NEAR EAST UNIVERSITY

FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

COM400

VOICE AND VIDEO ON INTERNET

SUBMITTED BY: YASIR ALI

SUBMITTED TO: EKREM VAROGLU

JUNE2000

Acknowledgment

First of all I am thankful to gracious Allah, the all Almighty, who enable me to complete this project.

I would like to thanks all of my teacher because of whom I able to complete my graduation especially to Mr. Ekrem Varoglu who help me lot and encouraging me to complete this project

Also I would like to thank my parents, for their encouragement support and prayer for me.

I am also thankful to all of my friend who help a lot to complete this project especially to M.Nauman, Adnan Rizvi, Shahid Butt, Baber Rahman, Hafiz Zullifqar Ali, Syed Jawad Ali, Naveed Mustafa and Rizwan Ahmed ch.

Yasir Ali

ABSTRACT

The Internet is under rapid growth and continuous evolution in order to accommodate an increasingly large number of applications with diverse service requirements. In particular, voice and video on IP is one of the most promising services currently being deployed. Besides the potentially significant cost reduction, multimedia can offer many new features and easier integration with widely adopted Web-based services. Despite these advantages, there still exist a number of barriers to the widespread deployment of multimedia application such as the lack of control architectures and associated protocols for managing calls and video, a security mechanism for user authentication, and proper charging schemes. The most prominent one, however, is how to ensure the QOS needed for voice conversation. The purpose of this project is to survey the state-of-the-art technologies in enabling the QoS and protocols support for voice and video communications in the next-generation Internet. In this project, we first review the existing technologies in supporting voice and video over IP networks, including the basic mechanisms in the IETF multimedia application architecture, and ITU-T H.323-related Recommendations.

Table of contents

Chapter number **Page number**

4
Chap four

5
Chap five

Chapter one

1.1-INTRODUCTION

The project is about voice and video on Internet. In this project first of all I will discuss about the "Internet "

What is Internet? When it was established and who established it. What was the main purpose of Internet at that time and why they needed to establish it. The Internet is under rapid growth and continuous evolution in order to accommodate an increasingly large number of applications with diverse service requirements. While the Internet has served as a research and education vehicle for more than two decades, the last few years have witnessed its tremendous growth and its great potential for providing a wide variety of services. In particular, using the Internet to carry video and phone conversations, known as Internet telephony or voice over IP (VoIP), is taking the telecommunications industry by storm. Not only does it represent the best opportunity so far for companies and Telco's to facilitate voice and data convergence, but it also promises to deliver a new era in cheap telephone calls. Five years ago. Many to be far too unreliable for mass-market deployment regarded Internet telephony. But over the past few years, reliability and quality have quickly improved, and Internet telephony is now one of the fastest growing industries.

Voice on Internet is beginning to link the worlds of data and voice. The goal is to combine the strengths of telephones, telephone lines, computers and computer networks to achieve the benefits of both worlds.

THE first two projects in this area are:

1- The Internet Connection Phone, which lets computer users hold voice communications over networks, using their PCs instead of telephones. It can work over the Internet, using either a LAN or a telephone line to connect to the Internet.

2-The Voice Data Gateway, which takes this concept one step further: it lets computer users use a single telephone line to connect to the Internet and still continue to use the telephone line for normal voice calls.

Eventually, goal is to make the Internet a full function platform for voice communications, in addition to text, graphics and multimedia - and to integrate voice and data communications seamlessly within the same network.

Unfortunately, Internet telephony technology is also relatively immature, with quality and latency still being major issues. They are, however, both being addressed. Voice quality has improved greatly from early versions of the technology, which was characterized by distortions and disruptions in speech. Improved technologies for voice coding and lost packet reconstruction have also yielded products where speech is easy to understand. Latency, a factor that affects the pace of a conversation, is also being addressed. Humans can tolerate about 250 ms of latency before it has a noticeable effect, and voice services over the public Internet today typically exceed this figure. Latency will, however, continue to improve, driven by three factors: improved gateways (developers are just beginning to squeeze latency out of the first generation of products); deployment over private networks — by deploying gateways on private circuits, organizations and service providers can control the bandwidth utilization; and hence latency; Internet development (today's Internet was not designed with real-time communications in mind). The Internet Engineering Task Force (IETF), together with Internet backbone equipment providers, is addressing this with technologies like Resource Reservation Protocol (RSVP), which will let bandwidth be reserved. While it will take some time for the world's routers to be upgraded and operational aspects (e.g., how to bill for high quality of service, QoS) to be resolved

Chapter two

Internet

2.1-The Internet

The Internet, sometimes called simply "the Net," is a worldwide system of computer networks - a network of networks in which users at any one computer can, if they have permission, get information from any other computer (and sometimes talk directly to users at other computers). It was conceived by the Advanced Research Projects Agency (ARPA) of the U.S. government in 1969 and was first known as the ARPANet. The original aim was to create a network that would allow users of a research computer at one university to be able to "talk to" research computers at other universities. A side benefit of ARPANet's design was that, because messages could be routed or rerouted in more than one direction, the network could continue to function even if parts of it were destroyed in the event of a military attack or other disaster.

Today, the Internet is a public, cooperative, and self-sustaining facility accessible to hundreds of millions of people worldwide. Physically, the Internet uses a portion of the total resources of the currently existing public telecommunication networks. Technically, what distinguishes the Internet is its use of a set of protocols called TCP/IP (Transmission Control Protocol/Internet Protocol). Two recent adaptations of Internet technology, the intranet and the extranet, also make use of the TCP/IP protocol.

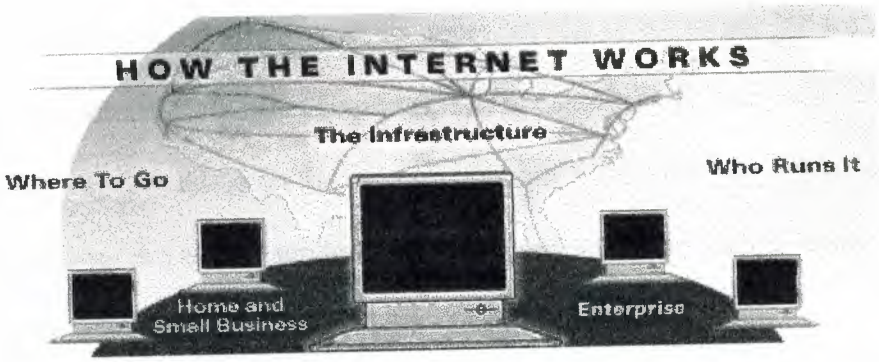


Figure2.1how the internet work

For many Internet users, electronic mail (e-mail) has practically replaced the Postal Service for short written transactions. Electronic mail is the most widely used application on the Net. You can also carry on live "conversations" with other computer users, using IRC (Internet Relay Chat). More recently, Internet telephony hardware and software allows real-time voice conversations.

