CHAPTER 1

Cisco CallManager Architecture

Cisco Architecture for Voice, Video and Integrated Data (Cisco AVVID) is a suite of components that includes Internet Protocol (IP) telephony communications. Cisco CallManager is the call routing and signaling component for *IP telephony* in a Cisco AVVID IP Telephony network. The term IP telephony describes telephone systems that place calls over the same type of data network that makes up the Internet.

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 Cisco 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 are different, 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 (PBXs)
- Class 5 switches
- Class 1 to 4 switches

A *key system* is a very 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-switched 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's 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 (called *interexchange carriers* [IECs or IXCs]) use *Class 1 to 4 switches*. They process truly mammoth levels of calls and connect calls from one Class 5 switch to another.

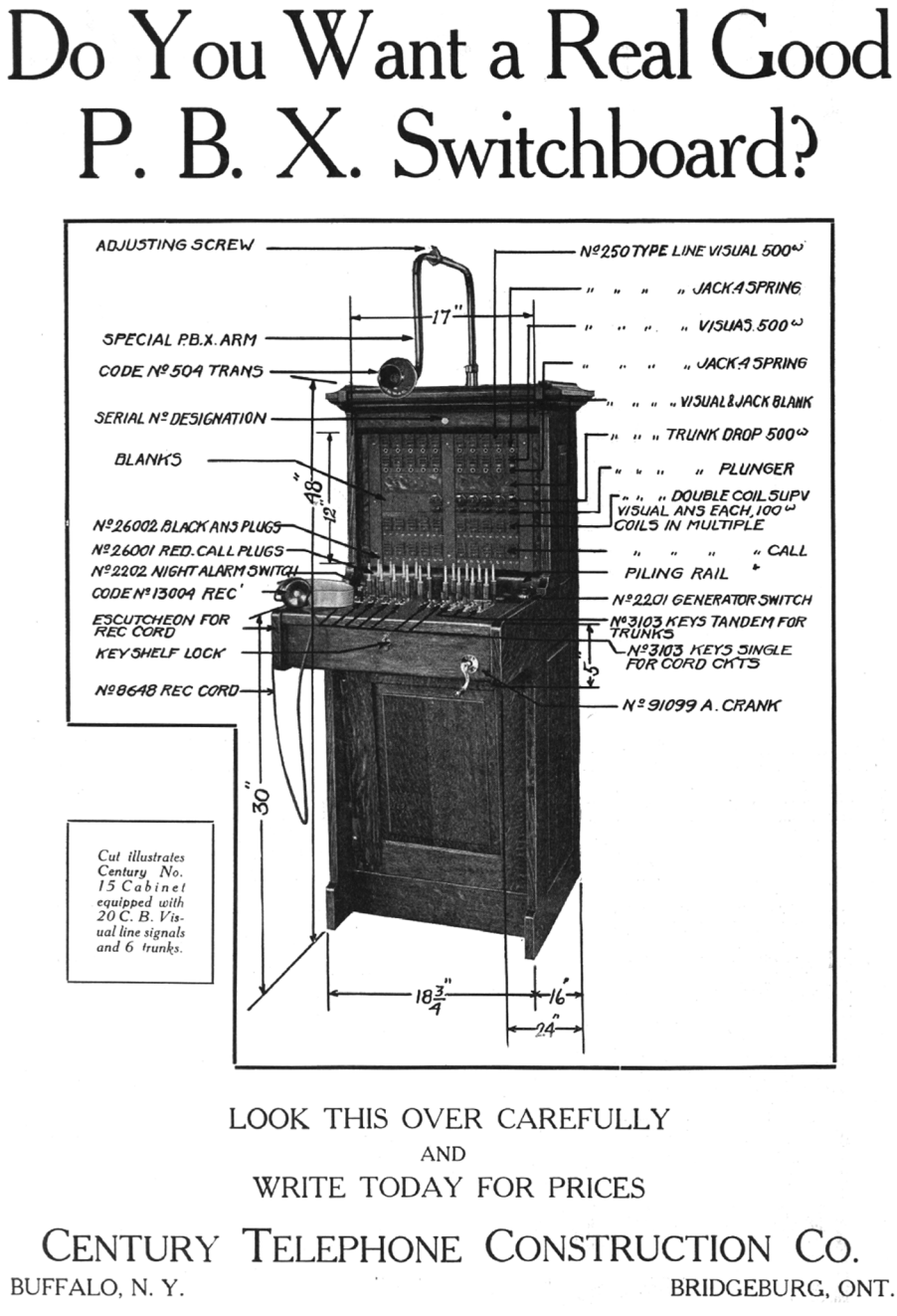
Despite the large disparity in the number of users supported by these types of traditional networks, the core technology is circuit-based. Consider an old-time telephone operator. He or she sits in front of a large plugboard with hundreds of metal sockets and plugs. (Figure 1-1 shows a picture of an 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 if the called party is busy. If not, the operator then 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 if the line is busy, and then completes the circuit between the calling and called parties.

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, while 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 once 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 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.

Figure 1-1 *An Early PBX*



Like the central switching office, CallManager is a computerized replacement for a human operator. However, CallManager relies on packet switching to transmit conversations. *Packet switching* is the mechanism by which data is transmitted through the Internet. Web pages, e-mail, and instant messaging are all conveyed through the fabric of the Internet by packet switching.

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 backup connection can forward the information to the backup router, which in turn will look at the address of the recipient and determine 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 PCM-encoded voice stream fills the capacity of a single voice circuit.

An interesting wrinkle about voice encoding is introduced by packet-switched communications. Even if circuit-switched systems encoded the voice stream according to a more efficient scheme, there is little incentive to do so, because a circuit is fully reserved no matter how much 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 a smaller number of 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 that the resulting media stream requires. This statement does not hold so true for those codecs that attempt to minimize bandwidth. 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 (DSPs) can handle the computations efficiently. As a result, codec support often differs substantially from device to device in the Voice over IP (VoIP) network.

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 rest of this chapter discusses the following topics:

- Circuit-switched systems
- Cisco AVVID IP Telephony networks
- Enterprise deployment of Cisco 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.

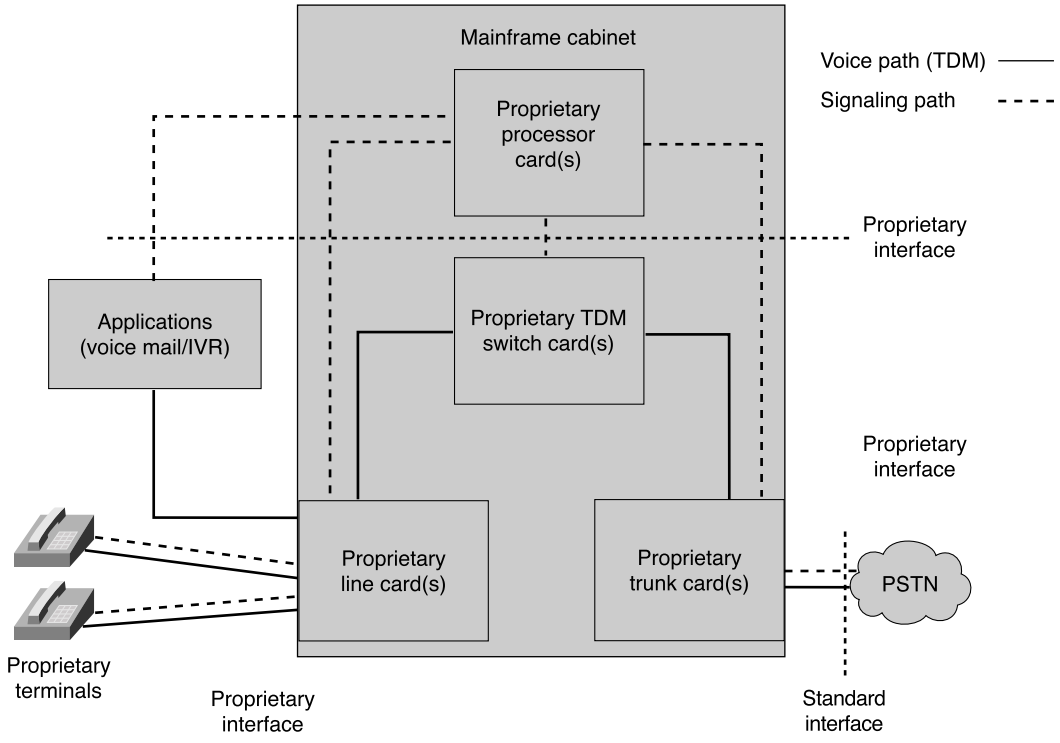
Call Establishment in a Circuit-Switched Telephone System

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 soon as the telephone system determines that the called party wants to take the call, it connects an 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.

Figure 1-2 *Traditional Circuit-Switched Architecture*



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. There are dozens of call signaling protocols, of which the following are just a sample:

- 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.

Once 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, where endpoints generally have more voice-encoding schemes to choose from.

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, 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 AVVID IP Telephony Networks

A Cisco AVVID IP Telephony network is a packet-based system. Cisco 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 IP Auto Attendant (Cisco IP AA), Cisco IP SoftPhone, Cisco IP Interactive Voice Response (IVR), Cisco IP Contact Center (IPCC), and Cisco WebAttendant.

A Cisco AVVID IP Telephony 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:

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

Cisco CallManager History

There have been several releases of the software that would become Cisco CallManager release 3.1. It started in 1994 as a point-to-point video product, but it was recast as an IP-based telephony system in 1997. By 2001, it is able to support as many as 100,000 users.

1994—Multimedia Manager

Cisco CallManager release 3.1 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 was a signaling source and sink only. 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 3.51 Server, 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.

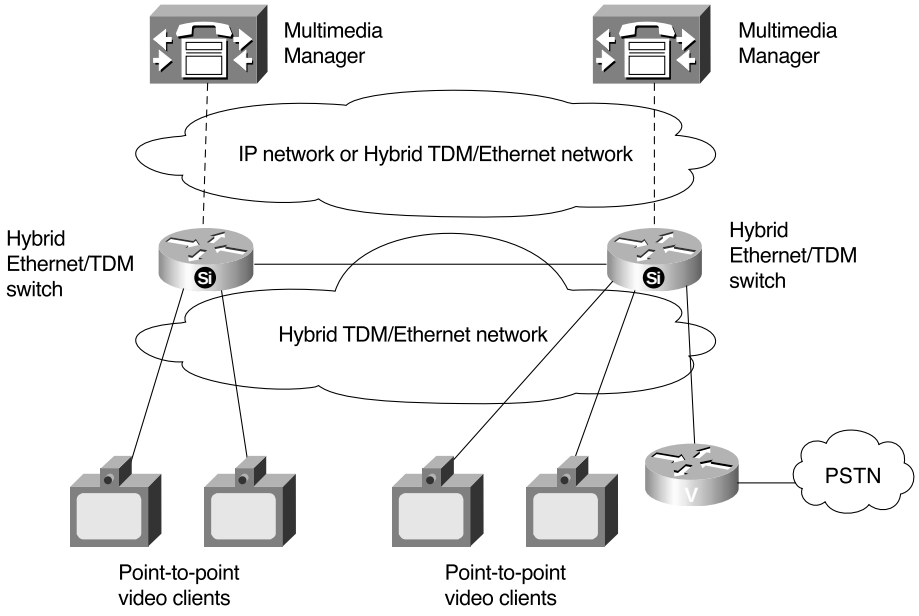
Figure 1-3 depicts CallManager as it existed in 1994.

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.

