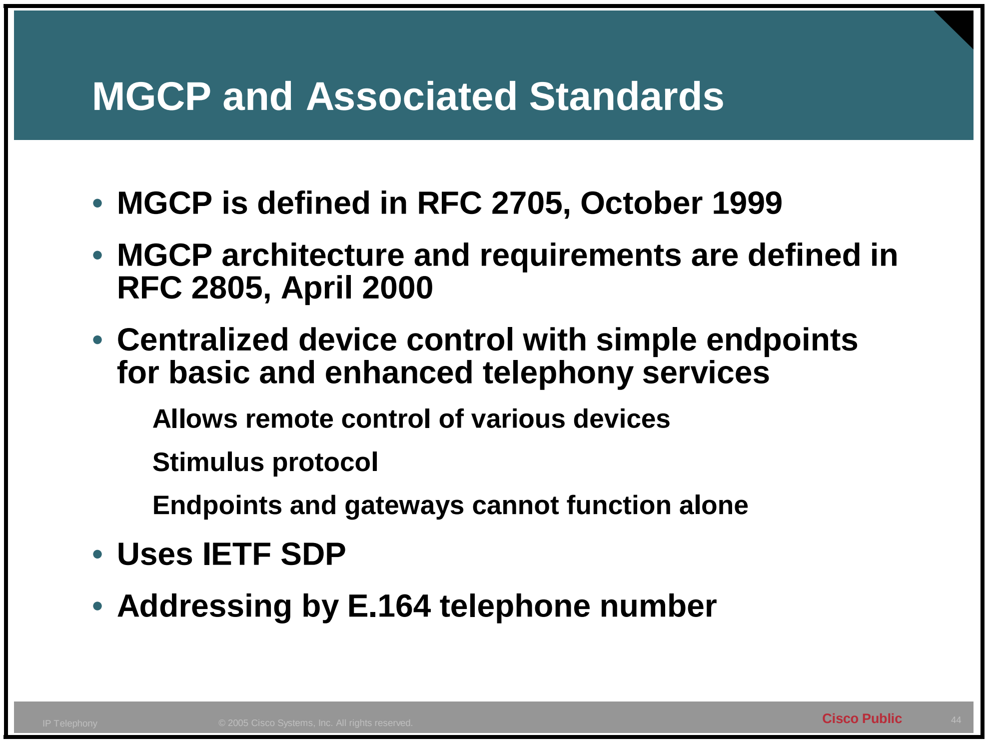
Configuring MGCP

MGCP and Its Associated Standards

MGCP controls telephony gateways from a centralized call agent. This topic describes MGCP and identifies its associated standards.

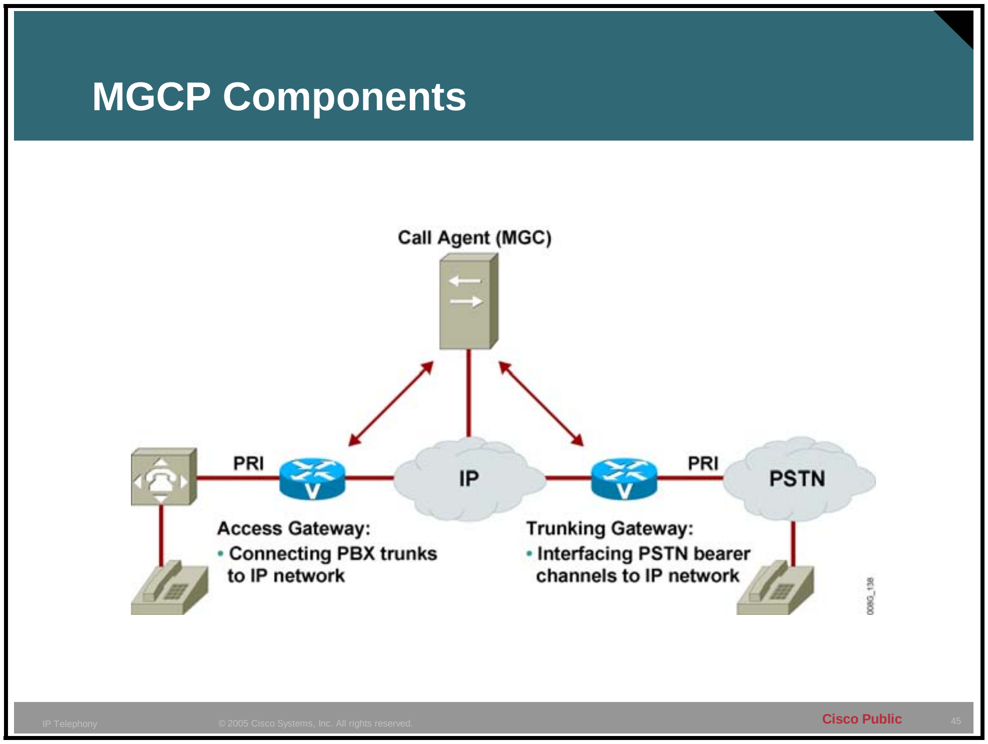


MGCP defines an environment for controlling telephony gateways from a centralized call control component known as a call agent. An MGCP gateway handles the translation of audio between the telephone SCN and the packet-switched network of the Internet. Gateways interact with a call agent that performs signaling and call processing.

IETF RFC 2705 defines MGCP. RFC 2805 defines an architecture for MGCP. These IETF standards describe MGCP as a centralized device control protocol with simple endpoints. The MGCP protocol allows a central control component, or call agent, to remotely control various devices. This protocol is referred to as a stimulus protocol because the endpoints and gateways cannot function alone. MGCP incorporates the IETF SDP to describe the type of session to initiate.

Basic MGCP Components

MGCP defines a number of components and concepts. You must understand the relationships between components and how the components use the concepts to implement a working MGCP environment. This topic describes the basic MGCP components.



The following components are used in an MGCP environment:

- **Endpoints:** Represent the point of interconnection between the packet network and the traditional telephone network
- **Gateways:** Handle the translation of audio between the SCN and the packet network
- **Call agent:** Exercises control over the operation of a gateway

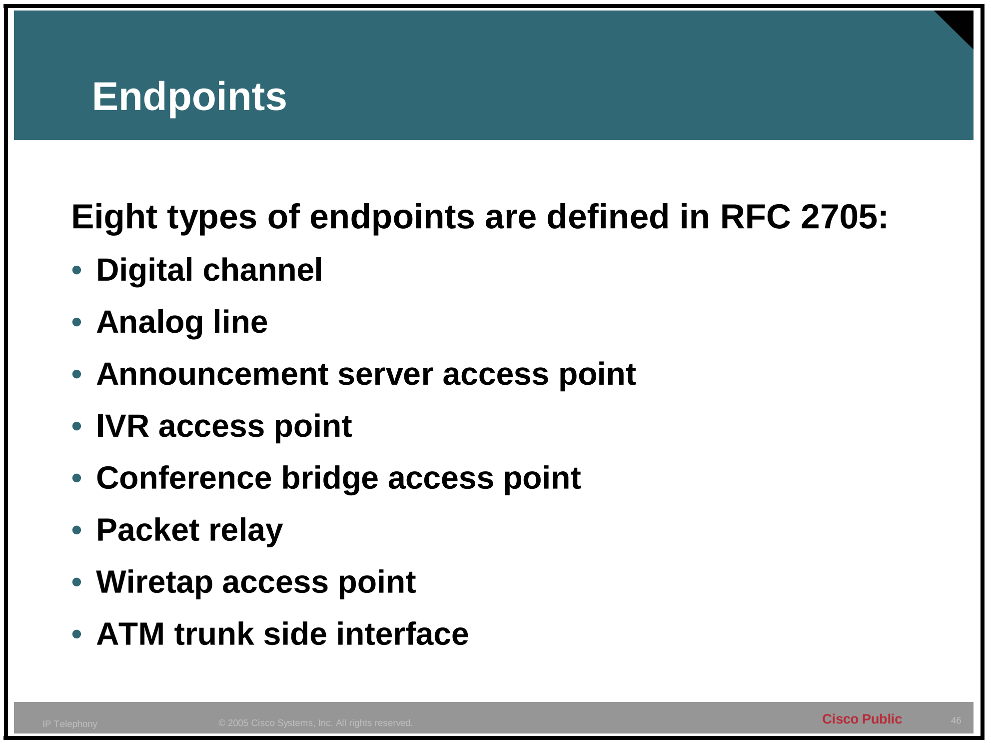
The figure shows an MGCP environment with all three components.

Example: Cisco MGCP Components

Cisco voice gateways can act as MGCP gateways. Cisco CallManager acts as an MGCP call agent.

MGCP Endpoints

This topic lists the standard endpoints and defines the way that identifiers are associated with an endpoint.



