CHAPTER 1

Cisco CallManager Architecture

A Cisco IP Communications network is a suite of components that includes Internet Protocol (IP) telephony communications. Cisco CallManager is a core component of a Cisco IP Communications network, the primary function of which is to serve as the call routing and signaling component for IP telephony.

The term *IP telephony* describes telephone systems that place calls over the same type of data network that makes up the Internet. Although strictly speaking, IP telephony primarily enables users to have voice conversations, CallManager also has the capability to enable users with PCs associated with their phones, users with video-only endpoints, and users with H.323-based video systems to have end-to-end video conversations.

Telephone systems have been around for more than 100 years. Small, medium, and large businesses use them to provide voice communications between employees within the business and to customers outside the business. The public telephone system itself is a very large network of interconnected telephone systems.

What makes IP telephony systems in general, and CallManager in particular, different is that they place calls over a computer network. The phones that CallManager controls plug directly into the same IP network as your PC, rather than into a phone jack connected to a telephones-only network.

Phone calls placed over an IP network differ fundamentally from those placed over a traditional telephone network. To understand how IP calls differ, you must first understand how a traditional telephone network works.

In many ways, traditional telephone networks have advanced enormously since Alexander Graham Bell invented the first telephone in 1876. Fundamentally, the traditional telephone network is about connecting a long, dedicated circuit between two telephones.

Traditional telephone networks fall into the following four categories:

- Key systems
- Private Branch Exchanges (PBX)

- Class 5 switches
- Class 1 to 4 switches

A *key system* is a small-scale telephone system designed to handle telephone communications for a small office of 1 to 25 users. Key systems can be either analog, which means they use the same 100-year-old technology of your home phone, or digital, which means they use the 30-year-old technology of a standard office phone.

A *PBX* is a corporate telephone office system. These systems scale from the small office of 20 people to large campuses (and distributed sites) of 30,000 people. However, because of the nature of the typical circuit-based architecture, no PBX vendor manufactures a single system that scales throughout the entire range. Customers must replace major portions of their infrastructure if they grow past their PBX limits.

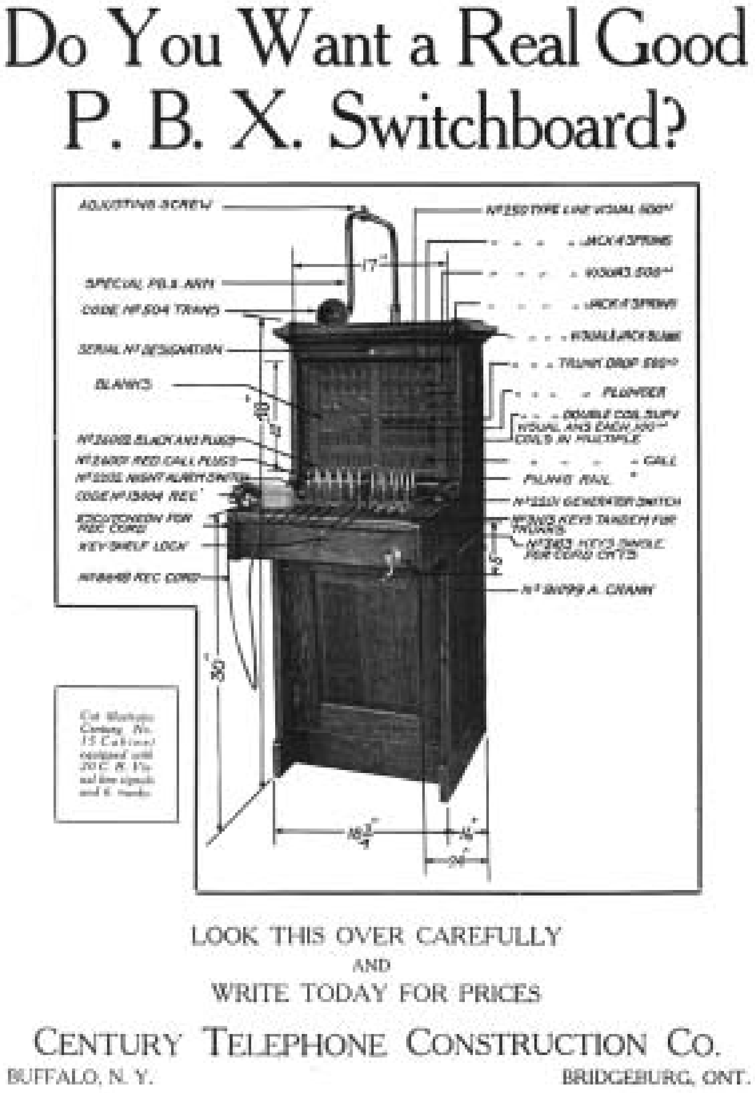
A *Class 5 switch* is a national telephone system operated by a local telephone company (called a local exchange carrier [LEC]). These systems scale from about 2000 to 100,000 users and serve the public at large.

Long distance companies and national carriers (called *interexchange carriers* [IEC or IXC]) use *Class 1 to 4 switches*. They process truly mammoth levels of calls and connect calls from one Class 5 switch to another.

Despite the large disparity in the number of users supported by these types of traditional networks, the core technology is circuit-based. Consider an old-time telephone operator. He or she sits in front of a large plugboard with hundreds of metal sockets and plugs. (Figure 1-1 shows a picture of an early PBX.) When a subscriber goes off-hook, a light illuminates on the plugboard. The operator plugs in the headset and requests the number of the party from the caller. After getting the number of the called party and finding the called party's socket, the operator checks to see whether the called party is busy. If the called party is not busy, the operator connects the sockets of the calling and called parties with a call cable, thus completing a circuit between them. The circuit provides a conduit for the conversation between the caller and the called party.

Today's central switching office—specifically, its call processing software—is simply a computerized replacement for the old-time telephone operator. Obeying a complex script of rules, the call processing software directs the collection of the number of the called party, looks for the circuit dedicated to the called party, checks to see whether the line is busy, and then completes the circuit between the calling and called parties.

Figure 1-1 *An Early PBX*



Do You Want a Real Good P. B. X. Switchboard?

LOOK THIS OVER CAREFULLY
AND
WRITE TODAY FOR PRICES

CENTURY TELEPHONE CONSTRUCTION CO.
BUFFALO, N. Y. BRIDGEBURG, ONT.

In the past, this circuit was an analog circuit from end to end. The voice energy of the speaker was converted into an electrical wave that traveled to the listener, where it was converted back again into a sound wave. Even today, the vast majority of residential telephone users still have an analog circuit that runs from their phone to the phone company’s central switching office, whereas digital circuits run between central switching offices.

This reliance on circuits characterizes traditional telephone systems and gives rise to the term *circuit switching*. A characteristic of circuit switching is that after the telephone system collects the number of the called party, and establishes the circuit from the calling party to the called party, this circuit is dedicated to the conversation between those calling and called parties. The resources allocated to the conversation cannot be reused for other purposes, even if the calling and called parties are silent on the call. Furthermore, if something happens to disrupt the circuit between the calling and called parties, they can no longer communicate.

Like the central switching office, CallManager is a computerized replacement for a human operator. CallManager, however, relies on packet switching to transmit conversations. *Packet switching* is the mechanism by which data is transmitted through the Internet, which encapsulates packets according to the Internet Protocol (IP). Web pages, e-mail, and instant messaging are all conveyed through the fabric of the Internet by packet switching. The term *voice over IP (VoIP)* specifically refers to the use of packet switching using IP to establish voice communications between IP-enabled endpoints on LANs and IP WANs, as well as the Internet (although CallManager is generally not deployed in configurations that route voice traffic over the Internet).

In packet switching, information to be conveyed is digitally encoded and broken down into small units called *packets*. Each packet consists of a header section and the encoded information. Among the pieces of header information is the network address of the recipient of the information. Packets are then placed on a router-connected network. Each router looks at the address information in each packet and decides where to send the packet. The recipient of the information can then reassemble the packets and convert the encoded data back into the original information.

Packet switching is more resilient to network problems than circuit switching because each packet contains the network address of the recipient. If something happens to the connection between two routers, a router with a redundant connection can forward the information to a secondary router, which in turn looks at the address of the recipient and determines how to reach it. Furthermore, if the sender and recipient are not communicating, the resources of the network are available to other users of the network.

In circuit-switched voice communications, an entire circuit is consumed when a conversation is established between two people. The system encodes the voice in a variety of manners, but the standard for voice encoding in the circuit-switched world is *pulse code modulation (PCM)*. Because PCM is the de facto standard for voice communications in the circuit-switched world, it comes as no surprise that a single voice circuit has been defined as the amount of bandwidth required to carry a single PCM-encoded voice stream.

Video communications require that significantly more information be sent from one end of a connection to another. In circuit-switched video communications, multiple circuits are usually simultaneously reserved for a single call to allow endpoints to exchange high-quality video.

An interesting complication involving voice encoding is introduced by packet-switched communications. Even if circuit-switched systems encode the voice stream according to a more efficient scheme, little incentive exists to do so, because, in most instances, a circuit is fully reserved no matter how little data you place on it. In the packet-switched world, however, a more efficient encoding scheme means that for the same amount of voice traffic, you can place smaller packets on the network, which in turn means that the same network can carry a larger number of conversations. As a result, the packet-switched world has given rise to several different encoding schemes called *codecs*.

Different types of voice encoding offer different benefits, but generally the more high fidelity the voice quality, the more bandwidth the resulting media stream requires. As the amount of bandwidth that you are willing to permit the voice stream to consume decreases, the more clever and complex the codec must become to maintain voice quality. The codecs that attempt to minimize the bandwidth required for a voice stream require complex mathematical calculations that attempt to predict in advance information about the volume and frequency level of an utterance. Such codecs are highly optimized for the spoken voice. Furthermore, these calculations are often so computationally intensive that software cannot perform them quickly enough; only specialized hardware with digital signal processors (DSP) can handle the computations efficiently. As a result, codec support often differs substantially from device to device in the VoIP network, because devices that do not incorporate DSPs can generally support only easy-to-encode and easy-to-decode codecs such as G.711.

Because not all network devices understand all codecs, an important part of establishing a packet voice call is the negotiation of a voice codec to be used for the conversation. This codec negotiation is a part of a packet-switched call that does not assume nearly the same importance on a circuit-switched call. Chapter 5, “Media Processing,” discusses codecs in more detail.

The information contained in a video call is also encoded using a particular codec; unlike voice codecs, however, of which a handful of variants must be interworked, for interactive video-conferencing, video in the IP world has widely adopted H.263 to encode end-to-end video information (although most products are moving toward H.264).

The rest of this chapter discusses the following topics:

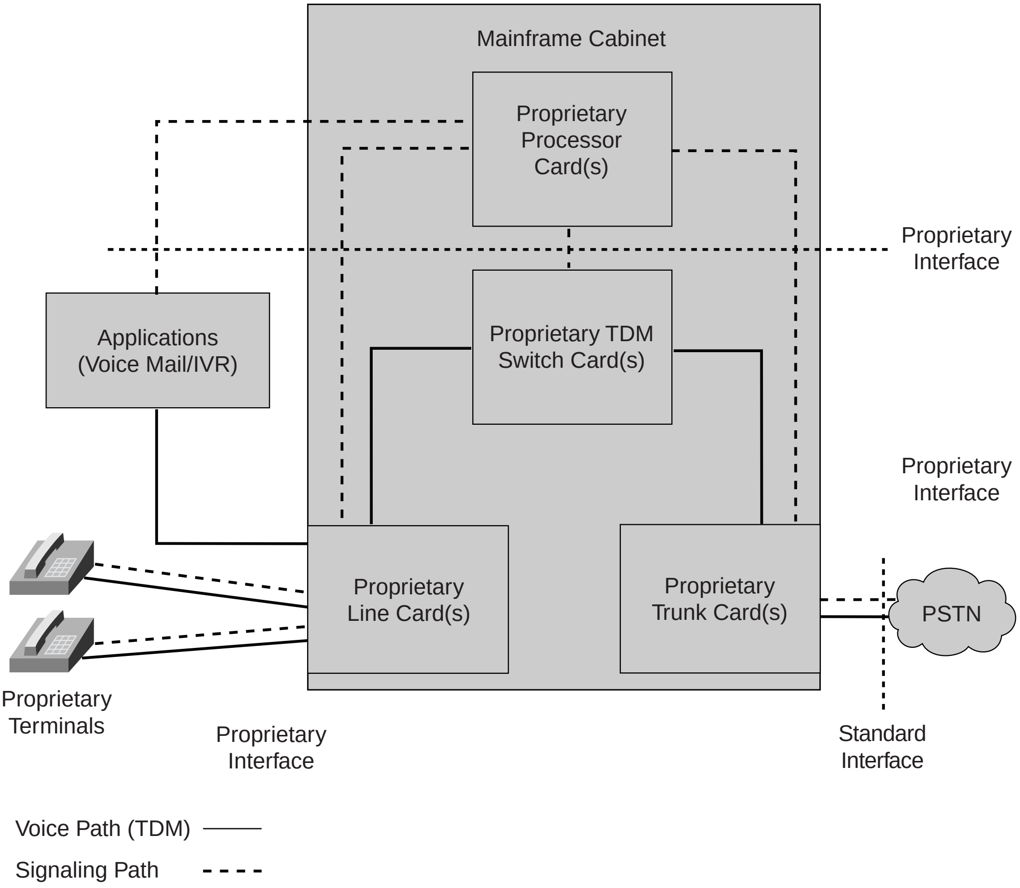
- Circuit-switched systems
- Cisco IP Communications networks
- Enterprise deployment of CallManager clusters

Circuit-Switched Systems

A *circuit-switched system* is typically a vertically integrated, monolithic computer system. A mainframe cabinet houses a proprietary processor, often along with a redundant processor, which in turn is connected with a bus to cabinets containing switch cards, line cards, and trunk cards.

Line cards control station devices (usually phones), and trunk cards control trunk devices (connections to other telephone systems). A wire runs from a station into a line card and carries both the call signaling and the encoded voice of the station device. Similarly, wires called *trunks* connect circuit-switched systems together with trunk cards. Line and trunk cards forward received call signaling to the call processing software, while the encoded media is available to the switch cards. Figure 1-2 demonstrates this architecture.

Figure 1-2 *Traditional Circuit-Switched Architecture*



Call Establishment in a Circuit-Switched Telephone System

Call establishment with a circuit-switched system consists of two phases: a session establishment phase and a media exchange phase.

The session establishment phase is the phase in which the telephone system attempts to establish a conversation. During this phase, the telephone system finds out that the caller wants to talk to someone, locates and alerts the called party, and waits for the called party to accept the call. As part of the call establishment, the telephone system also establishes a circuit back from the telephone switch closest to the caller to the caller itself. This circuit permits the caller to hear a ringback tone in the earpiece of the handset and also ensures that, if the called party answers, the end-to-end circuit can be connected as quickly as possible. This optimization eliminates *clipping*, a condition that occurs when the called party speaks before the circuit is completely formed, causing the caller to miss the initial utterance.

As soon as the telephone system determines that the called party wants to take the call, it completes the end-to-end circuit between the caller and called user, which permits them to begin the media exchange phase. The media exchange phase is the phase in which the endpoints actually converse over the connection that the session establishment phase forges.

Session establishment is the purview of *call signaling protocols*. Call signaling protocol is just a fancy term for the methods that coordinate the events required for a caller to tell the network to place a call, provide the telephone number of the destination, ring the destination, and connect the circuits when the destination answers. The following represent just a sample of the dozens of call signaling protocols:

- Rudimentary indications that can be provided over analog interfaces
- Proprietary digital methods
- Various versions of ISDN Basic Rate Interface (BRI), which are implementations of ITU-T Q.931
- Various versions of ISDN Primary Rate Interface (PRI), which are also implementations of ITU-T Q.931
- Integrated Services User Part (ISUP), which is part of Signaling System 7 (SS7)

All of these protocols serve the purpose of coordinating the establishment of a communications session between calling and called users.

As part of the session establishment phase, the telephone network reserves and connects circuits from the caller to the called user. Circuit-switched systems establish circuits with commands to their switch cards. Switch cards are responsible for bridging the media from one line or trunk card to another card in response to directives from the call processing software.

After a circuit-switched system forges an end-to-end connection, the end devices (also called *endpoints*) can begin the media exchange phase. In the media exchange phase, the endpoints encode the spoken word into a data stream. By virtue of the circuit connection, a data stream encoded by one endpoint travels to the other endpoint, which decodes it.

One feature to note is that in a circuit-switched system, the telephone network's switches are directly involved in both the call signaling and the media exchange. The telephone system must process the events from the caller and called user as part of the session establishment, and then it issues commands to its switch cards to bridge the media. Both the call signaling and the media follow the same path.

Call signaling protocols sometimes embed information about the voice-encoding method to be used to ensure that the endpoints communicate using a common encoding scheme. For voice communications, however, this media negotiation does not assume the importance it does in a packet-based system, in which endpoints generally have more voice-encoding schemes from which to choose.

In summary, a circuit-switched system goes through the following steps (abstracted for clarity) to establish a call:

Step 1 **Call signaling**—Using events received from the line and trunk cards, the telephone system detects an off-hook event and dialed digits from the caller, uses the dialed digits to locate a destination, establishes a circuit between a ringback tone generator and the caller, offers the call to the called user, and waits for the called user to answer. When the called user answers, the telephone system fully connects a circuit between the caller and called user.

Step 2 **Media exchange**—By virtue of their connected circuit, the calling and called users can converse. The calling user's phone encodes the caller's speech into a data stream. The switch cards in the telephone system forward the data stream along the circuit until the called user's phone receives and decodes it. Both the call signaling and the media follow a nearly identical path.

Cisco IP Communications Networks

A Cisco IP Communications network is a packet-based system. CallManager is a member of a class of systems called *softswitches*. In a softswitch-based system, the call signaling components and device controllers are not separated by a hardware bus running a proprietary protocol but instead are separate boxes connected over an IP network and talking through open and standards-based protocols.

CallManager provides the overall framework for communication within the corporate enterprise environment. CallManager handles the signaling for calls within the network and calls that originate or terminate outside the enterprise network. In addition to call signaling, CallManager provides call feature capabilities, the capability for voice mail interaction, and an application programming interface (API) for applications. Among such applications are Cisco Unity, Cisco IP Communicator, Cisco IP Contact Center (IPCC) Enterprise edition, Cisco CallManager Attendant Console, Cisco IP Manager Assistant (IPMA), Cisco Emergency Responder (CER), Cisco Personal Assistant (PA), Cisco MeetingPlace, Cisco IP Queue Manager, and a variety of third-party applications.

A Cisco IP Communications network is by nature more open and distributed than a traditional telephone system. It consists of a set number of servers that maintain static provisioned information, provide initialization, and process calls on behalf of a larger number of client devices. Servers cooperate with each other in a manner termed *clustering*, which presents administrators with a single point of provisioning, offers users the illusion that their calls are all being served by the same CallManager node, and enables the system to scale and provide reliability.

The remainder of this section discusses the following topics:

- "CallManager History" presents a short history of CallManager.
- "Cisco-Certified Servers for Running Cisco IP Communications" describes the Windows 2000 servers that CallManager runs on.
- "Windows 2000 Services and Tomcat Services on Cisco IP Communications Servers" presents the services that run on the server devices in a Cisco IP Communications network.
- "Client Devices That CallManager Supports" presents the station, trunking, and media devices that CallManager supports.
- "Call Establishment in a Cisco IP Communications Network" describes how a Cisco IP Communications system places telephone calls.
- "Cisco IP Communications Clustering" describes the concept of clustering servers in a Cisco IP Communications system.

CallManager History

There have been several releases of the software that would become CallManager release 4.1. It started in 1994 as a point-to-point video product, but it was recast as an IP-based telephony system in 1997. By 2004, CallManager could support, via multiple clusters, hundreds of thousands of users with a full suite of enterprise-class features.