The most widely used part of the Internet is the World Wide Web (often abbreviated "WWW" or called "the Web"). Its outstanding feature is hypertext, a method of instant cross-referencing. In most Web sites, certain words or phrases appear in text of a different color than the rest; often this text is also underlined. When you select one of these words or phrases, you will be transferred to the site or page that is relevant to this word or phrase. Sometimes there are buttons, images, or portions of images that are "clickable." If you move the pointer over a spot on a

Web site and the pointer changes into a hand, this indicates that you can click and be transferred to another site.

Using the Web, you have access to millions of pages of information. Web "surfing" is done with a Web browser, the most popular of which are Netscape Navigator and Microsoft Internet Explorer. The appearance of a particular Web site may vary slightly depending on the browser you use. Also, later versions of a particular browser are able to render more "bells and whistles" such as animation, virtual reality, sound, and music files, than earlier versions.

2.2- IP (Internet Protocol)

The Internet Protocol (IP) is the method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one address that uniquely identifies it from all other computers on the Internet. When you send or receive data (for example, an e-mail note or a Web page), the message gets divided into little chunks called packets. Each of these packets contains both the sender's Internet address and the receiver's address. Any packet is sent first to a gateway computer that understands a small part of the Internet. The gateway computer reads the destination address and forwards the packet to an adjacent gateway that in turn reads the destination address and so forth across the Internet until one gateway recognizes the packet as belonging to a computer within its immediate neighborhood or domain. That gateway then forwards the packet directly to the computer whose address is specified.

Because a message is divided into a number of packets, each packet can, if necessary, be sent by a different route across the Internet. Packets can arrive in a different order than the order they were sent in. The Internet Protocol just delivers them. It's up to another protocol, the Transmission Control Protocol (TCP) to put them back in the right order.

IP is a connectionless protocol, which means that there is no established connection between the end points that are communicating. Each packet that travels through the Internet is treated as an independent unit of data without any relation to any other unit of data. (The reason the packets do get put in the right order is because of TCP, the connection-oriented protocol that keeps track of the packet sequence in a message.) In the Open Systems Interconnection (OSI) communication model, IP is in layer 3, the Networking Layer.

The most widely used version of IP today is Internet Protocol Version 4 (IPv4). However, IP Version 6 (IPv6) is also beginning to be supported. IPv6 provides for much longer addresses and therefore for the possibility of many more Internet users. IPv6 includes the capabilities of IPv4 and any server that can support IPv6 packets can also support IPv4 packets.

IP networks run over a variety of devices, including simple hubs, high-function routers, and sophisticated policy-enabled switches. These platforms offer a rich set of functions in their hardware and software to perform encryption, sequencing, classification, prioritization, compression, and so on. Driven by the incredible performance of these platforms, IP networks are able to offer an increasingly rich set of functions to the applications that run on them.

Many of these functions---in particular, those related to the delivery of traffic and the broad reach of IP---are considered "fundamental" services today. Functions such as forwarding, sequencing, service location, and so on might have been "exotic" a few short years ago, but today, you will find them in nearly all good quality networking equipment.

A new set of functions, known as Intelligent Network Services, builds on the functions these platforms offer to deliver high-level network intelligence to e-business applications. Intelligent Network Services include voice, video, legacy integration, load balancing, caching, and more.

With Intelligent Network Services in place, information technology professionals can deploy Internet Application Technologies like real-time trading, distance learning, and unified messaging.

The forwarding functions of IP networks are

Security services might use a *specialized security application-specific integrated circuit (ASIC)* to factor large prime numbers or encrypt data securely at sustained data rates.

- Voice and video services use coders-decoders (CODECs), switching, and high-speed cable plants to make efficient use of capacity.

- SNA services rely on link spoofing, tunnels, and other prioritization facilities in devices in order to simulate legacy environments or to present legacy information to enterprise applications (data mining, Web front-ends).
- QoS services leverage differentiated switching hardware that can classify and queue traffic at wire speed.
- Intelligent Network Classification services use technologies such as Network Based Application Recognition (NBAR), Context-Based Access Control (CBAC), Multimedia Conference Manager (MCM) proxy's to uniquely identify network traffic and handle it accordingly.

High-availability applications *use* redundant architectures and fault-tolerant equipment to ensure maximum service availability

2.3-TCP (Transmission Control Protocol)

TCP (Transmission Control Protocol) is a method (protocol) used along with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. While IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the individual units of data (called packets) that a message is divided into efficient routing through the Internet.

For example, when an HTML file is sent to you from a Web server, the Transmission Control Protocol (TCP) program layer in that server divides the file into one or more packets, numbers the packets, and then forwards them individually to the IP program layer. Although each packet has the same destination

IP address, it may get routed differently through the network. At the other end (the client program in your computer), TCP reassembles the individual packets and waits until they have arrived to forward them to you as a single file.

TCP is known as a connection-oriented protocol, which means that a connection is established and maintained until such time as the message or messages to be exchanged by the application programs at each end have been exchanged. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end. In the Open Systems Interconnection (OSI) communication model, TCP is in layer 4, the Transport Layer.

2.4-OSI (Open System interconnection)

OSI (Open Systems Interconnection) is a standard description or "reference model" for how messages should be transmitted between any two points in a telecommunication network. Its purpose is to guide product implementers so that their products will consistently work with other products. The reference model defines seven layers of functions that take place at each end of a communication. Although OSI is not always strictly adhered to in terms of keeping related functions together in a well-defined layer, many if not most products involved in telecommunication make an attempt to describe them in relation to the OSI model. It is also valuable as a single reference view of communication that furnishes everyone a common ground for education and discussion.

Developed by representatives of major computer and telecommunication companies beginning in 1983, OSI was originally intended to be a detailed specification of interfaces. Instead, the committee decided to establish a common reference model for which others could develop detailed interfaces that in turn could become standards. OSI was officially adopted as an international standard by the International Organization of Standards (ISO). Currently, it is Recommendation X.200 of the International Telecommunication Union.

The main idea in OSI is that the process of communication between two end users in a telecommunication network can be divided into layers, with each layer adding its own set of special, related functions. Each communicating user is at a computer equipped with these seven layers of function. So, in a given message between users, there will be a flow of data through each layer at one end down through the layers in that computer and, at the other end, when the message arrives, another flow of data up through the layers in the receiving computer and ultimately to the end user. The actual programming and hardware that furnishes these seven layers of function is usually a combination of the computer operating system, applications (such as your Web browser), TCP/IP or alternative transport and network protocols, and the software and hardware that enable you to put a signal on one of the lines attached to your computer.