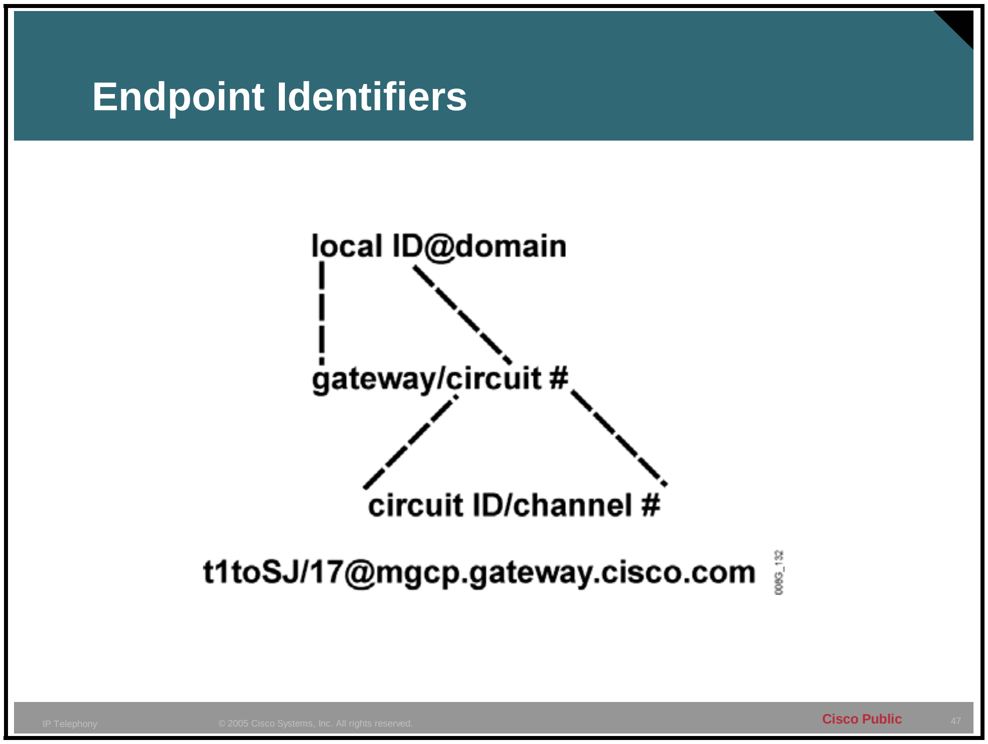
Endpoints represent the point of interconnection between the packet network and the traditional telephone network. Endpoints can be physical, representing an FXS port or a channel in a T1 or E1, or they can be logical, representing an attachment point to an announcement server.

To manage an endpoint, the call agent must recognize the characteristics of an endpoint. To aid in this process, endpoints are categorized into several types. The intent is to configure a call agent to manage a type of endpoint rather than to manage each endpoint individually.

There are several types of endpoints. RFC 2705 defines eight types, as follows:

- **Digital service level zero (DS0):** Represents a single channel (DS0) in the digital hierarchy. A digital channel endpoint supports more than one connection.
- **Analog line:** Represents the client-side interface, such as FXS, or switch-side interface, such as FXO, to the traditional telephone network. An analog line endpoint supports more than one connection.
- **Announcement server access point:** Represents access to an announcement server, for example, to play recorded messages. An announcement server endpoint may have only one connection. Multiple users of the announcement server are modeled to use different endpoints.
- **Interactive voice response (IVR) access point:** Represents access to an IVR service. An IVR endpoint has one connection. Multiple users of the IVR system are modeled to use different endpoints.

- **Conference bridge access point:** Represents access to a specific conference. Each conference is modeled as a distinct endpoint. A conference bridge endpoint supports more than one connection.
- **Packet relay:** Represents access that bridges two connections for interconnecting incompatible gateways or relaying them through a firewall environment. A packet relay endpoint has two connections.
- **Wiretap access point:** Represents access for recording or playing back a connection. A wiretap access point endpoint has one connection.
- **ATM trunk side interface:** Represents a single instance of an audio channel in the context of an ATM network. An ATM interface supports more than one connection.



When interacting with a gateway, the call agent directs its commands to the gateway for the express purpose of managing an endpoint or a group of endpoints. An endpoint identifier, as its name suggests, identifies endpoints.

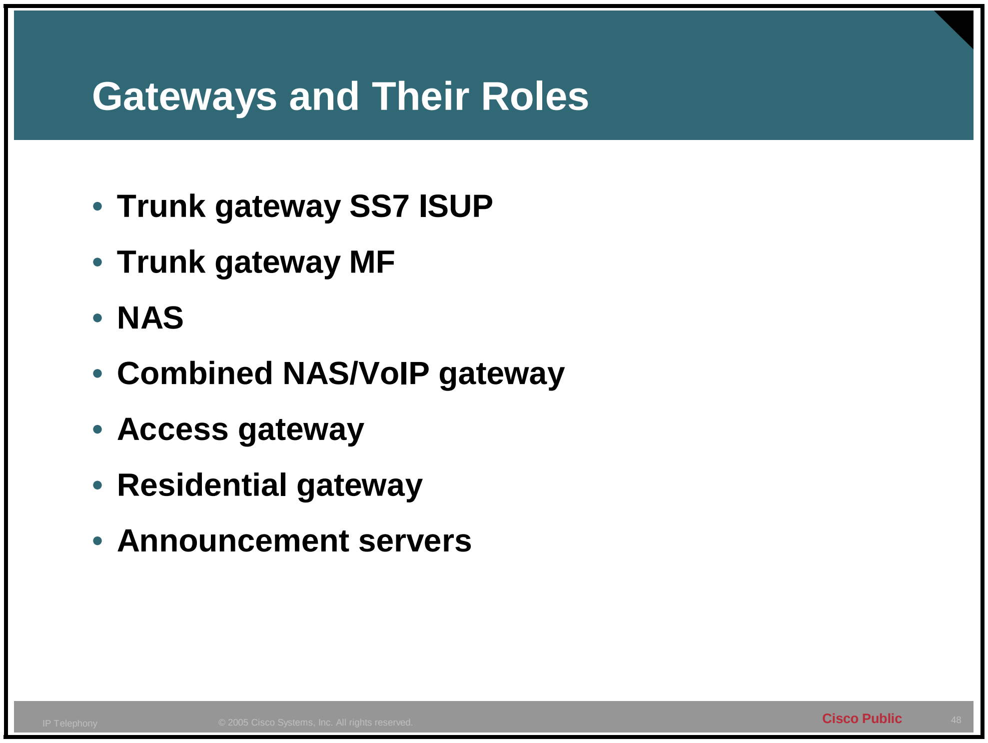
Endpoint identifiers consist of two parts: a local name of the endpoint in the context of the gateway and the domain name of the gateway itself. The two parts are separated by an "at" sign (@). If the local part represents a hierarchy, the subparts of the hierarchy are separated by a slash. In the graphic, the "local ID" may be representative of a particular "gateway/circuit #," and the "circuit #"may in turn be representative of a "circuit ID/channel #."

Example: Endpoint Identifiers

In the figure, **mgcp.gateway.cisco.com** is the domain name and **t1toSJ/17** refers to channel 17 in the T1 to San Jose.

MGCP Gateways

This topic lists several standard gateways and describes their functions.



Gateways are clustering points for endpoints. Gateways handle the translation of audio between the SCN and the packet network.

Although gateways are implemented in real systems, from a modeling point of view gateways are logical components. In this context, gateways represent a clustering of a single type and profile of endpoints.

A gateway interacts with one call agent only; therefore, it associates with one call agent at a time.