1994—Multimedia Manager

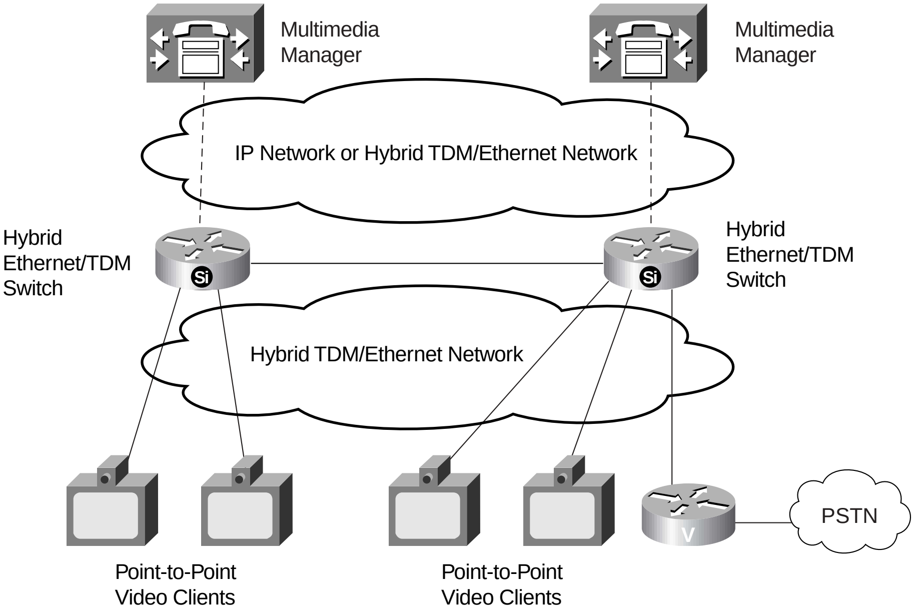
The application that would become CallManager release 4.1 began in 1994 as Multimedia Manager 1.0. Multimedia Manager was the signaling controller for a point-to-point video product. Multimedia Manager was developed under HP-UX in the language SDL-88.

Specification and Description Language (SDL) is an International Telecommunication Union (ITU)-standard (Z.100) graphical and textual language that many telecommunications specifications use to describe their protocols. An SDL system consists of many independent state machines, which communicate with other state machines solely through message passing and are thus object-oriented. Furthermore, because SDL is specifically designed for the modeling of real-time behavior, it is extremely suitable for call processing software.

Although Multimedia Manager 1.0 was developed in HP-UX, it was produced to run on Microsoft Windows NT 3.51. Each Multimedia Manager server served only as a call signaling source and destination. Multimedia Manager 1.0 managed connections by sending commands to network hubs, which contained the matrix for the video connections. Each hub contained 12 hybrid Ethernet/time-division multiplexing (TDM) ports. Each port could serve either a PC running videoconferencing software or a subhub that managed four PRI interfaces for calls across the public network. In addition, hubs could be chained together using hybrid Ethernet/TDM trunks. At that point in time, the software was somewhat of a hybrid system; Multimedia Manager, running on a Microsoft Windows NT Server 3.51, handled the call signaling and media control over IP like a softswitch, but the media connections were still essentially circuit-based in the network hubs.

Figure 1-3 depicts CallManager as it existed in 1994.

Figure 1-3 *CallManager in 1994*


Multimedia Manager
Multimedia Manager
IP Network or Hybrid TDM/Ethernet Network
Hybrid Ethernet/TDM Switch
Si
Hybrid Ethernet/TDM Switch
Si
Hybrid TDM/Ethernet Network
V
PSTN
Point-to-Point Video Clients
Point-to-Point Video Clients

1997—Selsius-CallManager

Although Multimedia Manager 1.0 worked wonderfully, by 1997 it was clear that Multimedia Manager was not succeeding in the marketplace. Customers were reluctant to replace their Ethernet-only network infrastructure with the hybrid Ethernet/TDM hubs required to switch the bandwidth-hungry video applications. At that point, Multimedia Manager 1.0 changed from a videoconferencing solution to a system designed to route voice calls over an IP network. Unlike the hybrid solution, which required intervening hubs to connect a virtual circuit between endpoints, media signaling traveled over the IP infrastructure directly from station to station. In other words, the system became a packet-switched telephone system.

The change required the development of IP phones and IP gateways. The database, which had been a software application running under Windows NT, became a set of web pages connected to a Microsoft Access database. The new interface permitted administrators to modify the network configuration from any remote machine's web browser.

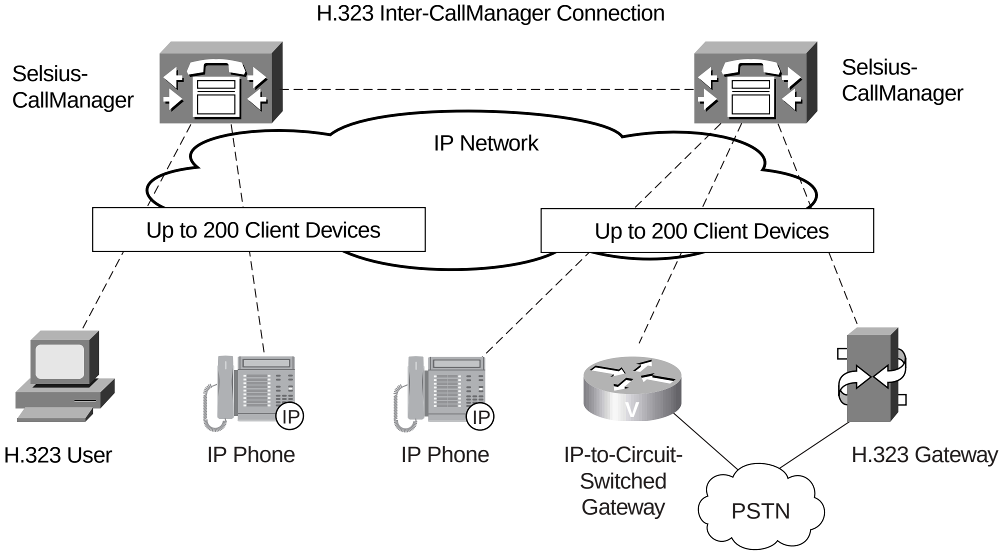
The call processing software changed, too. It incorporated new code to control the IP phones and gateways. For this purpose, the Skinny Client Control Protocol (SCCP) and Skinny Gateway Control Protocol (SGCP) were invented. In addition, the software supported Microsoft NetMeeting, an application that uses the H.323 protocol to support PC-to-PC packet voice calls.

At the same time, the call processing software had finally outgrown the SDL development tools. To ensure that the code base could continue to grow, the pure SDL code was converted into an SDL application engine based on C++ that duplicated all of the benefits that the previous pure SDL environment had provided.

Selsius-CallManager 1.0 was born. It permitted SCCP station-to-station and station-to-trunk calls. Each Selsius-CallManager supported 200-feature phones with features such as transfer and call forward.

Figure 1-4 depicts CallManager as it existed in 1997.

Figure 1-4 *CallManager in 1997*



2000—Cisco CallManager Release 3.0

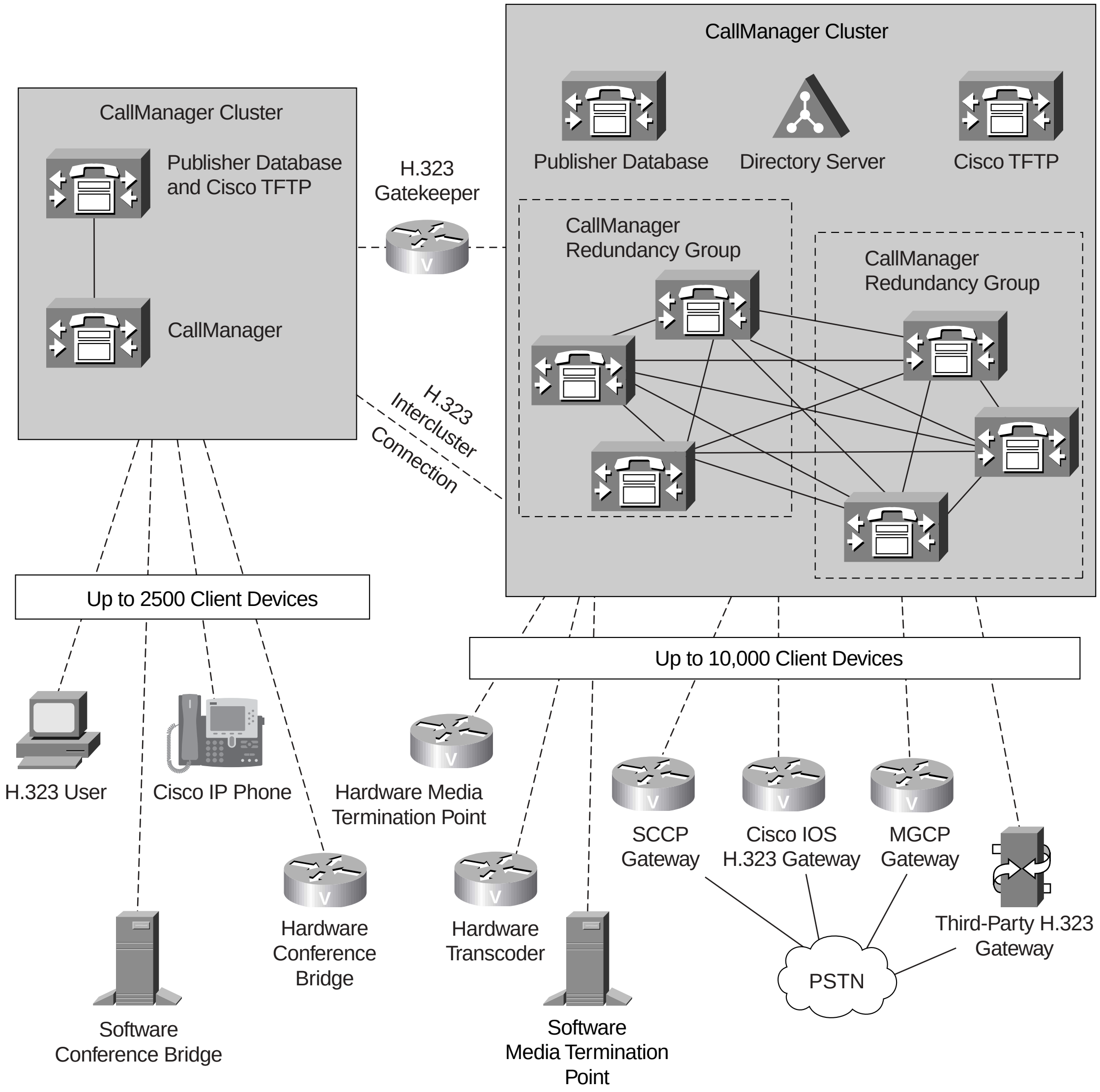
CallManager received a great deal of attention from the marketplace. By 1998, Selsius-CallManager 2.0 had been released, and Cisco Systems, Inc., had become interested in the potential of the product.

After acquiring the CallManager product as a result of its acquisition of Selsius Systems in 1998, Cisco concentrated on enhancing the product. Cisco also simultaneously undertook a huge design and re-engineering effort to provide both scalability and redundancy to the system. Clustering was introduced, and the SDL engine became the Signal Distribution Layer (SDL) engine, which permits the sending of signals directly from one CallManager to another. A redundancy scheme allowed stations to connect to any CallManager in a cluster and operate as if they were connected to their primary CallManager. Support for Media Gateway Control Protocol (MGCP) was added, as was the Cisco IP Phones 7910, 7940, and 7960, which provided a large display, softkeys (virtual buttons on the phone's display), and access to voice mail, phone settings, network directories, and services.

By mid-2000, Cisco CallManager release 3.0 was complete. It permitted feature-rich calls between H.323 stations and gateways, MGCP gateways, and SCCP stations and gateways. Each cluster supported up to 10,000 endpoints, and multiple cluster configurations permitted the configuration of up to 100,000 endpoints.

Figure 1-5 depicts CallManager release 3.0.

Figure 1-5 *CallManager in 2000*


CallManager Cluster
Publisher Database and Cisco TFTP
CallManager
H.323 Gatekeeper
H.323 Intercluster Connection
CallManager Cluster
Publisher Database
Directory Server
Cisco TFTP
CallManager Redundancy Group
CallManager Redundancy Group
Up to 2500 Client Devices
Up to 10,000 Client Devices
H.323 User
Cisco IP Phone
Hardware Media Termination Point
SCCP Gateway
Cisco IOS H.323 Gateway
MGCP Gateway
Third-Party H.323 Gateway
Hardware Conference Bridge
Hardware Transcoder
PSTN
Software Conference Bridge
Software Media Termination Point

2001—Cisco CallManager Release 3.1

CallManager release 3.1 built on the foundation of CallManager 3.0. The platform supported more gateway devices and station devices, added enhancements to serviceability, and added more features. Among the specific enhancements were the following:

- Music on hold (MOH)
- Media resource devices available to the cluster, rather than to individual CallManager servers

- Support for digital interfaces on MGCP gateways
- Call preservation between IP phones and MGCP gateways on server failure
- Database support for third-party devices
- Extension mobility
- ISDN overlap sending and T1-CAS support in a variety of VoIP gateways
- Support for Extensible Markup Language (XML) and HTML applications in Cisco IP Phones
- Support for telephony applications through Telephony Application Programming Interface (TAPI) and Java TAPI (JTAPI) and JTAPI/TAPI call processing redundancy support

2001—Cisco CallManager Release 3.2

CallManager 3.2 was a small-scale release that improved the following areas:

- **Scalability**—Improvements to support up to 20,000 IP phone endpoints per cluster, to improve the number of simultaneous H.323 calls, and to permit CallManager to simultaneously connect to multiple voice messaging systems
- **Language**—Localization of end-user–visible interfaces, such as phones, end-user applications, gateways, and user-accessible configuration pages to U.K. English and many non-English language and tone sets
- **Supported devices**—Support for station-oriented analog gateways such as the VG224 and VG248, as well as the Cisco IP Phone 7905
- **Features**—Auto-answer at destination IP phone for hands-free intercom service, Automated Alternate Routing (AAR) to route calls over the Public Switched Telephone Network (PSTN) when network bandwidth is no longer available, the ability to drop the most recently joined conference participant from an Ad Hoc conference, consultation transfer from applications, and message waiting enhancements

2002—Cisco CallManager Release 3.3

Like CallManager 3.2, CallManager 3.3 was a reasonably small-scale release, but which improved the following areas:

- **Scalability**—Improvements to support up to 30,000 IP phones per cluster and hundreds of thousands of IP phones using multiple clusters with an H.323 gatekeeper
- **H.323 support**—Improved ability to support H.323 gatekeeper-controlled connections between CallManager nodes and better scalability and redundancy through support for multiple gatekeepers and alternate H.323 gateways

- **Application improvements**—Support for Cisco IP Manager/Assistant and Cisco Call Back on Busy applications
- **QSIG support**—Support for basic call and line identification services using QSIG, a protocol designed to foster feature transparency between different PBXs
- **Feature improvements**—Distinctive ring per line appearance and configurable call waiting tones for consecutive calls

2004—Cisco CallManager Release 4.0

CallManager 4.0 was a large-scale release that focused quite strongly on features. Chief among the feature changes was a fundamental change in the way that Cisco IP Phones could manage calls. Prior to CallManager 4.0, Cisco IP Phones abided by two main restrictions:

- For any given line appearance, a Cisco IP Phone could have at most two calls, of which one could be actively streaming voice.
- When a Cisco IP Phone was actively streaming voice, other Cisco IP Phones that shared a directory number with the active Cisco IP Phone could not place or receive calls on the shared directory number (although their other directory numbers, if any, could be used to place and receive calls).

In CallManager release 4.0, Cisco IP Phones are no longer restricted to at most two calls per line appearance. Instead, the maximum number of calls per line appearance is configurable, although phones are still restricted to at most one actively streaming call. (An exception is models such as the Cisco IP Phone 7905, 7910, and 7912, which lack a display that would permit a user to efficiently manage more than two calls—these devices are still limited to, at most, two calls.)

Furthermore, in CallManager 4.0, devices that share line appearances are no longer restricted from placing and receiving calls if other devices that share the directory number are actively streaming voice on a call. A phone can continue to place and receive calls until it reaches the maximum threshold (up to 200 calls) configured by the system administrator.

In addition to continuing to support end-user features provided by earlier releases (transfer, Ad Hoc conference, Meet-Me conference, drop the last conference party, call park, call pickup, group call pickup, call back on busy, redial, speed dials, and others), CallManager 4.0 added the following features to Cisco IP Phones:

- **Call join**—Allows a user to select several calls from the same line on a Cisco IP Phone and conference them all at once.
- **Direct transfer**—Allows a user to select two calls from the same line on a Cisco IP Phone and transfer (connect) them together.

- **Barge and cBarge**—Allow a user at one IP phone (the “barger”) to automatically conference himself or herself into a call with two other conversing parties, one of which shares a line with the barger. cBarge relies on an external conference bridge resource; barge mixes the voice on IP phones that contain a built-in bridge, namely the Cisco IP Phones 7940, 7941, 7960, 7961, 7970, and 7971.
- **Privacy**—Allows a user at one Cisco IP Phone to prevent other users who share a line appearance from viewing the connected name and number identification of parties with which he or she is conversing.
- **Abbreviated dialing**—Permits a user to quickly dial preconfigured numbers by entering a one- or two-digit index code that represents the speed dial number.
- **Conference drop any party**—Permits a user who has created a conference to select from a list of currently connected parties and drop one from the conference.
- **Immediate diversion**—Permits a user who is receiving or already conversing on a call to divert the caller to the diverter’s voice mailbox.
- **Malicious call identification**—Permits a user to press a button on an active or recently terminated call to notify the system administrator (and service provider) that a harassing or threatening call has been received.
- **Multilevel Precedence and Preemption**—Permits users to preempt lower priority calls already occurring at the called number with calls designated as higher priority. This feature is used primarily by the military.
- **Hunt groups**—Native hunt group capability in CallManager. Hunt groups support broadcast (ring all members), top down, circular, and longest idle hunting. A ring no answer timer can be applied to determine the time to wait before proceeding to the next point.

You can learn more about these features in Chapter 3, “Station Devices.” Learn more about hunt groups in Chapter 2, “Call Routing.”

In addition to focusing on features, CallManager 4.0 improved the following areas:

- **QSIG support**—Addition of QSIG supplementary services for call diversion and call transfer to permit display updates when calls across multiple PBXs are transferred or forwarded and to support delivery of message waiting indications between PBXs
- **Video support**—Addition of media control capabilities to support the establishment of video calls from either video-enabled Cisco IP Phones, third-party SCCP video endpoints, H.323-based video endpoints, and audio-only Cisco IP Phones that have an associated PC for video display

- **Security**—Support for signaling authentication of Cisco IP Phones to prevent rogue phones from registering with CallManager or impersonating other devices, support for Cisco IP Phone signaling, support for media encryption between phones, and integrated support for multiple levels of administrator access
- **SIP support**—Addition of the Session Initiation Protocol (SIP) call signaling protocol specifically for connections to phone systems outside of a CallManager cluster

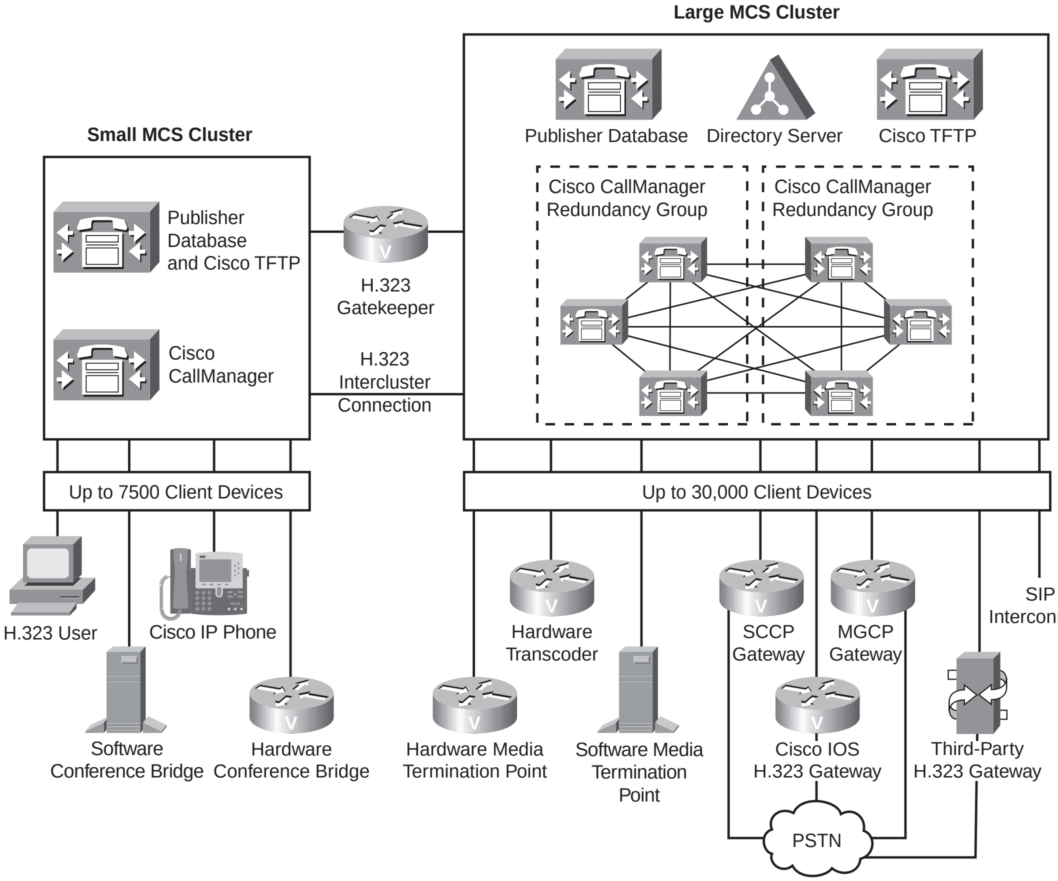
2004—Cisco CallManager Release 4.1

CallManager 4.1 continues to focus on support for new features. The following list summarizes the new additions:

- **QSIG enhancements**—CallManager continued implementing the QSIG protocol for feature transparency with other PBXs.
  - **Path replacement** optimizes the path between two parties to remove circuit hairpins that form when one party transfers a call to another party.
  - **Call forward by rerouting** prevents circuit hairpins from forming when a phone on one system forwards a call to a phone on another system.
  - **Call back on no reply and call back on busy** allow a caller to set a monitor on a called number that has not accepted a call and receive a prompt to redial the called number when it becomes available.
  - **Called name** allows a calling user to see the name of the party he or she is calling, even when the called party is served by a different call agent.
- **Dialed number analyzer** is a service that allows you to enter dial strings on behalf of calling devices and analyze how the call may route.
- **Time of day routing** provides a flexible mechanism by which you can activate and deactivate route partitions according to a schedule. Chapter 2, describes this feature in more detail.
- **Client matter codes** allow you to define post-dial strings that are associated with specific clients that users can dial to attribute the cost of the call to the client. The codes show up in call detail records to achieve billing traceability.
- **Forced authorization codes** allow you to define post-dial strings that users must dial to reach their destinations.