OSI divides telecommunication into seven layers. The layers are in two groups. The upper four layers are used whenever a message passes from or to a user. The lower three layers (up to the network layer) are used when any message passes through the host computer. Messages intended for this computer pass to the upper layers. Messages destined for some other host are not passed up to the upper layers but are forwarded to another host. The seven layers are:

Layer 7: The application layer...This is the layer at which communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. (This layer is *not* the application itself, although some applications may perform application layer functions.)

Layer 6: The presentation layer...This is a layer, usually part of an operating system, that converts incoming and outgoing data from one presentation format to another (for example, from a text stream into a popup window with the newly arrived text). Sometimes called the syntax layer.

Layer 5: The session layer...This layer sets up, coordinates, and terminates conversations, exchanges, and dialogs between the applications at each end. It deals with session and connection coordination.

Layer 4: The transport layer...This layer manages the end-to-end control (for example, determining whether all packets have arrived) and error-checking. It ensures **complete data transfer**.

Layer 3: The network layer...This layer handles the routing of the data (sending it in the right direction to the right destination on outgoing transmissions and receiving incoming transmissions at the packet level). The network layer does **routing and forwarding**.

Layer 2: The data link layer...This layer provides error control and synchronization for the physical level and does bit-stuffing for strings of 1's in excess of 5. It furnishes transmission protocol knowledge and management.

Layer 1: The physical layer...This layer conveys the bit stream through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier

2.5-E-mail explanation:

This document describes how electronic mail (e-mail) works. It begins by defining some terms and concepts, which are a vital part of e-mail. It then goes a layer deeper, explaining some lower-level concepts. Several specific applications are then discussed: some briefly, some in great detail.

2.6-High-level Concepts

2.6.1Mail-boxes

A mail-box is a file, or possibly a directory of files, where incoming messages are stored.

2.6.2User Agents

A mail user agent, or MUA, is an application run directly by a user. User agents are used to compose and send out-going messages as well as to display, file and print messages which have arrived in a user's mail-box. Examples of user agents are elm, mailx, mh, zmail, Netscape, ...; more information is provided about these in the Specific Applications section below.

2.6.3-Transfer Agents

Mail transfer agents (MTAs) are used to transfer messages between machines. User agents give the message to the transfer agent, who may pass it onto another transfer agent, or possibly many other transfer agents. Users may give messages to transfer agents directly, but this requires some expertise on the part of the user and is only recommended for experts.

Transfer agents are responsible for properly routing messages to their destination. While their function is hidden from the average user, theirs is by far the most complex part of getting messages from their source to their destination. The most common transfer agent is send mail (1m).

2.6.4-Delivery Agents

Delivery agents are used to place a message into a user's mail-box. When the message arrives at its destination, the final transfer agent will give the message to the appropriate delivery agent, who will add the message to the user's mailbox.

The standard delivery agent for Solaris, starting with 2.5, is mail. Local (1m).

2.7-Low-level Concepts

2.7.1-Character Sets

A character set is simply a mapping of byte values to characters.

The most common character set is US-ASCII, which has 32 non-printable control characters and 96 printable characters, for a total of 128. These 128 characters can be encoded in 7 bits of data, so each 8-bit byte representing one of these characters has the lower 7 bits set to the appropriate value for the given character and the 8th (high) bit set to zero. US-ASCII is therefore considered a single-byte 7-bit character set.

Many European languages have accentuated characters (like the German ü, the French ç and é, the (Swedish?) ø and the Spanish ñ). Such languages are commonly represented by characters sets whose lower half (*i.e.*, values 0 - 127) are identical to those of US-ASCII, and whose upper half (*i.e.*, values 128 - 255) represent these accentuated characters. These are therefore considered single-byte 8-bit characters sets; an example is ISO-8859-1.

Many Asian languages have so many characters that they need multiple bytes to represent them all. They are therefore considered multiple-byte character sets.

2.7.2-Headers & Bodies

Each message consists of two parts. The headers contain information about who authored the message, the intended recipients, the time of creation, the subject of the message, delivery stamps, ... Each header is of the form "*keyword*: *value*", where *keyword* is a special word (like **From** or **Date**) identifying the type of information contained in that header, and *value* is the information itself. More information about message headers can be found in RFC 822 and RFC 1123, section 5.

A blank line always separates the headers from the body.

The body contains the information the sender is trying to communicate. The "message" as most people think of it is really the body of the message.

2.7.3-MIME

For many years, most messages were plain text in the US-ASCII character set, so no structure was needed for message bodies. The recent explosion of messaging in Europe and Asia and the transmission of multi-media messages have brought about such a need.

MIME (Multipurpose Internet Mail Extensions, specified in RFCs 2045 - 2049, especially RFC 2045 and RFC 2046, defines such a body structure. It specifies how a Content-Type header can be used to specify a particular character set or other non-textual data type for a message. For example, the header:

```
Content-Type: text/plain; charset=us-ascii
```

indicates that the message consists of plain text in the US-ASCII character set. MIME also specifies how to encode data when necessary (more on this below). It is the responsibility of the receiving user agent to use this information to display the message in a form that will be understood by the user.

2.8-Transfer Protocols

The language spoken between transfer agents is known as a transfer protocol. There are many in existence; the most common is *SMTP* (Simple Mail Transfer Protocol); also well-known are UUCP (Unix-to-Unix copy) and X.400. This document studies SMTP at length. For further information about SMTP, refer to RFC 821 and RFC 1123, section 5.

2.9-Envelopes and Bodies

SMTP uses the concept of an envelope to transfer messages; this merely contains information about from whom the message originated and to whom it is destined. The originator address is important: in case there is a problem transferring or delivering the message, the originator can be notified.

The SMTP body is the entire message as defined above in Headers & Bodies. So the message headers plus the message body equals the SMTP body. The term *SMTP body* is not used that commonly, but it is important to distinguish it from the message body.

2.10-7-bit data *vs.* 8-bit data

For historical reasons relating to the US-ASCII character set, SMTP is a 7-bit protocol, which means it limits bytes of data sent to use only the low-order 7-bits. If the 8th (high) bit of a byte is set, SMTP dictates that the bit must be zeroed out. In order for a message containing 8-bit data to be transferred without data loss, the message must first be encoded into 7-bit data. As most early e-mail users spoke English, however, and most computers used the 7-bit US-ASCII character set, this was not a problem.

In recent years, however, several factors have increased the need for 8-bit message transfer. As mentioned above, European languages often use 8-bit character sets, and Asian language character sets often require multiple bytes; their transmission is greatly simplified if all 8 bits can be transferred unaltered. Finally, the explosion of multi-media messages like audio and video clips have brought about a two-fold need for 8-bit message transfer: encoding messages into 7-bit data is not only cumbersome, but the resultant encoded message is significantly (typically 33%) larger than the original message.