RFC 2705 identifies the following seven types of gateways:

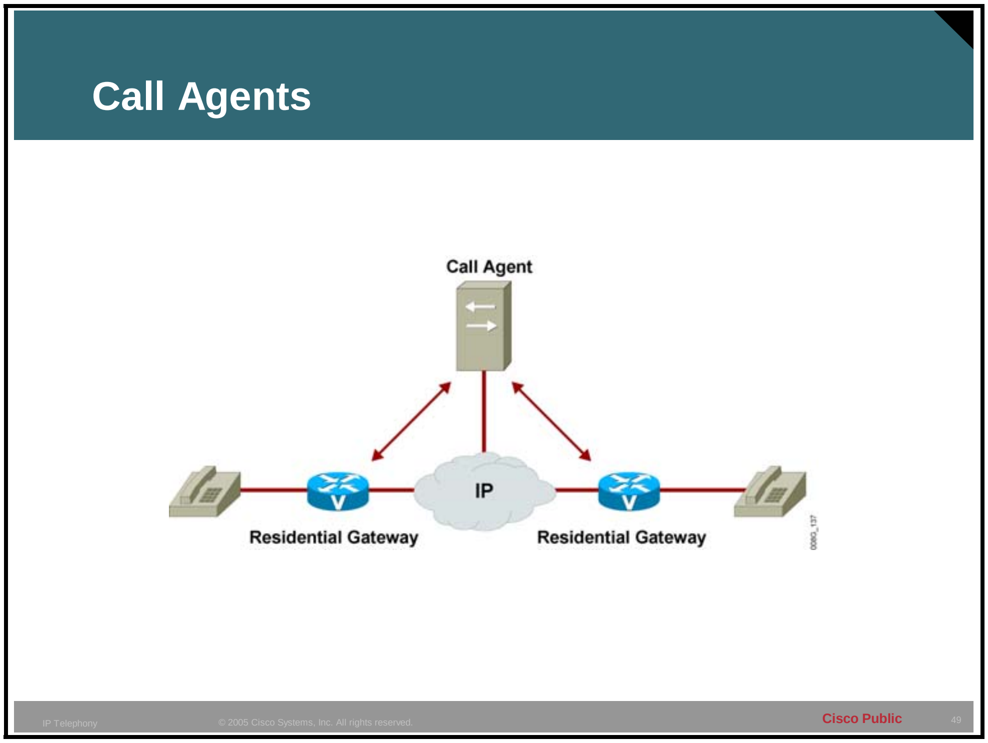
- **Trunk gateway SS7 ISUP:** Supports digital circuit endpoints subject to ISDN signaling
- **Trunk gateway multifrequency (MF):** Typically supports digital or analog circuit endpoints that are connected to a service provider of an enterprise switch that is subject to MF signaling
- **Network access server (NAS):** Supports an interconnect to endpoints over which data (modem) applications are provided
- **Combined NAS/VoIP gateway:** Supports an interconnect to endpoints over which a combination of voice and data access is provided
- **Access gateway:** Supports analog and digital endpoints connected to a PBX
- **Residential gateway:** Supports endpoints connected to traditional analog interfaces

- **Announcement servers:** Supports endpoints that represent access to announcement services

Multiple gateway types, and multiple instances of the same type, can be incorporated into a single physical gateway implementation.

MGCP Call Agents

This topic describes how a call agent controls gateways and endpoints.



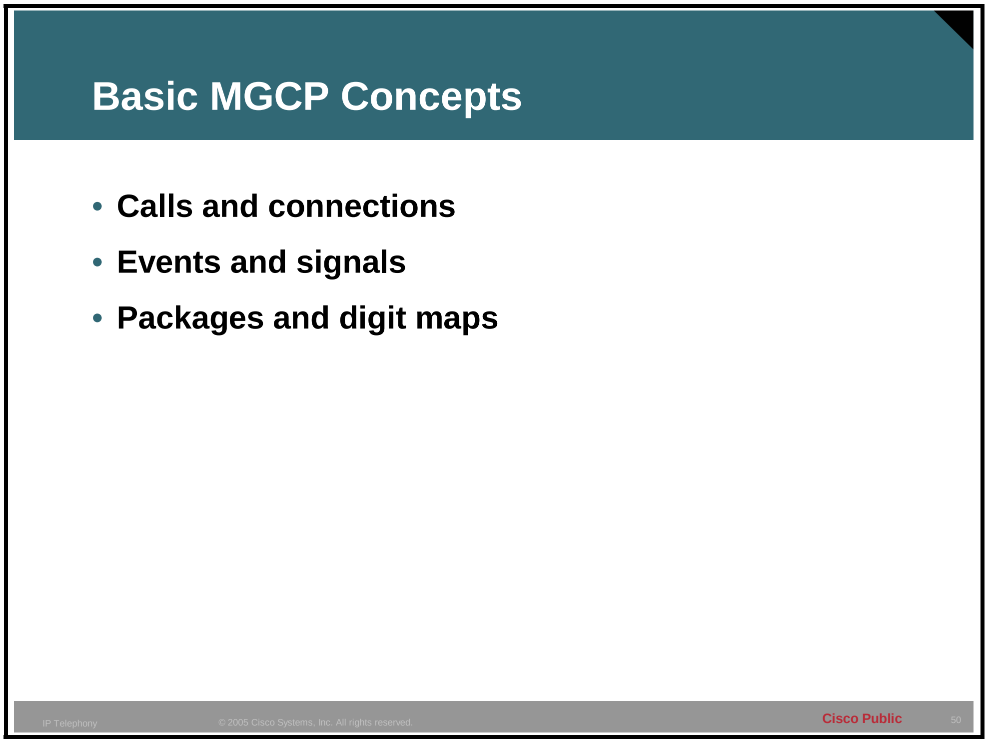
A call agent, or Media Gateway Controller (MGC), represents the central controller in an MGCP environment.

A call agent exercises control over the operation of a gateway and its associated endpoints by requesting that a gateway observe and report events. In response to the events, the call agent instructs the endpoint what signal, if any, the endpoint should send to the attached telephone equipment. This requires a call agent to recognize each endpoint type that it supports and the signaling characteristics of each physical and logical interface that is attached to a gateway.

A call agent uses its directory of endpoints and the relationship that each endpoint has with the dial plan to determine call routing. Call agents initiate all VoIP call legs.

Basic MGCP Concepts

This topic introduces the basic MGCP concepts.

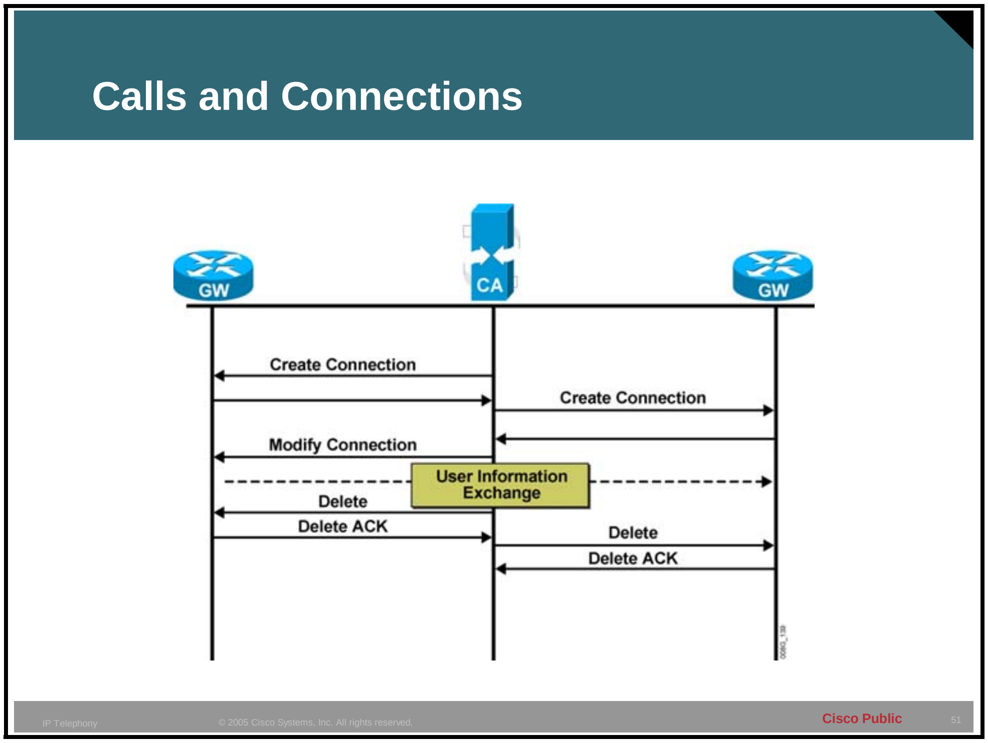


The basic MGCP concepts are listed below:

- **Calls and connections:** Allow end-to-end calls to be established by connecting two or more endpoints
- **Events and signals:** Fundamental MGCP concept that allows a call agent to provide instructions for the gateway
- **Packages and digit maps:** Fundamental MGCP concept that allows a gateway to determine the call destination