Figure 1-6 depicts CallManager release 4.1.

Figure 1-6 *CallManager in 2005*


Large MCS Cluster
Publisher Database
Directory Server
Cisco TFTP
Cisco CallManager Redundancy Group
Cisco CallManager Redundancy Group
Small MCS Cluster
Publisher Database and Cisco TFTP
Cisco CallManager
H.323 Gatekeeper
H.323 Intercluster Connection
Up to 7500 Client Devices
Up to 30,000 Client Devices
H.323 User
Cisco IP Phone
Software Conference Bridge
Hardware Conference Bridge
Hardware Transcoder
Hardware Media Termination Point
Software Media Termination Point
SCCP Gateway
MGCP Gateway
Cisco IOS H.323 Gateway
SIP Interconnect
Third-Party H.323 Gateway
PSTN

Cisco-Certified Servers for Running Cisco IP Communications

CallManager and its associated services run on a Windows 2000 server. Because voice applications are so critical to an enterprise's function, however, Cisco Systems requires that CallManager be installed only on certified server platforms.

Cisco Systems provides a suite of certified servers called Media Convergence Servers (MCS). In addition to these servers, Cisco allows users to install Cisco IP Communications software on servers offered by HP and IBM. Customer-provided servers must match exact server configurations provided by Cisco, because any deviations from the specifications might result in an incomplete install and an unsupported system.

NOTE At times, this book uses the term *server* and, at other times, it uses the term *node*, particularly when describing CallManager clustering.

A CallManager cluster consists of networked *servers* running a variety of services that together provide an enterprise VoIP system. Some of these servers in the cluster are generally dedicated to the CallManager database or TFTP service. Others run CallManager, the call processing component of a Cisco IP Communications system.

This book uses the term *node* to refer specifically to the servers in a CallManager cluster that are running the CallManager service. It's not uncommon to read a sentence referencing both *nodes* and *servers*. For example, it's consistent to state both that a CallManager cluster can consist of a maximum of 20 servers and that it can consist of a maximum of 8 nodes, because the 12 non-call processing servers handle services such as the Publisher database, TFTP, Cisco IP Voice Media Streaming App, and applications.

The current list, as of the release of 4.1, of MCSs that Cisco ships are as follows:

- Cisco MCS 7815I-2000
- Cisco MCS 7825H-3000
- Cisco MCS 7835H-3000
- Cisco MCS 7835I-3000
- Cisco MCS 7845H-3000
- Cisco MCS 7845I-3000

In addition to MCS, users can build Cisco IP Communications systems based off of the following HP and IBM platforms. (You can find the latest system information and specific parts lists on Cisco.com at http://www.cisco.com/go/swonly.)

- Compaq DL320-G2 Pentium 4 3060 MHz
- Compaq DL320 Pentium III 800 MHz
- Compaq DL320 Pentium III 1133 MHz
- Compaq DL320-G2 Pentium 4 2.26 GHz
- Compaq DL380 Pentium III 1000 MHz
- Compaq DL380 G2 Pentium III 1266 MHz
- HP DL380-G3 Xeon 3060 MHz

- HP DL380-G3 Dual Xeon 3.06 GHz
- HP DL380-G2 Pentium III 1400 MHz
- HP DL380-G2 Pentium III 1400 MHz
- HP DL380-G3 Xeon 2400 MHz
- HP DL380-G2 Dual Pentium III 1400 MHz
- HP DL380-G3 Dual Xeon 2.4 GHz
- IBM xSeries 306 Single-Processor 3.06 GHz
- IBM xSeries 346 Single-Processor 3.4 GHz
- IBM xSeries 346 Dual-Processor 3.4 GHz

Cisco MCS ships with an installation disk that contains all of the Windows 2000 services that are required to create a working IP telephony network. The HP and IBM servers are hardware-only; you must order a software-only version of CallManager (and the Windows 2000 installation) from Cisco to install on these servers.

Cisco IP Communications consists of a suite of applications that you can provision in numerous ways for flexibility. For example, although a server contains applications for managing the database, device initialization, device control, software conferencing, and voice mail, you might decide to reserve an entire server for just one of these functions in a large, differentiated Cisco IP Communications deployment. Servers that perform a sole function are called *dedicated servers*. For an overview of the services CallManager supports, see the section "Windows 2000 and Tomcat Services on Cisco IP Communications Servers."

The following list describes Cisco MCS 7800 series servers (two other servers, the MCS-7855-1500 and MCS-7865I-1500, are Cisco Unity-specific).

- **MCS-7815I-2000 server**—The only tower system that Cisco ships. It is suitable for smaller installations and can be configured to run CallManager, Unity, or Unity Bridge. This server can support up to 300 Cisco IP Phones. MCS 7815 server can only be deployed in the minimal cluster configuration, with one MCS running a Publisher database, Cisco TFTP, and backup CallManager and with the other MCS handling active call processing services.
- **MCS-7825H-3000 server**—A rack-mountable system, requiring a single rack space. This system can be configured to run CallManager, Cisco Conference Connection (CCC), Cisco Emergency Responder (CER), Cisco IPCC Express (Integrated Contact Distribution [ICD]), Cisco IP Interactive Voice Response (IP IVR), Cisco Personal Assistant, Cisco Queue Manager, and Cisco Unity Unified Messaging. This server can support up to 1000 IP phones or, via clustering, up to 4000 IP phones.

- **MCS-7835H-3000 and MCS-7835I-3000 servers**—Rack-mountable systems that require two rack spaces and have a single 3.06-GHz processor. These servers can be configured to run CallManager, Cisco Conference Connection (CCC), Cisco Emergency Responder (CER), Cisco IPCC Express (ICD), Cisco IP Interactive Voice Response (Cisco IP IVR), Cisco Personal Assistant, Cisco Queue Manager, and Cisco Unity Unified Messaging. These servers can support up to 2500 IP phones or, via clustering, up to 10,000 IP phones.

- **MCS-7845H-3000 and MCS-7845I-3000 servers**—Rack-mountable systems that require two rack spaces and have dual 3.06-GHz processors. These servers can be configured to run CallManager, Cisco Conference Connection (CCC), Cisco Emergency Responder (CER), Cisco Internet Service Node (ISN), Cisco IPCC Express (ICD), Cisco IP Interactive Voice Response (Cisco IP IVR), Cisco Personal Assistant, Cisco Queue Manager, and Cisco Unity Unified Messaging. These servers can support up to 7500 IP phones or, via clustering, up to 30,000 IP phones.

Windows 2000 and Tomcat Services on Cisco IP Communications Servers

Cisco IP Communications relies on several Windows 2000 services, of which Cisco CallManager is only one. Cisco IP Communications uses the Windows 2000 services described in Table 1-1.

Table 1-1 *Windows 2000 Services That Run on a Cisco IP Communications Server*

Service	**Description**
Cisco CallManager	Provides call signaling and media control signaling for up to 7500 devices. You can have up to eight instances of the CallManager service per cluster.
Cisco Certificate Authority Proxy Function	Manages security certificates for Cisco IP Phones such as the Cisco 7940 and 7960 that do not directly support installed certificates.
Cisco CTIManager	Provides support for the TAPI and JTAPI application interfaces.
Cisco IP Voice Media Streaming App	Provides media termination, RFC 2833 tone interworking, inband tone services for SIP, MOH, and G.711 media mixing capabilities.
Cisco Messaging Interface	Permits Simple Message Desk Interface (SMDI) communications to voice messaging systems over an RS-232 connection.
Cisco MOH Audio Translator	Converts any audio file format compatible with DirectShow and converts it to G.711, G.729a, and wideband codec for MOH to IP telephony endpoints.
Cisco RIS Data Collector	Collects serviceability information from all cluster members for improved administration.
Cisco Telephony Call Dispatcher	Allows users such as receptionists and attendants to receive and quickly transfer calls to other users in the organization; provides automated routing capabilities.

continues

Table 1-1 *Windows 2000 Services That Run on a Cisco IP Communications Server (Continued)*

Service	Description
Cisco TFTP	Provides preregistration information to devices, including a list of CallManager nodes with which the devices are permitted to register, firmware loads, and device configuration files.
Cisco Database Layer Monitor (provides database notification)	A change notification server and watchdog process that ensures that all Cisco IP Communications applications on a server are working properly.
Publisher database	Serves as the primary read-write data repository for all Cisco IP Communications applications in the cluster. The Publisher database replicates database updates to all Subscriber databases in the cluster.
Subscriber database	Serves as a backup read-only database for Cisco IP Communications applications running on the server, should the applications lose connectivity to the Publisher database.
Cisco CDR Insert	Periodically scans local call detail record (CDR) files logged by CallManager nodes and inserts them into the CDR database.
Cisco CTL Provider	Accepts connections from the CTL Client utility, which allows you to change the cluster security mode and update the cluster's Certificate Trust List (CTL).
Cisco Extended Functions	Provides the Quality Reporting Tool service, which allows users to report problems with their phone via the **QRT** softkey.
Cisco Serviceability Reporter	Generates a daily serviceability summary report for the cluster, including server performance, alerts generated by system, call activities, and other information.

While Table 1-1 indicates native Windows 2000 services that provide call-related services, Cisco IP Communications also supports applications that run as Java servlets hosted by the Apache plug-in Tomcat. Table 1-2 lists the Tomcat applications that Cisco IP Communications supports in the 4.1 release.

Table 1-2 *Tomcat Applications That Run on a Cisco IP Communications Server*

Name	Description
Cisco Web Dialer	Allows corporate directories to support click-to-dial functionality in which a user viewing a directory page can click a link to have his or her IP phone automatically call the selected person.
Cisco IP Manager Assistant	Provides an enhanced suite of services especially suited for managing the relationship between managers and assistants. This suite includes call filtering, immediate diversion, and send all calls functions.
Cisco Extension Mobility	Allows a user at a Cisco IP Phone to provide a user ID and password to log in to the phone and retrieve his or her extension and customized line settings.

Client Devices That CallManager Supports

In a Cisco IP Communications network, CallManager is the telephone operator, and it places calls on behalf of many different endpoint devices. These devices can be classified into the following categories:

- **Station devices**—Station devices are generally, but not always, telephone sets. CallManager offers a variety of sets, which it controls with SCCP.

  Cisco IP Phone 7902 is a cost-effective, single-line, entry-level station with no display.

  Cisco IP Phones 7905G and 7912G are single-line, entry-level phones (with a format different from the Cisco IP Phone 7910G) with a graphical display.

  Cisco IP Phone 7920 is a mobile 802.11b phone that enables voice communications over wireless LANs.

  Cisco IP Phone 7935 and 7936 are console speakerphones with softkey displays designed for use in conference rooms. They do not support inline power and do not have a switch for supporting an associated PC.

  Cisco IP Phone 7940G supports two line/feature buttons and offers a nine-line display with softkeys and status lines.

  Cisco IP Phone 7941G supports two line/feature buttons with lighted keys and offers a high-resolution display with softkeys and status lines.

  Cisco IP Phone 7960G supports six line/feature buttons and has the same display as the Cisco IP Phone 7940G.

  Cisco IP Phone 7961G supports up to six line/feature buttons with lighted keys and has the same display as the Cisco IP Phone 7941G.

  Cisco IP Phone 7970G offers eight line/feature buttons and an 11-line backlit, high-resolution color display with touch screen and additional softkeys.

  Cisco IP Phone 7971G-GE provides unconstrained bandwidth to desktop applica-tions via Gigabit Ethernet (GE) and features eight line/feature buttons and an 11-line backlit, high-resolution color display with touch screen and additional softkeys.

  Cisco IP Phone 7914 expansion modules can be added to Cisco IP Phone 7960G. Each expansion module adds 14 buttons and up to 2 modules can be added to a Cisco IP Phone.

  Station devices need not be physical handsets. CallManager also supports H.323 user clients, such as the following:

  - — NetMeeting, which runs as a software application on a user's PC
  - — Cisco IP Communicator, a software-based phone that connects to CallManager using SCCP
  - — Cisco IP SoftPhone, which connects to CallManager using the TAPI application interface

  Chapter 3 goes into more detail about station devices.

- **Gateway devices**—Gateways provide a bridge between two end users whose endpoints utilize different protocols. Gateways allow IP phones to interact with the billions of already deployed phones in the world.

  Gateway devices generally provide one of two types of interconnections. One type of interconnection is from one telephone system to another. This access can be from one network of CallManager nodes to another, from a CallManager network to a PBX or from a CallManager network to a public network such as a Class 4 or Class 5 switch. (But note that intercluster H.323 trunks provide an alternative for connecting CallManager networks together without requiring a gateway device.)

  Gateways do not necessarily need to provide access to other networks, however. Gateways can also be used to interwork VoIP directly with traditional telephones (POTS phones).

  CallManager controls gateways via three protocols: H.323, MGCP, and the legacy Skinny Gateway Control Protocol (SGCP). On their circuit interfaces, gateways provide both digital—for example, BRI, T1/E1 Channel Associated Signaling (CAS) and T1/E1 Primary Rate Interface (PRI)—and analog (the same type of telephone interface that probably runs into your home) interfaces.

  Cisco gateways fall into three general categories:

  - — Cisco IOS integrated routers are gateways that provide IP routing in addition to their gateway services. These can be viewed as IP routers that just happen to provide support for analog phones or for analog or digital trunk interfaces. Cisco IOS routers accept voice interface cards (VIC) and voice/WAN interface cards (VWIC) that can provide connectivity to the PSTN using many telephony protocols (as well as media services such as transcoding, media termination, and conference mixing).
  - — Cisco standalone voice gateways operate solely as end devices; they do not route IP traffic from network to network. The Cisco ATA 186, ATA 188, VG224, and VG248 provide CallManager with gateway services from its IP phones to analog phones or trunks.

  - — Cisco Catalyst voice gateway modules also operate solely as end devices. These modules are inserted into the Cisco Catalyst 6xxx chassis. Cisco Catalyst 6xxx can accept the Communication Media Module (CMM), the 6608 module, and the 6624 module. The CMM, in turn, can take port adapters that support the T1, E1, or FXS telephony interfaces.

  Chapter 4, "Trunk Devices," goes into more detail about trunk devices.

- **Media processing devices**—Media processing devices perform codec conversion, media mixing, and media termination functions. CallManager controls media processing devices using SCCP. Five types of media processing devices exist.

  - — **Transcoding resources**—These exist to perform codec conversions between devices that otherwise could not communicate because they do not encode voice conversations using a common encoding scheme. If CallManager detects that two endpoints cannot interpret each other's voice-encoding schemes, it inserts a transcoder into the conversation. Transcoders serve as interpreters. When CallManager introduces a transcoder into a conversation, it tells the endpoints in the conversation to send their voice streams to the transcoder instead of to each other. The transcoder translates an incoming voice stream from the codec that the sender uses into the codec that the recipient uses, and then forwards the voice stream to the recipient. The Catalyst 6xxx platform offers a blade that performs transcoding functions and the NM-HDV, NM-HDV2, and NM-HD-2VE modules support transcoding functions for IOS gateways.

  - — **Unicast conferencing devices**—These exist to permit Ad Hoc and Meet-Me conferencing. When an endpoint wants to start a multiple-party conversation, all the other parties in the conversation need to receive a copy of its voice stream. If several parties are speaking at once in a conversation, some component in the conversation needs to combine the independent voice streams present at a particular instant into a single burst of sound to be played through the telephone handset.

    Unicast conferencing devices perform the functions of both copying a conference participant's voice stream to other participants in the conference and mixing the voice streams into a single stream. When you initiate a conference, CallManager looks for an available Unicast conferencing device and dynamically redirects all participants' voice streams through the device. The Catalyst 6xxx platform offers a blade that performs mixing functions, and NM-HDV, NM-HDV2, and NM-HD-2VE modules support mixing functions for Cisco IOS gateways. In addition, the Cisco IP Voice Media Streaming App is a software application that can mix media streams encoded according to the G.711 codec.

  - — **Media termination point (MTP) resources**—These devices exist to allow users to invoke features such as hold and transfer, even when the person they are conversing with is using an H.323 endpoint such as NetMeeting. Devices that are only H.323v1-compatible do not tolerate interruptions in their media sessions very well. Attempts to place these devices on hold will cause them to terminate their active call. A media termination device serves as a proxy for these old H.323 devices and allows them to be placed on hold as part of feature operation.

CallManager also uses MTPs to interwork with SIP networks. SIP networks generally encode DTMF tones directly in the RTP stream using RFC 2833, while CallManager typically encodes tones directly in the signaling stream. An MTP can provide the interworking between these different types of tones as well as provide inband ringback when a Cisco IP Phone transfers a SIP caller.

The Catalyst 6xxx platform offers modules that perform media termination functions, and modules that provide media termination functions also exist for Cisco IOS routers. Furthermore, the Cisco IP Voice Media Streaming App is a software application that can perform media termination functions for calls that use the G.711 codec.

— **Music on Hold (MOH) resources**—These exist to provide users a music source when you place them on hold. When you place a user on hold, CallManager renegotiates the media session between the party you place on hold and the MOH device. For as long as you keep the user on hold, the MOH device transmits its audio stream to the held party. When you remove the user from hold, CallManager renegotiates the media stream between your device and the user.

— **Annunciator resources**—These exist to provide users audio announcements when error conditions occur such as preemption due to higher-priority calls, invalid dialed digit strings, or other problems CallManager encounters when placing calls.

Table 1-3 provides a comprehensive list of the Cisco IP Phones that CallManager supports.