Figure 1-3 *Cisco CallManager in 1994*



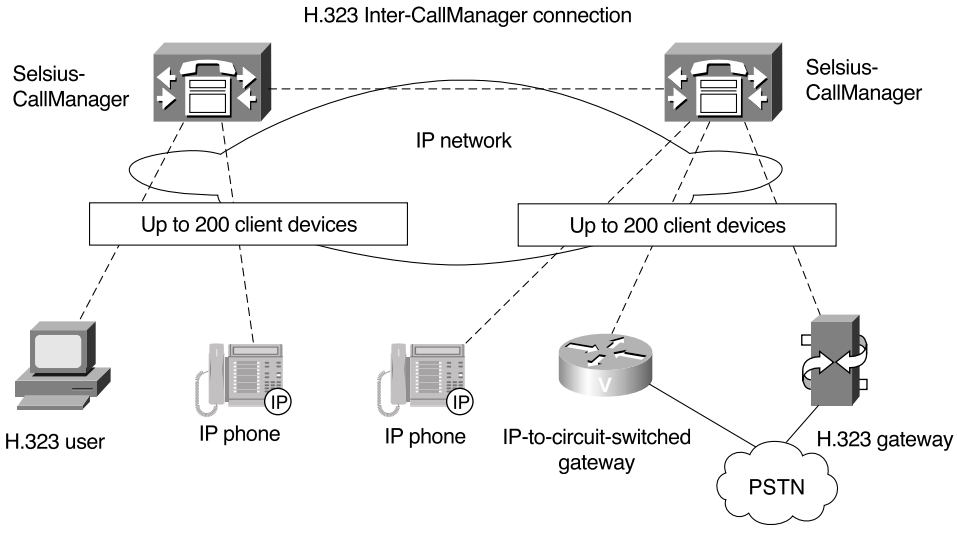
The call processing software, too, changed. It incorporated new code to control the IP phones and gateways. For this purpose, the Skinny Client Control Protocol (SCCP) and Skinny Gateway Control Protocol (SGCP) were invented. In addition, the software supported Microsoft NetMeeting, an application that 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 a C++ based SDL application engine that duplicated all of the benefits that the previous pure SDL environment had provided.

Selsius-CallManager 1.0 was born. It permitted Skinny Protocol 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 *Cisco CallManager in 1997*



2000—Cisco CallManager Release 3.0

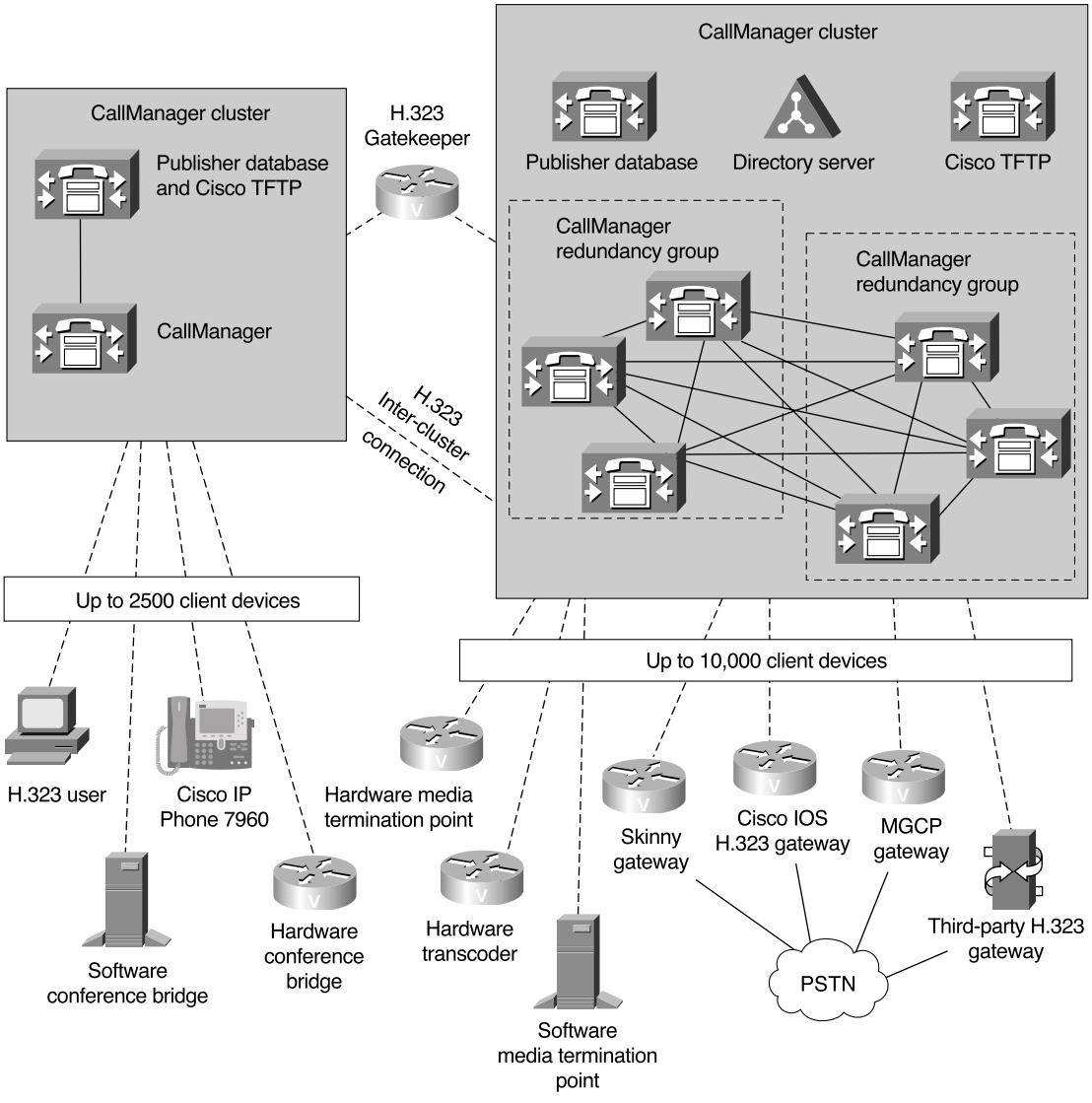
CallManager received a great deal of attention from the marketplace. By 1998, CallManager 2.0 had been released, and Cisco Systems, Inc., had become interested in the potential of the product.

After acquiring the CallManager product as a result of Cisco Systems' acquisition of Selsius Systems in 1998, Cisco concentrated on enhancing the product. Cisco was also simultaneously undertaking a huge design and re-engineering effort to provide both scalability and redundancy to the system. Clustering was introduced, and the Specification and Description Language (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 Phone 7960, which provided a large display, soft keys, and access to 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 Skinny Protocol 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 Cisco CallManager release 3.0.

Figure 1-5 *Cisco CallManager in 2000*



2001—Cisco CallManager Release 3.1

Cisco CallManager release 3.1 builds on the foundation of Cisco CallManager 3.0. The platform supports more gateway devices and station devices, adds enhancements to serviceability, and adds more features. Among the specific enhancements are

- Music on hold (MOH) servers
- Media resource devices available to the cluster, rather than individual CallManager nodes
- Support for digital interfaces on MGCP gateways
- Call preservation between IP Phones and MGCP gateways on server failure
- Generic database support for third-party devices
- Hoteling
- Overlap sending
- 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)
- Support for Cisco IP Phones 7910 and 7940

Cisco-Certified Servers for Running Cisco AVVID IP Telephony

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

Cisco has certified the following servers:

- Cisco MCS-7825-800
- Cisco MCS-7835-1000
- Compaq DL320
- Compaq DL380
- IBM xSeries 330
- IBM xSeries 340

The Cisco MCS-7825-800 and Cisco MCS-7835-1000 ship with an installation disk that contains all of the Windows 2000 services that are required to create a working IP telephony network. The Compaq and IBM servers are hardware-only; you must order a software-only version on CallManager to install on these servers.

Cisco AVVID IP Telephony 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 AVVID IP Telephony deployment.

Cisco offers the following MCS 7800 series servers:

- The *MCS-7825-800 server* is a slim but powerful server, suitable for smaller installations. It ships with either Cisco CallManager or Cisco IP/IVR. Through Cisco CallManager's clustering (see the section "Cisco AVVID IP Telephony Clustering") architecture, you can achieve high levels of availability and scalability.
- The *MCS-7835-1000 server* is a high-availability server platform that delivers the high performance required by enterprise networks. In addition to using CallManager clustering to achieve scalability and availability, it features a redundant hot-plug power supply and redundant hot-plug hard drives using RAID-1 disk mirroring to ensure maximum availability. If a power supply or hard drive fails, you can replace it without powering down the server or affecting service. In the case of the hard drive, as soon as you insert a replacement drive, the integrated RAID controller mirrors the data from the primary drive to the new drive without any user intervention.

Table 1-1 presents specifications for these MCS 7800 series servers.

Table 1-1 *Specifications for MCS 7800 Series Servers*

Feature	MCS-7825-800	MCS-7835-1000
Operating System	Windows 2000	Windows 2000
Intel Pentium III Processor	800 MHz	1 GHz
RAM	512 MB	1 GB
Hard Drives	Single 20 GB Fast ATA (7200 rpm)	Dual 18.2 GB Ultra3 SCSI 10,000 rpm
Hardware RAID Controller	No	Yes
Hot-Plug Redundant Power Supply	No	Yes
Size (1 U = 4.3 cm)	1 U	3 U
Maximum IP Phones	500	2500
Participation in Cisco CallManager Clusters	Yes	Yes
Maximum Users in Cluster	2500	10,000
Maximum IP IVR/Auto Attendant ports	30	48

The Compaq servers that Cisco has certified have the specifications shown in Table 1-2. These servers mirror the Cisco MCS 7800 series servers in function. The Compaq DL320 is suitable for smaller installations, while the Compaq DL380 supports larger clusters.

Table 1-2 *Specifications for Cisco-Certified Compaq Servers*

Feature	DL320	DL380
Operating System	Windows 2000	Windows 2000
Intel Pentium III Processor	800 MHz	1 GHz
RAM	512 MB	1 GB
Hard Drives	Single 20 GB Fast ATA (7200 rpm)	Dual 18.2 GB Ultra3 SCSI 10,000 rpm
Hardware RAID Controller	No	Yes
Hot-Plug Redundant Power Supply	No	Yes
Size (1 U = 4.3 cm)	1 U	3 U
Maximum IP Phones	500	2500
Participation in Cisco CallManager Clusters	Yes	Yes
Maximum Users in Cluster	2500	10,000
Maximum IP IVR/Auto Attendant ports	30	48

At the time of this writing, information on the IBM servers was not yet available.

Windows 2000 Services on Cisco AVVID IP Telephony Servers

Cisco AVVID IP Telephony relies on several Windows 2000 services, of which Cisco CallManager is only one. Cisco AVVID IP Telephony uses the Windows 2000 services described in Table 1-3.

Table 1-3 *Windows 2000 Services that Run on a Cisco AVVID IP Telephony Server*

Name	Description
Cisco CallManager	Provides call signaling and media control signaling for up to 2500 devices.
Cisco IP Voice Media Streaming Application	Provides H.323 media termination, music on hold, and G.711 media mixing capabilities.

continues

Table 1-3 *Windows 2000 Services that Run on a Cisco AVVID IP Telephony Server (Continued)*

Name	Description
Cisco Messaging Interface	Permits Simple Message Desk Interface (SMDI) communications to voice-mail systems over an RS-232 connection.
Cisco MOH Audio Translator	Converts G.711 music source files to G.729a music source files for providing music on hold to G.729a-capable devices.
Cisco RIS Data Collector	Collects serviceability information from all cluster members for improved administrability.
Cisco Telephony Call Dispatcher	Allows users such as switchboard attendants to receive and quickly transfer calls to other users in the organization; provides automated routing capabilities.
Cisco TFTP	Provides preregistration information to devices, including a list of CallManager servers with which the devices are permitted to register, firmware loads, and device configuration files.
Database Notification	A change notification server and watchdog process that ensures that all Cisco AVVID IP Telephony applications on a server are working properly.
Publisher Database	Serves as the primary read-write data repository for all Cisco AVVID IP Telephony 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 AVVID IP Telephony applications running on the server, should the applications lose connectivity to the Publisher database.

Client Devices that Cisco CallManager Supports

In a Cisco AVVID IP Telephony 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 telephone handsets. CallManager offers four different types of handsets, which it controls with Skinny Protocol. The Cisco IP Phone 7910 is an entry-level station with a single line appearance and a two-line display. The Cisco IP Phone 7935 is a speakerphone console designed for use in conference rooms. The Cisco IP Phone 7940 supports two line appearances and offers a more powerful nine-line display with soft keys and status lines. The Cisco IP Phone 7960 supports up to six line appearances and has the same display as the Cisco IP Phone 7940.

However, station devices need not be physical handsets. CallManager also supports H.323 user clients, such as NetMeeting, which runs as a software application on a user's PC, and Cisco SoftPhone, which connects to CallManager using the TAPI application interface.

Chapter 3, "Station Devices," goes into more detail about station devices.

- **Gateway devices**—Gateway devices provide access from one telephone system to another. This access can be from one network of CallManager servers 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.)

  CallManager supports a wide range of gateway devices. The Cisco 2600, 3600, and 5300 series routers can connect to CallManager using the H.323 protocol. The Cisco VG200 gateway communicates to CallManager using MGCP. The Cisco Catalyst 4000 and 6000 switches offer a set of Voice Interface Cards (VICs) that communicate with CallManager using MGCP also in 3.1.