To meet this need, SMTP has been extended to allow 8-bit data to be properly transferred between consenting transfer agents. The negotiating process used to verify consent is specified in RFC 1869, which describes the general extension mechanism to SMTP (called *ESMTP*), and RFC 1652, which describes the specific extension to allow 8-bit data transfer, called *8BITMIME*. If a transfer agent has a message containing 8-bit data and it cannot negotiate the proper transfer of that data, it must either encode the message into 7-bit data using MIME, or return the message to the sender indicating the reason for the return.

It is no coincidence that MIME and ESMTP have common rationales and goals; they were developed in conjunction with each other towards the same end.

2.11-Routing

RFC 974 describes Mail Routing and the Domain Name System; a brief overview of how send mail implements this is given here.

Mail exchangers (**MX**) records are maintained by domain name servers (DNS) to tell MTAs where to send mail messages. An MX record can be specified for a specific host, or a wild-card MX record can specify the default for a specific domain. The MX record tells an MTA where a message, whose ultimate target is a given host in a given domain, should be sent to next, i.e., which intermediate hosts should be used to ultimately deliver a message to the target host. These MX records vary depending on the domain. To illustrate, here is an an example of how a message from a.eng.sun.com destined for b.ucsb.edu might be routed:

The MTA on a.eng.sun.com looks up the MX record for b.ucsb.edu, which tells it to route the message to venus.sun.com. The MTA on venus.sun.com looks up the MX record for b.ucsb.edu, which tells it to route the message to hub.ucsb.edu. The MTA on hub.ucsb.edu looks up the MX record for b.ucsb.edu, which tells it to route the message directly to b.ucsb.edu. The MTA on b.ucsb.edu recognizes that the message has arrived at its intended destination and processes the message for local delivery.

2.12-send mail specifics

MX records are maintained by DNS only (i.e., not hosts files or NIS). If no MX records are available for a given host, sendmail will try to send to that host directly. Once sendmail determines which host to attempt to send the message to: an intermediate host as indicated by an MX record, or a direct connection to the target host, it uses gethostbyname() to determine the IP-address of the target machine in order to make a connection.

The gethostbyname library routine may use DNS, an /etc/hosts file, or NIS to perform its name-to-IP-address look-up, as configured by the /etc/nsswitch.conf file. N.B.: the host name passed to gethostbyname may have been derived from an MX record if a domain name server is running, even though gethostbyname() may not use DNS to resolve this name's address. Remember that MX records are only available from DNS, and the name service switch does not affect a search for MX records. This is as required by RFC 1123, section 5.3.5. This situation may be most noticeable when DNS is not first in the /etc/nsswitch.conf file. It may then be possible that a host name only in /etc/hosts or NIS be redirected by a wild-card MX record to another host

Chapter Three

Voice on Internet

3.1- Introduction:

Voice on Internet is beginning to link the worlds of data and voice. The goal is to combine the strengths of telephones, telephone lines, computers and computer networks to achieve the benefits of both worlds.

THE first two projects in this area are:

1- The Internet Connection Phone, which lets computer users hold voice communications over networks, using their PCs instead of telephones. It can work over the Internet, using either a LAN or a telephone line to connect to the Internet.

2-The Voice Data Gateway, which takes this concept one step further: it lets computer users use a single telephone line to connect to the Internet and still continue to use the telephone line for normal voice calls.

Eventually, our goal is to make the Internet a full function platform for voice communications, in addition to text, graphics and multimedia - and to integrate voice and data communications seamlessly within the same network.

3.2- Voice Mail

Voice mail provides the basic ability to record, store, and manipulate spoken messages. Callers can leave messages for others that can be retrieved at a later time, so problems arising from time zone differences are reduced. Call recipients (subscribers) can leave detailed greetings that tell callers when they will be available. Businesses can receive orders and deliver information during non-business hours or when no one is available. Organizations can use voice mail to distribute general information efficiently to large numbers of employees or customers.

When implemented with screening options, voice mail systems can give busy users the freedom to choose between answering any call immediately and deferring response to a more convenient time.

The automated attendant function commonly packaged with voice mail performs the duties of an operator/receptionist: supervising transfers, screening calls and offering the caller directory assistance to the proper extension. On the extension side, voice mail systems commonly offer employees the ability to program their voice mail boxes with call-forwarding and paging options. They also allow consultants and visitors to maintain mailboxes which are not linked to any particular extension, but may be used for leaving and retrieving messages from any phone, at any time.

In countries where rotary dialing is still prevalent, the addition of a speech recognition resource makes automated attendant features available to all callers. It does this by augmenting touchtone input, allowing callers to speak the extension numbers of persons they wish to reach. Dial pulse detection (DPD), available on some boards, is another way to accept telephone responses when touchtone is not available.

For low-density systems, the D/41™ family provides four channels of voice processing per slot. The D/41ESC is the correct choice for four ports of voice in countries with high-voltage protection requirements, and for incorporating calling line ID into voice mail applications.

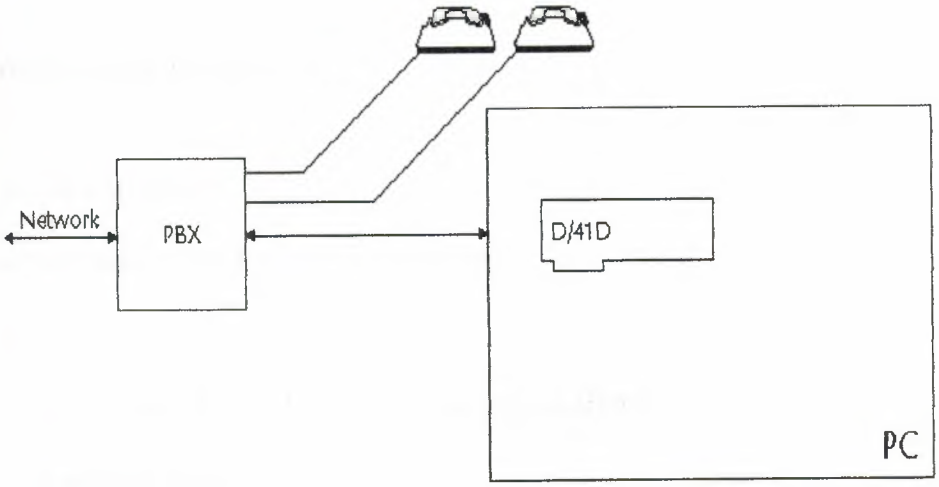


Fig. 3.1 For higher densities, the D/160SC-LS™ provides 16 channels of voice per slot.