Table 1-3 *Cisco IP Phones That CallManager Supports*

Name	Description
Cisco IP Phone 12SP+	Legacy phone with 12 feature buttons and 2-line text display
Cisco IP Phone 30VIP	Legacy phone with 30 feature buttons and 2-line text display
Cisco IP Phone 7902	Single-line appearance phone with no display
Cisco IP Phone 7905G	Single-line appearance phone with 2-line graphical display
Cisco IP Phone 7910G	Legacy single-line appearance phone with 2-line black-and-white alphanumeric display
Cisco IP Phone 7912G	Single-line appearance phone with 2-line graphical display
Cisco IP Phone 7920	6-line appearance wireless LAN phone (802.11b) with 9-line grayscale graphical display
Cisco IP Phone 7935	Speakerphone console with alphanumeric display designed for use in conference rooms
Cisco IP Phone 7940G	Dual-line appearance phone with 9-line grayscale graphical display
Cisco IP Phone 7941G	Lighted button, dual-line appearance phone with high resolution graphical display
Cisco IP Phone 7960G	6-line appearance phone with 9-line grayscale graphical display

Table 1-3 *Cisco IP Phones That CallManager Supports (Continued)*

Name	Description
Cisco IP Phone 7961G	Lighted button, 6-line appearance phone with high resolution grayscale graphical display
Cisco IP Phone 7970G	Lighted button, 8-line appearance phone with 9-line color graphical touch screen display
Cisco IP Phone 7971G-GE	Gigabit Ethernet lighted button 8-line appearance phone and 9-line color graphical touch screen display
Microsoft NetMeeting	Windows-based H.323 software client application
Cisco IP SoftPhone	Windows-based JTAPI software client application
Cisco IP Communicator	Windows-based SCCP software client application

Table 1-4 provides a list of the gateway devices that CallManager supports.

Table 1-4 *Cisco Gateways That CallManager Supports*

Gateway Model	Gateway Control Protocol	Trunk Interface	Port Types
Cisco IOS Integrated Routers			
Cisco 1750	H.323	FXS FXO	Loop start or ground start
Cisco 1751 Cisco 1760	MGCP H.323 SIP	FXS FXO T1/E1 PRI T1 CAS E1 CAS R2	Loop start or ground start E&M T1 PRI E1 PRI
Cisco 2600 series Cisco 2800 series	MGCP H.323 SIP (Only MGCP supports QSIG.) (Only H.323 supports E1 CAS R2.)	FXS FXO BRI T1/E1 PRI T1 CAS E1 CAS R2 QSIG (Not all Cisco 2600 series gateways support QSIG. Refer to your gateway documentation.)	Loop start or ground start T1/E1 PRI E&M

continues

Table 1-4 *Cisco Gateways That CallManager Supports (Continued)*

Gateway Model	Gateway Control Protocol	Trunk Interface	Port Types
Cisco 3600 series Cisco 3700 series Cisco 3800 series	MGCP H.323 SIP (Only MGCP supports QSIG.) (Only H.323 supports E1 CAS R2.)	FXS FXO BRI T1/E1 PRI T1 CAS E1 CAS R2 QSIG (Not all Cisco 3600 series gateways support QSIG. Refer to your gateway documentation.)	Loop start or ground start T1/E1 PRI E&M T1/E1 PRI
Cisco 7200 series Cisco 7500 series	MGCP H.323 SIP	T1/E1 CAS T1/E1 PRI QSIG	T1/E1 CAS T1/E1 PRI
Cisco AS5300 Cisco AS5350 Cisco AS5400	H.323	T1/E1 CAS T1/E1 PRI	T1/E1 CAS T1/E1 PRI
Cisco Standalone Voice Gateways			
Cisco Voice Gateway 200 (VG200)	MGCP or H.323 (Only MGCP supports QSIG.)	FXO FXS T1/E1 PRI T1 CAS QSIG	Loop start or ground start T1/E1 PRI E&M T1/E1 PRI
Cisco Voice Gateway 224 (VG224)	MGCP or SCCP	FXS	FXS
Cisco Access Digital Trunk Gateway DE-30+	MGCP	E1 PRI QSIG	E1 PRI E1 PRI
Cisco Access Digital Trunk Gateway DT-24+	MGCP	T1 PRI T1 CAS FXO QSIG	T1 PRI E&M Loop start or ground start T1 PRI
Cisco Access Analog Trunk Gateway (AT-2, AT-4, AT-8)	Skinny Gateway Control Protocol	FXO	Loop start
Cisco Access Analog Station Gateway (AS-2, AS-4, AS-8)	Skinny Gateway Control Protocol	FXS	Loop start

Table 1-4 *Cisco Gateways That CallManager Supports (Continued)*

Gateway Model	Gateway Control Protocol	Trunk Interface	Port Types
Cisco VG248 Analog Phone Gateway	SCCP	FXS	Loop start
Cisco IAD2420	MGCP	FXS FXO T1 PRI T1 CAS QSIG	Loop start or ground start T1 PRI E&M T1 PRI
Cisco Catalyst Voice Gateway Modules			
Cisco Catalyst 4000 Access Gateway Module (WS-X4604-GWY)	MGCP or H.323 (Only MGCP supports QSIG.)	FXS FXO T1 CAS T1/E1 PRI QSIG	POTS Loop start or ground start E&M T1/E1 PRI T1/E1 PRI
Cisco Catalyst 4224 Voice Gateway Switch	MGCP or H.323 (Only MGCP supports QSIG.)	FXS FXO T1/E1 PRI T1 CAS QSIG	POTS Loop start or ground start T1/E1 PRI E&M T1/E1 PRI
Cisco Catalyst 6000 8-Port Voice T1/E1 and Services Module (WS-X6608-T1) (WS-X6608-E1)	MGCP	T1/E1 PRI T1 CAS QSIG	T1/E1 PRI E&M, loop start, ground start T1/E1 PRI
Cisco Catalyst 6000 24-Port FXS Analog Interface Module (WS-X6624-FXS)	MGCP	FXS	POTS
Cisco Communication Media Module (WS-X6600-24FXS)	MGCP	FXS	POTS
Cisco Communication Media Module (WS-X6600-24FXS)	MGCP	T1 PRI T1 CAS E1 PRI	T1 PRI E&M E1 PRI

Call Establishment in a Cisco IP Communications Network

Call establishment between circuit-switched and VoIP systems is more similar than different. While a circuit-switched system relies on a two-phase process that consists of a call signaling phase (into which commands to connect circuits are included) and a media exchange phase, a VoIP system usually deconstructs call establishment into the following three phases:

Step 1 **Call signaling**—Like a circuit-switched call, VoIP systems need to coordinate the placing, offering, and answering of a call; that is, given a person named Alice who wants to call another person named Bob, the call signaling step answers the question, "Do Alice and Bob want to talk?"

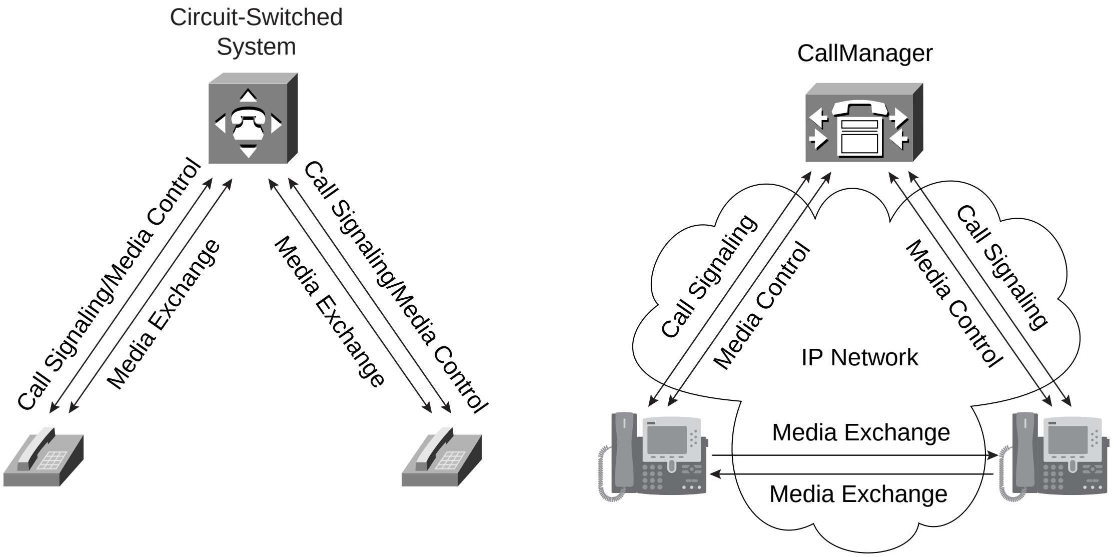
Step 2 **Media control**—Unlike traditional circuit-switched systems, however, VoIP systems enable the endpoints to talk directly to each other over the IP infrastructure. While a circuit-switched system has a sort of tacit media control phase in which it asks the switching fabric to join two circuits, a VoIP system uses a more robust phase to enable the endpoints in the call to exchange IP and port information so that the endpoints can connect themselves. In this respect, the media control step answers the question, "How should Alice and Bob talk?"

> **NOTE** SIP and MGCP combine the exchange of media control information with the call signaling phase, although this behavior doesn't change the underlying fact that the endpoints are ultimately connecting themselves. H.323 also supports an integrated signaling and media control phase via its optional fast start procedure.

Step 3 **Media exchange**—After IP and port information has been exchanged, the endpoints encode information into Real-Time Transport Protocol (RTP) or Secure Real-Time Transport Protocol (SRTP) packets, which they stream directly to each other over the IP infrastructure. In the case of CallManager, this means that, although CallManager is handling the call signaling and media control phases, CallManager has nothing to do with the actual exchange of the conversation, which is a function of the phones and the IP routers that connect them. The media exchange phase answers the real important questions—questions such as "How about we go out for pizza Friday?"

Figure 1-7 shows a comparison between the circuit-switched and packet-switched call models.

Figure 1-7 *Circuit-Switched Call Versus Packet-Switched Call*



Phones are connected directly into the circuit-switched system.

1 Call signaling: The system detects a call request and extends the call to the destination. Negotiation of the type of connection usually occurs as part of the call signaling itself.

2 Media exchange: When the call is answered, the circuit-switched system must bridge the voice stream. Both call signaling and media exchange are centralized.

Phones connect to CallManager through a network of routers.

1 Call signaling: CallManager detects a call request and extends the call to the destination.

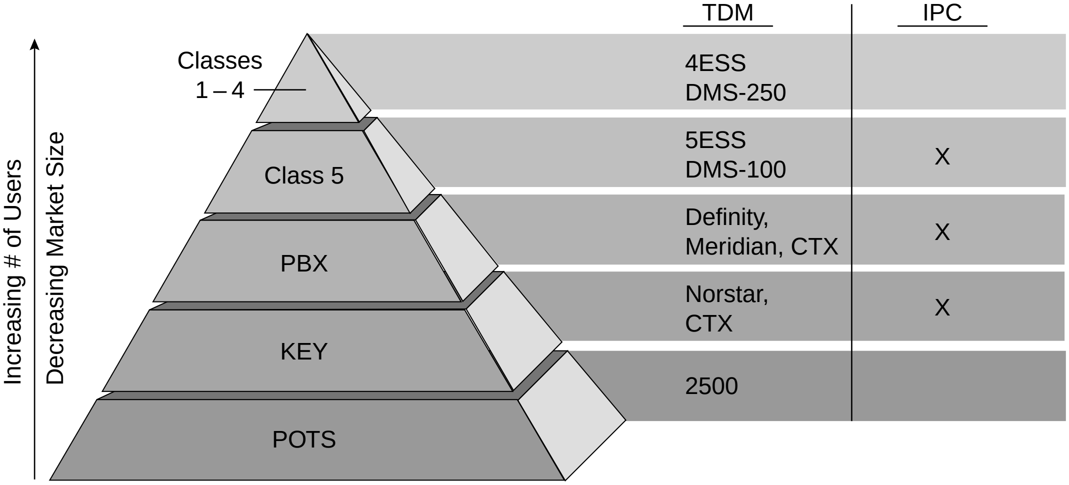
2 Media control (sometimes, but not always, part of call signaling): When the destination answers, the endpoints must negotiate a codec and exchange addresses for purposes of exchanging media.

3 Media exchange: The phones exchange media directly with each other. The media often follows a completely different set of routers than the call signaling. Call signaling and media control are centrally managed, but the high-bandwidth media is distributed.

Using the IP network as a virtual matrix offers some remarkable benefits. The Internet is an IP network that spans the globe. A computer on the Internet can talk to its neighbor as easily as it talks to a computer located 1000 miles away. Similarly, without the need to connect circuits one leg at a time across long distances, one CallManager can connect calls between IP phones separated by area codes or even country codes as easily as it can connect two IP phones in the same building.

Furthermore, IP networks are distributed by their nature. A traditional circuit-based solution requires that all the wires for your voice network run into the same wiring closet. This means that the telephone system can intercept events from the line and trunk cards and gain access to the media information that the devices send to connect them in the matrix. CallManager can communicate with devices by establishing virtual wires through the fabric of the IP network, and the devices themselves establish virtual wires with each other when they start exchanging media. This feature makes CallManager more scalable than traditional circuit-switched systems. Figure 1-8 offers a comparison.

Figure 1-8 *Cisco IP Communications (IPC) Scalability*



TDM	IPC
4ESS DMS-250	
5ESS DMS-100	X
Definity, Meridian, CTX	X
Norstar, CTX	X
2500	

Another major benefit of CallManager is that it resides on the same network as your data applications. The Cisco IP Communications model is a traditional Internet client/server model. CallManager is simply a software application running on your data network with which clients (telephones and gateways) request services using IP interfaces. This co-residency between your voice and data applications allows you to integrate traditional data applications (such as web servers and directories) into the interface of your voice devices. The use of standard Internet protocols for such applications (HTML and XML) means that the skills for developing such applications are readily available, if you want to customize the services available to your voice devices.

Finally, CallManager interacts with IP devices on the network using call signaling protocols, which allows you to mix and match equipment from other vendors when building your voice network. For devices, CallManager supports SCCP to phones, gateways, and transcoding devices; MGCP to gateway devices; H.323 to user and gateway devices; and SIP to other SIP networks. For CTI applications, CallManager supports TAPI and JTAPI.

Cisco IP Communications Clustering

A traditional telephone system tends to come packaged in a large cabinet with racks of outlying cabinets to house the switch cards, line cards, and trunk cards. A Cisco IP Communications network, however, is composed of a larger number of smaller, more specialized components. This allows you to more closely tailor your telephone network to your organization's needs.

This focus on the combined power of small components extends to CallManager, the call processing component of a Cisco IP Communications network. Within a cluster, up to eight servers can be dedicated to running the CallManager service to handle the call routing, signaling, and media control for the enterprise, with other servers dedicated to providing database services, TFTP, applications, and media services such as conferencing, media termination, music on hold, or annunciation. Such a set of networked servers is called a *CallManager cluster*. Clustering helps provide the wide scalability of a Cisco IP Communications network, redundancy in the case of network problems, ease of use for administrators, and feature transparency between users.

Clustering allows for flexibility and growth of the network. In release 4.1, clusters can contain up to eight call processing nodes, which together can support 30,000 endpoints. If your network serves a smaller number of users, you can buy fewer servers. (Using multiple clusters served by an H.323 gatekeeper, CallManager can support larger networks—up to hundreds of thousands of phones.) As your network grows, you can simply add more servers. Clustering allows you to expand your network seamlessly.

The idea behind a cluster is that of a virtual telephone system. A cluster allows administrators to provision much of their network from a central point. Cluster cooperation works so effectively that users might not realize that more than one CallManager node handles their calls. A guiding philosophy of clustered operation is that if a user's primary CallManager node experiences an outage, the user cannot distinguish any change in phone operation when it registers with a secondary or tertiary CallManager. Thus, to the users and the administrators, the individual nodes in the cluster appear as one large telephone system, even if your users reside in completely different geographical regions.

CallManager cluster members do not need to be co-resident. In fact, geographically separating the cluster members can provide even greater device survivability. If a disaster occurs in one geographic site (if, for instance, the CallManager system administrator receives one too many special executive requests and takes a fire ax to the Media Convergence Servers), nodes in other geographic sites can take over the phones. Separating CallManager cluster members in this fashion is called *clustering over the WAN*.

Clustering over the WAN currently requires a high-performance network between the cluster members. The following list summarizes the guidelines:

- At most a 40-ms round-trip packet delay between any two CallManager nodes
- At most four active CallManager nodes (with four standby CallManager nodes for failover)

- 900 kbps per each 10,000 Busy Hour Call Attempts (BHCA) in the cluster (with more bandwidth required if you want to support device failover across the WAN)

If your network doesn't meet the guidelines for clustering over the WAN, deployment options are still available to you:

- Remote sites should run independent clusters—a model called *distributed call processing*.
- Devices in remote sites should be managed by a cluster of servers that reside in a central site, a model called *centralized call processing*.
- Both the centralized and distributed models should be used in a combined model.

Large networks tend to deploy a combination of distributed and centralized call processing systems.

Because performance characteristics and supported deployment models change from release to release, be sure to check http://www.cisco.com/go/srnd for current models and additional information.

Clustering and Reliability

Clustering provides for high reliability of a Cisco IP Communications network. In a traditional telephone network, a fixed association exists between a telephone and the call processing software that serves it. Traditional telephone vendors provide reliability through the use of redundant components installed in the same chassis. Table 1-5 draws a comparison between a traditional telephone system's redundant components and Cisco IP Communications redundancy.

Table 1-5 *Comparison Between Traditional Telephone System Redundancy and Cisco IP Communications Redundancy*

Function	**PBX**	**Cisco IP Communications**
Processor unit	Redundant	Up to eight call processing nodes (running CallManager) with one Publisher database, up to two TFTP servers, and other application and media servers as needed
Media switching	Redundant TDM switch	Distributed IP network (multiple path)
Intercabinet interfaces	Redundant	Distributed IP interfaces (multiple path)
Intracabinet buses	Redundant TDM bus	Redundant Ethernet buses
Power supplies	Redundant	Redundant
Line cards	Single (usually 24)	Not applicable

Table 1-5 *Comparison Between Traditional Telephone System Redundancy and Cisco IP Communications Redundancy (Continued)*

Function	PBX	Cisco IP Communications
Power to phones	Inline (phantom)	Inline (phantom), third pair, or external
Phones	Single interface	Capable of registering with up to three CallManagers and one SRST for retention of service during network outages

CallManager redundancy works differently. The redundancy model differs by Cisco IP Communications component. Clustering has one meaning with regard to the database, another meaning with regard to CallManager nodes, and a third meaning with regard to the client devices.

Database Clustering

To serve calls for client devices, CallManager needs to retrieve settings for those devices. In addition, the database is the repository for information such as service parameters, features, and the route plan. The database layer is a set of dynamic link libraries (DLL) that provide a common access point for data insertion, retrieval, and modification of the database. The database itself is Microsoft SQL 2000.

If the database were to reside on a single machine, the phone network would be vulnerable to a machine or network outage. Therefore, the database uses a replication strategy to ensure that every server can access important provisioning information even if the network fails.

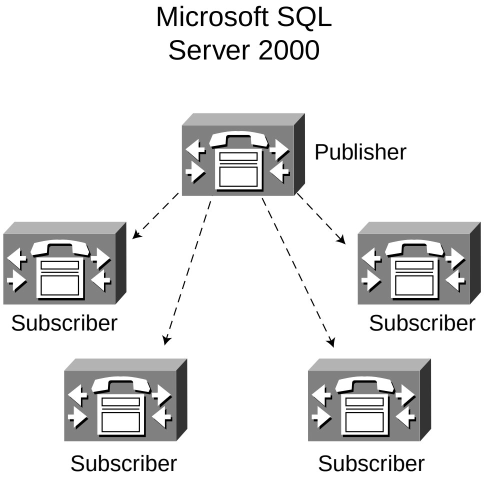
Each CallManager cluster consists of a set of networked databases. One database, the Publisher, provides read and write access for database administrators and for CallManager nodes themselves. For large installations, it is recommended that the Publisher reside on a separate server to prevent database updates from impacting the real-time processing that CallManager does as part of processing calls.

In normal operations, all CallManager nodes in a cluster retrieve information from the Publisher. However, the Publisher maintains a TCP connection to each node in the cluster that runs a CallManager. When database changes occur, the Publisher database replicates the changed information to Subscriber databases on each of these connected nodes. The Publisher replicates all information other than Publisher call detail records (CDR). In addition, the Publisher serves as a repository for CDRs written by all CallManager nodes in the cluster.

In a large campus deployment, a server is often dedicated to handling the Publisher database. This server is often a high-availability system with hardware redundancy, such as dual power supply and Redundant Array of Independent Disks (RAID) disk arrays.