  Each gateway type manages a set of traditional telephony interfaces. These interfaces can be analog interfaces (the same type of telephone interface that probably runs into your home), digital interfaces such as T1 and E1 Call Associated Signaling (CAS), or any of eight flavors of ISDN Primary Rate Interface (PRI).

  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 Skinny Protocol. Four 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 4000 and 6000 platforms offer a blade that performs transcoding functions. *Blades* are cards that are the width of the chassis that they are going into, and they contain the digital signal processors (DSPs) that perform codec conversion and media mixing.

— **Unicast conferencing devices**—These exist to permit Ad Hoc and Meet-Me conferencing. When an endpoint wishes to start a multiparty conversation, all of 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 4000 and 6000 platforms offer a blade that performs mixing functions. In addition, the Cisco IP Voice Media Streaming Application 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. Older H.323 devices 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. The Catalyst 4000 and 6000 platforms offer blades that perform media termination functions, and the Cisco IP Voice Media Streaming Application 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 music on hold device. For as long as you keep the user on hold, the music on hold 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.

Table 1-4 provides a comprehensive (at the time of this writing) list of the devices that CallManager supports.

Table 1-4 *Client Devices that Cisco CallManager Supports*

Name	Type	Description
Cisco IP Phone 7910	Station	Single-line appearance phone with 2-line black-and-white alphanumeric display
Cisco IP Phone 7940	Station	Dual-line appearance phone with 9-line grayscale graphical display
Cisco IP Phone 7935	Station	Speakerphone console with alphanumeric display designed for use in conference rooms
Cisco IP Phone 7960	Station	6-line appearance phone with 9-line grayscale graphical display
Microsoft NetMeeting	Station	Windows-based H.323 software client application
Cisco SoftPhone	Application	Windows-based TAPI software client application
Cisco 1750 Gateway	Gateway	H.323 gateway FXS, FXO, and E&M analog interfaces
Cisco 2600 Series Gateways	Gateway	H.323 gateway with FXS, FXO, and E&M analog interfaces, and T1 and E1 CAS, user- and network-side PRI digital interfaces
Cisco 3600 Gateway	Gateway	MGCP and H.323 gateway with FXS, FXO, and E&M analog interfaces, and T1 and E1 CAS, user- and network-side PRI digital interfaces
Cisco 3810 V3 Gateway	Gateway	H.323 gateway with FXS, FXO, and E&M analog interfaces, and T1 and E1 CAS digital interfaces
Cisco 5300 Series Gateways	Gateway	H.323 gateway with T1 and E1 CAS, user- and network-side PRI digital interfaces
Cisco 7200 Gateway	Gateway	H.323 gateway with T1 CAS, user- and network-side PRI digital interfaces
Cisco DT-24+	Gateway	MGCP gateway with user- and network-side PRI digital interfaces
Cisco DE-30+	Gateway	MGCP gateway with user- and network-side PRI digital interfaces
Cisco Catalyst 4000	Gateway	A platform for which blades controlled with MGCP are available that provide FXS, FXO, and E&M analog interfaces, T1 and E1 CAS, and user- and network-side PRI digital interfaces

continues

Table 1-4 *Client Devices that Cisco CallManager Supports (Continued)*

Name	Type	Description
Cisco Catalyst 6000	Gateway	A platform for which blades controlled with MGCP are available that provide FXS analog interfaces and user- and network-side PRI digital interfaces
Cisco VG200 Gateway	Gateway	MGCP gateway with FXS and FXO analog interfaces and T1 CAS, user- and network-side PRI digital interfaces
Cisco Catalyst 4000	Media	A platform for which blades controlled with Skinny Protocol are available that provide conferencing, transcoding, and media termination
Cisco Catalyst 6000	Media	A platform for which blades controlled with Skinny Protocol are available that provide conferencing, transcoding, and media termination
Cisco IP Voice Media Streaming Application	Media	G.711 conferencing and media termination software application

Call Establishment in a Cisco AVVID IP Telephony Network

Although a circuit-based system relies on a switch card to forge a media connection between two devices, a packet-based system uses no switch cards at all. Rather, the calling device streams media over the IP network directly to the called device. This point bears repeating, because it is a fundamental difference between circuit-switched and packet-switched systems in the enterprise: In a traditional circuit-switched system, both the signaling path and the media path run into the central cabinet, with the call processing software controlling the media on behalf of the devices by talking to a switch card. In a softswitch, the call processing software terminates the signaling path and coordinates the media session directly with the calling and called devices, which initiate the media exchange on their own. In processing a call, Cisco AVVID IP Telephony performs the following steps:

Step 1 **Call signaling**—An IP telephony device sends a request to CallManager to originate a call. The request contains the address of the destination to be called. CallManager locates the called party, sends a new call event to the called device, and waits for the called device to respond with an answering event.

Step 2 **Media control**—When the called device answers, CallManager determines the details of the media session to be established. CallManager must ensure that the two devices can communicate with a

common voice-encoding scheme, and it must provide each device with the IP address and port on which the other device has chosen to receive media.

Step 3 **Media exchange**—After the media session is negotiated and the addresses exchanged, each device streams media directly through the IP network to the other device. Unlike a circuit-switched system, CallManager does not bridge the media streams. Media termination is a function of the endpoints themselves.

Figure 1-6 illustrates a comparison between the circuit-switched and packet-switched call models.

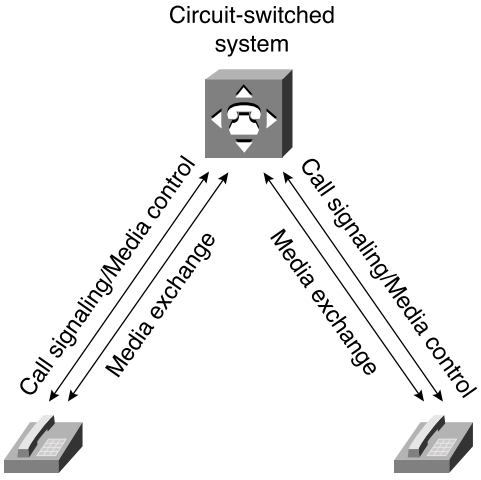
Using the IP network as a virtual matrix offers some remarkable benefits. The Internet is an IP network that spans the globe. A computer on the Internet can talk to its neighbor as easily as it talks to one that is 1000 miles away. Similarly, without the need to connect circuits one leg at a time across 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 of 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 is able to 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 1-7 offers a comparison.

Another major benefit of CallManager is that it resides on the same network as your data applications. The Cisco AVVID IP Telephony 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 standard or open interfaces. This coresidency 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 wish to customize the services available to your voice devices.

Finally, CallManager interacts with IP devices on the network using standard or open protocols, which allows you to mix and match equipment from other vendors when building your voice network. For devices, CallManager supports the open Skinny Protocol to phones, gateways, and transcoding devices; MGCP to gateway devices; and H.323 to user and gateway devices. For server applications, CallManager supports TAPI and JTAPI.

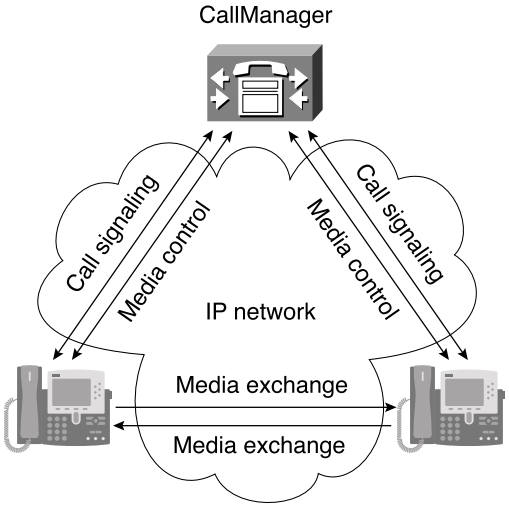
Figure 1-6 *Circuit-Switched Call vs. Packet-Switched Call*



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.



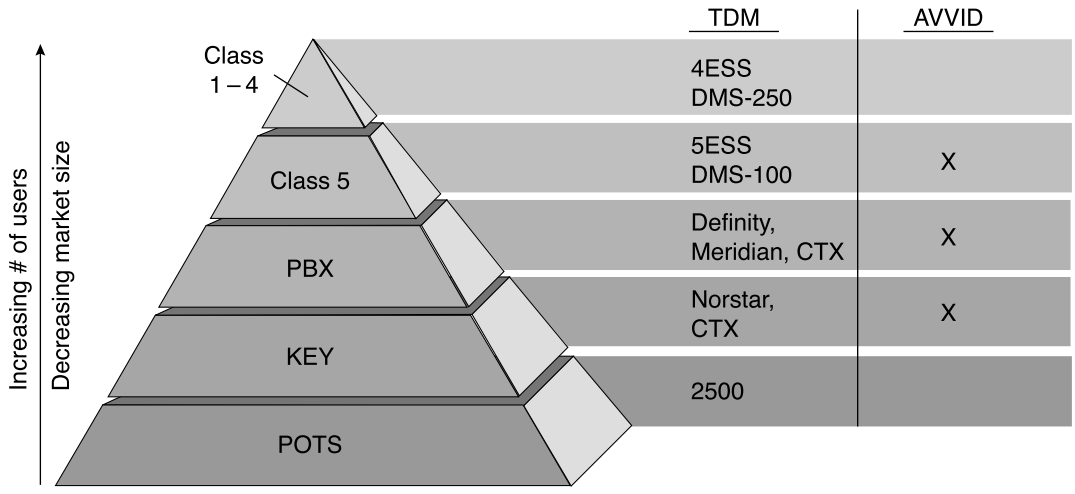
Phones connect to CallManager through a network of routers.

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 from the call signaling. Call signaling and media control are centrally managed, but the high-bandwidth media is distributed.

Figure 1-7 *Cisco AVVID IP Telephony Scalability*



Cisco AVVID IP Telephony 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 AVVID IP Telephony 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 the call processing component of a Cisco AVVID IP Telephony network: Cisco CallManager. Up to eight servers can cooperatively manage the call processing for the enterprise. Such a set of networked servers is called a CallManager cluster. Clustering helps provide the wide scalability of a Cisco AVVID IP Telephony 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 3.1, clusters can contain up to eight servers, which together can support 10,000 endpoints. If your network serves a smaller number of users, you can buy fewer servers. (Cisco CallManager can support larger networks—up to 100,000 users—through the use of *intercluster trunking*.) 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 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 servers in the cluster appear as one large telephone system, even if your users reside in completely different geographical regions.

Clustering requires a certain amount of bandwidth. Unless a LAN connects two servers in the network, they should not be part of the same cluster. Only if your network includes highly reliable, high-bandwidth—T3 or better—connections between two remote sites should you consider putting cluster members on either side of the connection. Rather, either remote sites should run independent clusters—a model called *distributed call processing*—or devices in remote sites should be managed by a cluster of servers that reside in a central site, a model called *centralized call processing*. Large networks tend to deploy a combination of distributed and centralized call processing systems.

Clustering and Reliability

Clustering provides for high reliability of a Cisco AVVID IP Telephony network. In a traditional telephone network, there is a fixed association 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 AVVID IP Telephony redundancy.

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

Function	PBX	Cisco AVVID
Processor unit	Redundant	Up to 8 servers (1 for the Publisher database service, 1 for Cisco TFTP, 6 for Cisco CallManager)
Media switching	Redundant TDM switch	Distributed IP network (multipath)
Intercabinet interfaces	Redundant	Distributed IP interfaces (multipath)
Intracabinet buses	Redundant TDM bus	Redundant Ethernet buses
Power supplies	Redundant	Redundant
Line cards	Single (usually 24)	Not applicable
Power to phones	In-line (phantom)	In-line (phantom), third pair, or external
Phones	Single interface	Can be triple-homed

CallManager redundancy works differently. The redundancy model differs by Cisco AVVID IP Telephony component. Clustering has one meaning in regard to the database, another meaning in regard to CallManager nodes, and a third meaning in 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 (DLLs) that provide a common access point for data insertion, retrieval, and modification of the database. The database itself is Microsoft SQL Server 7.0 standard edition.

If the database resided 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.