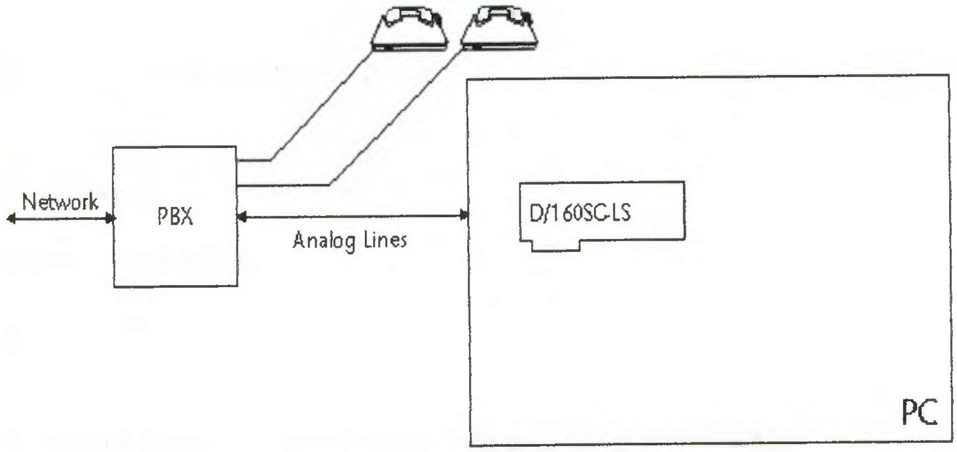


Fig. 3.2: Boards such as the D/42-NS (See p. xx, "Dialogic PBX Integration

Boards") exploit the call control signals of particular switches, allowing application control of such PBX features as

message waiting lights and call transferring.

3.2.1 Component Choices

- Voice boards
- Automatic speech recognition boards
- Fax boards
- Antares board with ASR or text-to-speech algorithm
- Dial pulse detection

Software

- Development packages

Development Tools And Utilities

- Application Generation ToolKits
- Voice Starter Kit
- PromptMaster

- System Density2 to 96 (typical) with additional voice boards

3.2.2 Application Enhancements

- Automated attendant
- Unified messaging
- Preview/predictive dialing
- PBX integration

3.2.3Application Tasks

- Receive a call
- Transfer a call automatically
- Define a menu of choices callers will hear
- Recognize DTMF tones
- Play message to callers
- Screen a call
- Route call to a live operator
- Record, play, forward and delete messages from callers
- Play message to answerer
- Allow users to adjust speed and volume of voice playback
- Copy messages to third parties
- Generate DTMF tones (outdial)
- Establish a call

3.3- Internet telephony

3.3.1- History:

The Multimedia Networking Applications group has been working on IP Telephony Technology since 1995.Our main areas of interest are Internet telephony

and telephony-based applications. The group delivered its first product - IBM Internet Connection Phone (known as ICPhone), an Internet Phone, built on technologies from the Audio\Video group. IP Telephony was identified by HRL as an emerging technology with very vast opportunities, as early as 1995.

The ICPhone used proprietary protocols, since a standard did not exist at the time, and we developed a global directory based on Light Directory Access Protocol (LDAP), which has since, became a standard. End users on one of the IGN/Advantis servers have used this directory without any support since 1996.

At the end of 1996, the IP Telephony development community accepted the ITU H.323 standard. The Haifa Research Lab, in cooperation with Zurich

Research Lab, moved all related activities, such as Internet Phone clients as well as IP/PSTN Gateways and Gatekeepers, to this standard. Voice over IP Technology To achieve high quality voice over IP, several audio technologies were developed in HRL and integrated within HRL IP Telephony components. Below are some of these components:

GSM and G.723.1 codecs

Echo Cancellation

Echo Suppression

Voice Activity Detection

Silence Suppression

Automatic Gain Control

Comfort Noise Generation

Jitter Control

Low Latency

Packet Trucking

DTMF Detection, Generation and Transfer within RTP

Packets

Simultaneous Voice and Data Protocol (SDP) over PPP

3.3.2 Current Activities IP/PSTN Gateway

Haifa Research Lab developed a prototype-based IP/PSTN Gateway during 1997, and a product-based IP/PSTN Scaleable Gateway during 1998.The IP/PSTN Scaleable Gateway is a PRPQ with Telecommunication and Media ISU. The gateway is based on the Direct Talk (DT/6000) product and on the Multi Service Platform (MSP/6000) product. The main features of the gateway are as follows:

Service creation environment based on Direct Talk Programmability tools providing high level Programming as well as APIs to create new Services.

1xT1/E1 to 4xT1/E1 ports in the first version. Both GSM and G.723.1 are deployed on the DSP (IBM SPN256 card).

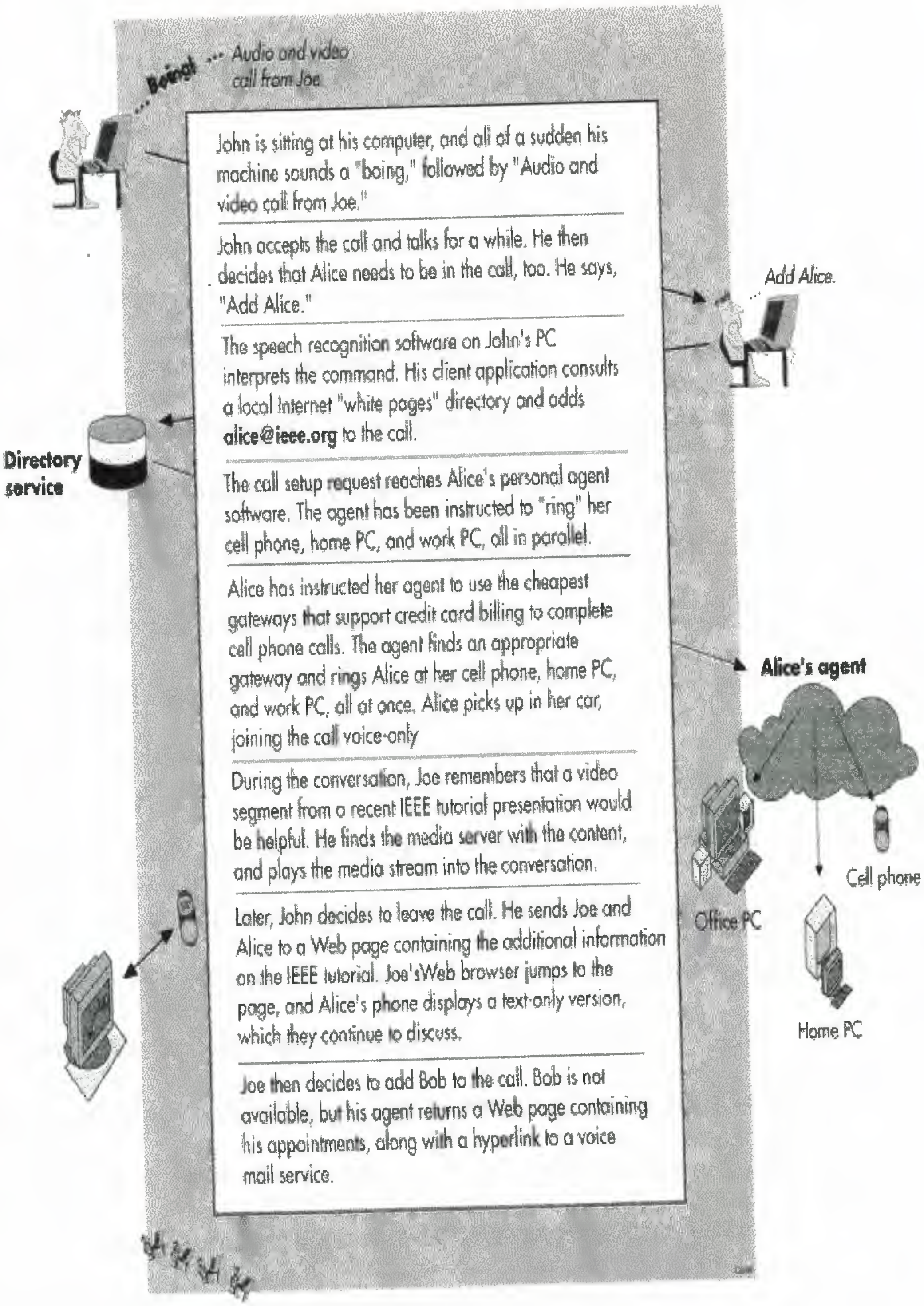
All audio technology features as listed above. Packet trucking which gains up to 70% on Bandwidth traffic as well as TCP/UDP I/O Overhead. The IP/PSTN Scaleable Gateway was developed using Distributed architecture concepts and function