Subscriber databases are read-only. CallManager nodes access the Subscriber databases only in cases when the Publisher is not available. Even so, CallManager nodes continue operating with almost no degradation. If the Publisher is not available, Subscriber nodes write CDRs locally and replicate them to the Publisher when it becomes available again. Figure 1-9 shows database clustering.

Figure 1-9 *Database Clustering*



CallManager Clustering

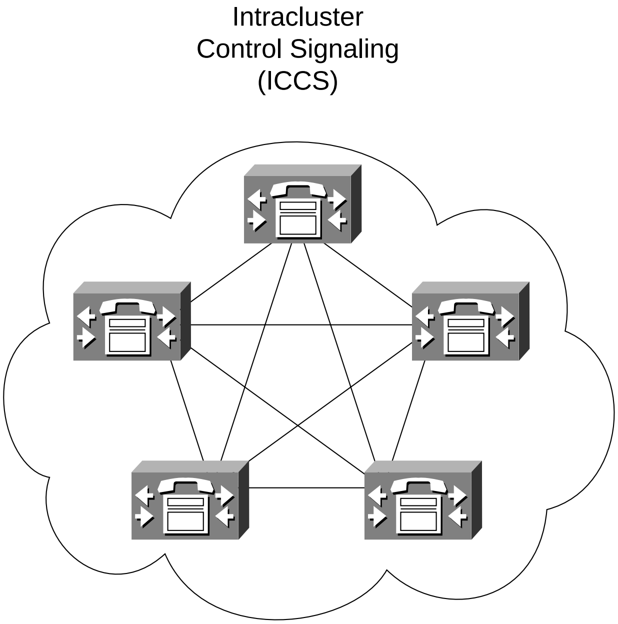
Although the database replicates nearly all information in a star topology (one Publisher, many Subscribers), CallManager nodes replicate a limited amount of information in a fully-meshed topology (every node publishes information to every other node).

CallManager uses a fully-meshed topology rather than a star topology because it needs to be able to respond dynamically and robustly to changes in the network. Database information changes relatively rarely, and the information in the database is static in nature. For example, the database allows you to specify which CallManager nodes can serve a particular device, but the information does not specifically indicate to which node a device is currently registered. Therefore, a star topology that prevents database updates but permits continued operation if the Publisher database is unreachable serves nicely.

CallManager, on the other hand, must respond to the dynamic information of where devices are currently registered. Furthermore, because processing speed is paramount to CallManager, it must store this dynamic information locally to minimize network activity. Should a node fail or the network have problems, a fully-meshed topology allows devices to locate and register with backup CallManager nodes. It also permits the surviving reachable CallManager nodes to update their routing information to extend calls to the devices at their new locations.

Figure 1-10 shows the connections between CallManager nodes in a cluster.

Figure 1-10 *CallManager Clustering*



When devices initialize, they register with a particular CallManager node. The CallManager node to which a device registers must get involved in calls to and from that device. Each device has an address, either a directory number or a route pattern. (See Chapter 2 for more information about call routing). The essence of the inter-CallManager replication is the advertisement of the addresses of newly registering devices from one CallManager to another. This advertisement of address information minimizes the amount of database administration required for a Cisco IP Communications network. Instead of having to provision specific ranges of directory numbers for trunks between particular CallManager nodes in the cluster, the cluster as a whole can automatically detect the addition of a new device and route calls accordingly.

The other type of communication between CallManager nodes in a cluster is not related to locating registered devices. Rather, it occurs when a device controlled by one CallManager node calls a device controlled by a different CallManager node. One CallManager node must signal the other to ring the destination device. The second type of communication is hard to define. For lack of a better term, it is called Intracluster Control Signaling (ICCS).

Understanding this messaging requires knowing more about CallManager architecture. CallManager is roughly divided into six layers:

- Link
- Protocol
- Aggregator
- Media Control
- Call Control
- Supplementary Service

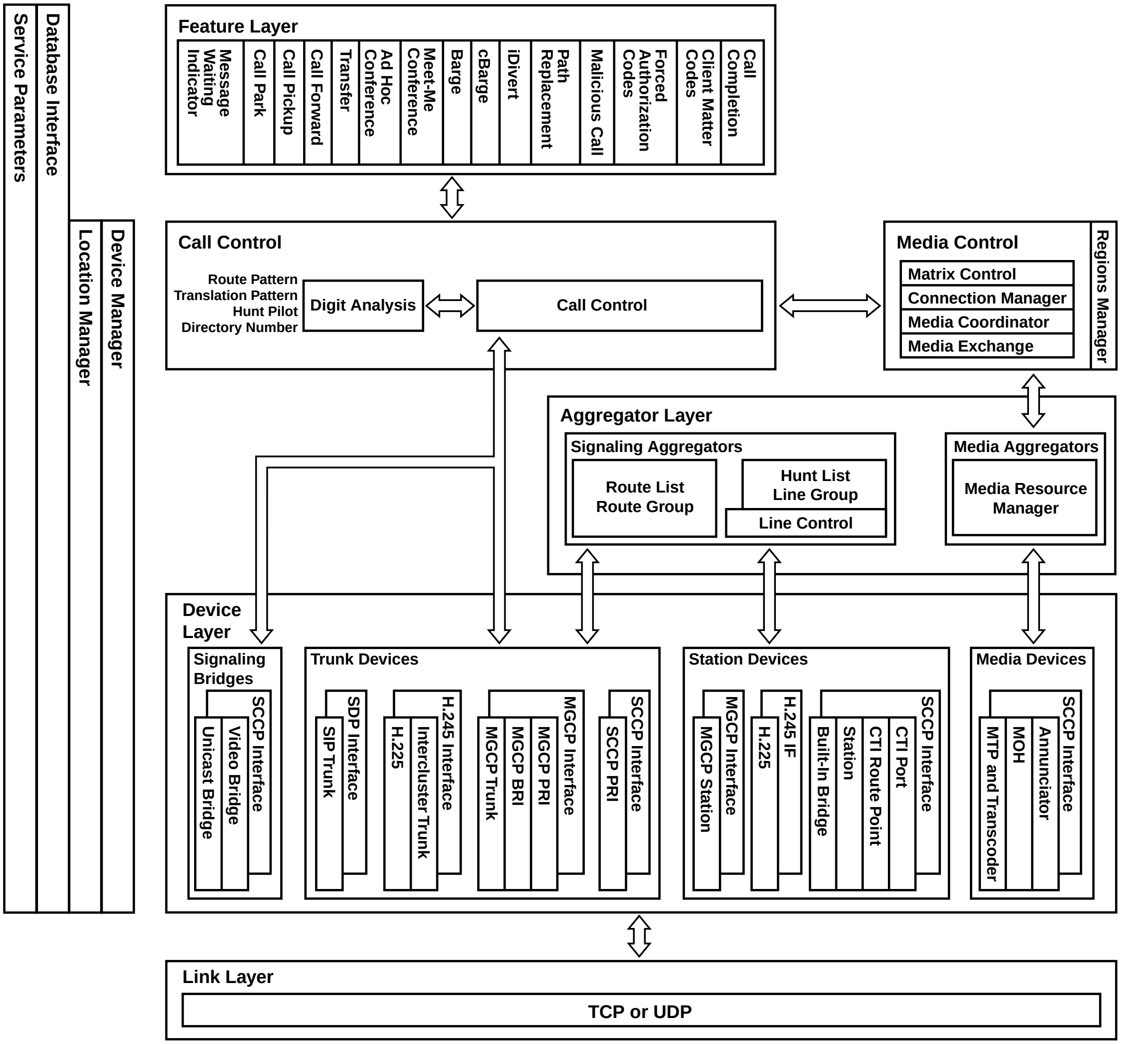
Figure 1-11 depicts this architecture. At the beginning of each subsequent chapter of this book, there is a copy of this figure with shading to indicate the components of CallManager that are covered in that particular chapter.

The Link Layer is the most basic. Its function is to ensure that if a device sends a packet of information to CallManager, or CallManager sends a packet of information to a device, the sent packet is received. CallManager uses two methods of communication. The Transmission Control Protocol (TCP) is by far the most commonly used. TCP underlies much communication on the Internet. It provides for reliable communication between peers using the Internet Protocol. CallManager uses TCP for call signaling and media control with CallManager nodes, media devices, IP phones, H.323 gateways, and ISDN call signaling originating from MGCP gateways. The User Datagram Protocol is a protocol in which a sent packet is not guaranteed to be received. CallManager uses UDP for communication with MGCP gateways and SIP proxies. Although UDP itself is not reliable, MGCP or SIP is designed to handle instances where the IP network loses the message; in such a case, MGCP or SIP retransmit its last message.

The Protocol Layer includes the logic that CallManager uses to manage the different types of devices that it supports. These devices include media devices, trunk devices, and station devices. The Protocol Layer also supports third-party integration with CallManager through the TAPI and JTAPI protocols.

The Aggregator Layer allows CallManager to properly handle the interactions between groups of related devices. The media resource manager, for example, permits one CallManager node to locate available media devices, even if they are registered to other CallManager nodes. The route list performs a similar function for gateways. Line control permits CallManager to handle IP phones that share a line appearance, even if the IP phones are registered with different CallManager nodes.

Figure 1-11 *Layers Within CallManager*



The Media Control Layer handles the actual media connections between devices. It handles the media control portion of setting up a call, but it also handles more complicated tasks. For instance, sometimes CallManager must introduce a transcoding device to serve as an interpreter for two devices that don't communicate via the same codec. In this case, one call between two devices consists of multiple media hops through the network. The Media Control Layer coordinates all the media connections.

The Call Control layer handles the basic call processing of the system. It locates the destination that a caller dials and coordinates the Media Control, Aggregator, and Protocol Layers. Furthermore, it provides the primitives that the Supplementary Service Layer uses to relate independent calls. The Supplementary Service Layer relates independent calls together as part of user-requested features such as transfer, conference, and call forwarding.

Within each layer, the SDL application engine manages *state machines,* which are essentially small event-driven processes, but they do not show up on the Microsoft Windows 2000 Task Manager. Rather, the SDL application engine manages state machine tasks. These state machines each handle a small bit of the responsibility of placing calls in a CallManager network. For example, one kind of state machine is responsible for handling station devices, whereas another type is responsible for handling individual calls on station devices.

These state machines perform work through the exchange of proprietary messages. Before CallManager release 3.0 was created, these messages were strictly internal to CallManager. With the 3.0 release, these messages could travel from a state machine in one CallManager node directly to another state machine managed by a different CallManager node. This mechanism is, in fact, what allows a CallManager cluster to operate with perfect feature transparency. The same signaling that occurs when a call is placed between two devices managed by the same CallManager node occurs when a call is placed between two devices managed by different CallManager nodes.

Architecturally, intracluster communication tends to occur at the architectural boundaries listed in Figure 1-11. Take, for example, the situation that occurs when two devices that share a line appearance register with different CallManager nodes. When someone dials the directory number of the line appearance, both devices ring. Even though the state machine responsible for managing each station is on its own CallManager node, both of these state machines are associated with a single state machine that is responsible for managing line appearances. (These can reside on one of the two CallManager nodes in question, or possibly on a third CallManager node.) The ICCS, however, guarantees that the feature operates the same, no matter how many CallManager nodes are handling a call.

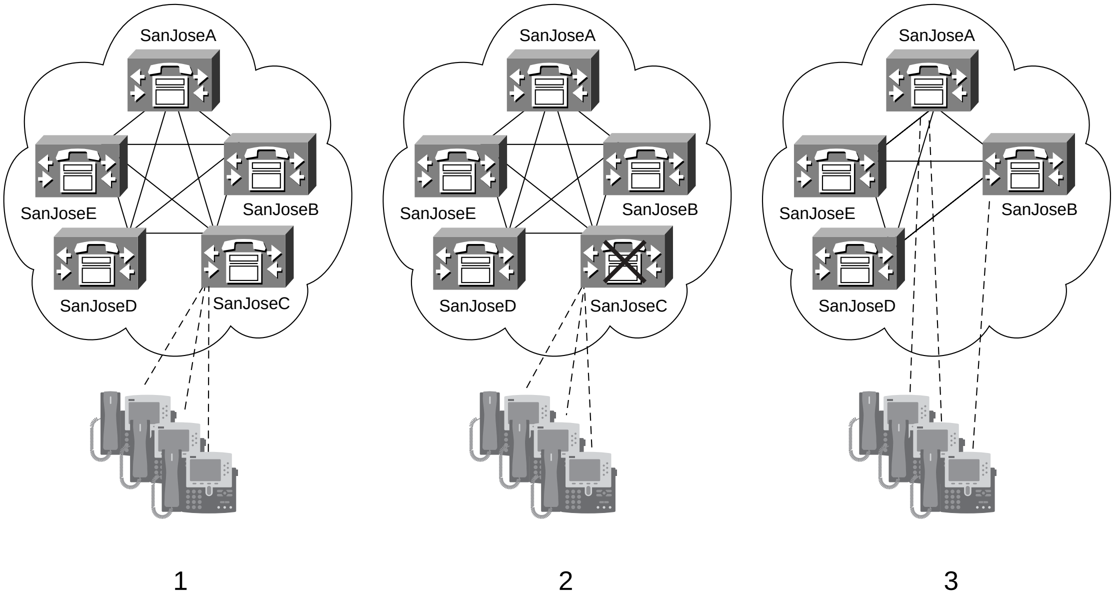
The architectural layers are rather loosely coupled. In theory, a call between two devices registered to different CallManager nodes in the cluster could involve up to seven CallManager nodes, although in practice, only two are required.

Device Redundancy

In a traditional telephone system, the phone is a slave to the call processing logic in the cabinet; it is unaware of the operating condition of its master. Consequently, the secondary master must maintain the state of the endpoint. For this reason, traditional telephone system architectures are redundant architectures rather than distributed architectures: Maintaining state across more than a single backup processor is excessively complex and difficult. In the Cisco IP Communications architecture, the endpoint is aware of the operational status of the server, as well as its own connectivity states. As a result, the endpoints determine which CallManager nodes serve them. You can provision each endpoint with a list of candidate nodes. If the node to which an endpoint is registered has a software problem, or a network connectivity glitch prevents the endpoint from contacting the node, the endpoints move their registration to a secondary or even tertiary CallManager. Phones in active conversations, assuming that the media path is not interrupted, maintain their audio connection to the party to which they are streaming. However, because CallManager is not available to the phone during this interim, users cannot access features on the preserved call. When the call terminates and the phone reregisters, the phone regains access to CallManager features.

Figure 1-12 shows an example of this behavior in action. On Step 1 on the left, three phones are homed to CallManager SanJoseC in a cluster, and each has multiple CallManager nodes configured for redundancy. In Step 2, CallManager SanJoseC fails. As a result, Step 3 shows that all phones that were registered with CallManager SanJoseC switch over to their secondary CallManagers. One phone moves to CallManager SanJoseB, and the other phones move to CallManager SanJoseA.

Figure 1-12 *Device Redundancy*



Deployment of Servers Within a CallManager Cluster

Each CallManager node in a cluster can support up to 7500 phones. A CallManager cluster can support up to 30,000 phones. Adding multiple clusters permits as many phones as you need. Within a cluster, several strategies exist for deployment of servers. Servers can be arranged into clusters, built up of small "molecular" (for lack of a better word) units. Any individual cluster contains at most one Publisher database. Every non-Publisher server in the cluster contains a Subscriber database. Furthermore, each cluster must contain at least one TFTP server to provide Cisco IP Phones and gateways their configurations.

Often, a cluster needs to contain individual servers that run applications or media services (for annunciation or music on hold). From a call agent architectural standpoint, these servers are more akin to end devices than direct participants in the clustering model.

Any single cluster must be composed according to the following rules:

- A cluster can contain at most 20 servers.
- A cluster can contain at most eight nodes running CallManager.
- A cluster must have a Publisher database.
- A cluster must have at least one TFTP service running.

For survivability purposes, a given cluster must contain at least two CallManager nodes. In case one node fails, IP phones and gateways can fail over to the backup node for call processing services. Cisco recommends two models for call processing redundancy. You can compose a cluster using a combination of the two models, but, in general, only one of the two is employed.

In the 1:1 model, you can have either one node entirely in reserve or split the load evenly between the primary and secondary node. If the primary node should fail, all devices registered to it fail over to the secondary node. A 1:1 redundancy model allows you to support the maximum number of phones—7500—per individual node.

In the 2:1 model, you hold one node in reserve for every two nodes that host active devices. If either primary node should fail, the devices registered to that node fail over to the backup node. Because both primary nodes could, in theory fail, causing all devices on both primary nodes to rehome to the secondary node, any individual primary node cannot host the maximum number of devices without unduly stressing the secondary node should both primary nodes fail. Therefore, a 2:1 model allows you to support 5000 phones per primary node, yielding a total of 10,000 per 2:1 redundancy group.

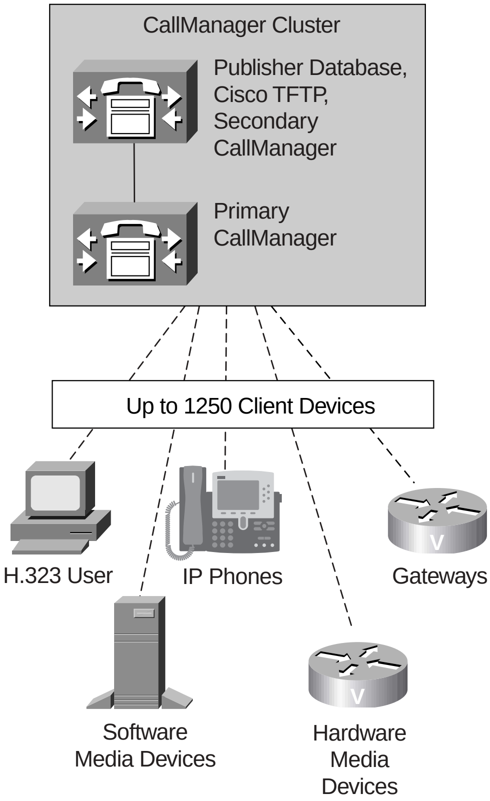
The following sections describe and depict the different configurations.

Minimum Configuration—Up to 1250 Users

The minimum configuration consists of merely two servers. However, because these servers must host a primary CallManager, backup CallManager, Publisher and Subscriber databases, and TFTP server, the maximum number of Cisco IP Phones and gateways that can be supported is 1250.

In this model, one server houses the Publisher and Cisco TFTP, and it serves as a backup CallManager. The other server houses a primary CallManager. Under normal operating conditions, all devices in the cluster register to the second server, but if the second server is unavailable, the first server takes over CallManager responsibilities. Figure 1-13 shows this deployment model.

Figure 1-13 *Deployment Model 1 for up to 1250 Users*

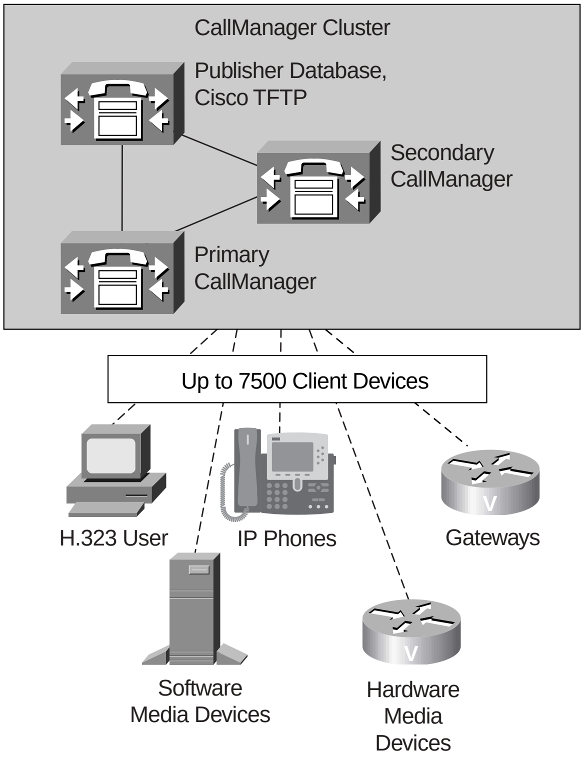


1:1 Redundancy—Up to 7500 Users

To support more than 1250 users, you must dedicate at least one server to both a Publisher database and Cisco TFTP server. When the data management services are offloaded onto a separate server, you can dedicate servers specifically to call processing.