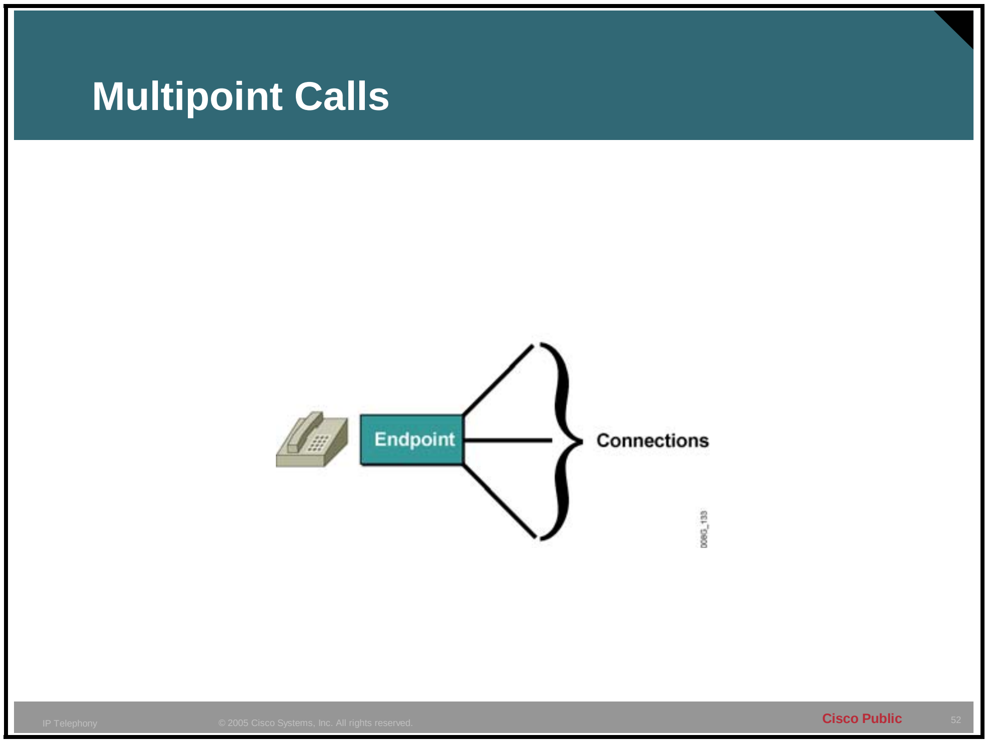
MGCP Calls and Connections

This topic discusses how end-to-end calls are established by connecting multiple endpoints.



End-to-end calls are established by connecting two or more endpoints. To establish a call, the call agent instructs the gateway that is associated with each endpoint to make a connection with a specific endpoint or an endpoint of a particular type. The gateway returns the session parameters of its connection to the call agent, which in turn sends these session parameters to the other gateway. With this method, each gateway acquires the necessary session parameters to establish RTP sessions between the endpoints. All connections that are associated with the same call will share a common call ID and the same media stream.

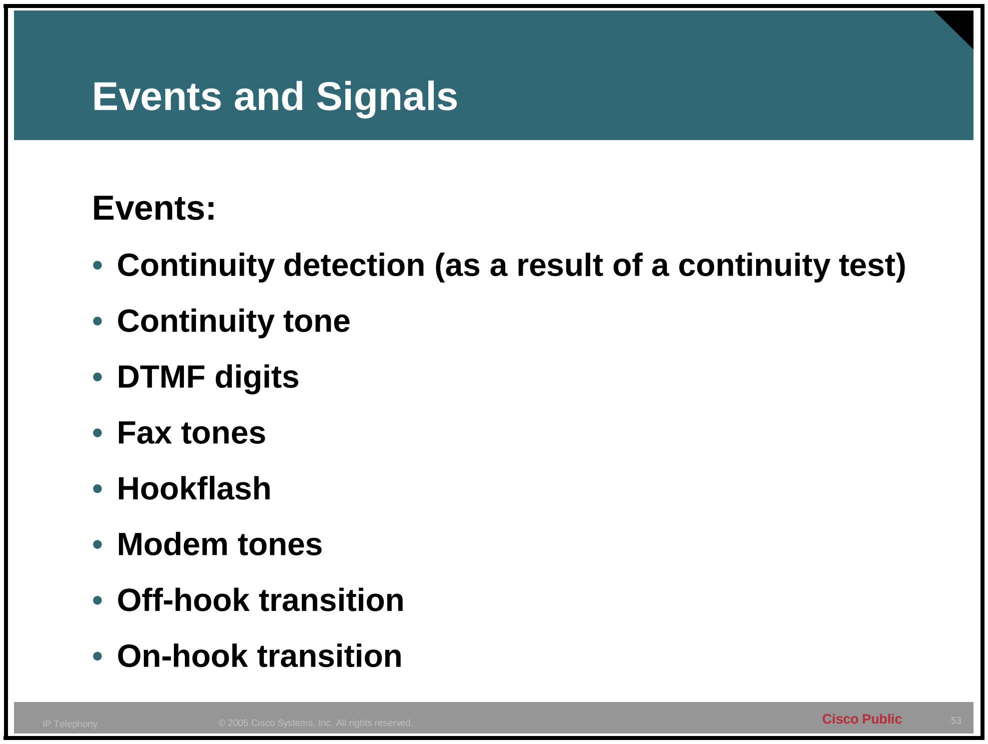
At the conclusion of a call, the call agent sends a delete connection request to each gateway.



To create a multipoint call, the call agent instructs an endpoint to create multiple connections. The endpoint is responsible for mixing audio signals.

MGCP Events and Signals

This topic describes how a call agent uses events and signals to provide instruction to the gateway.

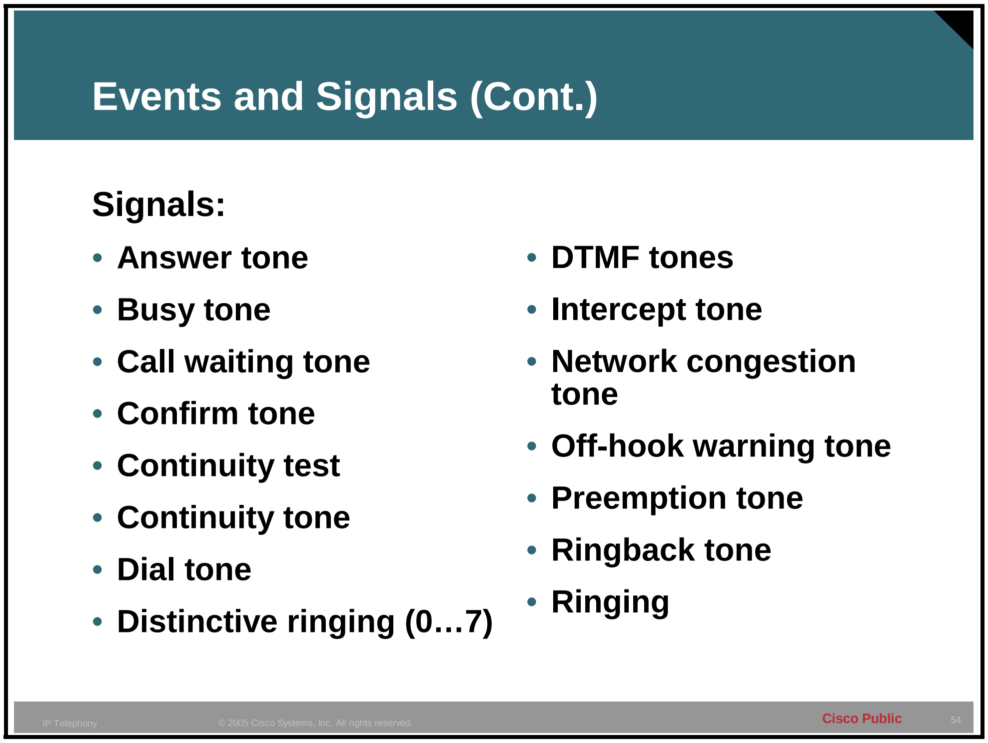


Events and signals help the call agent instruct the gateway on the call control and signaling procedures. The call agent has complete control over the gateway. The call agent informs the gateway of every action to take, including the following:

- Events that the gateway monitors on an endpoint
- What to do if an event occurs
- When to generate a notification to the call agent

An example of an event on an analog line is an off-hook condition. Using signals, the call agent requests that the gateway provide dial tone upon observing the off-hook event.

The figure lists examples of events used in MGCP environments. Events and signals are assigned simple, case-insensitive codes; for example, the code for an off-hook transition event is “hd”, and the code for the dial tone signal is “dl”.



The figure gives examples of signals used in MGCP environments.

MGCP Packages

This topic describes how events and signals are packaged in gateways and how they are used by digit maps and gateways.

Packages

- Basic packages (generic media, DTMF, MF, trunk, line, handset, RTP, NAS, announcement server, script)
- CAS packages (RFC 3064)
- Business telephone packages (RFC 3149)

Packages are collections of commonly occurring events and signals that are relevant to a specific type of endpoint; for example, “off hook,” an event, and “dial tone,” a signal, are unique to managing subscriber lines. Consequently, they are associated with the “line” package.

Each event and signal is placed in one particular package. The name of an event or signal acquires the code assigned to the package. The package code and the event code are separated by a slash. Therefore, the full specified name for the off-hook transition event is “L/hd”.

RFC 2705 defines packages, as shown in the table.

Packages

Reference	Package Name
G	Generic media
M	Multifrequency
D	DTMF
T	Trunk
N	Network access server
L	Line
A	Announcement server
R	RTP
H	Handset
S	Script

RFC 3064 defines CAS packages, while RFC 3149 defines business telephone packages. The CAS packages contain definitions for various CAS media gateways, including support for emulated E&M interfaces, direct inward dialing (DID) interfaces, FXO interfaces, and others. Business telephone packages contain support for business telephone functions (and buttons) such as hold, transfer, forward, conference, and others.