decomposition concepts, i.e. all its components communicate via message on sockets preceding even new emerging standards, such as MGCP.

3.3.3 Internet telephony

Internet telephony was first used as a simple way to provide point-to-point voice transport between two IP hosts, primarily to replace expensive international phone calls. However, the growing interest in providing integrated voice, data, and video services has caused its scope to be expanded. Internet telephony now encompasses a range of services. These services include not only traditional conferencing, call control supplementary services, multimedia transport, and mobility, but also new services that integrate Web, e-mail, presence, and instant messaging applications with telephony. Furthermore, it is generally accepted that Internet telephony and traditional circuit-switched telephony will coexist for quite some time, requiring gateways between the two worlds.

Figure 3.3gives an example scenario of an integrated IP telephony call. The services contained in the call scenario require many protocol components in order to work. In this article we examine the various protocols and discuss how they fit into the broader picture.



John is sitting at his computer, and all of a sudden his machine sounds a "boing," followed by "Audio and video call from Joe."

John accepts the call and talks for a while. He then decides that Alice needs to be in the call, too. He says, "Add Alice."

The speech recognition software on John's PC interprets the command. His client application consults a local Internet "white pages" directory and adds **alice@ieee.org** to the call.

The call setup request reaches Alice's personal agent software. The agent has been instructed to "ring" her cell phone, home PC, and work PC, all in parallel.

Alice has instructed her agent to use the cheapest gateways that support credit card billing to complete cell phone calls. The agent finds an appropriate gateway and rings Alice at her cell phone, home PC, and work PC, all at once. Alice picks up in her car, joining the call voice-only

During the conversation, Joe remembers that a video segment from a recent IEEE tutorial presentation would be helpful. He finds the media server with the content, and plays the media stream into the conversation.

Later, John decides to leave the call. He sends Joe and Alice to a Web page containing the additional information on the IEEE tutorial. Joe'sWeb browser jumps to the page, and Alice's phone displays a text-only version, which they continue to discuss.

Joe then decides to add Bob to the call. Bob is not available, but his agent returns a Web page containing his appointments, along with a hyperlink to a voice mail service.

Figure3.3:integerated internet telephony

First, a signaling protocol is needed to allow calls between participants, and to establish a multimedia session so that the participants can exchange audio and video. We discuss signaling protocols in the next section. The actual audio and video are exchanged between session participants using a transport protocol called RTP, which we discuss after that. Directory access protocols, used to access white pages services, for example, are also important, but we do not discuss them further here, since they are the same as for e-mail service, for example.[1]

During the call illustrated in Figure 1, Joe brings in a media server and instructs it to play a video segment. This is accomplished using a streaming media control protocol, called the Real Time Streaming Protocol (RTSP), which we describe. The intelligent agent concept described in the figure interacts with the signaling protocol to provide advanced services. To realize these agents, we briefly describe a call processing language. We then present the Gateway Location Protocol (GLP), which helps the agent in selecting a gateway for terminating the call from the Internet on the telephone network.

This tutorial does not cover the protocols that allow controllers and signaling gateways -- gateways connecting to the public switched telephone network (PSTN) at the signaling layer -- to communicate. Proposals for such protocols are being discussed within the Internet Engineering Task Force (IETF) at this time, including Media Gateway Control Protocol (MGCP)[2] and Media Description Control Protocol (MDCP).[3] Other protocols, such as those for billing and authentication, are also beyond the scope of this survey.

3.4-SIGNALING PROTOCOLS

Signaling protocols are at the heart of Internet telephony and distinguish it from other services. They play several roles,[4] discussed below.

- User location. If user *A* wishes to communicate with user *B*, *A* first needs to find out where *B* is currently located on the network, so that the session establishment request (below) can reach him. This function is known as *user location*. Users can be in different places at different times, and even reachable by several means at the same time (by work PC or traditional phone). This function is particularly important for users whose PCs do not have a permanent IP address. (Almost all modem connections, including asynchronous digital subscriber line (ADSL) and cable modems, assign addresses to PCs dynamically using the Dynamic Host Configuration Protocol (DHCP).[5]

- Session establishment. The signaling protocol allows the called party to accept the call, reject it, or redirect it to another person, voice mail, or a Web page. (Generally, the terms *call* and *session* are used interchangeably in this article, although session has a somewhat wider meaning, including, for example, a group of hosts listening to an "Internet radio" multicast.)

- Session negotiation. The multimedia session being set up can comprise different media streams, including audio, video, and shared applications. Each of these media streams may use a variety of different speech and video compression algorithms, and take place on different multicast or unicast addresses and ports. The process of session negotiation allows the parties involved to settle on a set of session parameters. This process is also sometimes known as *capabilities exchange*.

- Call participant management. New members can be added to a session, and existing members can leave a session.

- Feature invocation. Call features, such as hold, transfer, and mute, require communication between parties.

