true destination of the call, which is terminal 3 in this example.
7. Gatekeeper 1 instructs proxy 1 to call proxy 3.
8. Proxy 1 calls proxy 3.
9. Proxy 3 consults gatekeeper 3 for the true destination, which is terminal 3.
10. Gatekeeper 3 gives the address of terminal 3 to proxy 3.
11. Proxy 3 completes the call to terminal 3.

Notice that the resulting path between terminal 1 and terminal 3 involves three separate legs; one between terminal 1 and proxy 1, one between proxy 1 and proxy 3, and one between proxy 3 and terminal 3. Both the media and any signaling are carried over these three legs.

Cisco Implementation of H.323

This topic discusses how Cisco implements H.323.

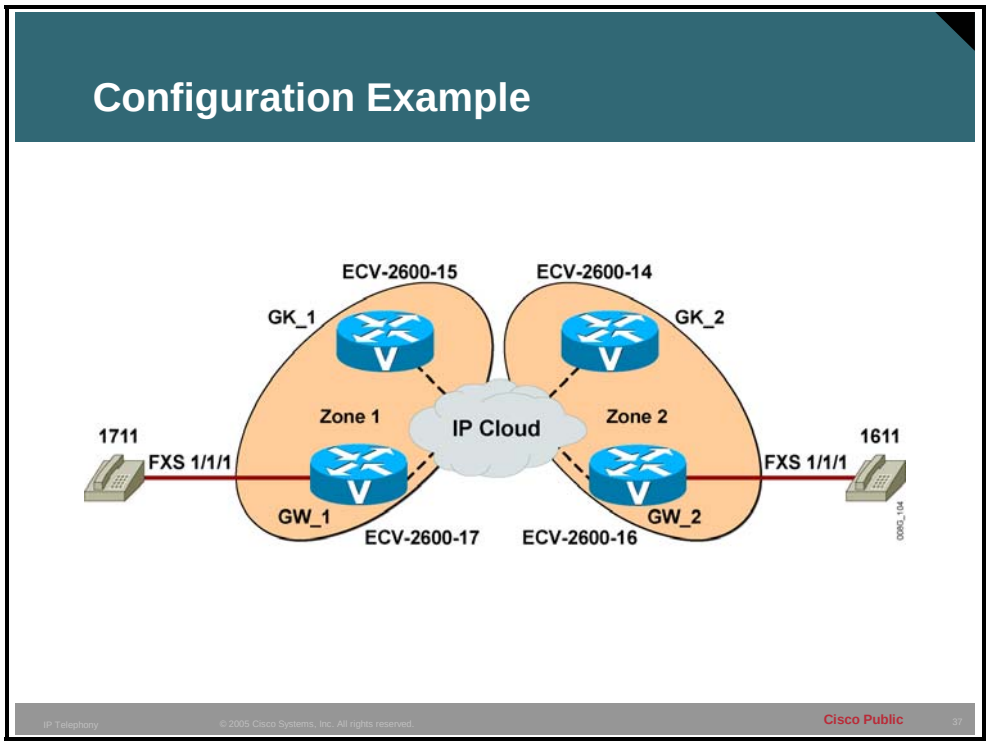


Cisco provides support for all H.323 components. These H.323 components include the following:

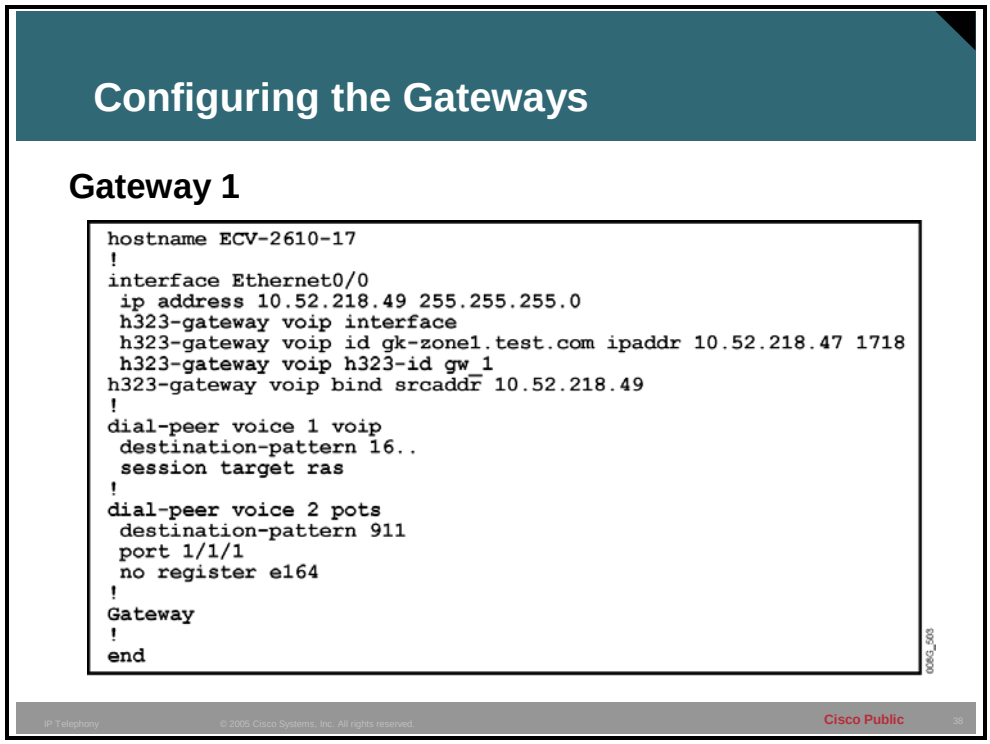
- **H.323 terminals:** Cisco provides support for H.323 terminals in Cisco IP Phone.
- **Gateways:** Cisco implements H.323 gateway support in:
  - Cisco voice-enabled routers (first available in Cisco IOS Release 11.3)
  - Cisco SC2200 Signaling Controllers
  - Cisco PGW 2200 PSTN gateways
  - Voice-enabled Cisco AS5*xx*0 access servers
  - Cisco BTS 10200 Softswitch
- **Gatekeepers:** Cisco implements gatekeeper support in:
  - Cisco Multimedia Conference Manager
  - Cisco CallManager
  - Routers (first available in Cisco IOS Release 11.3)
- **Multipoint control unit:** The MC and MP of the Cisco IP/VC 3500 Series MCU support all H.323 conference types. The IP/VC 3500 also incorporates a gatekeeper.
- **Other support:** Cisco PIX 500 Series firewalls and Context-Based Access Control (CBAC) support in the Cisco Secure Integrated Software monitor the logical channel handshaking of the H.245 control function and dynamically open conduits for the RTP sessions.

Configuring H.323 Gateways

This topic illustrates and describes the configuration commands used to create a two-zone, two-gatekeeper scenario.