Gateways and Their Packages

Gateway Type	Packages
Trunk gateway (ISUP)	G, D, T, R
Trunk gateway (MF)	G, M, D, T, R
Network access server (NAS)	G, M, T, N
Combined NAS/VoIP gateway	G, M, D, T, N, R
Access gateway (VoIP)	G, D, M, R
Access gateway (VoIP & NAS)	G, D, M, N, R
Residential gateway	G, D, L, R
Announcement server	A, R

Packages cluster events and signals by their relevance to various types of endpoints. Conceptually, gateways also cluster endpoints of different types. It is appropriate then to associate packages with gateways. The table in the figure lists the gateways and identifies the packages that are associated with them.

MGCP Digit Maps

This topic describes the function of digit maps in MGCP.

Digit Maps

Dial	To Reach
0	Operator
xxxx	Local extension
9011 + up to 15 digits	International number
91xxxxxxxxxx	Domestic long distance
9 + 7 or 10 digits	Local PSTN

would be implemented as
Digit map = (0 | [1-8]xxx | 9[2-9]x.T | 91xxxxxxxxxx | 9011x.T)

A digit map is a specification of the dial plan. When you download a digit map to a gateway for use on an endpoint or a group of endpoints, a digit map allows the gateway to collect digits until the gateway either finds a match or concludes that the dialed digits could not possibly match the specification. When either condition occurs, the gateway notifies the call agent.

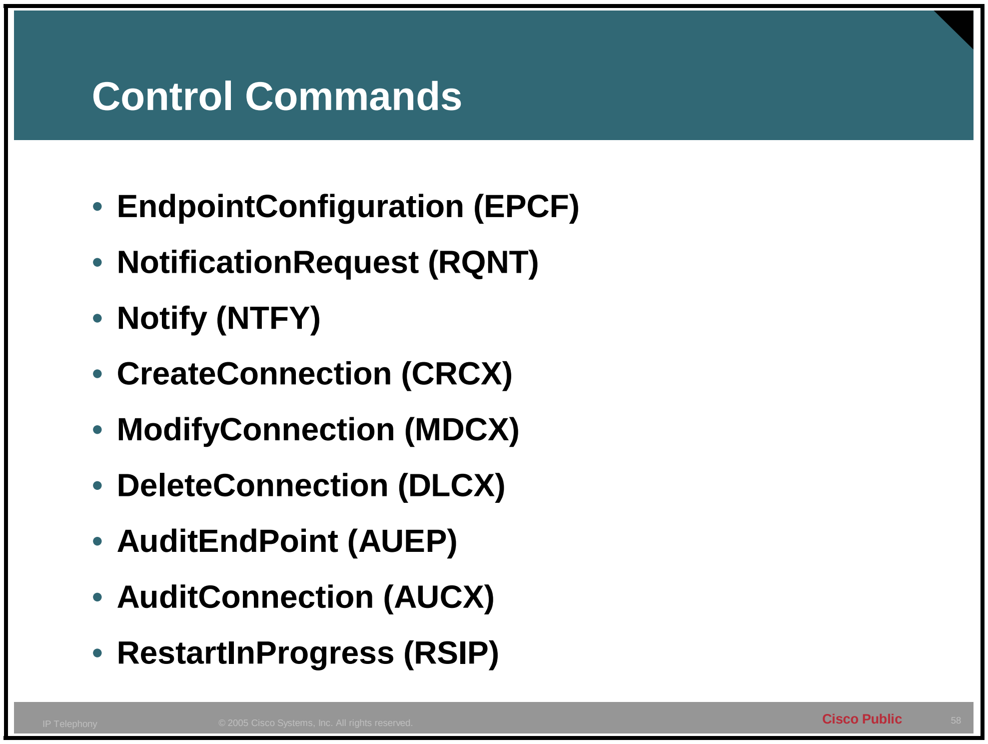
Without a digit map, a gateway must notify the call agent on each digit dialed, which places a heavy burden on the call agent and the network connecting the gateway and call agent. The figure shows an example of a digit map.

Example: Digit Map Translation

According to the figure, if a user dials 9 as the first digit, three of the possible digit maps may be invoked, depending on the next digit dialed. If the next digit is a 1, then the user is dialing domestic long distance, and the digit map 91*xxxxxxxxxx* will be used.

MGCP Control Commands

MGCP defines nine messages to control and manage endpoints and their connections. This topic describes these control messages.



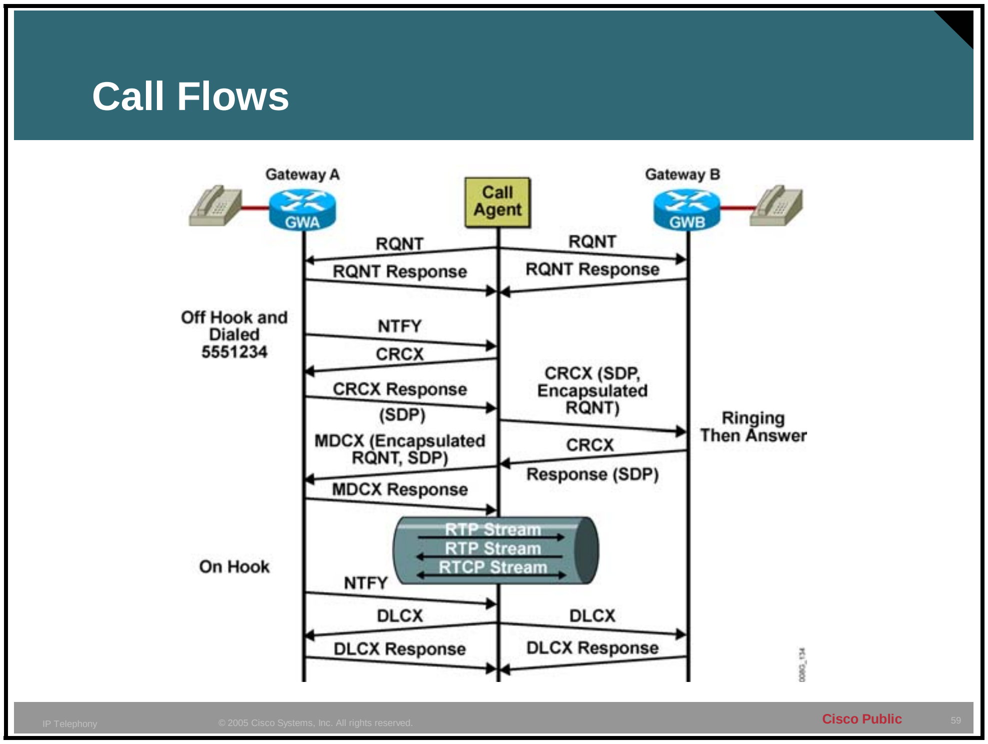
A call agent uses control commands or messages to direct its gateways and their operational behavior. Gateways use the following control commands in responding to requests from a call agent and notifying the call agent of events and abnormal behavior:

- **EndpointConfiguration (EPCF):** Identifies the coding characteristics of the endpoint interface on the line side of the gateway. The call agent issues the command.
- **NotificationRequest (RQNT):** Instructs the gateway to watch for events on an endpoint and the action to take when they occur. The call agent issues the command.
- **Notify (NTFY):** Informs the call agent of an event for which notification was requested. The gateway issues the command.
- **CreateConnection (CRCX):** Instructs the gateway to establish a connection with an endpoint. The call agent issues the command.
- **ModifyConnection (MDCX):** Instructs the gateway to update its connection parameters for a previously established connection. The call agent issues the command.
- **DeleteConnection (DLCX):** Informs the recipient to delete a connection. The call agent or the gateway can issue the command. The gateway or the call agent issues the command to advise that it no longer has the resources to sustain the call.
- **AuditEndpoint (AUEP):** Requests the status of an endpoint. The call agent issues the command.
- **AuditConnection (AUCX):** Requests the status of a connection. The call agent issues the command.

- **RestartInProgress (RSIP):** Notifies the call agent that the gateway and its endpoints are removed from service or are being placed back in service. The gateway issues the command.

Call Flows

This topic illustrates and explains the interactions between a call agent and its associated gateways.



The figure illustrates a dialog between a call agent and two gateways. Although the gateways in this example are both residential gateways, the following principles of operation are the same for other gateway types:

1. The call agent sends a notification request (RQNT) to each gateway. Because they are residential gateways, the request instructs the gateways to wait for an off-hook transition (event). When the off-hook transition event occurs, the call agent instructs the gateways to supply dial tone (signal). The call agent asks the gateway to monitor for other events as well. By providing a digit map in the request, the call agent can have the gateway collect digits before it notifies the call agent.