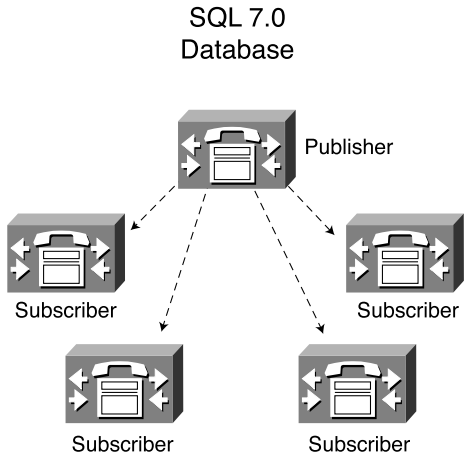
Each CallManager cluster consists of a set of networked databases. One database, the Publisher, provides read and write access for database administrators and for CallManager nodes themselves. For large installations, it is recommended that the Publisher reside on a separate server to prevent database 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 server 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 servers. The Publisher replicates all information other than Publisher Call Detail Records (CDRs). 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, CallManager nodes write CDRs locally and replicate them to the Publisher when it becomes available again. Figure 1-8 shows database clustering.

Figure 1-8 *Database Clustering*



Cisco CallManager Clustering

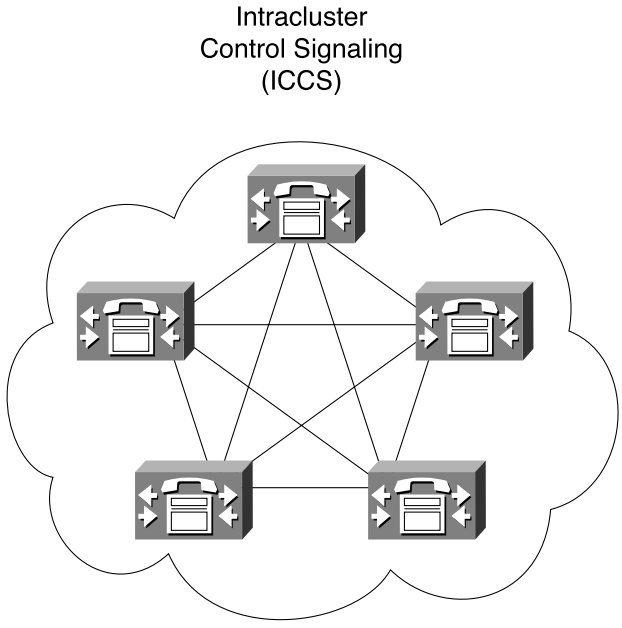
Although the database replicates nearly all information in a star topology (one Publisher, many Subscribers), CallManager nodes replicate a limited amount of information in a fully meshed topology (every server publishes information to every other server).

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 server 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 server 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 1-9 shows the connections between CallManager nodes in a cluster.

Figure 1-9 *Cisco 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, “Call Routing,” 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 AVVID IP Telephony 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 peg. 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 1-10 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.

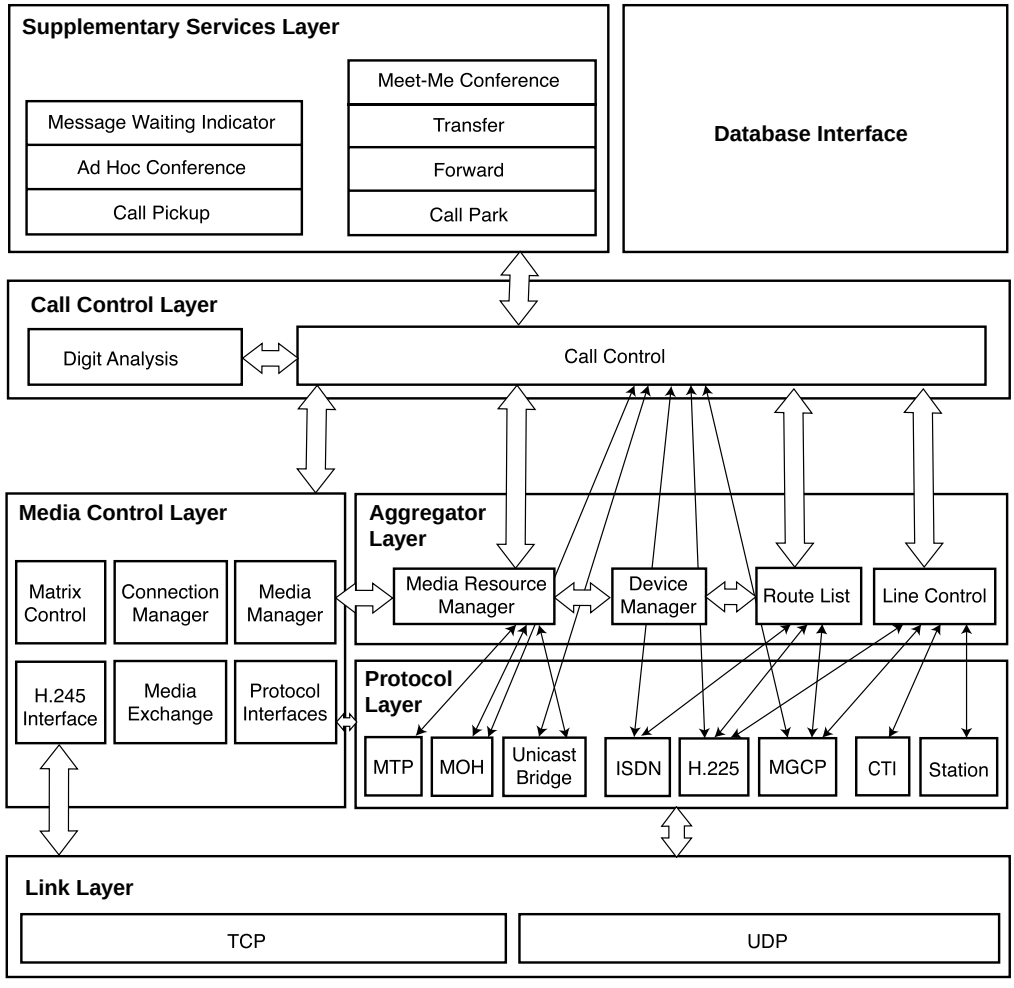
The Link Layer is the most basic. Its function is to ensure that if a device sends a packet of information to CallManager, or CallManager sends a packet of information to a device, the sent packet is received. CallManager uses two methods of communication. The TCP protocol is by far the most commonly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

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 talk 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 of the media connections.

Figure 1-10 *Layers Within Cisco CallManager*



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 call transfer, conference, and call forwarding.

Within each layer, the SDL application engine manages state machines. 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, while another type is responsible for handling individual calls on station devices. These state machines are essentially small event-driven processes, but they do not show up on Microsoft Windows 2000's Task Manager. Rather, the SDL application engine manages these tasks.

These state machines perform work through the exchange of proprietary messages. Before Cisco CallManager release 3.0 was created, these messages were strictly internal to CallManager. With the release of Cisco CallManager 3.0, 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 1-10. 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, though in practice, only two are required.

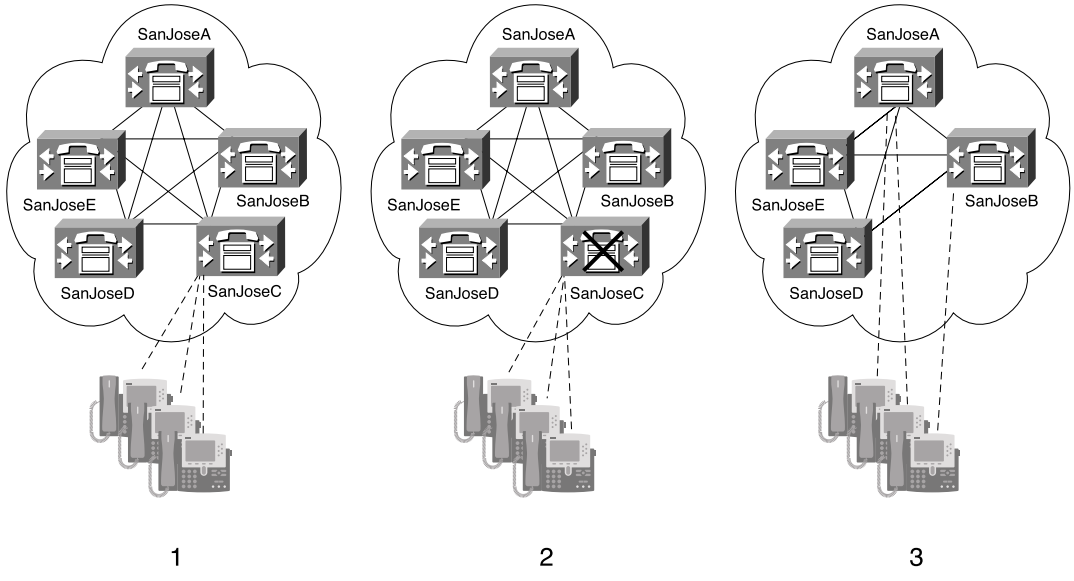
Device Redundancy

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 AVVID IP Telephony 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 servers. If the server to which an endpoint is registered has a software problem, or a network connectivity glitch prevents the endpoint from contacting the server, 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. Once the call terminates and the phone reregisters, the phone regains access to CallManager features.

Figure 1-11 shows an example of this behavior in action. On the left, three phones are homed to CallManager SanJoseC in a cluster, and each has multiple CallManager nodes configured for redundancy. CallManager SanJoseC fails. As a result, 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 1-11 *Device Redundancy*



Deployment of Servers Within a Cisco CallManager Cluster

Each CallManager node in a cluster can support up to 2500 phones. A CallManager cluster can support up to 10,000 phones. Adding multiple clusters permits as many phones as you need.

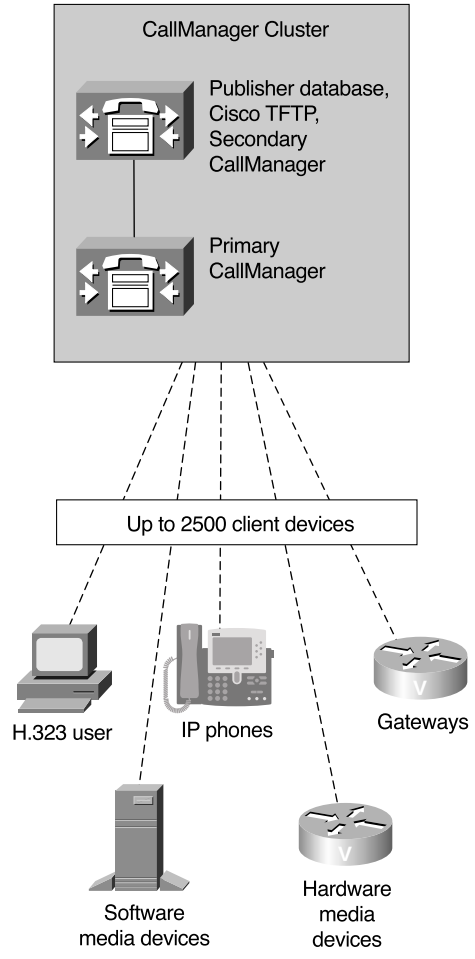
Within a cluster, several strategies exist for deployment of servers. Different strategies exist for clusters up to 2500 users, sites between 2500 and 5000 users, sites between 5000 and 10,000 users, and sites above 10,000 users.

Up to 2500 Users

Two deployment models exist for sites up to 2500 users.

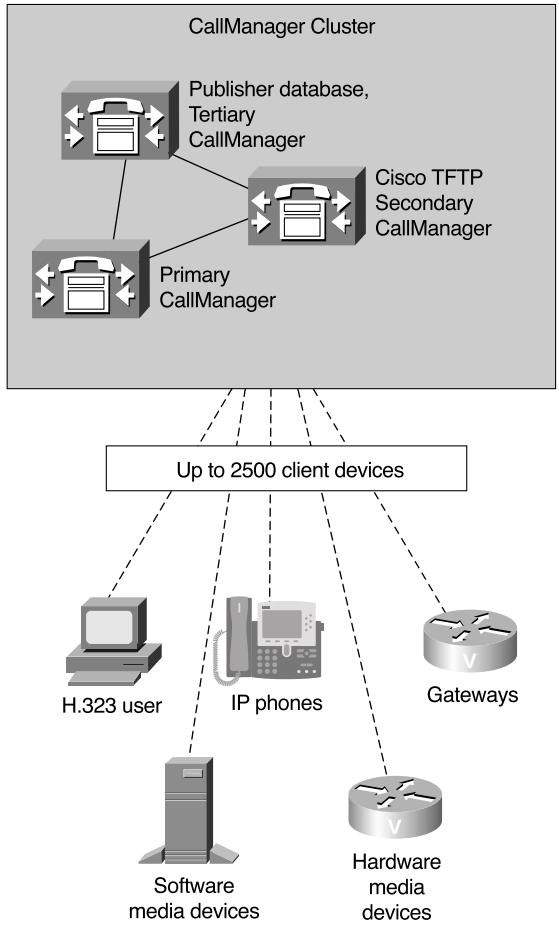
The first model requires two servers. In this model, one server houses the Publisher and Cisco TFTP, and it serves as a backup CallManager. The other server houses a primary CallManager. Under normal operating conditions, all devices in the cluster register to the second server, but if the second server is unavailable, the first server takes over CallManager responsibilities. Figure 1-12 shows this deployment model.