In a 1:1 redundancy group, one server acts as a primary call processing server, with a second server prepared to take over call processing services should the primary server fail. This model also permits load sharing—you can spread your users across both servers. Should a server fail, you can permit users served by the failed CallManager server to fail to the other server. Figure 1-14 shows this deployment model.

Figure 1-14 *1:1 Redundancy Group with Separate Publisher and TFTP Server (7500 Users)*



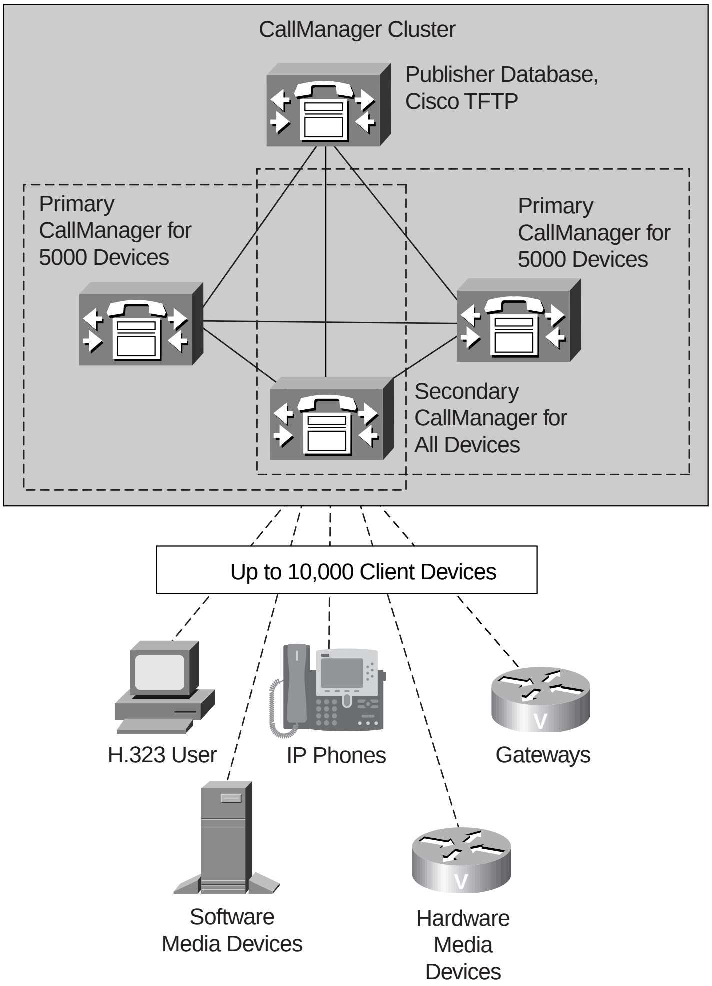
2:1 Redundancy—Up to 10,000 Users

To support more than 1250 users, you must dedicate at least one server to both a Publisher database and Cisco TFTP server. When the data management services are offloaded onto a separate server, you can dedicate servers specifically to call processing.

In a 2:1 redundancy group, two servers act as primary call processing nodes with one server reserved to take over call processing services should one or both primaries fail. The backup call processing node must be prepared to take the devices handled by both active call processing nodes. To prevent overloading the secondary nodes in case both primaries fail, the maximum number of devices that any primary can support must be reduced from 7500 to 5000, yielding a maximum load on the secondary of 10,000 devices should both primary nodes fail.

Figure 1-15 shows this deployment model.

Figure 1-15 *2:1 Redundancy Model with Separate Publisher and TFTP Server (10,000 Users)*


CallManager Cluster
Publisher Database, Cisco TFTP
Primary CallManager for 5000 Devices
Primary CallManager for 5000 Devices
Secondary CallManager for All Devices
Up to 10,000 Client Devices
H.323 User
IP Phones
Gateways
Software Media Devices
Hardware Media Devices

Separating the Publisher and TFTP server from the call processing nodes has the advantage of eliminating the risk that database activity on the Publisher node degrades performance of CallManager if the primary CallManager is unavailable.

Up to 30,000 Users

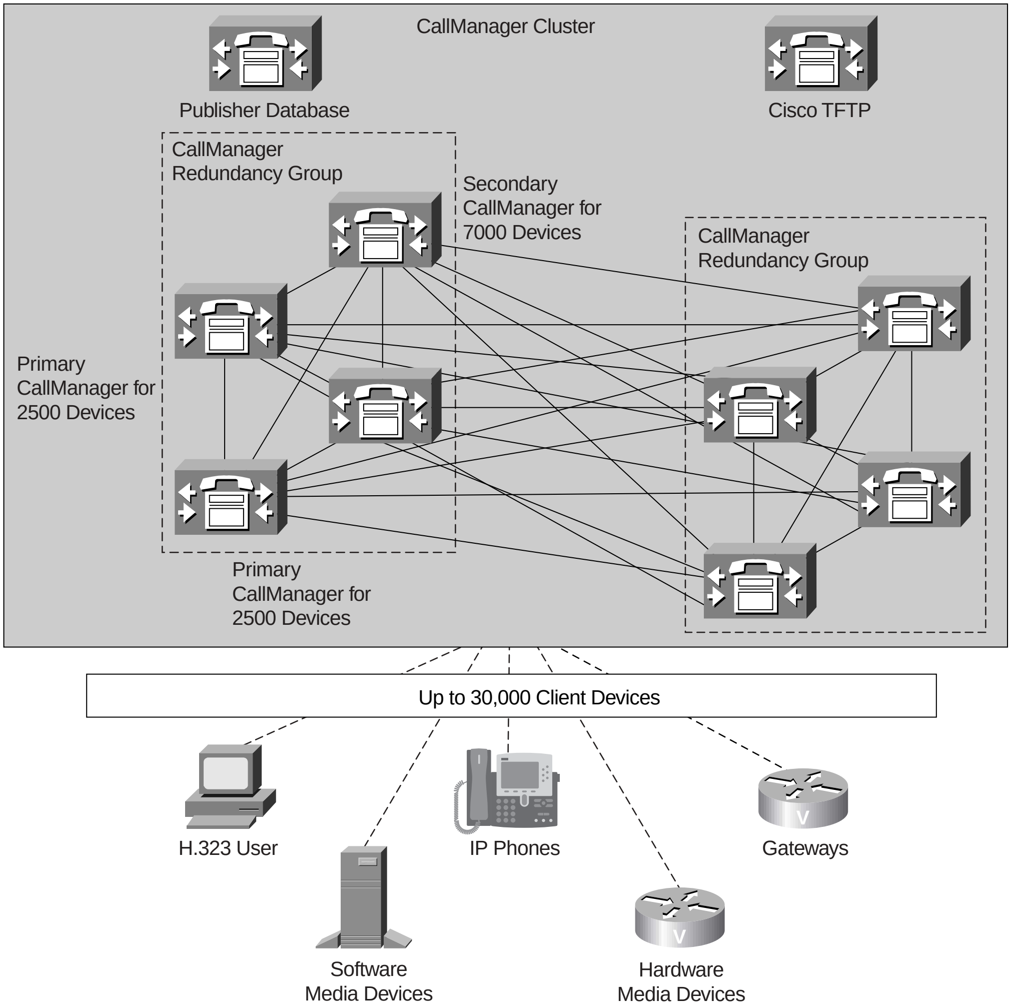
When you exceed 7500 users, it is advisable to configure one server as the Publisher database and one as a TFTP server.

After doing so, you can construct a cluster using either the 1:1 redundancy model or 2:1 redundancy model for call processing nodes.

The 1:1 redundancy model permits you to achieve the cluster maximum of 30,000 IP phones, with 1 server dedicated for a Publisher database, at least one server dedicated for TFTP, and four 1:1 redundancy groups. If you have 7500 IP phones per group, that yields 30,000 IP phones for each of 4 primary servers.

Figure 1-16 shows this deployment model.

Figure 1-16 *Deployment Model for 15,000 to 30,000 Users*

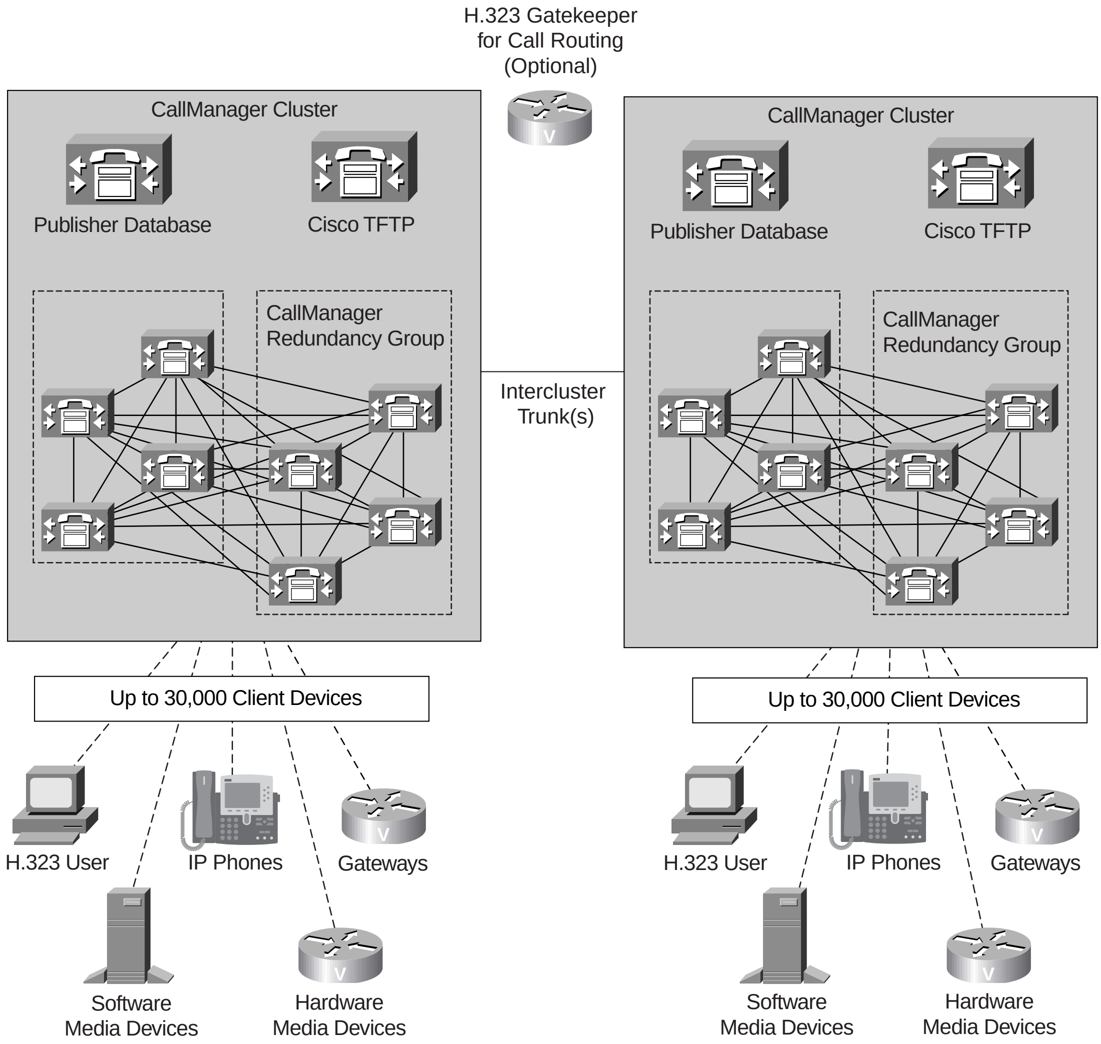


More than 30,000 Users

When the number of users climbs above 30,000, a single cluster cannot manage all devices. However, you can connect CallManager clusters together through either gateways or direct CallManager-to-CallManager connections called *intercluster trunks*. Intercluster trunks run a variant of the H.323 protocol. Figure 1-17 shows this configuration.

Between clusters, you can achieve dial plan management either by configuring your route plan to route calls across the appropriate intercluster trunks or through the use of an H.323 gatekeeper. If you need admissions control, you achieve it through the use of an H.323 gatekeeper.

Figure 1-17 *Deployment Model for More than 30,000 Users*



Enterprise Deployment of CallManager Clusters

This section provides an overview of the ways in which you can deploy CallManager throughout your enterprise. It addresses network infrastructure, admissions control, and supported CallManager topologies.

The excellent Cisco Solutions Reference Network Design guide "IP Telephony SRND," available at http://www.cisco.com/go/srnd, addresses all of the content in this section in far greater detail. The contents of this section have been stolen shamelessly from it. If you are already thoroughly acquainted with the aforementioned Cisco document, you might want to skip the rest of this chapter. In any case, we strongly recommend you read the document to supplement the information contained here.

This section covers two main topics:

- "Network Topologies" describes the supported deployment strategies for a CallManager network.
- "Quality of Service (QoS)" describes the methods by which you can ensure that voice traffic does not experience degradation when the network becomes congested.

Network Topologies

CallManager can be deployed in several different topologies. This section provides an overview of the following topologies:

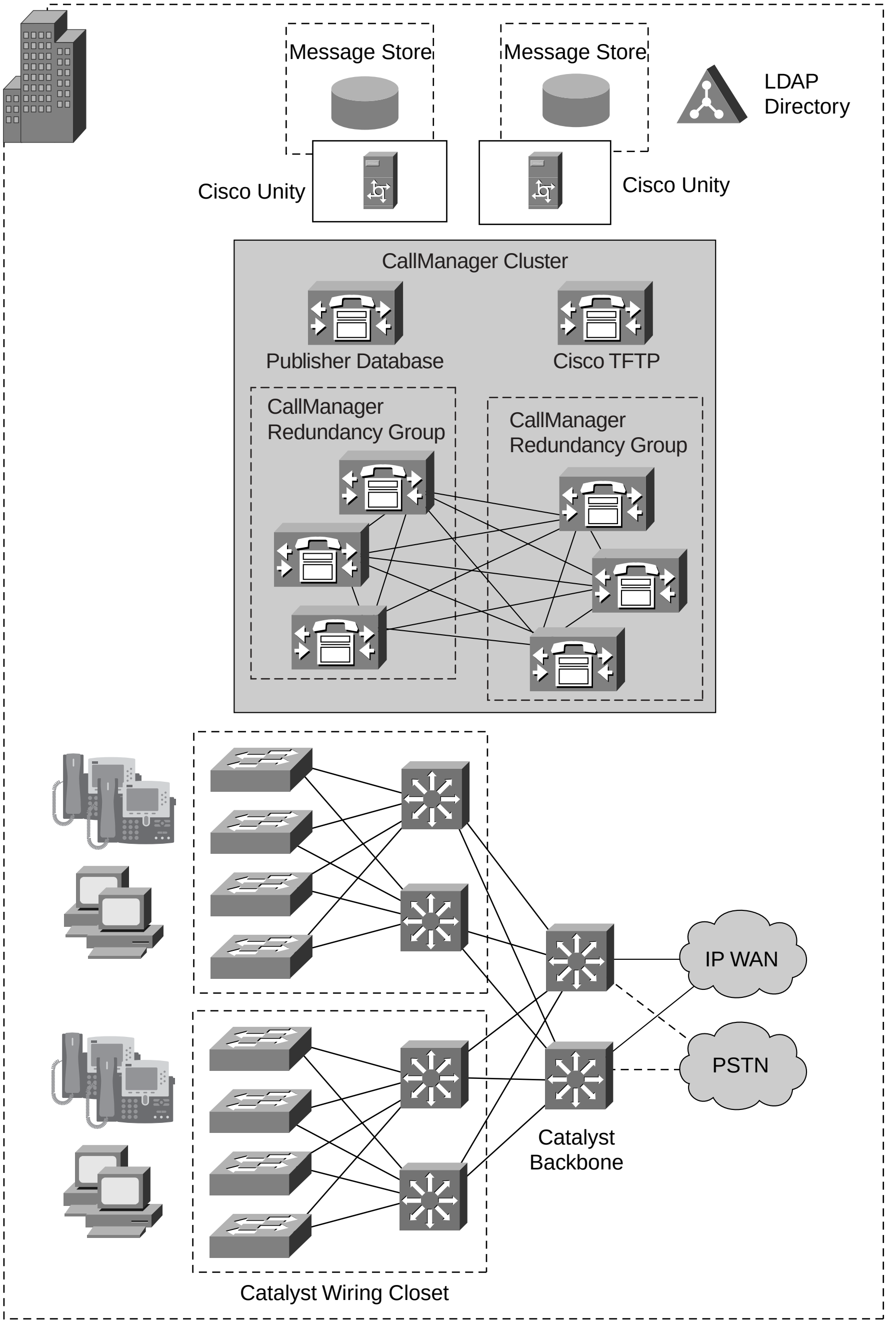
- Single-site model
- Multiple-site model with independent call processing
- Multiple-site IP WAN model with distributed call processing
- Multiple-site model with centralized call processing
- Combined multiple-site model

Single-Site Model

The single-site model consists of a single site or campus served by a LAN. A cluster of up to ten servers (one dedicated to the Publisher database, one dedicated to the TFTP service, and eight running the CallManager service) provides telephony service to up to 30,000 IP-enabled voice devices within the campus. Calls outside of the campus environment are served by IP-to-Public Switched Telephone Network (PSTN) gateways. Because bandwidth is often overprovisioned and undersubscribed on the LAN, there is usually no need to worry about admissions control.

Figure 1-18 depicts the single-site model.

Figure 1-18 *Single-Site Model*



Multiple-Site Model with Independent Call Processing

The multiple-site model consists of multiple sites or campuses, each of which runs an independent cluster of up to ten servers. Each cluster provides telephony service for up to 30,000 IP-enabled voice devices within a site. Because bandwidth is often overprovisioned and undersubscribed on the LAN, there's usually no need to worry about admissions control.

IP-to-PSTN gateways handle calls outside or between each site. The multiple-site model with independent call processing allows you to use the same infrastructure for both your voice and data. However, because of the absence of an IP WAN, you cannot take advantage of the economies of placing voice calls on your existing WAN, because these calls must pass through the PSTN.

Figure 1-19 depicts the multiple-site model with independent call processing.

Multiple-Site IP WAN Model with Distributed Call Processing

From CallManager's point of view, the multiple-site IP WAN model with distributed call processing is identical to the multiple site model with independent call processing. From a practical point of view, they differ markedly.

Whereas the multiple-site model with independent call processing uses only the PSTN for carrying voice calls, the multiple-site IP WAN model with distributed call processing uses the IP WAN for carrying voice calls when sufficient bandwidth is available. This allows you to take advantage of the economies of routing calls over the IP WAN rather than the PSTN.

In such a case, you can set up each site with its own CallManager cluster and interconnect the sites with PSTN-enabled H.323 routers, such as Cisco 2600, 3600, and 5300 series routers. Each cluster provides telephony service for up to 30,000 IP-enabled voice devices. You can add other clusters, which allows your network to support vast numbers of users.

This type of deployment allows you to bypass the public toll network when possible and guarantees that remote sites retain survivability should the IP WAN fail. Using an H.323 gatekeeper allows you to implement a QoS policy that guarantees the quality of voice calls between sites. The same voice codec must apply to all intersite calls. Two chief drawbacks of this approach are increased complexity of administration, because each remote site requires its own database, and less feature transparency between sites.

Because each site is an independent cluster, for all users to have access to conference bridges, MOH, and transcoders, you must deploy these resources in each site. Figure 1-20 presents a picture of the multiple-site IP WAN model with distributed call processing.