2. The gateways respond to the request. At this point, the gateways and the call agent wait for a triggering event.

3. A user on gateway A goes off hook. As instructed by the call agent in its earlier request, the gateway provides dial tone. Because the gateway is provided with a digit map, it begins to collect digits (as they are dialed) until either a match is made or no match is possible. For the remainder of this example, assume that the digits *match* a digit map entry.

4. Gateway A sends a notify (NTFY) to the call agent to advise the call agent that a requested event was observed. The notify identifies the endpoint, the event, and, in this case, the dialed digits.

5. After confirming that a call is possible based on the dialed digits, the call agent instructs gateway A to create a connection (CRCX) with its endpoint.

6. The gateway responds with a session description if it is able to accommodate the connection. The session description identifies at least the IP address and UDP port for use in a subsequent RTP session. The gateway does not have a session description for the remote side of the call, and the connection enters a wait state.

7. The call agent prepares and sends a connection request to gateway B. In the request, the call agent provides the session description obtained from gateway A. The connection request is targeted to a single endpoint—if only one endpoint is capable of handling the call—or to any one of a set of endpoints. The call agent also embeds a notification request that instructs the gateway about the signals and events that it should now consider relevant. In this example, in which the gateway is residential, the signal requests ringing and the event is an off-hook transition.

Note The interaction between gateway B and its attached user has been simplified.

---

8. Gateway B responds to the request with its session description. Notice that gateway B has both session descriptions and recognizes how to establish its RTP sessions.

9. The call agent relays the session description to gateway A in a modify connection request (MDCX). This request may contain an encapsulated notify request that describes the relevant signals and events at this stage of the call setup. Now gateway A and gateway B have the required session descriptions to establish the RTP sessions over which the audio travels.

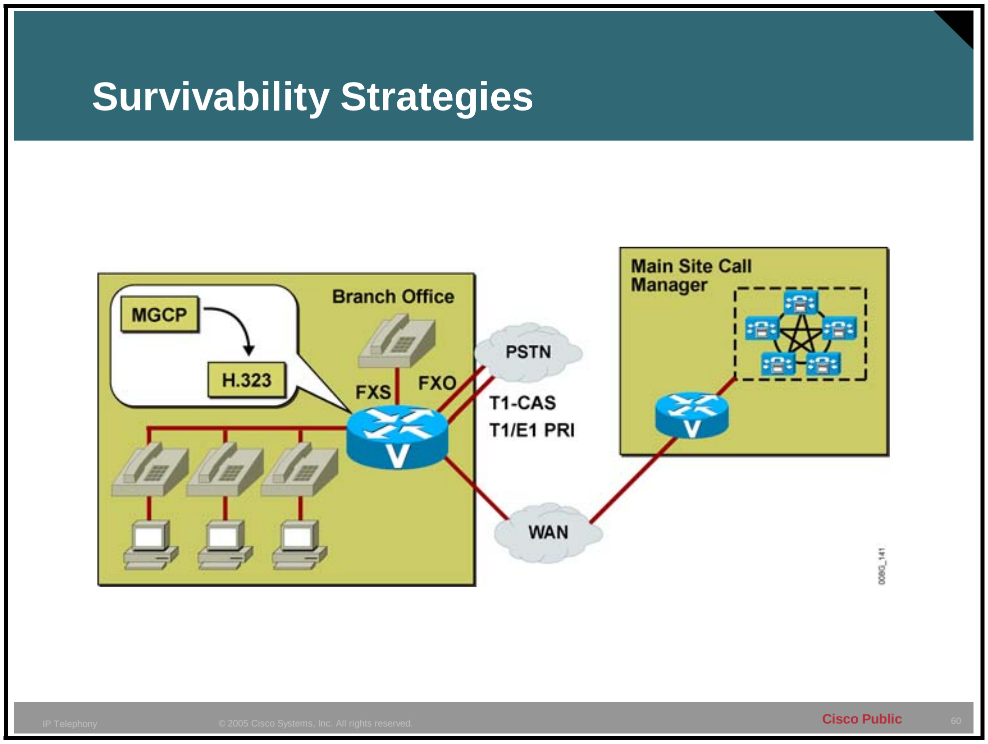
10. At the conclusion of the call, one of the endpoints recognizes an on-hook transition. In the example, the user on gateway A hangs up. Because the call agent requested the gateways to notify in such an event, gateway A notifies the call agent.

11. The call agent sends a delete connection (DLCX) request to each gateway.

12. The gateways delete the connections and respond.

Survivability Strategies

Maintaining high availability in an MGCP environment requires a design that accommodates the failure of a call agent. This topic describes two strategies for managing the loss of a call agent.



In the MGCP environment, the call agent controls all call setup processing on the IP and the telephony sides of a gateway. Because a gateway is associated with only one call agent at a time, if that call agent fails or is inaccessible for any reason, the gateway and its endpoints are left uncontrolled and, for all practical purposes, useless. Cisco Systems has developed two methods to handle lost communication between a call agent and its gateways: MGCP switchover and switchback and MGCP gateway fallback. These features operate in the following manner:

- **MGCP switchover and switchback:** MGCP switchover permits the use of redundant MGCP call agents. This feature requires two or more Cisco CallManager servers to operate as MGCP call agents. One Cisco CallManager server becomes the primary server and functions as the MGCP call agent. The other Cisco CallManager servers remain available as backup servers.

  The MGCP gateway monitors MGCP messages sent by the Cisco CallManager server. If traffic is undetected, the gateway transmits keepalive packets to which the Cisco CallManager server responds. If the gateway does not detect packets from the Cisco CallManager for a specified period, it tries to establish a new connection with a backup Cisco CallManager server.

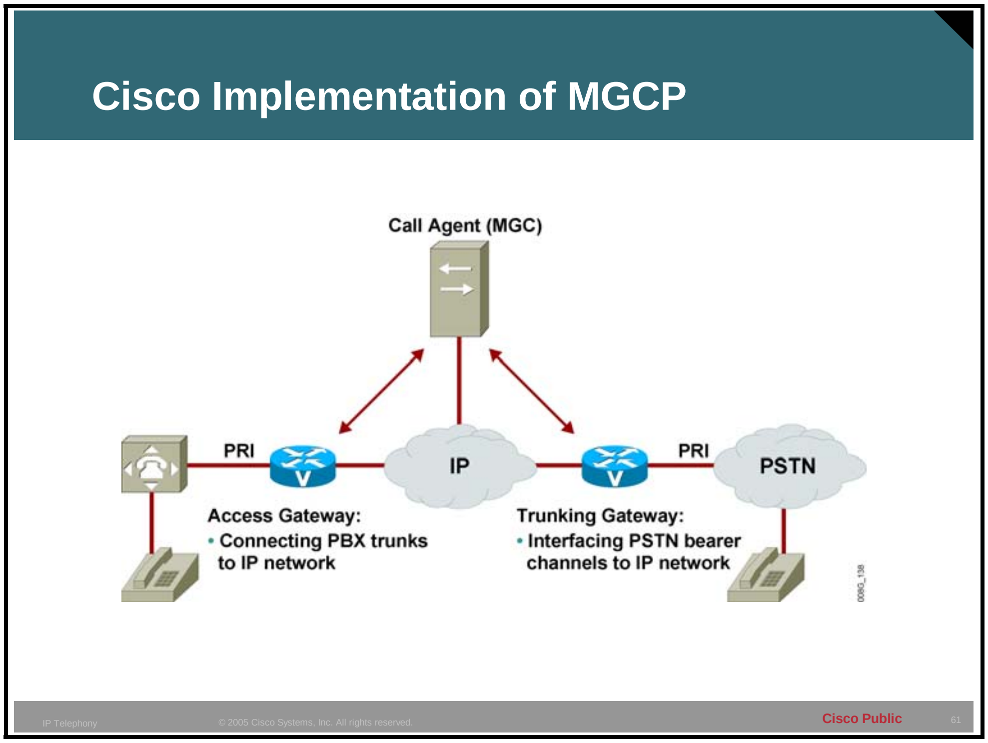
  You can configure a Cisco voice gateway to reestablish connection with the primary Cisco CallManager server when it becomes available again. This is the switchback function.

- **MGCP gateway fallback:** MGCP gateway fallback is a feature that improves the reliability of MGCP branch networks. A WAN link connects the MGCP gateway at the remote site to the Cisco CallManager at the central sites (the MGCP call agent). If the WAN link fails, the fallback feature keeps the gateway working as an H.323 gateway.

  MGCP gateway fallback works in conjunction with the Survivable Remote Site Telephony (SRST) feature. SRST allows Cisco gateways and routers to manage connections temporarily for Cisco IP Phones when a connection to a Cisco CallManager is unavailable.

Cisco Implementation of MGCP

This topic describes how Cisco implements MGCP.