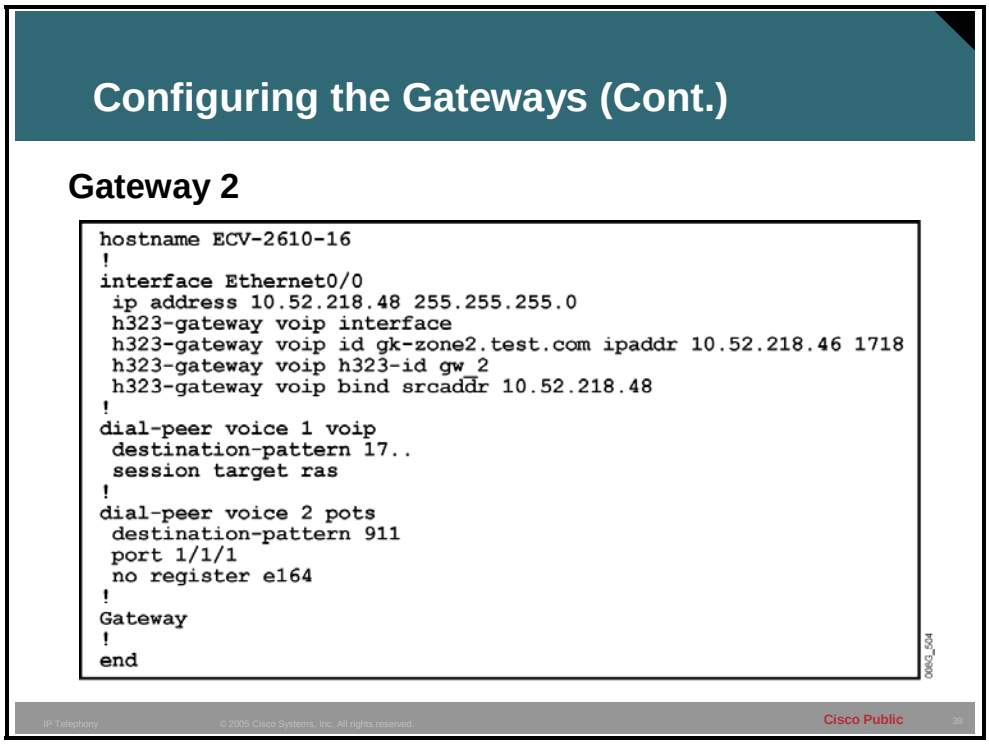


The figure illustrates the scenario on which the gateway configurations are based.



To use a gatekeeper, the user must complete the following three tasks on the gateway:

1. Enable the gateway with the **gateway** command.
2. Configure the relationship with the gatekeeper. This requires three interface subcommands:
   - **h323-gateway voip interface:** Tells the router that this interface should be enabled for H.323 packet processing.
   - **h323-gateway voip id:** Identifies the ID of the gatekeeper.
   - **h323-gateway voip h323-id:** Configures the ID of this router. When the router registers with the gatekeeper, the gatekeeper recognizes the gateway by this ID.
3. Configure a dial peer to use the gatekeeper with the **ras** parameter on the dial peer subcommand **session target**.



You can use other interface subcommands, one of which is illustrated in the configuration for both gateways. This command performs the following function:

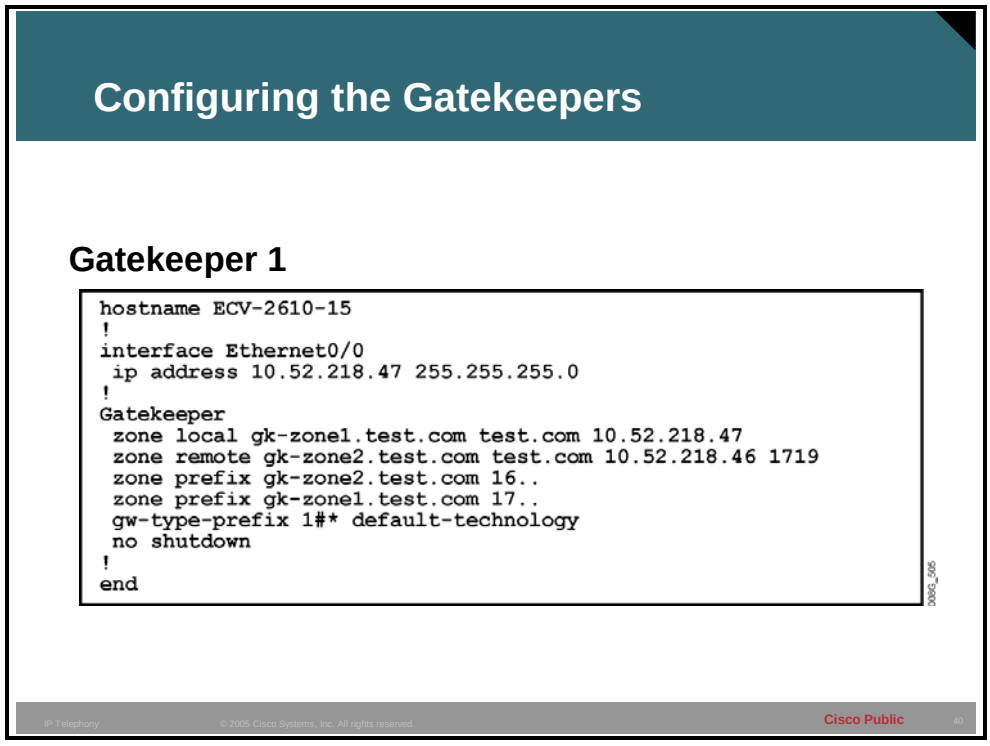
- **h323-gateway voip tech-prefix 1#:** Registers a technology prefix
- **h323-gateway voip bind srcaddr 10.52.218.48:** Sets the source address in H.323 packets