Figure 1-12 *Deployment Model 1 for up to 2500 Users*



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

Figure 1-13 *Deployment Model 2 for up to 2500 Users*



The second deployment model has the advantage of eliminating the risk that database activity on the Publisher database degrades performance of CallManager if the primary CallManager is unavailable. Furthermore, both deployment models permit the use of the locations feature of CallManager as an admissions control mechanism. (Admissions

control is a means by which you can prevent your IP network from carrying so much voice traffic that the quality of individual calls degrades because of dropped packets.)

2500 to 5000 Users

Cluster sizes of between 2500 and 5000 users require four servers. One server houses the Publisher database and Cisco TFTP. The second server houses a CallManager that serves as primary CallManager for the first 2500 users. The third server houses a CallManager that serves as primary CallManager for the second 2500 users. The fourth server runs a CallManager that serves as a backup if either of the other CallManager servers becomes unavailable. Figure 1-14 shows this deployment model.

5000 to 10,000 Users

Cluster sizes of between 5000 and 10,000 users require eight servers. One server houses the Publisher database. Another server houses Cisco TFTP so that devices can get their settings and firmware loads from Cisco TFTP without competing with CallManager for processor resources.

CallManager runs on the remaining six servers. These servers consist of two replication groups of three servers each. The three CallManager nodes within a replication group together control 5000 users. In each replication group, the first server handles 2500 users, the second server handles 2500 users, and the third server provides a backup in case either of the other servers in the group becomes unavailable. Figure 1-15 shows this deployment model.

More than 10,000 Users

When the number of users climbs above 10,000, 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*. These trunks run the H.323 protocol. Figure 1-16 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 1-14 *Deployment Model for 2500 to 5000 Users*

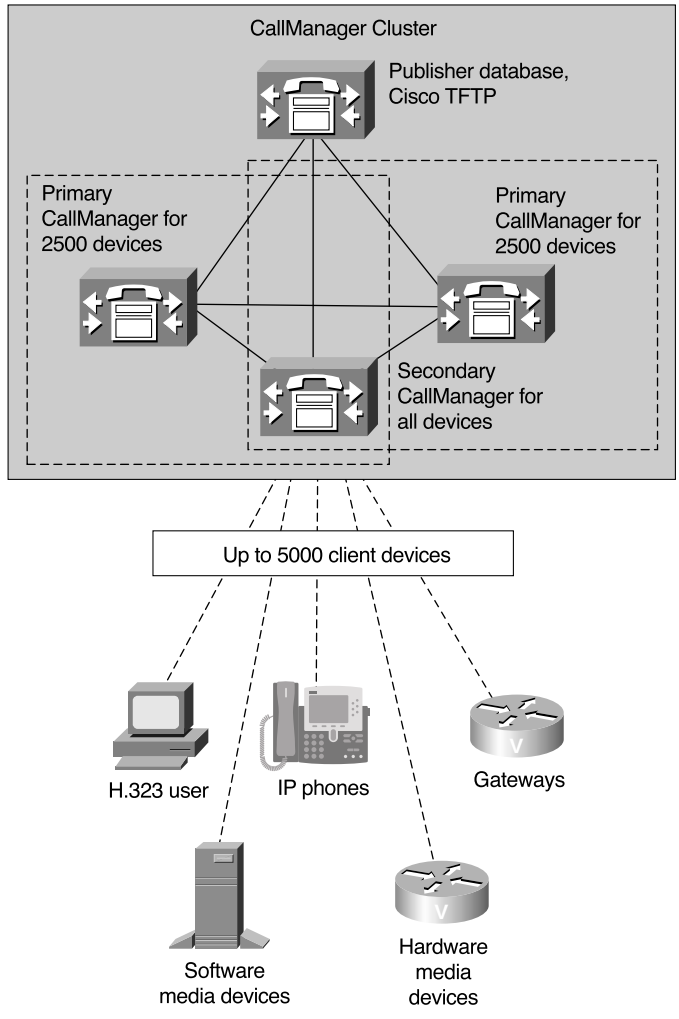


Figure 1-15 *Deployment Model for 5000 to 10,000 Users*

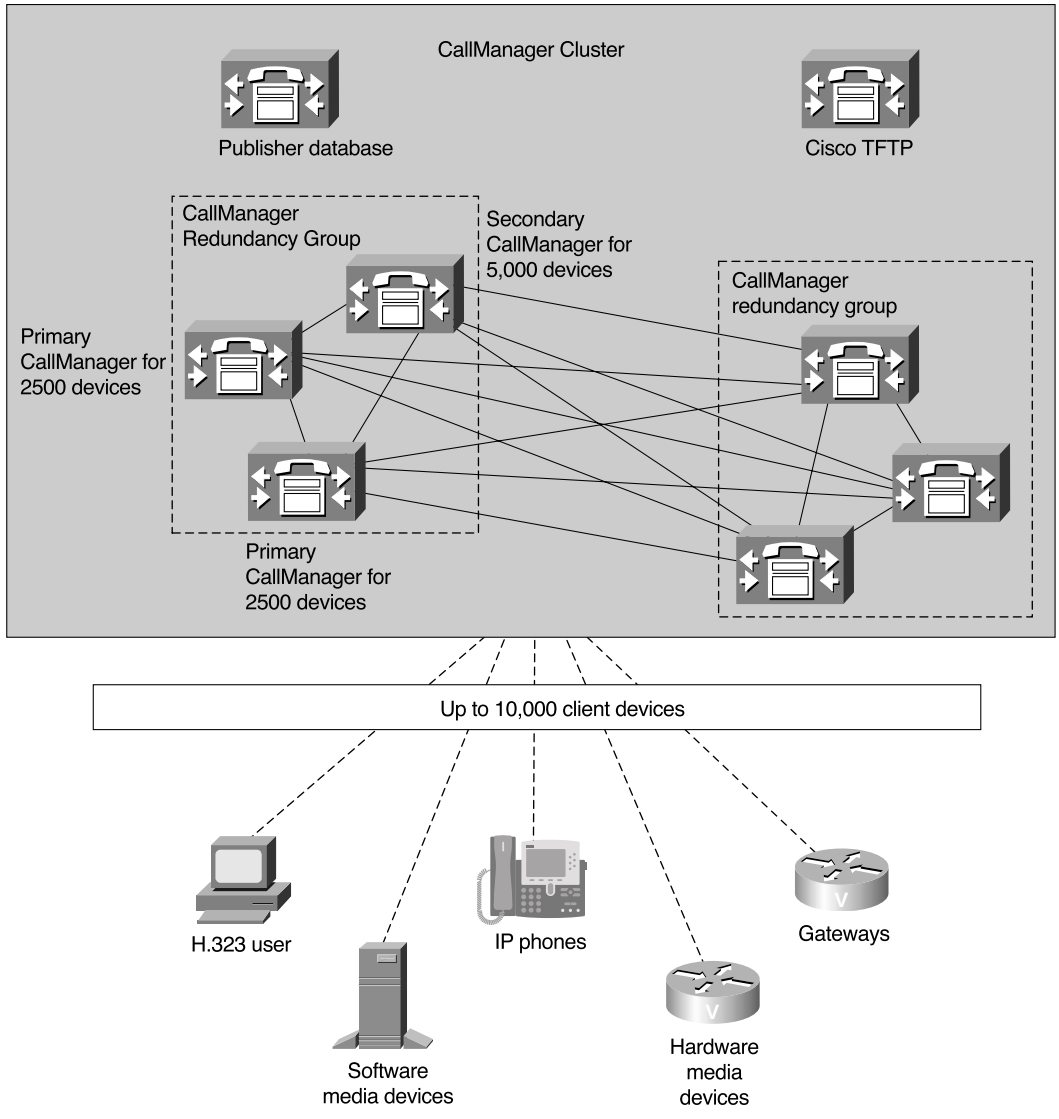
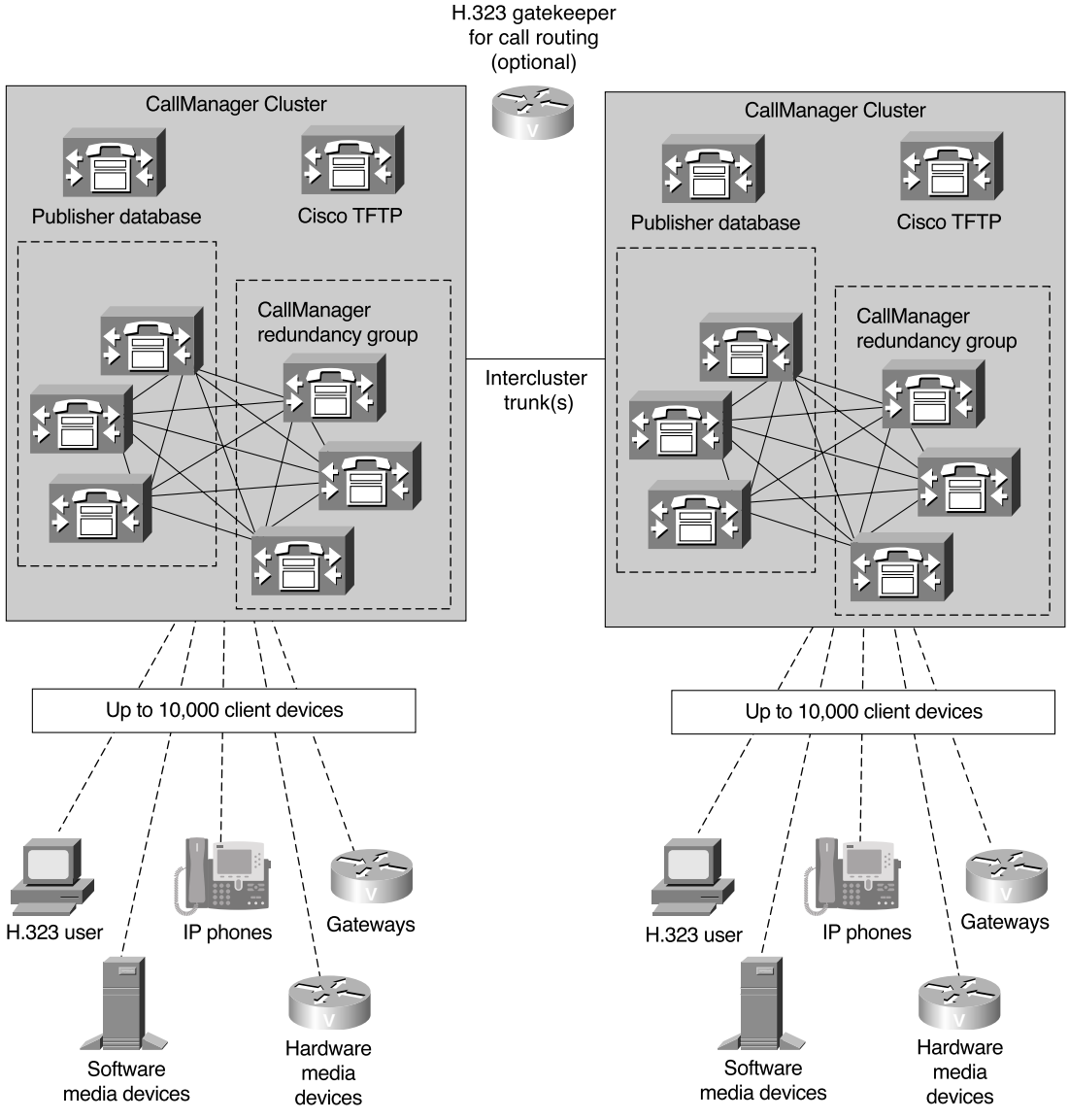


Figure 1-16 *Deployment Model for More than 10,000 Users*



Enterprise Deployment of Cisco CallManager Clusters

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

The excellent Cisco document DOC-7811103, *Cisco IP Telephony Network Design Guide*, which is also publicly available at www.cisco.com/univercd/cc/td/doc/product/voice/ip_tele/, 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 wish 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" 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:

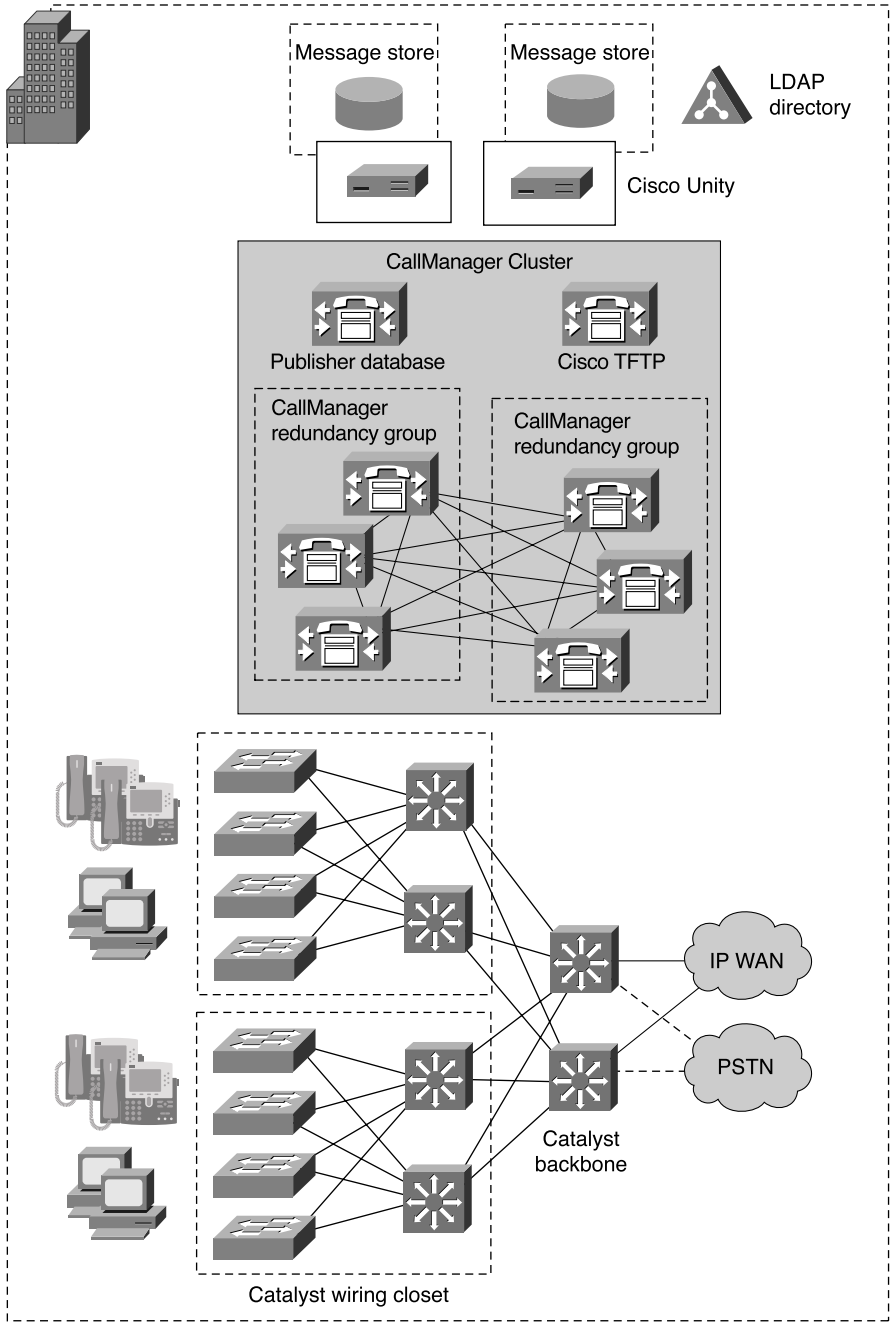
- Single-site model
- Multiple-site model with independent call processing
- Multiple-site IP WAN model with distributed call processing
- Multiple-site model with centralized call processing
- Combined multiple-site model

Single-Site Model

The single-site model consists of a single site or campus served by a LAN. A cluster of up to eight servers (one dedicated to the Publisher database, one dedicated to the TFTP service, and six running CallManager) provides telephony service to up to 10,000 IP-enabled voice devices within the campus. Calls outside of the campus environment are served by IP-to-Public Switched Telephony Network (PSTN) gateways. Because bandwidth is often overprovisioned and undersubscribed on the LAN, there is usually no need to worry about admissions control.

Figure 1-17 presents a picture of the single-site model.

Figure 1-17 *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 eight servers. Each cluster provides telephony service for up to 10,000 IP-enabled voice devices within a site. Because bandwidth is often over-provisioned 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 1-18 presents a picture of 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 instead of 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 gateways, such as Cisco 2600, 3600, and 5300 series routers, under H.323 gatekeeper control. Each cluster provides telephony service for up to 10,000 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 also guarantees that remote sites retain survivability should the IP WAN fail. Using an H.323 gatekeeper allows you to implement a quality of service (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, music on hold (MOH) servers, and transcoders, you must deploy these resources in each site. Figure 1-19 presents a picture of the multiple-site IP WAN model with distributed call processing.

Figure 1-18 *Multiple-Site Model with Independent Call Processing*

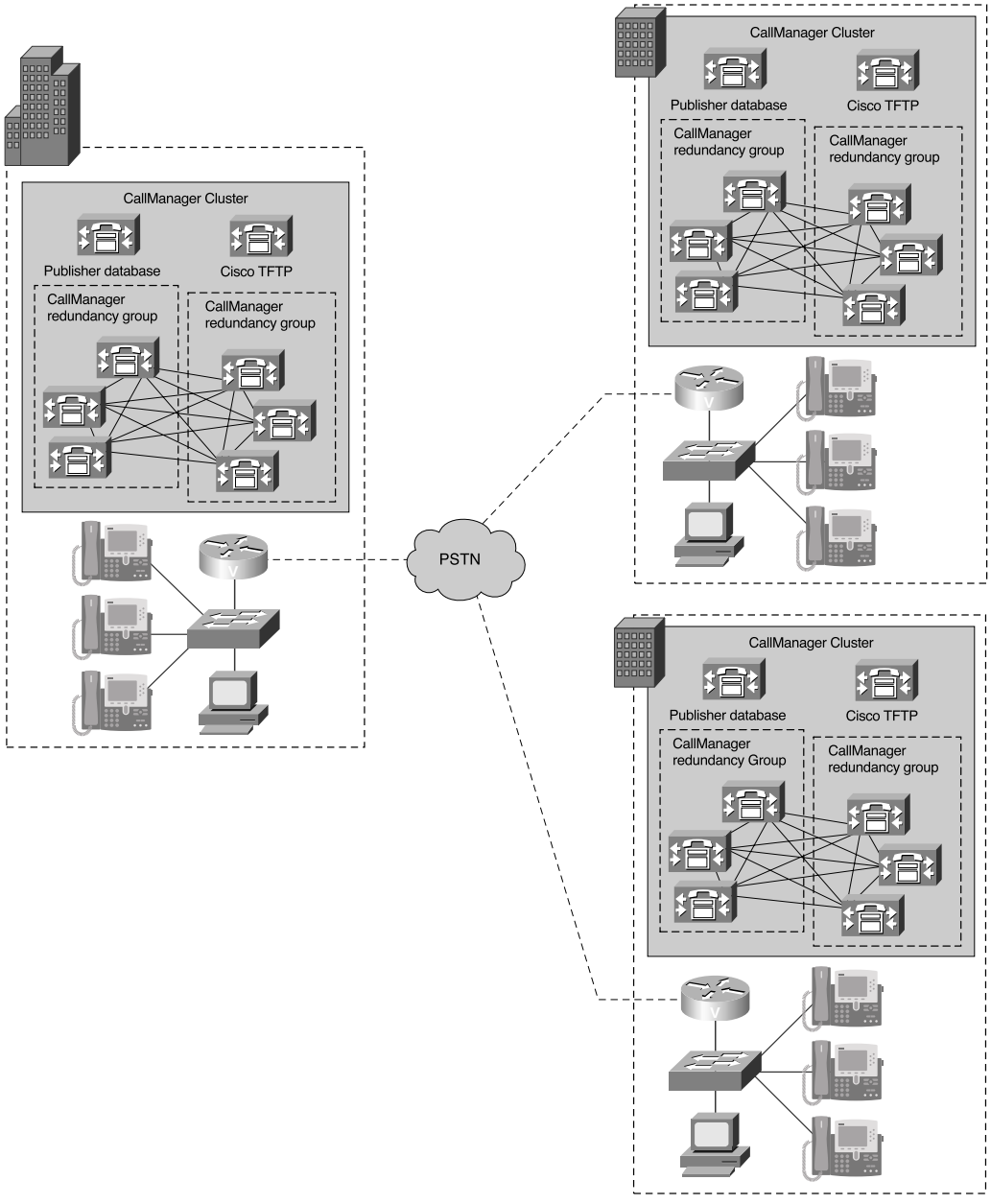
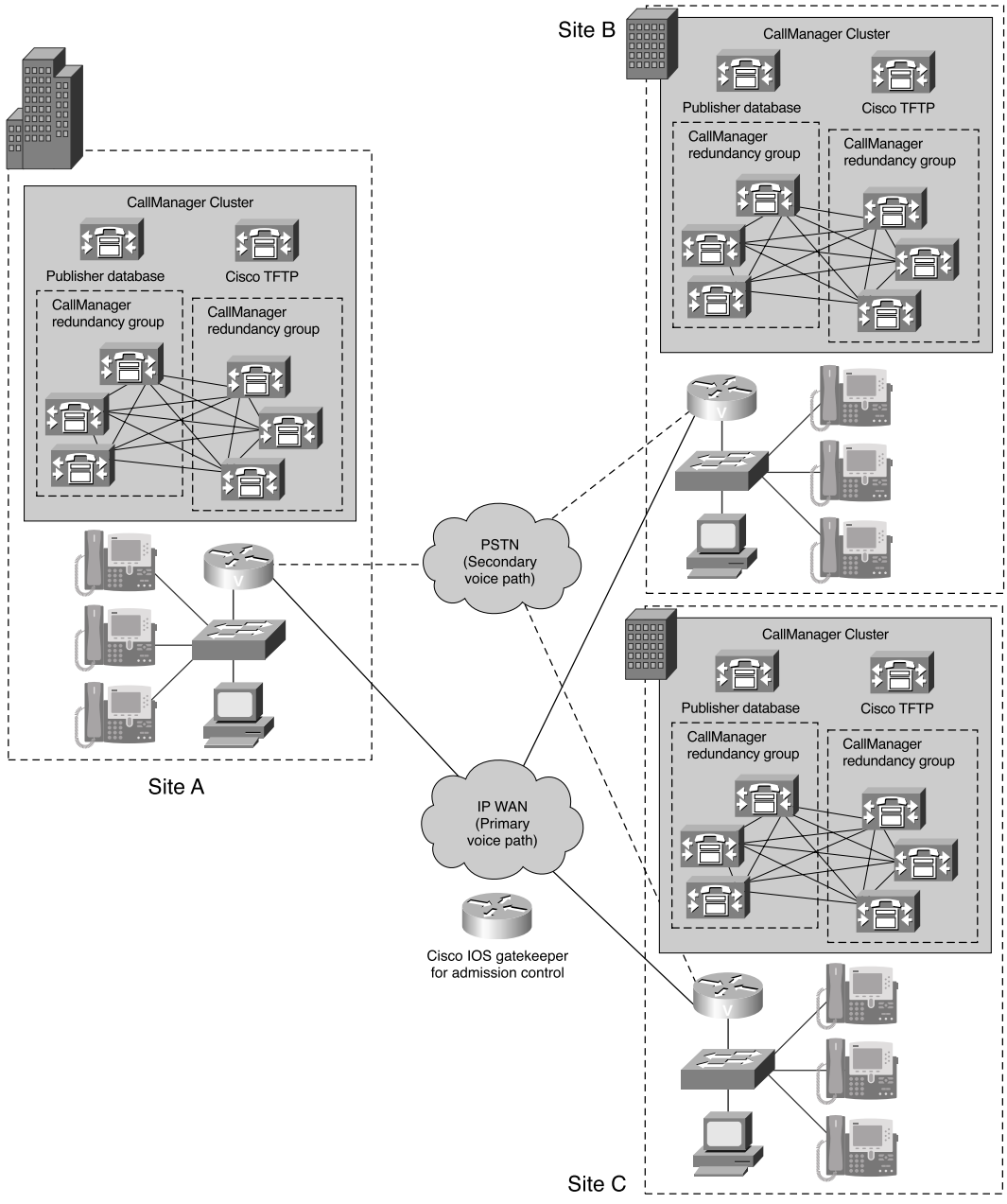


Figure 1-19 *Multiple-Site IP WAN Model with Distributed Call Processing*



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 also 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, while 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. Locations requires that all phones must register with a single CallManager. As a result, when using the multiple-site model with centralized call processing, the maximum supported cluster size consists of three CallManagers, and the secondary and tertiary CallManagers must operate purely as redundant servers. At any one time, the same CallManager node must serve all devices in the cluster.