Several protocols exist to fill this need. One is International Telecommunications Union (ITU) Recommendation H.323,[6] which describes a set of protocols. The IETF has defined two protocols to perform many of the above tasks: the Session Initiation Protocol (SIP)[7] and the Session Description Protocol (SDP).[8]

3.5 BASIC MECHANISMS IN H.323 protocol:

International Telecommunication Union —Telecommunication Standardization Sector (IITU-T) H.323-related Recommendations for enabling multimedia communications in packet-based networks. We then discuss the IETF QoS framework, specifically the integrated services model (Intserv) and differentiated services (Diff-serv) architecture

H.323 are a series of Recommendations of the ITU-T to enable multimedia communications in packet-switched networks [4]. H.323 is designed to extend the traditionally circuit-based services including audiovisual and multimedia conferencing services into packet-based networks. The Internet Telephony can be based on a subset function of the H.323 for voice only support. Therefore, one of the primary objectives of H.323 is the interoperability with the existing circuit-switching systems (PSTN and ISDN).

The basic elements defined in H.323 architecture are: terminals, gateways, gatekeepers, and multipoint control units (MCUs), in which the terminals, gateways, and MCUs are collectively referred as endpoints.

A terminal is an end user device, which can be a simple telephone or PC/workstation. Its main responsibility is to participate in H.323-defined communications, including both point-to-point calls and multipoint conferences.

A gateway, as the name suggests, is an intermediate device to provide interoperation between H.323 compliant devices and non-H.323 device in particular PSTN and ISDN devices. The main functionalities contain the translation of signaling media encoding, and packetization. There exist number of different types of gateways, for example gateways for PSTN devices and gateways for ISDN (H.320) videoconferencing devices.

A gatekeeper manages a set of register endpoints, collectively referred as a zone in main functions include call admission (or call authorization), address resolution, and other management-related functions (e.g., bandwidth allocation). Each endpoint before initiating call or conference has to register with the designated gatekeeper within the zone. The gatekeeper provides the address resolution to a special transport address of the target recipient. It also determines whether to accept or reject the call connection request based on the available bandwidth or other system parameters.

An MCU provides the necessary control needed for multiparty videoconferences. It contains two logical components: a multipoint controller (MC) for call control coordination and multipoint processor (MP) to handle audio ' video mixing.

The key protocols used in the call setup are the Registration Admission Status (RAS) protocol. Q.931-based signaling protocol, and an H.245 media and conference control protocol.

RAS protocol is responsible for registration of endpoints (terminals, gateways, and MCUs) to the correspondent gatekeeper. RAS message carried in User Datagram Protocol (UDP) packets contain a number of request/reply message exchanged between the endpoints and gatekeeper. Besides the registration, RAS protocol also provide a means for the gatekeeper to monitor the endpoints within the zone and manage tl' associated resources.

The Q.931-based signaling protocol is derived from the integrated services digital network (ISDN) 0.931 signaling protocol tailored for n in the H.323 environment. The signaling messages are carried in reliable TCP packets. It provides the logical connection between the calling and called parties.

The H.245 media and conference control protocol is used for the two connected parties (after 0.931 establishment) to exchange various information related to their communications; for instance, type of messages (audio, video, or data) and format. In addition, it provides a set of control functions for multiparty videoconferences.RTF and RTCP, described earlier, are used for actual message transmission.

A summary of the H.323 protocol phases is given in. fig The RAS protocol is used in phases 0, 1, and 6 for registration and shutdown process. Signaling protocol is involved in phases 2, 5, and 6. The H.245 media and conference control protocol is active during phases 3 and 5, and media exchanged based on RTP/RTCP is carried out in phase 4 [4].

3.6-SESSION INITIATION PROTOCOL

As its name implies, SIP is used to initiate a session between users. It provides for user location services (this is its greatest strength, in fact), call establishment, call participant management (using a SIP extension [10]), and limited feature invocation. Interestingly, SIP does not define the type of session that is established.

SIP can just as easily establish an interactive gaming session as an audio/video conference.

Each SIP request consists of a set of header fields that describe the call as a whole followed by a message body that describes the individual media sessions that make up the call. Currently, SDP (described below) is used, but consenting parties may agree on another capability exchange protocol.

SIP is a client-server protocol, similar in both syntax and semantics to Hypertext Transport Protocol (HTTP). [11] Requests are generated by one entity (the client) and sent to a receiving entity (the server). The server processes the requests, and then sends a response to the client. A request and the responses that follow it are called a *transaction*. The software on an end system that interacts with a human user is known as a *user agent*. A user agent contains two components, a user agent client (UAC) and a user agent server (UAS). The UAC is responsible for initiating calls (sending requests), and the UAS for answering calls (sending responses). A typical Internet telephony appliance or application contains both a UAS and a UAC. (Note that this differs from a Web browser, which acts only as a client.)

Within the network, there are three types of servers. A registration server receives updates on the current locations of users. A *proxy server* receives requests and forwards them to another server (called a *next-hop* *server)*, which has more precise location information about the callee. The next-hop server might be another proxy server, a UAS, or a redirect server. A *redirect server* also receives requests, and determines a next-hop server. However, instead of forwarding the request there, it returns the address of the next-hop server to the client. The primary function of proxy and redirect servers is *call routing* -- the determination of the set of servers to traverse in order to complete the call. A proxy or redirect server can use any means at its disposal to determine the next-hop server, including executing programs and consulting databases. A SIP proxy server can also *fork* a request, sending copies to multiple next-hop servers at once. This allows a call setup request to try many different locations at once. The first location to answer is connected with the calling party.

As in HTTP, the client requests invoke *methods* (commands) on the server. SIP defines several methods. INVITE invites a user to a call. BYE terminates a connection between two users in a call. OPTIONS solicits information about capabilities, but does not set up a call. ACK is used for reliable message exchanges for invitations. CANCEL terminates a search for a user. Finally, REGISTER conveys information about a user's location to a SIP registration server.

A client sets up a call by issuing an INVITE request. This request contains header fields used to convey information about the call. The most important header fields are To and From, which contain the callee's and caller's address, respectively. The Subject header field identifies the subject of the call. The Call-ID header field contains a unique call identifier, and the CSeq header field contains a sequence number. The Contact header field lists addresses where a user can be contacted. It is placed in responses from a redirect server, for example. The Require header field is used for negotiation of protocol features, providing extensibility. The Content-Length and Content-Type header fields are used to convey information about the body of the message. The body contains a description of the session which is to be established.