A technology prefix advises the gatekeeper that this gateway can handle type 1# destinations. For routing purposes, a technology prefix may be assigned to a multimedia type, such as video. By registering type 1# support, the gateway supports the video applications.

In the dial peer, the **no register e164** subcommand causes the gateway not to register the destination pattern when communicating with the gatekeeper. When the dial peer does not register its prefix, the dial peer requires an alternative mechanism for the gatekeeper to acquire this information.

Configuring H.323 Gatekeepers

This topic illustrates the gatekeeper configuration for a two-zone, two-gatekeeper scenario.



The gatekeeper application is enabled with the **gatekeeper** command.

For this example, the gateways are configured to withhold their E.164 addresses, so the gatekeepers must define the addresses locally. This is done with the **zone prefix** command. In the example, each gatekeeper has two **zone prefix** commands, the first pointing to the other gatekeeper and the second pointing to the local zone (meaning the prefix is in the local zone). The **zone prefix** command that points to itself is configured with the name of the gateway used to direct traffic to the destination. The address of the gateway is not required, because it is determined automatically when the gateway registers. The commands in the figure perform the following functions:

- **zone local gk-zone1.test.com test.com 10.52.218.47:** Defines the ID of the local gatekeeper
- **zone remote gk-zone2.test.com test.com 10.52.218.46 1719:** Defines the identity and IP address of neighboring gatekeepers

Configuring the Gatekeepers (Cont.)

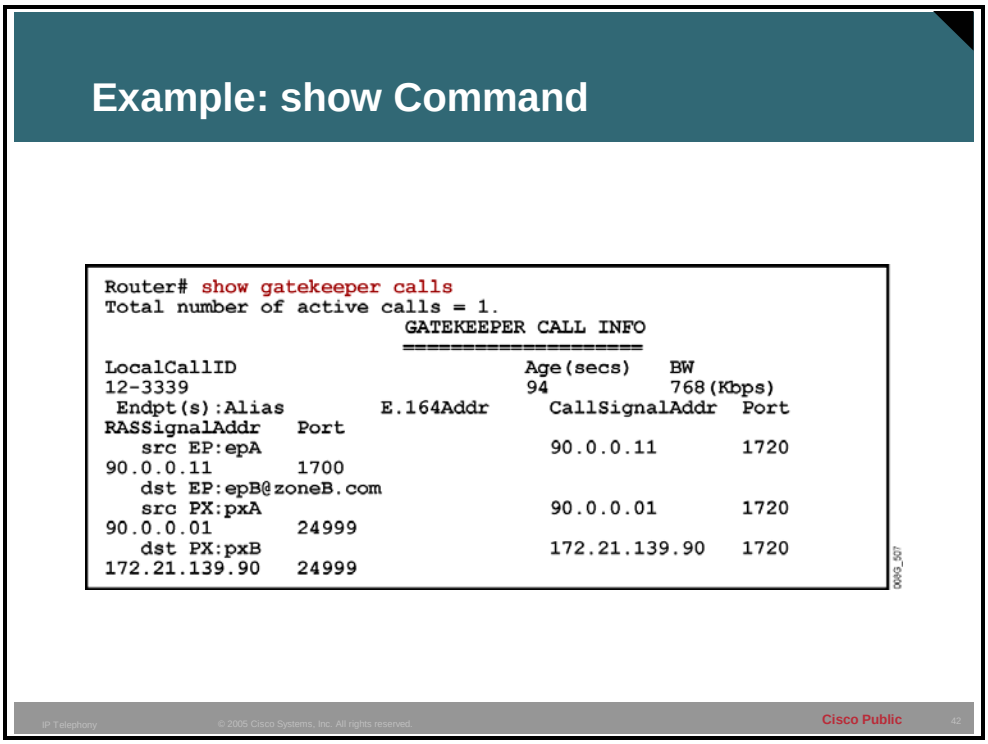
Gatekeeper 2

```
hostname ECV-2610-14
!
interface Ethernet0/0
 ip address 10.52.218.46 255.255.255.0
!
Gatekeeper
 zone local gk-zone2.test.com test.com 10.52.218.46
 zone remote gk-zone1.test.com test.com 10.52.218.47 1719
 zone prefix gk-zone2.test.com 16..
 zone prefix gk-zone1.test.com 17..
 gw-type-prefix 1#* default-technology
 no shutdown
!
end
```