Figure 1-19 *Multiple-Site Model with Independent Call Processing*

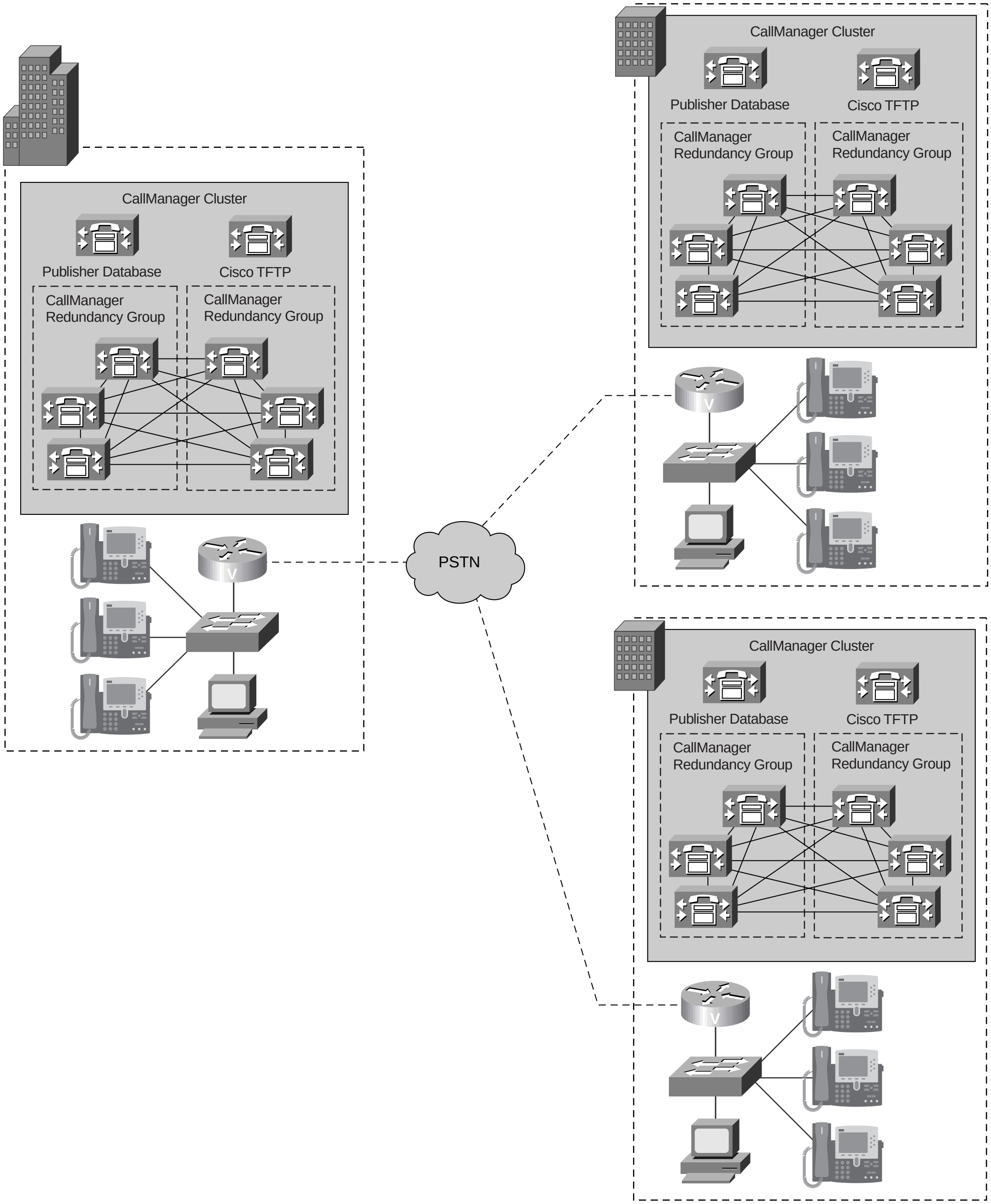
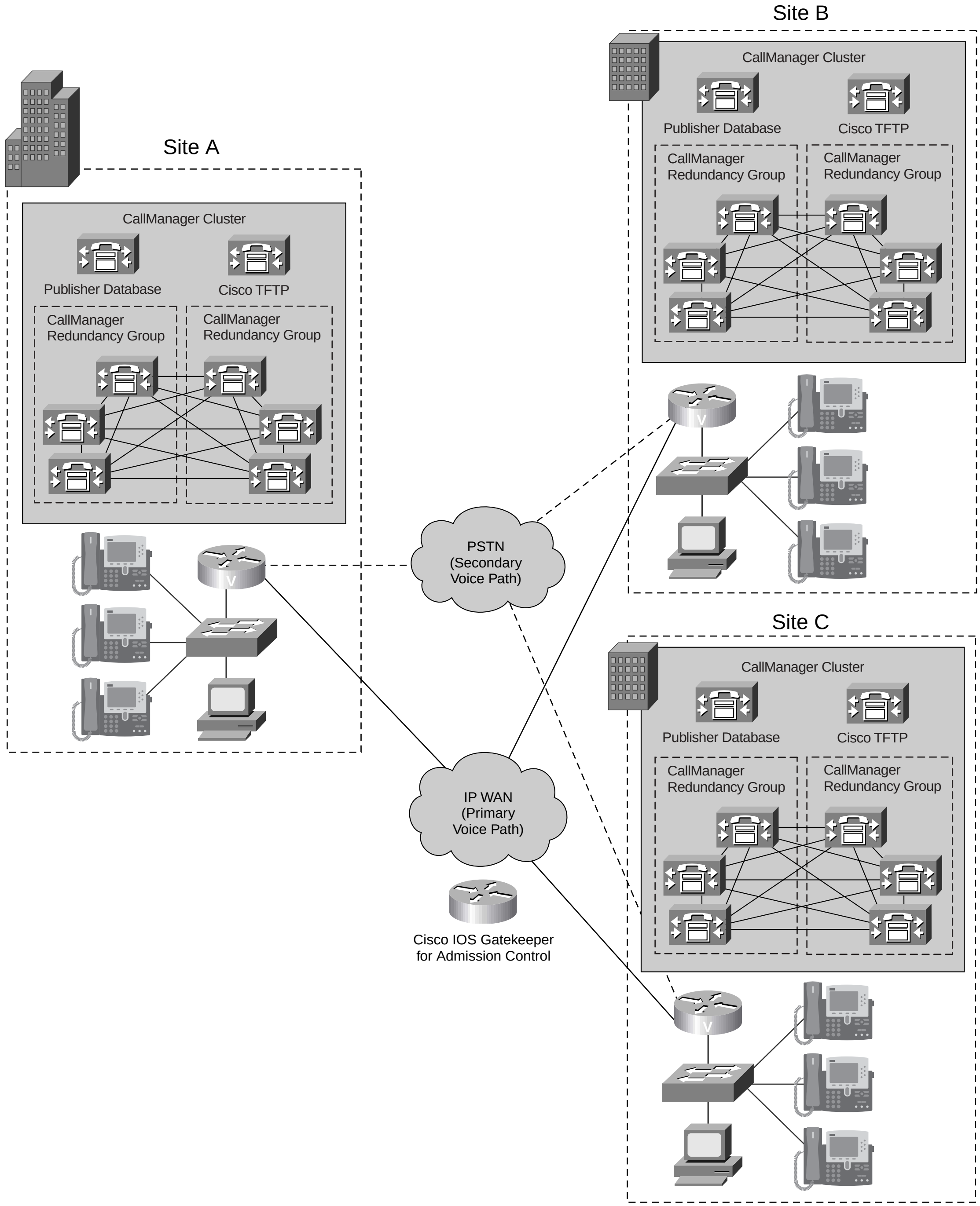

CallManager Cluster
Publisher Database
Cisco TFTP
CallManager Redundancy Group
CallManager Redundancy Group
PSTN
CallManager Cluster
Publisher Database
Cisco TFTP
CallManager Redundancy Group
CallManager Redundancy Group
CallManager Cluster
Publisher Database
Cisco TFTP
CallManager Redundancy Group
CallManager Redundancy Group

Figure 1-20 *Multiple-Site IP WAN Model with Distributed Call Processing*


Site B
CallManager Cluster
Publisher Database
Cisco TFTP
CallManager Redundancy Group
CallManager Redundancy Group
Site A
CallManager Cluster
Publisher Database
Cisco TFTP
CallManager Redundancy Group
CallManager Redundancy Group
PSTN (Secondary Voice Path)
Site C
CallManager Cluster
Publisher Database
Cisco TFTP
CallManager Redundancy Group
CallManager Redundancy Group
IP WAN (Primary Voice Path)
Cisco IOS Gatekeeper for Admission Control

Multiple-Site Model with Centralized Call Processing

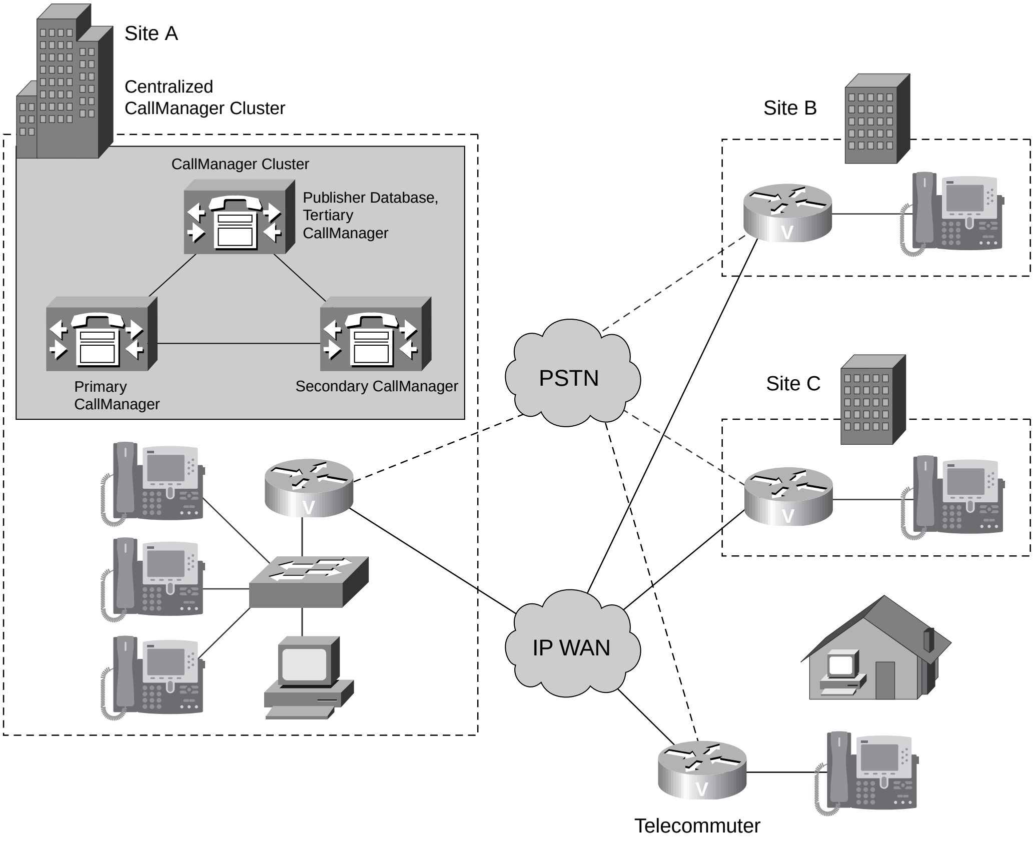
In a multiple-site model with centralized call processing, a CallManager cluster in a centralized campus processes calls placed by IP telephony devices both in the centralized campus and in remote sites connected by an IP WAN. This type of topology is called a *hub-and-spoke topology*: The centralized campus is the hub, and the branch offices sit at the end of IP WAN spokes radiating from the campus.

To CallManager, the multiple-site model with centralized call processing is nearly identical to the single-site model. However, guaranteeing voice quality between branch sites and the centralized site requires the use of a QoS policy that integrates the locations feature of CallManager.

Deploying a multiple-site model with centralized call processing offers easier administration and true feature transparency between the centralized and remote sites.

Because all sites are served by one cluster, you need to deploy only voice mail, conference bridges, and transcoders in the central site, and all remote sites can access these features. Figure 1-21 depicts the multiple-site model with centralized call processing.

Figure 1-21 *Multiple-Site Model with Centralized Call Processing*



If the IP WAN should fail, Cisco Survivable Remote Site Telephony (SRST) can ensure that the phones in remote sites can continue to place and receive calls to each other and to the PSTN. SRST is a feature of Cisco IOS that allows a Cisco router to act as a CallManager when the primary or secondary CallManager nodes are not reachable. SRST requires minimal configuration because it derives most of its settings from the CallManager database.

When the IP WAN is available, SRST acts as a data router for Cisco IP Phones (and outbound PSTN gateway for local calls) and simply ensures connectivity between the branch office devices and CallManager. If the IP WAN fails, however, SRST takes over control of the phones, allowing them to call each other and the PSTN. While phones are registered to SRST, they have access to a reduced feature set. When the IP WAN again becomes available, the phones reconnect to the CallManager cluster.

Table 1-6 shows the router platforms that support SRST and the maximum number of phones that each supports.

Table 1-6 *Routers That Support SRST*

Router	Number of Phones Supported
Cisco 1751-V Cisco 1760 Cisco 1760-V Cisco 2801	24
Cisco 2600XM Cisco 2811	36
Cisco 2650XM Cisco 2651XM Cisco 2821	48
Cisco 2691 Cisco 3640 Cisco 3640A	72
Cisco 2851	96
Cisco 3725	144
Cisco 3660	240

Table 1-6 *Routers That Support SRST (Continued)*

Router	Number of Phones Supported
Cisco 3825	336
Cisco 3745 Catalyst 6500 CMM	480
Cisco 3845	720

Combined Multiple-Site Model

You can deploy the centralized and distributed models in tandem. If you have several large sites with a few smaller branch offices all connected by the IP WAN, for example, you can connect the large sites using a distributed model, while serving the smaller branch offices from one of your main campuses using the centralized model. This hybrid model relies on complementary use of the locations feature of CallManager and gatekeepers for call admission control. Figure 1-22 depicts the combined multiple-site model.

Quality of Service (QoS)

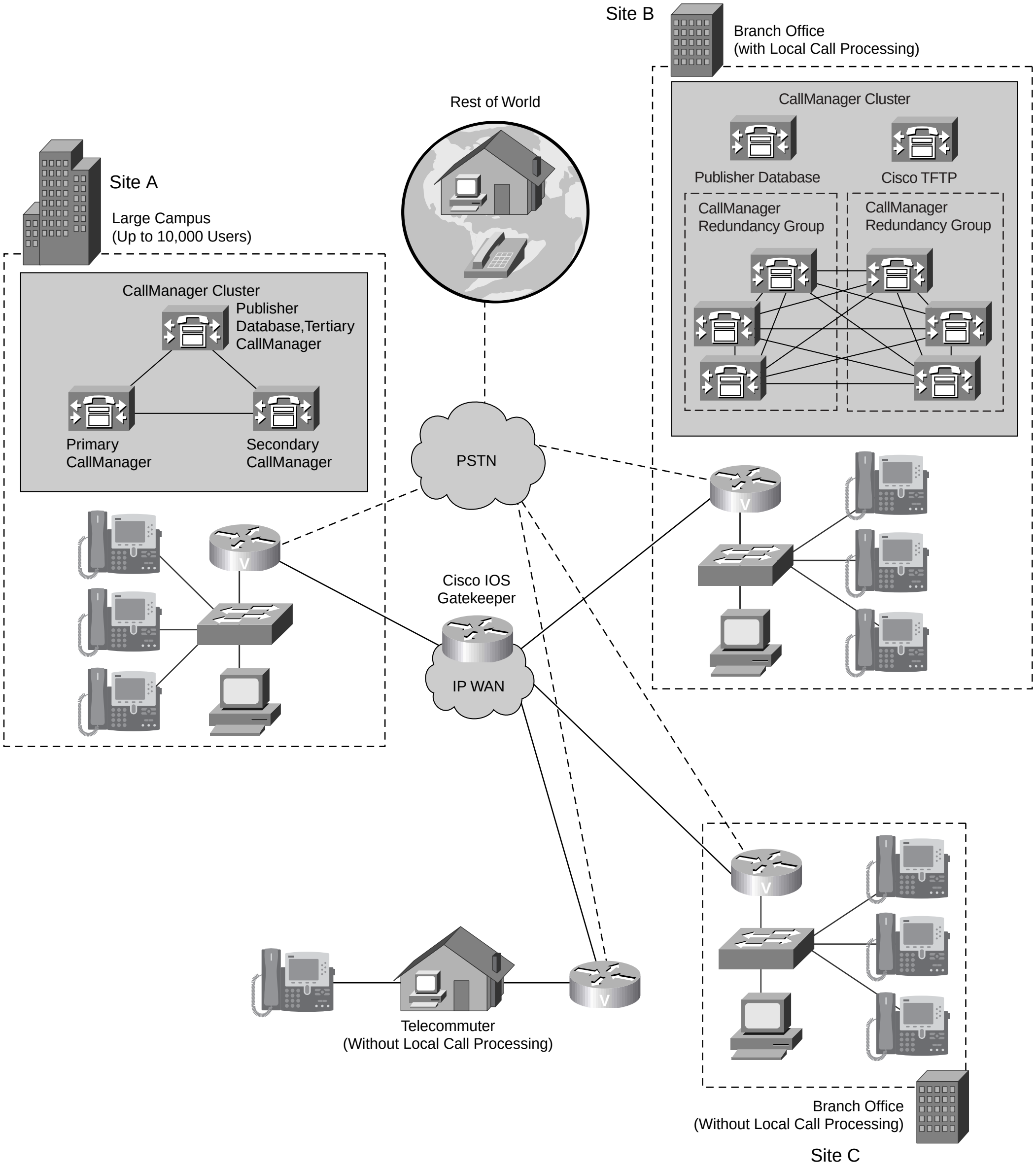
Your network's available bandwidth ultimately determines the number of VoIP calls that your network can handle. As the amount of traffic on an IP network increases, individual data streams suffer packet loss and packet latency. In the case of voice traffic, this can mean clipped, choppy, and garbled voice. QoS mechanisms safeguard your network from such conditions.

Unlike data traffic, voice traffic can survive some loss of information. Humans are good at extracting information from an incomplete data stream, whereas computers are not. Data traffic, on the other hand, can deal with delayed transmission, whereas delayed transmission can destroy the intelligibility of a conversation. *Traffic classification permits* you to categorize your traffic into different types. Traffic classification is a prerequisite to *traffic prioritization*, the process of applying preferential treatment to certain types of traffic. Traffic prioritization allows you to minimize the latency that a voice connection experiences at the expense of the latency that a data connection experiences.

The design guide "Enterprise Quality of Service SRND" at http://www.cisco.com/go/srnd covers QoS in a Cisco IP Communications network in much greater detail than this section, which just provides an overview.

Call admission control (CAC) mechanisms prevent an IP network from becoming clogged with traffic to the point of being unusable. When a network's capacity is consumed, admissions control mechanisms prevent new traffic from being added to the network.

Figure 1-22 *Combined Multiple-Site Model*



When calls traverse the WAN, admissions control assumes paramount importance. Within the LAN, on a switched network, life is good; if you classified your information properly, then either you have enough bandwidth or you do not. Links to remote sites across the IP WAN, however, can be a scarce resource. A 10-Mbps or 100-Mbps Ethernet connection can support hundreds of voice calls, but a 64-kbps ISDN link can route only a few calls before becoming overwhelmed.

This section describes the mechanisms that CallManager uses to enhance voice traffic on the network. It covers the following topics:

- "Traffic Marking" discusses traffic classification and traffic prioritization, features that enable you to give voice communications preferential treatment on your network.
- "Regions" discusses how you can conserve network bandwidth over bandwidth-starved IP WAN connections.
- "CallManager Locations" describes a method of call admissions control that functions within CallManager clusters.
- "H.323 Gatekeeper" describes a method of call admissions control that functions between CallManager clusters.

Traffic Marking

Traffic marking is important in configuring your VoIP network. By assigning voice traffic a routing priority higher than data traffic, you can ensure that latency-intolerant voice packets are passed through your IP fabric more readily than latency-tolerant data packets.

Routers that detect marked packets can place them in higher-priority queues for servicing before lower-priority packets. This strategy ensures that latency-sensitive voice and video traffic does not encounter undue delay between the endpoints. Marking voice and video streams at the highest priority helps ensure that users do not experience drops or delays in the end-to-end media stream. Marking call signaling higher than best-effort data helps ensure that users do not experience undue delay in receiving dial tone upon going off-hook.

CallManager supports two types of traffic marking. *IP Precedence* is the older type of traffic marking. In CallManager 4.0, a type of marking called Differentiated Services (or DiffServ), which is backward compatible with the older style of traffic marking, has essentially replaced it.

IP Precedence

The Cisco 79*xx* series phones (as well as the older Cisco 12SP+ and 30 VIP phones) all send out 802.1Q packets with the type of service field set to 5 for the voice stream and 3 for the signaling streams. CallManager permits you to set its type of service field to 3. In contrast, most data devices encode either no 802.1Q information or a default value of 0 for the type of service field.