Cisco provides support for MGCP gateways and the call agent in the following way:

- **Gateways:** Cisco implements MGCP trunk gateway and residential gateway support in the following devices:
  - Cisco voice-enabled routers (first available in Cisco IOS Release 12.1)
  - Cisco PGW 2200 PSTN gateways
  - Cisco Voice Gateway 224 (VG224)
  - Voice-enabled AS5*xx*0 access servers
  - BTS 10200 Softswitch

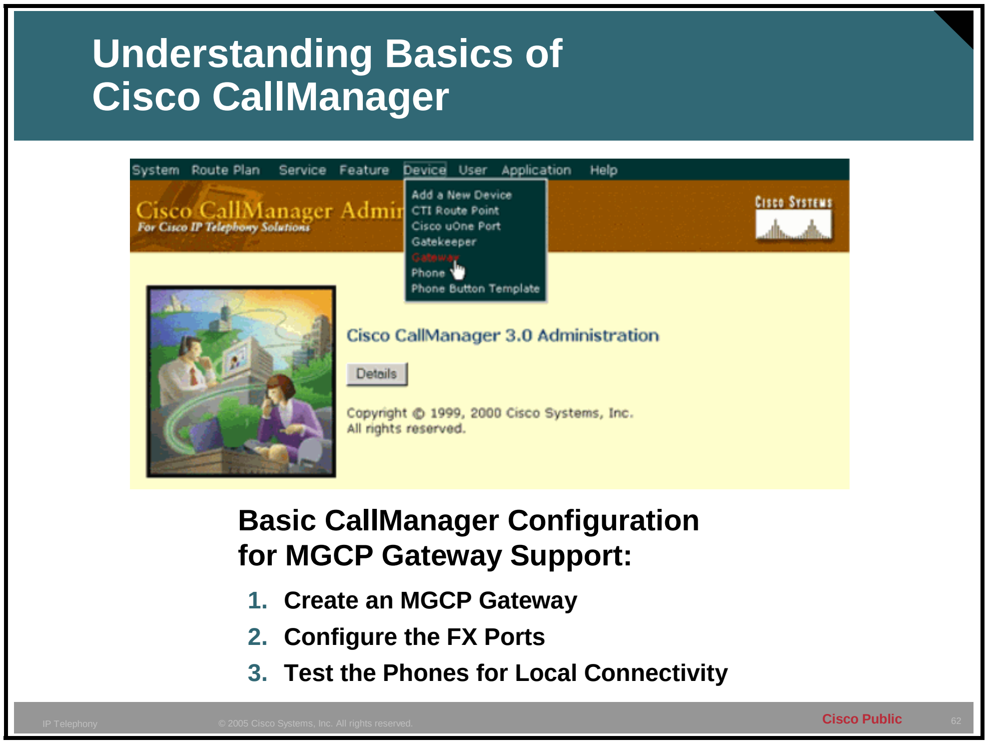
Note	Cisco CallManager interworking requires Cisco IOS Release 12.2.

- **Call agent:** Cisco implements call agent support in the following applications:
  - Cisco CallManager
  - BTS 10200 Softswitch

Note	Residential gateway and trunk gateway support does *not* include all analog and digital signaling types on the telephone interfaces. Check Cisco.com for an up-to-date list.

Understanding Basics of Cisco CallManager

This topic describes basic configuration of Cisco CallManager to support MGCP gateways. For detailed information regarding configuration of CallManager, refer to the *Cisco IP Telephony* (CIPT) course.



There are three basic tasks for configuring a Cisco CallManager for MGCP. The basic steps for each task are as follows:

- Task 1: Create an MGCP gateway

1. Use the Device Wizard to create an MGCP gateway. Select **Device** > **Gateway**.
2. Click **Add New Gateway**.
3. Select the appropriate gateway type (such as 26*xx*, 36*xx*, or 37*xx*).
4. Click **Next**.
5. For the **MGCP Domain Name**, use the actual host name you have assigned to the gateway. Specify the carrier module that the gateway has installed.
6. Click **Insert**.

- Task 2: Configure the FX ports

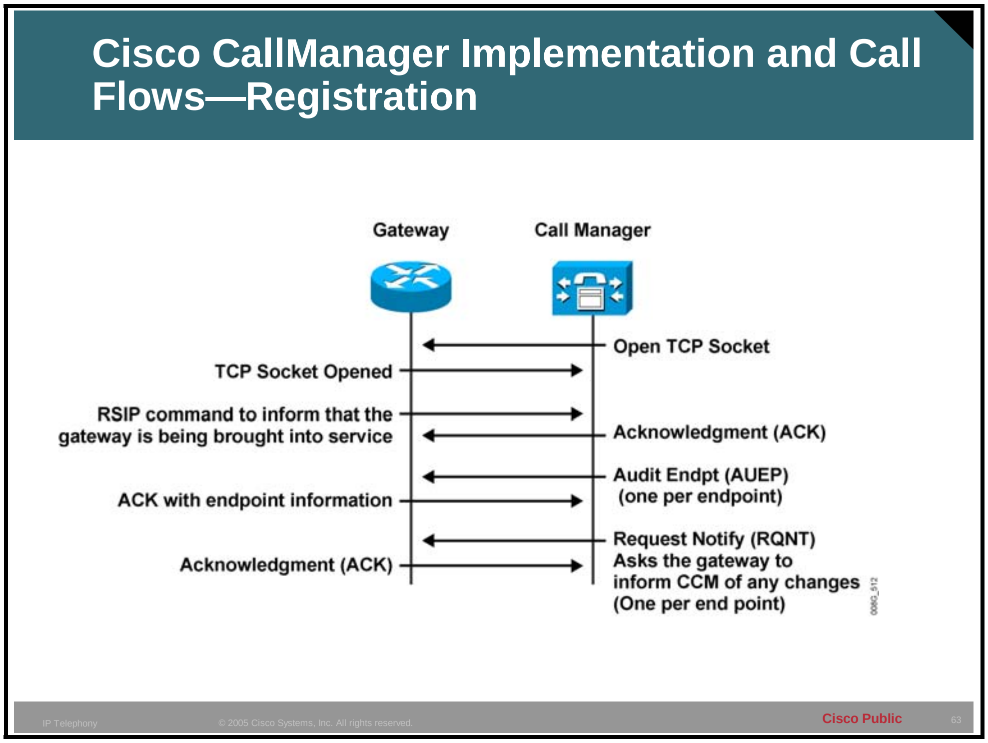
1. Identify the voice interface card (VIC) modules installed in the gateway.
2. Click **Update** to activate the changes.
3. The FXO and FXS ports appear at the bottom right of the next screen. These are also referred to as EndPoint Identifiers.
4. Select one of the ports.
5. Select the correct signaling type, **loop** or **ground start**.
6. Configure the parameters of the MGCP Member Configuration screen as required. For example, if this was an FXS port, you would be required to select loop or ground start signaling.
7. Click **Insert**.
8. For FXS ports only, click the **Add DN** text. Add the directory number (phone number) as appropriate, then click **Insert and Close**.
9. Select the **Back To MGCP Configuration** option.
10. Click **Reset Gateway**.

- Task 3: Test the phones for local connectivity

You should now have dial tone on the analog phones connected to the FXS ports. Try dialing from one FXS port to another. You should be able to make and receive calls between these ports.

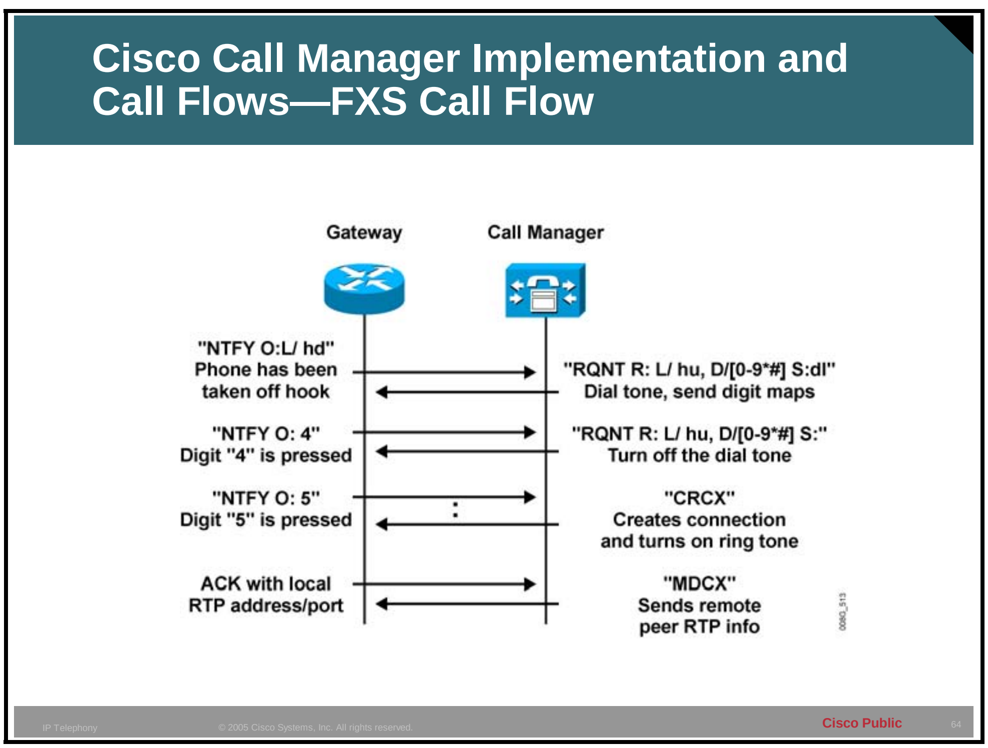
Registration and FXS Call Flow

Cisco CallManager implementation of MGCP uses specific command sequences to perform a variety of tasks. The figures are examples of how calls are made and how the gateways are registered.