Because the gateways register their technology prefixes, the gatekeeper does not need to be configured. If a technology prefix is required, the **gw-type-prefix** defines a technology prefix, and can manually update technology prefix knowledge in the gatekeeper. In the example configuration in this figure, the gatekeeper attempts to define a technology prefix as the default with the command **gw-type-prefix 1#* default-technology**. Any unknown destination is assumed to be of the default technology type, and calls are forwarded to any gateway that registered the default technology type.

Monitoring and Troubleshooting

The **show** and **debug** commands are valuable when examining the status of the H.323 components and during troubleshooting. This topic lists many **show** and **debug** commands that are used to provide support for monitoring and troubleshooting H.323.



Following are some of the **show** commands used for H.323:

- **show call active voice** [**brief**]: Displays the status, statistics, and parameters for all active voice calls
- **show call history voice** [**last** *n*|**record**|**brief**]: Displays call records from the history buffer
- **show gateway**: Displays the current status of the H.323 gateway configured in the router
- **show gatekeeper calls**: Displays the active calls for which the gatekeeper is responsible (illustrated in the figure)
- **show gatekeeper endpoints**: Lists the registered endpoints with ID and supported prefixes
- **show gatekeeper gw-type-prefix**: Displays the current technology prefix table
- **show gatekeeper status**: Displays the current status of the gatekeeper
- **show gatekeeper zone prefix**: Displays the gateways and their associated E.164 prefixes
- **show gatekeeper zone status**: Displays the status of the connections to gateways in the local zone and the status of the connections to gatekeepers in other zones

Selected debug Commands

The **debug** commands used for H.323 include the following:

- **debug voip ccapi inout:** Shows every interaction with the call control application programming interface (API) on the telephone interface and the VoIP side. Monitoring the

debug voip ccapi inout command output allows users to follow the progress of a call from the inbound interface or VoIP peer to the outbound side of the call. Because this debug is highly active, use it sparingly in a live network.

- **debug cch323 h225:** Traces the transitions in the H.225.0 state machine during the establishment of the call control channel. The first step in establishing a relationship between any two components is to bring up the call control channel. Monitoring the output of the **debug cch323 h225** command allows users to follow the progress and determine if the channel is established correctly.
- **debug cch323 h245**: Traces the state transitions in the H.245 state machine during the establishment of the H.245 control channel. Monitoring the output of the **debug cch323 h245** command allows users to follow the progress to see if the channel is established correctly.
- **debug cch323 ras**: Traces the state transition in the establishment of the RAS control channel. Monitoring the output of the **debug cch323 ras** command allows users to determine if the channel is established correctly.
- **debug h225 asn1**: Displays an expansion of the ASN.1-encoded H.225.0 messages. When investigating VoIP peer association problems, this **debug** command helps users monitor the activity of the call-signaling channel. Because H.225.0 encapsulates H.245, this is a useful approach for monitoring both H.225.0 and H.245.
- **debug h225 events**: Similar to the ASN.1 version of the command but does not expand the ASN.1. Debugging events usually imposes a lighter load on the router.
- **debug h245 asn1**: Similar to the H.225.0 variant except that it displays only the H.245 messages.
- **debug h245 events**: Similar to the H.225.0 variant except that it displays only the H.245 messages.