When present, the type of service field permits the routers in your IP network to place incoming packets into processing queues according to the priority values encoded in the packet. By more quickly servicing queues into which higher-priority packets are placed, a router can guarantee that higher-priority packets experience less delay. Because all Cisco IP Phones encode their packets with a type of service value of 5 and data devices do not, in effect, the type of service and class of service fields permit you to classify the type of data passing through your network. This allows you to ensure that voice transmissions experience less latency. Figure 1-23 presents an example.

Figure 1-23 *IP Precedence Example*

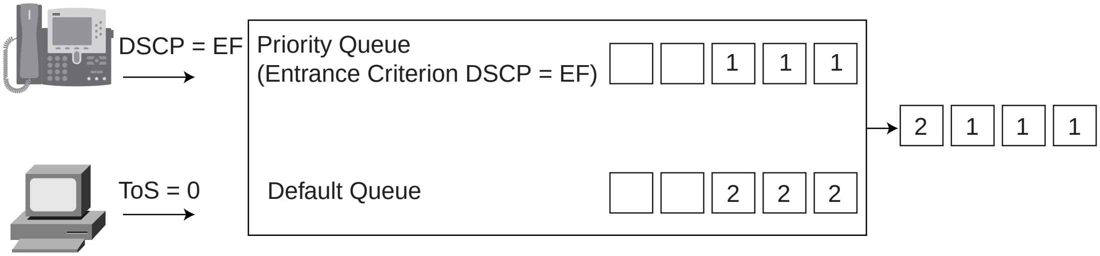


Figure 1-23 depicts two devices that send information through a network router. The Cisco IP Phone 7960 categorizes its traffic with type of service 5, while the PC categorizes its traffic with type of service 0. The router reads packets from both devices from the network and places them in queues based on the type of service field. Packets classified with type of service 5 go on a priority queue; other packets go on the default queue.

When the router decides to forward the packet out to the network again, it sends packets from the priority queue in preference to those on the default queue. Therefore, even if the Cisco IP Phone 7960 and PC send their packets to the router at the same time, the router forwards all of the packets sent by the IP Phone before forwarding any of the packets from the PC. This minimizes the latency (or end-to-end trip time) required for packets from the IP Phone, but increases the latency experienced by the PC. Thus, the router properly handles the latency-intolerant voice packets.

Differentiated Services

Differentiated Services (or DiffServ) is a traffic classification method that has essentially superseded the older IP Precedence traffic classification method. It permits a finer granularity classification than IP Precedence.

When a particular packet is marked, what is actually occurring is that a field in the IP packet header is being tagged with a particular value. The older IP Precedence field is 3 bits long, which permits IP Precedence values ranging from 0 to 7. The newer Differentiated Service Code Point (DSCP) values are 6 bits long, but they use, as the high-order bits of the DSCP, the original 3 bits set aside for the Type of Service field in the IP header. Therefore, routers that do not pay attention to the newer method of traffic classification can still provide voice and video traffic preferential treatment, because the high-order bits of this DSCP-marked traffic roughly correspond with the older IP Precedence values.

Table 1-7 *Comparison Between Traffic Classification Values*

3 High-Order Bits	IP Precedence	CallManager-Settable DSCP Values	Comment
000	Default	Default	Best-effort traffic
001	IP Prec 1	CS1 (001000) AF11 (001010) AF12 (001100) AF13 (001110)	
010	IP Prec 2	CS2 (010000) AF21 (010010) AF22 (010100) AF23 (010110)	
011	IP Prec 3	CS3 (011000) AF31 (011010) AF32 (011100) AF33 (011110)	Recommended call signaling
100	IP Prec 4	CS4 (100000) AF41 (100010) AF42 (100100) AF43 (100110)	Recommended video
101	IP Prec 5	EF (101110)	Recommended voice
110	IP Prec 6		Reserved
111	IP Prec 7		Reserved

Regions

Like IP precedence, regions play an important role in ensuring the quality of voice calls within your network. Regions allow you to constrain the codecs selected when one device calls another. Most often, you use regions to limit the bandwidth used when calls are placed between devices connected by an IP WAN. However, you can also use regions as a way of providing higher voice quality at the expense of network bandwidth for a preferred class of users.

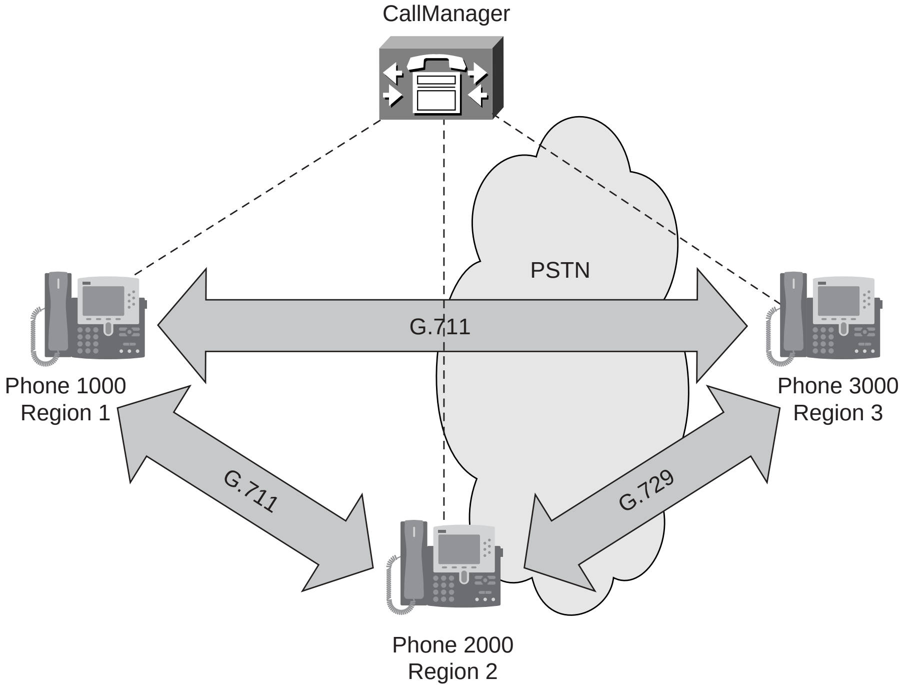
When you define a new region, Cisco CallManager Administration asks you to define the compression type used for calls between devices within the region. You also define, on a region-by-region basis, compression types used for calls between the region you are creating and all other regions.

You associate regions with device pools. All devices contained in a given device pool belong to the region associated with that device pool. When an endpoint in one device pool calls an endpoint in another, the codec used is constrained to what is defined in the region. If, for some reason, one of the endpoints in the call cannot encode the voice stream according to the specified codec, CallManager attempts to introduce a transcoder (see Chapter 5) to allow the endpoints to communicate.

Figure 1-24 depicts a configuration that uses three regions to constrain bandwidth between end devices. Phones 1000 and 2000 are in the main campus; phone 3000 is in a branch office. Calls within the main campus use the G.711 codec, as do calls from phone 1000 to phone 3000. Calls between phone 2000 and phone 3000 use the G.729 codec.

Figure 1-24 *Regions Overview*

	Region 1	Region 2	Region 3
Region 1	G.711		
Region 2	G.711	G.711	
Region 3	G.711	G.729	G.711



CallManager Locations-Based Call Admissions Control

Locations represent a form of admissions control. A location defines a topological area connected to other areas by links of limited bandwidth. With each location, you specify the amount of bandwidth available between users in that location and other locations in your network.

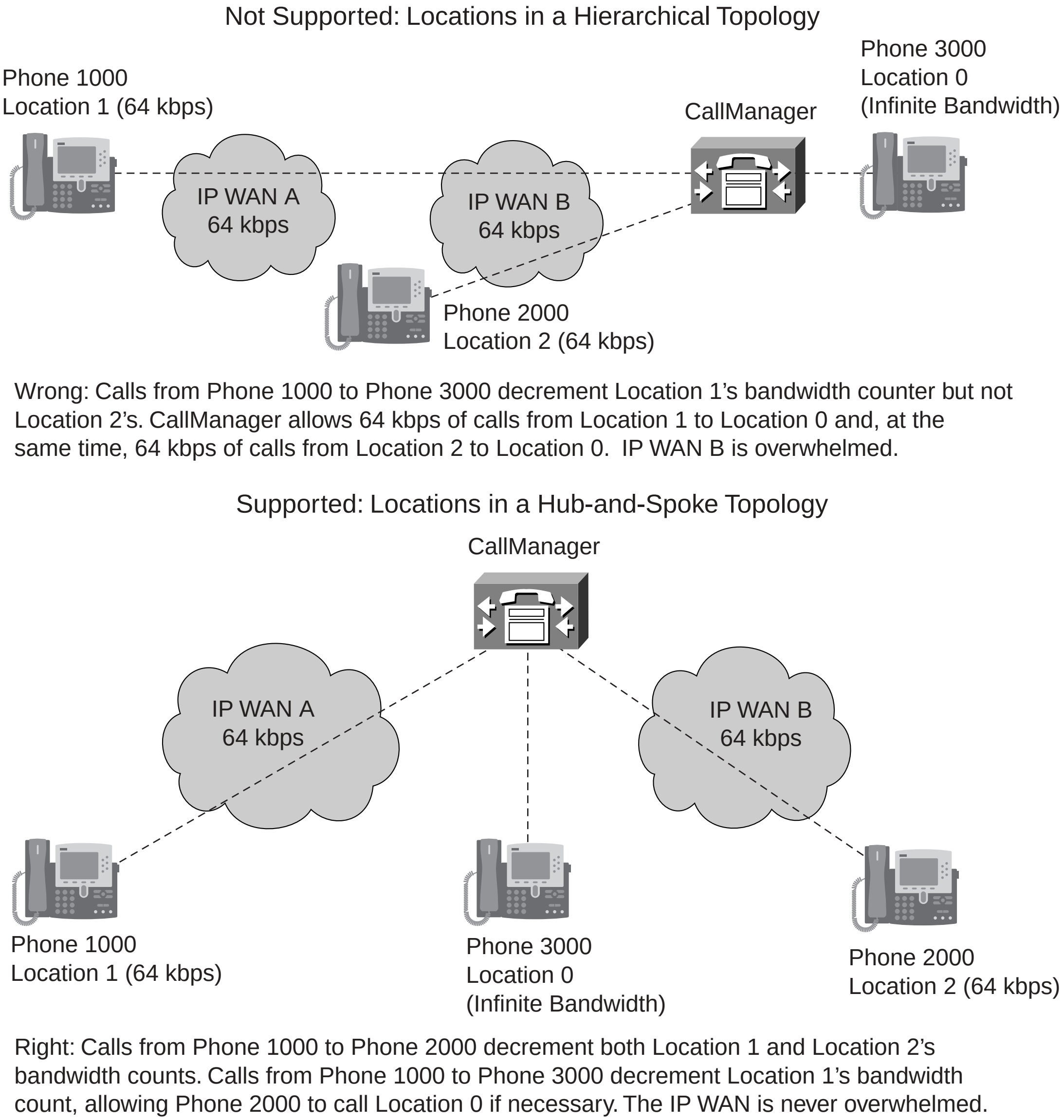
CallManager allows users to place an unlimited number of calls between devices within the same location; when a user places a call to another location however, CallManager temporarily deducts the bandwidth associated with the selected codec from the interlocation bandwidth remaining. When a user's call terminates, CallManager returns the allocated bandwidth to the pool of available bandwidth.

Users who attempt to place a call when no more bandwidth is available receive a fast busy tone (also called reorder tone), unless you enable a feature called Automated Alternate Routing (AAR). If AAR is properly configured, then instead of rejecting calls when the bandwidth between locations is oversubscribed, if the dialed destination has a PSTN number, CallManager automatically redials it to send the call to the local branch gateway for routing the call over the public network.

You must consider several design caveats before using locations-based CAC.

- Locations-based CAC requires that you deploy your voice network in a hub-and-spoke topology. Although locations allow you to configure admissions control, the locations mechanism is topologically ignorant. Having only one bandwidth counter for all interlocation calls means that all calls from one location to any other location must traverse only one logical network link, which limits deployment strictly to hub-and-spoke topologies. Figure 1-25 elaborates.
- When CallManager connects a call on behalf of a device that requires an MTP, CallManager does not account for the bandwidth between the device and the MTP. As a result, you must co-locate MTPs with the devices that require them and set up Media Resource Group Lists (MRGL) to use them.

Figure 1-25 *Hub-and-Spoke Topology Restriction*



Wrong: Calls from Phone 1000 to Phone 3000 decrement Location 1's bandwidth counter but not Location 2's. CallManager allows 64 kbps of calls from Location 1 to Location 0 and, at the same time, 64 kbps of calls from Location 2 to Location 0. IP WAN B is overwhelmed.

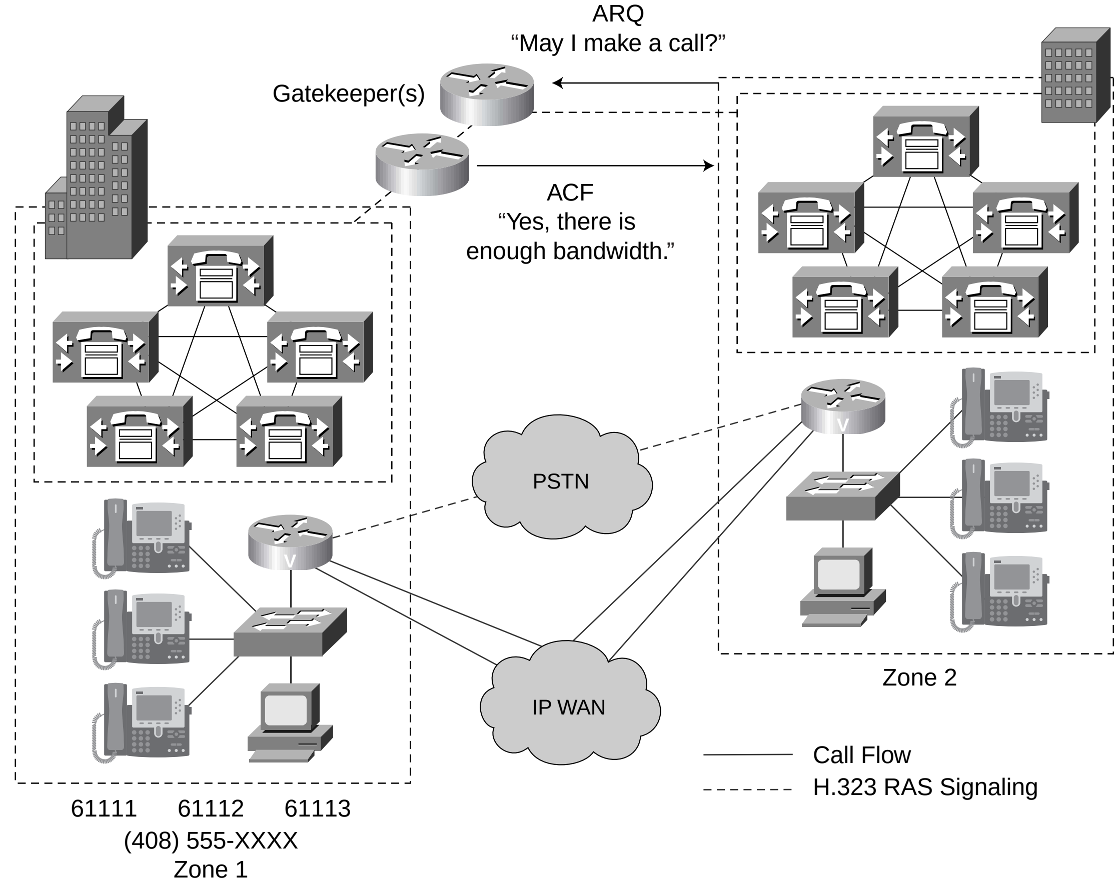
Right: Calls from Phone 1000 to Phone 2000 decrement both Location 1 and Location 2's bandwidth counts. Calls from Phone 1000 to Phone 3000 decrement Location 1's bandwidth count, allowing Phone 2000 to call Location 0 if necessary. The IP WAN is never overwhelmed.

H.323 Gatekeeper

CallManager can be configured to use an H.323 gatekeeper for call admissions control between CallManager clusters. Before placing an H.323 call, a gatekeeper-enabled CallManager makes a Registration, Admissions, and Status (RAS) protocol admissions request (ARQ) to the H.323 gatekeeper.

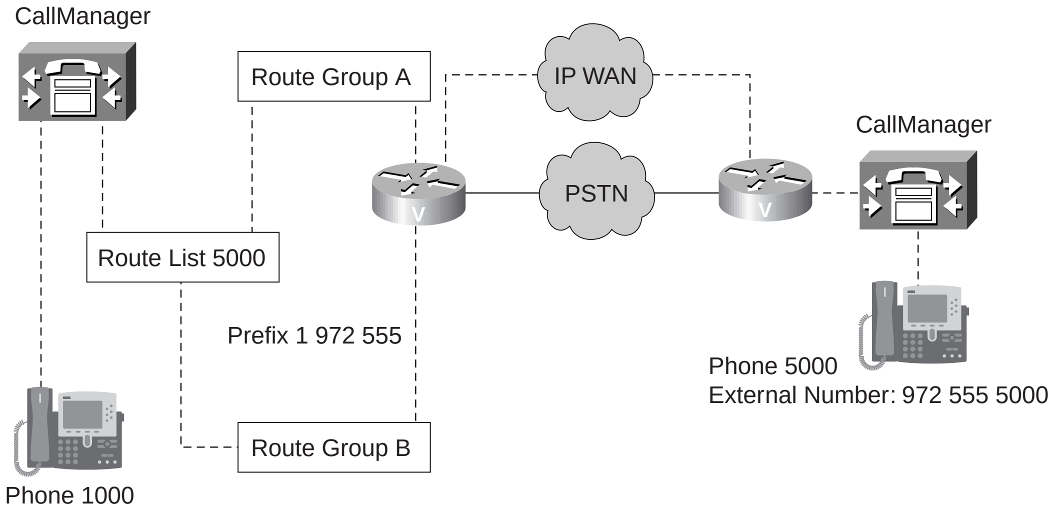
The H.323 gatekeeper associates the requesting CallManager with a zone and can track calls that come into and go out of the zone. If the bandwidth allocated for a particular zone is exceeded, the H.323 gatekeeper denies the call attempt, and the caller hears a fast busy tone. (Alternatively, using route lists, you can configure CallManager to offer the call to a local PSTN gateway if the gatekeeper denies the call.) Essentially, an H.323 gatekeeper provides a locations-like functionality for the H.323 domain. Figure 1-26 depicts a gatekeeper-enabled configuration.

Figure 1-26 *H.323 Gatekeeper-Based Call Admissions Control*



To configure fallback through the PSTN, you must configure the route plan to choose an alternate route if the gatekeeper rejects the call attempt. To configure PSTN fallback, you must configure a route list that contains two route groups. The first route group contains the intercluster trunk that routes outgoing calls over the IP WAN. If insufficient bandwidth is available, however, the H.323 gatekeeper rejects this outgoing call attempt. This call rejection triggers the alternate route associated with the route list. When CallManager selects this alternate route, it transforms the dialed digits to the destination's address as seen from the PSTN's point of view and offers the call to the PSTN gateway. Figure 1-26 demonstrates fallback routing through the PSTN.

Figure 1-27 *Fallback Routing Through the PSTN*



A call from Phone 1000 to Phone 5000 first attempts to route across the IP WAN. If the gatekeeper denies the call attempt, the route list modifies the dialed number and again offers the call to the gateway, which routes the call across the PSTN.

Chapter 2 discusses call routing in much more detail.

Summary

This chapter provided an overview of Cisco IP Communications, including VoIP and how Cisco IP Communications differs from traditional telephone systems, and how you can use VoIP to achieve savings by routing your telephone calls over the IP WAN.

You also learned about CallManager, the heart of Cisco IP Communications, including a short history of how CallManager has evolved and the following components of a Cisco IP Communications network:

- Cisco-certified servers on which CallManager runs
- Windows 2000 and Tomcat services that provide IP telephony in a Cisco IP Communications network
- Client devices that CallManager supports

Also discussed were the phases that CallManager goes through to set up a call and CallManager's clustering strategy for providing high availability and scalability. Several different deployment models for CallManagers within a cluster were also described, including several methods of deploying clusters to serve both campuses and campuses with remote offices. QoS, including traffic classification, traffic prioritization, and call admissions control (both locations-based and gatekeeper-controlled) by which you can guarantee good voice quality in your network were also discussed.