Example: Registration

The figure describes how Cisco CallManager registers voice gateways in its database using MGCP. The ACK commands are standard TCP acknowledgments of the received command.

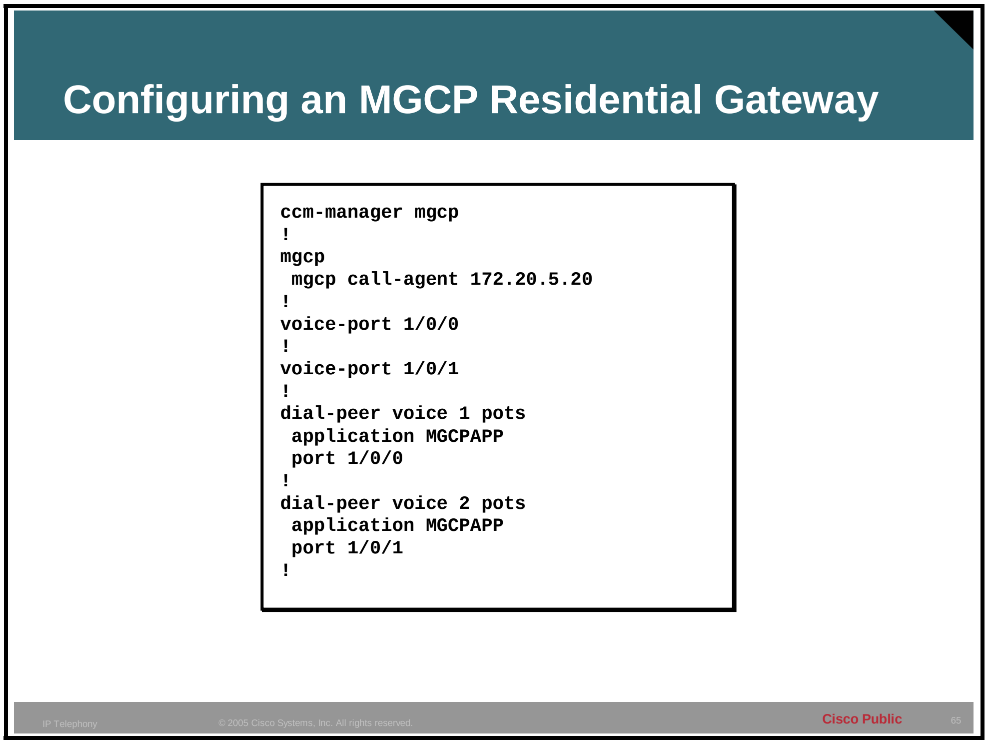


Example: FXS Call Flow

The figure shows a sample FXS call flow (dialing and connection).

Configuring MGCP

This topic illustrates the configuration commands that are required to implement MGCP residential gateway and trunk gateway capabilities on a Cisco router.



The figure highlights the commands required to configure an MGCP residential gateway.

MGCP is invoked with the **mgcp** command. If the call agent expects the gateway to use the default port (UDP 2427), the **mgcp** command is used without any parameters. If the call agent requires a different port, then the port must be configured as a parameter in the **mgcp** command; for example, **mgcp 5036** would tell the gateway to use port 5036 instead of the default port.

At least one **mgcp call-agent** command is required below the **mgcp** command. This command indicates the location of the call agent. The command identifies the call agent by an IP address or a host name. Using a host name adds a measure of fault tolerance in a network that has multiple call agents. When the gateway asks the DNS for the IP address of the call agent, the DNS may provide more than one address, in which case the gateway can use either one. If multiple instances of the **mgcp call-agent** command are configured, the gateway uses the first call agent to respond.

Other **mgcp** subcommands are optional.

Example: MGCP Residential Gateway Configuration

In the example, the configuration identifies the packages that the gateway expects the call agent to use when it communicates with the gateway. The last **mgcp** command specifies the default the gateway uses if the call agent does not share the capabilities. In this example, the command is redundant because the line package is the default for a residential gateway.

When the parameters of the MGCP gateway are configured, the active voice ports (endpoints) are associated with the MGCP. Dial peer 1 illustrates an **application mgcpapp** subcommand. This command binds the voice port (**1/0/0** in this case) to the MGCP. Also, notice that the dial peer does not have a destination pattern. A destination pattern is not used because the relationship between the dial number and the port is maintained by the call agent.

Configuring an MGCP Trunk Gateway

```
!
ccm-manager-mgcp
mgcp 4000
 mgcp call-agent 209.165.202.129 4000
!
controller T1 1/0
 framing esf
 clock source internal
 ds0-group 1 timeslots 1-24 type none service mgcp
!
controller T1 1/1
 framing esf
 clock source internal
 ds0-group 1 timeslots 1-24 type none service mgcp
!
voice-port 1/0:1
!
voice-port 1/1:1
!
```


The **ccm-manager-mgcp** command is required only if the call agent is a Cisco CallManager.

The second example illustrates the configuration of a trunk gateway.

Configuring trunk gateways requires the address or the name of the call agent, which is a requirement common to a residential gateway (RGW). The trunk package is the default for a trunk gateway and does not need to be configured. Again, other parameters are optional.

Example: Configuring MGCP Trunk Gateway

The figure illustrates commands for configuring a trunk gateway. Instead of using the **application mgcpapp** command in a dial peer, a trunk endpoint identifies its association with MGCP using the **service mgcp** parameter in the **ds0-group** controller subcommand. As always in MGCP, the call agent maintains the relationship between the endpoint (in this case a digital trunk) and its address.

Monitoring and Troubleshooting MGCP

Several **show** and **debug** commands provide support for monitoring and troubleshooting MGCP. This topic lists many useful **show** and **debug** commands.

Example: show Command

```
Router# show mgcp statistics

UDP pkts rx 8, tx 9
Unrecognized rx pkts 0, MGCP message parsing errors 0
Duplicate MGCP ack tx 0, Invalid versions count 0
CreateConn rx 4, successful 0, failed 0
DeleteConn rx 2, successful 2, failed 0
ModifyConn rx 4, successful 4, failed 0
DeleteConn tx 0, successful 0, failed 0
NotifyRequest rx 0, successful 4, failed 0
AuditConnection rx 0, successful 0, failed 0
AuditEndpoint rx 0, successful 0, failed 0
RestartInProgress tx 1, successful 1, failed 0
Notify tx 0, successful 0, failed 0
ACK tx 8, NACK tx 0
ACK rx 0, NACK rx 0
IP address based Call Agents statistics:
IP address 10.24.167.3, Total msg rx 8, successful 8,
failed 0
```


The figure illustrates the output of one of the **show** commands. The **show** and **debug** commands are valuable for examining the current status of the MGCP components and for troubleshooting. You should be familiar with the information provided from each command and how this information can help you.

The following **show** commands are useful for monitoring and troubleshooting MGCP:

- **show call active voice [brief]:** Displays the status, statistics, and parameters for all active voice calls. When the call is disconnected, this information is transferred to the history records.
- **show call history voice [last *n* | record | brief]:** Displays call records from the history buffer.
- **show mgcp:** Displays basic configuration information about the gateway.
- **show mgcp connection:** Displays details of the current connections.
- **show mgcp endpoint:** Displays a list of the voice ports that are configured for MGCP.
- **show mgcp statistics:** Displays a count of the successful and unsuccessful control commands (shown in the figure). You should investigate a high unsuccessful count.

The following **debug** commands are useful for monitoring and troubleshooting MGCP:

- **debug voip ccapi inout:** Shows every interaction with the call control API on the telephone interface and the VoIP side. Watching the output allows users to follow the progress of a call from the inbound interface or VoIP peer to the outbound side of the call. This debug is very active; you must use it sparingly in a live network.
- **debug mgcp [all | errors | events | packets | parser]:** Reports all **mgcp** command activity. You must use this debug to trace the MGCP request and responses.