Deploying a multiple-site model with centralized call processing offers easier administration and true feature transparency between the centralized and remote sites. However, because you cannot deploy a fully meshed cluster, maximum cluster size drops from 10,000 users to 2,500 users. Furthermore, should the IP WAN fail, devices in remote sites will be unable to place or receive calls, unless you configure dial backup.

Because all sites are served by one cluster, you only need to deploy voice mail, conference bridges, and transcoders in the central site, and all remote sites can access these features. Figure 1-20 presents a picture of the multiple-site model with centralized call processing.

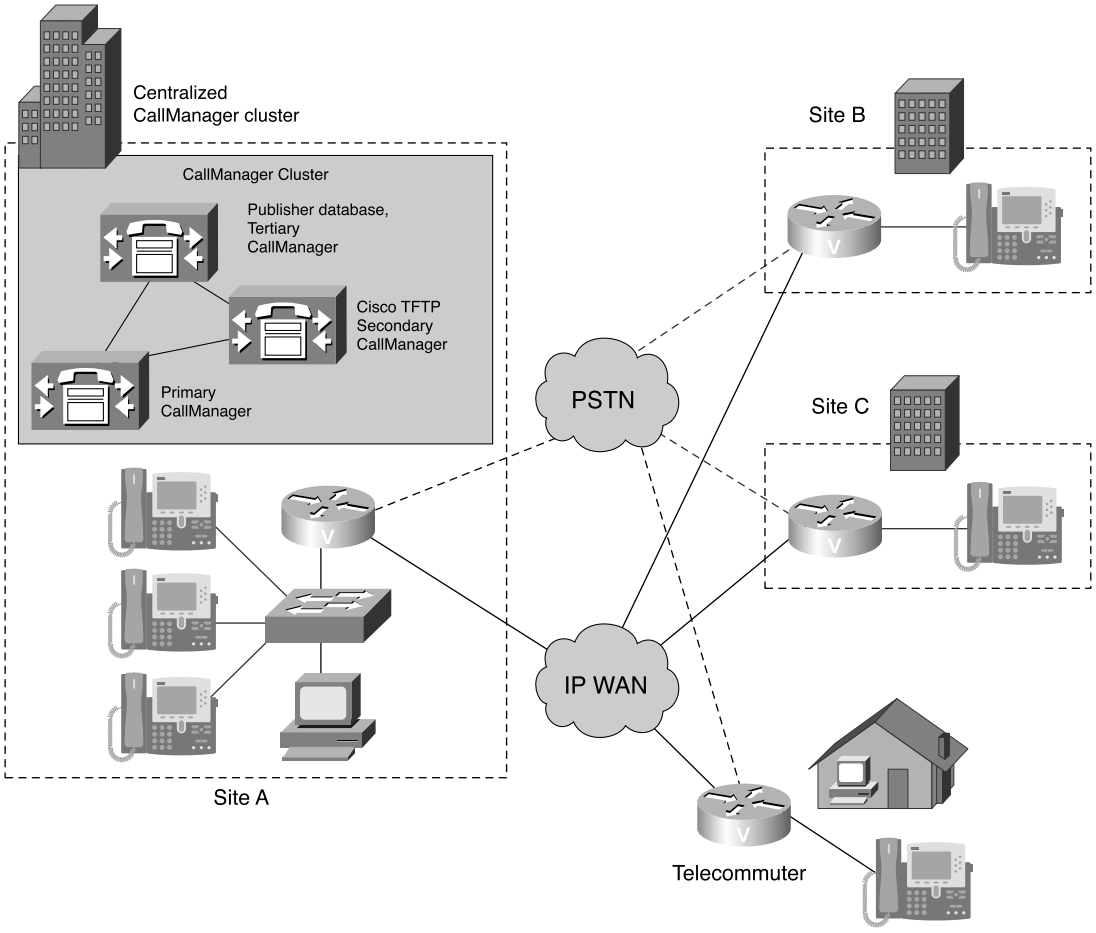
Combined Multiple-Site Model

You can deploy the centralized and distributed models in tandem. If, for example, you have several large sites with a few smaller branch offices all connected by the IP WAN, 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 1-21 presents a picture of the combined multiple-site model.

Quality of Service (QoS)

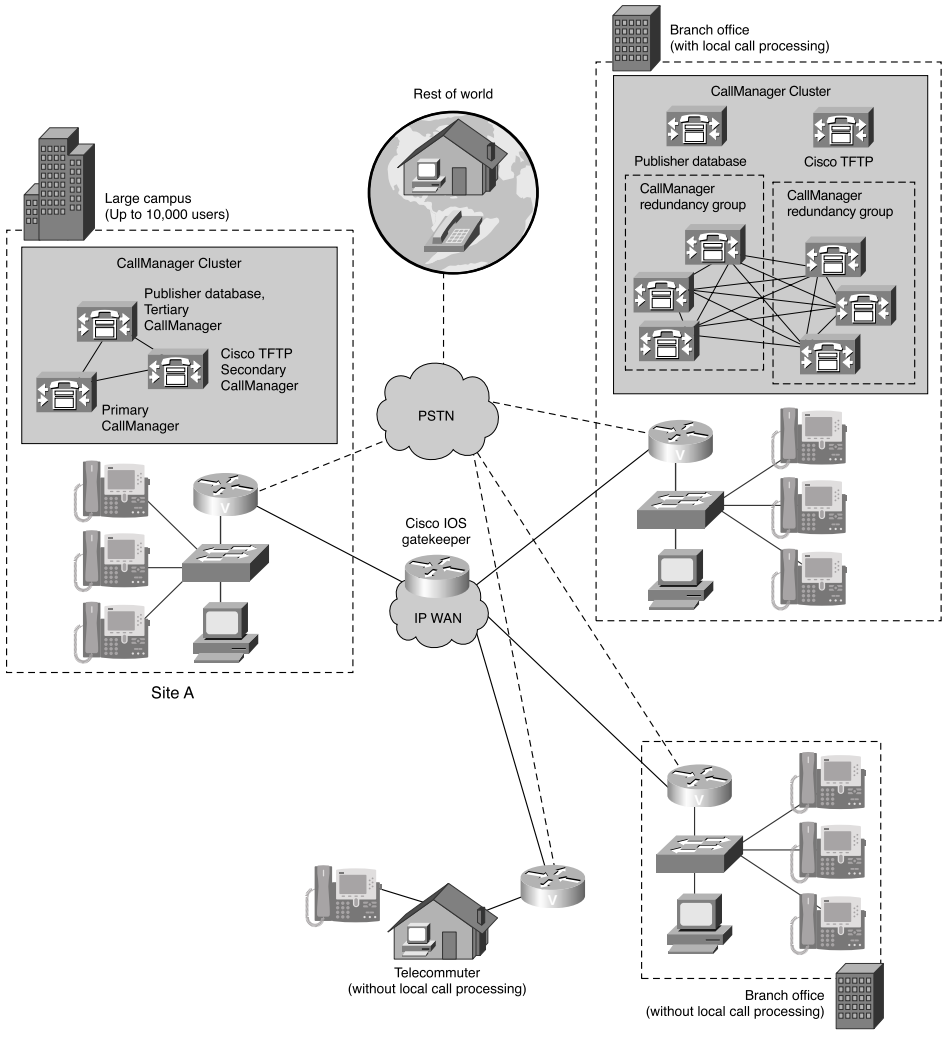
Your network's available bandwidth ultimately determines the number of VoIP calls that your network can handle. As the amount of traffic on an IP network increases, individual data streams suffer packet loss and packet latency. In the case of voice traffic, this can mean clipped, choppy, and garbled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

Figure 1-20 *Multiple-Site Model with Centralized Call Processing*



Unlike data traffic, voice traffic can survive some loss of information. Humans are good at extracting information from an incomplete data stream, while 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.

Figure 1-21 *Combined Multiple-Site Model*



The *Cisco AVVID QoS Design Guide* for CallManager 3.0(5), order number DOC-7811549, which is publicly available at www.cisco.com/univercd/cc/td/doc/product/voice/ip_tele/, covers QoS in a Cisco AVVID IP Telephony network in much greater detail than this section, which just provides an overview.

Admissions control mechanisms prevent an IP network from becoming clogged with traffic to the point of unusability. When a network's capacity is consumed, admissions control mechanisms prevent new traffic from being added to the network.

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:

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

IP Precedence

IP precedence provides a means of traffic classification and is important in configuring your traffic prioritization. By assigning voice traffic a routing priority higher than data traffic, you can ensure that the latency-intolerant voice packets are passed through your IP fabric more readily than latency-tolerant data packets. By assigning voice-related signaling a higher routing priority than data traffic, you guarantee that CallManager quickly provides a dial tone to users who go off-hook.

The Cisco 7900 series phones, as well as the Cisco 12SP+ and 30VIP phones, all send out 802.1Q packets with **class of service** and **type of service** fields set to 5 for the voice stream and 3 for the signaling streams. CallManager permits you to set its **class of service** and **type of service** fields to 3. In contrast, most data devices encode either no 802.1Q information or a default value of 0 for the **class of service** and **type of service** fields.

When present, the **class of service** and **type of service** fields permit 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 **type of service** and **class of service** values 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 1-22 presents an example.

Figure 1-22 *IP Precedence Example*

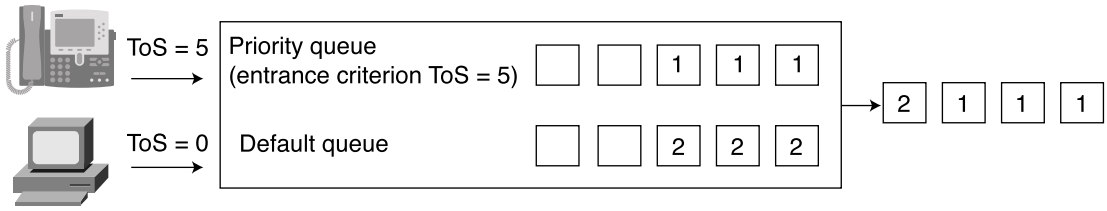


Figure 1-22 depicts two devices that send information through a network router. The Cisco IP Phone 7960 categorizes its traffic with **type of service** 5, while the PC categorizes its traffic with **type of service** 0. The router reads packets from both devices from the network and places them in queues based on the **type of service** field. Packets classified with **type of service** 5 go on a priority queue, while 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 will forward 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.

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.

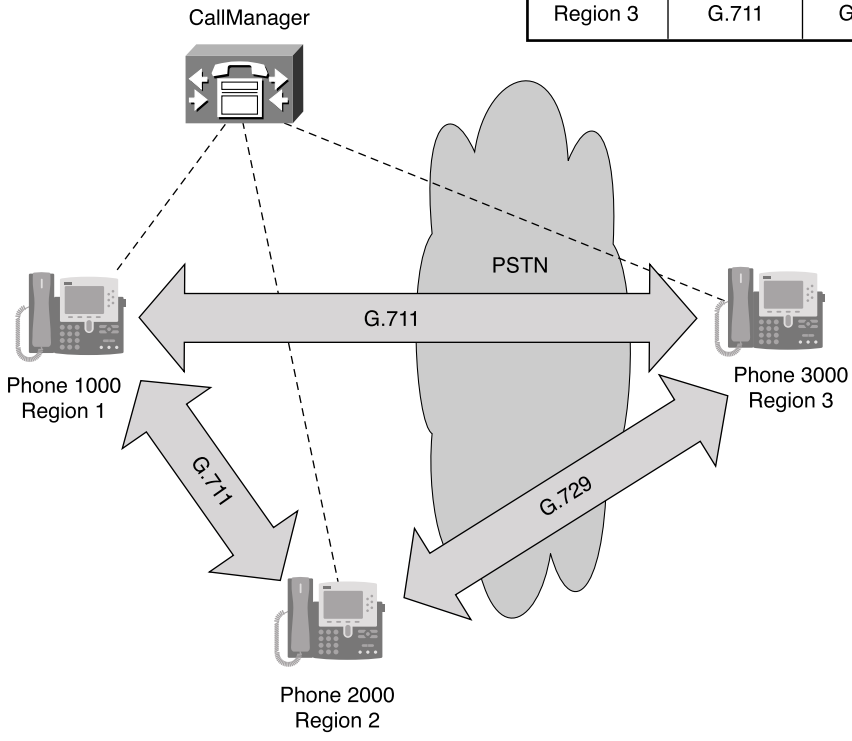
When you define a new region, Cisco CallManager Administration asks you to define the compression type used for calls between devices within the region. You also define, on a region-by-region basis, compression types used for calls between the region you are creating and all other regions.

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 1-23 depicts a configuration that uses three regions to constrain bandwidth between end devices. Phones 1000 and 2000 are in the main campus, while 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 1-23 *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



Cisco CallManager Locations

Locations is 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.

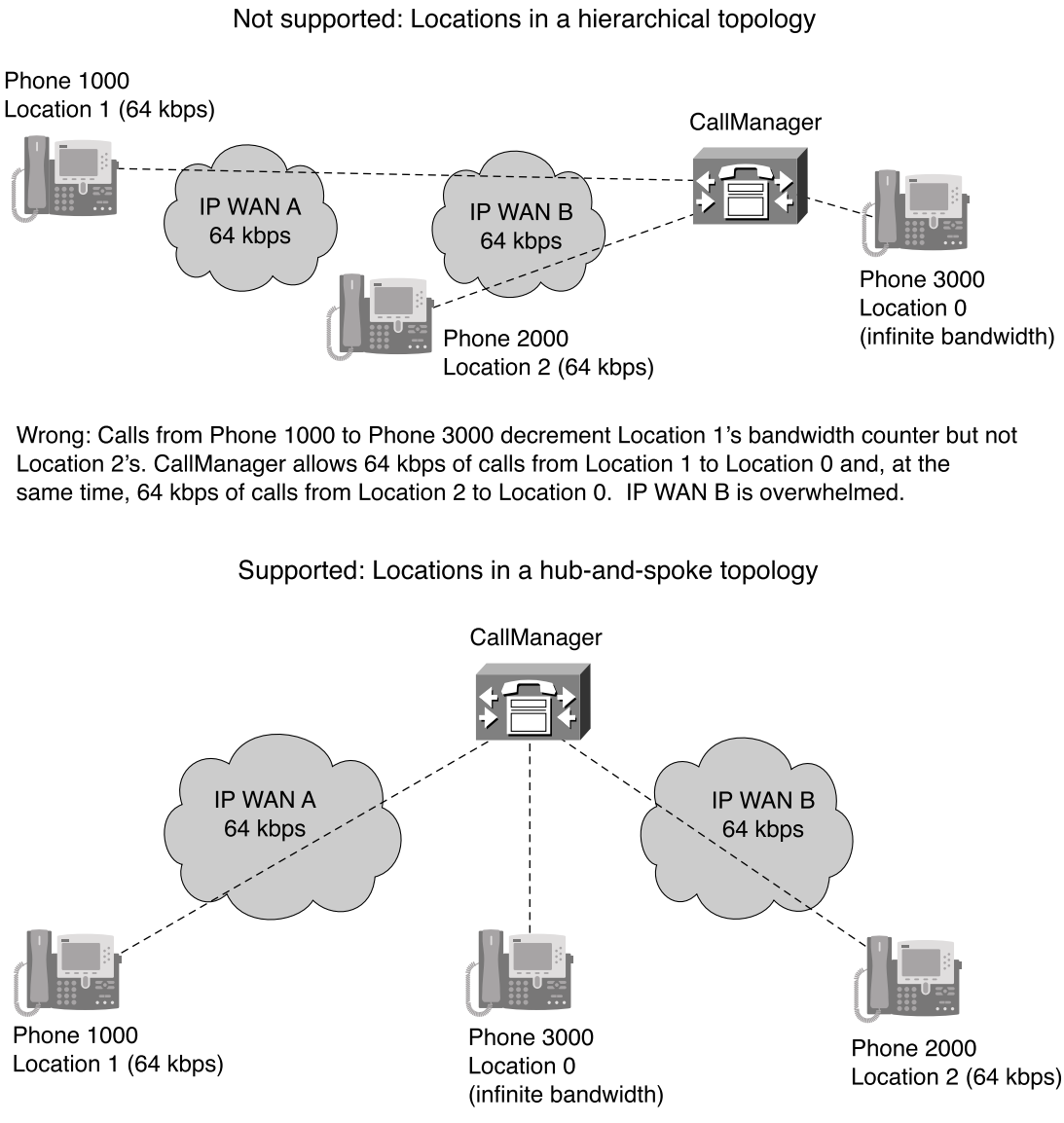
CallManager allows users to place an unlimited number of calls between devices within the same location, but when a user places a call to another location, CallManager temporarily deducts the bandwidth associated with the selected codec from the interlocation bandwidth remaining. When a user's call terminates, CallManager returns the allocated bandwidth to the pool of available bandwidth. Users who attempt to place a call when there is no more bandwidth available receive a fast busy tone. Several design caveats must be considered when using locations.

- Locations requires that all devices in a cluster register with the same CallManager node, with other CallManager nodes used solely for backup purposes. The locations feature is essentially a bandwidth counter that CallManager maintains. If CallManager notes that a call is being placed from one location to another, it decrements the call bandwidth from the bandwidth permitted for each location. The problem is that these counts are maintained on a basis that is strictly per CallManager node. Multiple CallManager nodes in a cluster means multiple independent bandwidth counters, and because any CallManager node in a cluster might serve a particular endpoint's call, there is no way that the counters can remain synchronized.
- Locations 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 1-24 elaborates.

Locations has two other features:

- CallManager permits calls that use media termination devices to complete, even if the bandwidth counter reports that no bandwidth is available.
- Bandwidth is deducted from the bandwidth count only after a call reaches the connected state. As a result, if multiple calls are established across a network link at the same time, locations can allow more calls to be established than the network link supports.

Figure 1-24 *Hub-and-Spoke Topology Restriction*

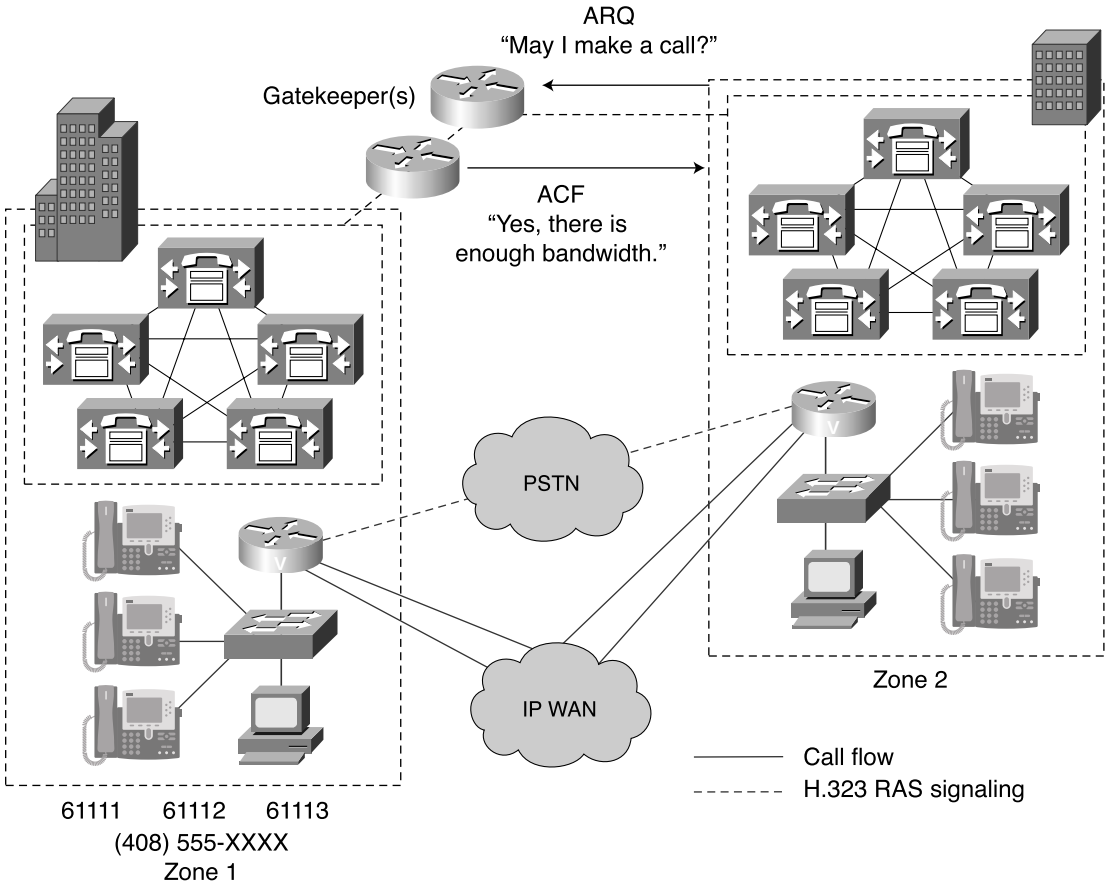


H.323 Gatekeeper

CallManager can be configured to use an H.323 gatekeeper for 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.

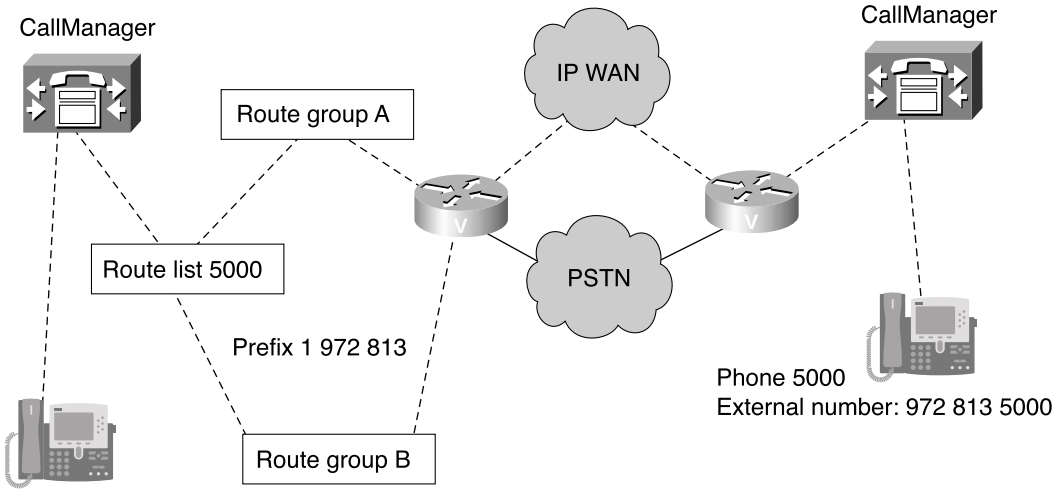
The H.323 gatekeeper associates the requesting CallManager with a zone and can track calls that come into and go out of the zone. If the bandwidth allocated for a particular zone is exceeded, the H.323 gatekeeper denies the call attempt, and the caller hears a fast busy tone. Essentially, an H.323 gatekeeper provides a locations-like functionality for the H.323 domain. However, unlike locations, a chief advantage of an H.323 gatekeeper configuration is that if not enough bandwidth is available to place the call across the IP WAN, you can configure the call to route out a PSTN gateway instead. Figure 1-25 presents a picture of a gatekeeper-enabled configuration.

Figure 1-25 *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 1-26 *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 has provided an overview of Cisco AVVID IP Telephony.

It covered VoIP and how Cisco AVVID IP Telephony differs from traditional telephone systems. It described how you can use VoIP to achieve savings by routing your telephone calls over the IP WAN.

The chapter discussed Cisco CallManager, the heart of Cisco AVVID IP Telephony. The chapter recounted a short history of how CallManager has evolved and discussed the following components of a Cisco AVVID IP Telephony network:

- The Cisco-certified servers on which CallManager runs
- The Windows 2000 services that provide IP telephony in a Cisco AVVID IP Telephony network
- The client devices that CallManager supports

The chapter covered the phases that CallManager goes through to set up a call and described CallManager's clustering strategy for providing high availability and scalability. It described several different deployment models for CallManagers within a cluster.

Finally, the chapter described several methods of deploying clusters to serve both campuses and campuses with remote offices. In addition, it summarized the methods of traffic classification, traffic prioritization, and admissions control (both locations-based and gatekeeper-based) by which you can guarantee good voice quality in your network.