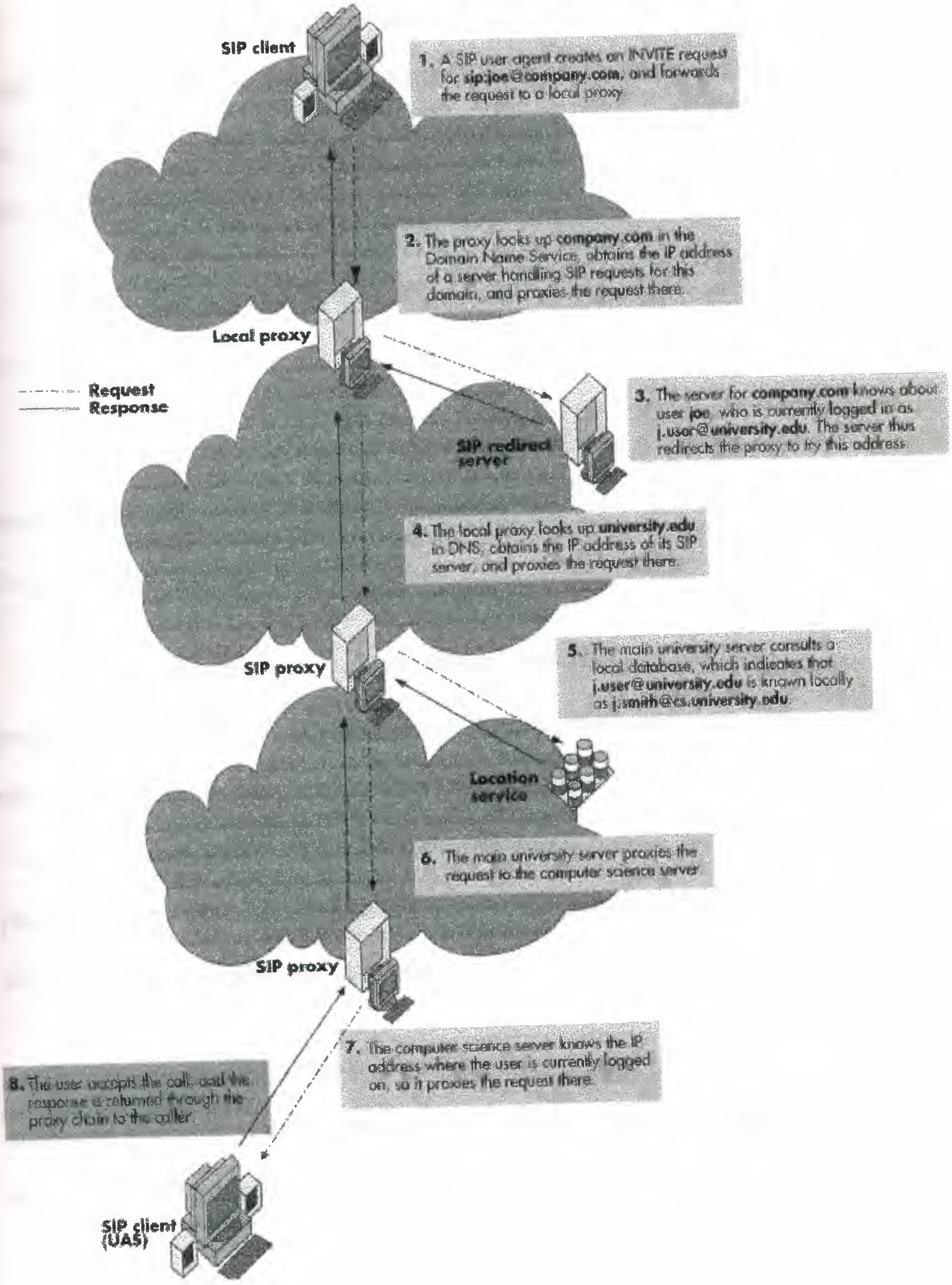
Extensions can be defined with new header fields. One such extension, used for call control,[10] defines several new headers used for feature invocation (such as call transfer) and multiparty conferencing.[12, 13]

A typical SIP transaction is depicted in Figure 3.4

3.7-SESSION DESCRIPTION PROTOCOL

SDP is used to describe multimedia sessions, for both telephony and distribution applications like Internet radio. The protocol includes information about:

- Media streams. A multimedia session can contain many media streams; for example, two audio streams, a video stream, and a whiteboard session. SDP conveys the number and type of each media stream. It currently defines audio, video, data, control, and application as stream types, similar to MIME types used for Internet mail.



1. A SIP user agent creates an INVITE request for **sip:joe@company.com,** and forwards the request to a local proxy.

2. The proxy looks up **company.com** in the Domain Name Service, obtains the IP address of a server handling SIP requests for this domain, and proxies the request there.

3. The server for **company.com** knows about user **joe**, who is currently logged in as **j.user@university.edu**. The server thus redirects the proxy to try this address.

4. The local proxy looks up **university.edu** in DNS, obtains the IP address of its SIP server, and proxies the request there.

5. The main university server consults a local database, which indicates that **j.user@university.edu** is known locally as **j.smith@cs.university.edu**.

6. The main university server proxies the request to the computer science server.

7. The computer science server knows the IP address where the user is currently logged on, so it proxies the request there.

8. The user accepts the call, and the response is returned through the proxy chain to the caller.

figure3.4:SIP transaction

- Addresses. For each stream, the destination address (unicast or multicast) is indicated. Note that the addresses for different media streams may differ, so a user may, for example, receive audio on a low-delay Internet telephone appliance and video on a workstation.

- Ports. For each stream, the UDP port numbers for sending and/or receiving are indicated.

- Payload types. The media formats that can be used during the session are also conveyed. For unicast sessions ("traditional" IP telephony), this list is called a *capability set*.

- Start and stop times. For broadcast-style sessions like a television program, the start, stop, and repeat times of the session are conveyed. Thus, one can announce or invite others to a weekly TV show or a Tuesday/Thursday lecture. (SIP can be used not just to make a traditional phone call, but a caller can also invite others to, say, a TV program, without the caller and callee talking to each other.)

- Originator. For broadcast-style sessions, the session description names the originator of the session and how that person can be contacted (e.g., in case of technical difficulties).

SDP conveys this information in a simple textual format. In fact, the acronym for SDP is a misnomer, since SDP is more of a description format. When a call is set up using SIP, the INVITE message contains an SDP body describing the session parameters acceptable to the caller. The response from the called party contains a modified version of this description, incorporating the capabilities of the called party.

The v line is a version identifier for the session. The o line is a set of values that form a unique identifier for the session. The u and e lines given the URL and e-mail addresses for further information about the session. The c line indicates the address for the session, the b line indicates the bandwidth (64 kbps in this case), and the t line the start and stop times (where 0 means that the session continues indefinitely). The k line conveys the encryption key for the session. There are three m lines, each of which identifies a media stream type (audio, video, and whiteboard application), the port number for that stream, the protocol, and a list of payload types. The a line specifies an attribute. For example, the line below the audio stream definition defines the codec parameters for RTP payload type 96.

3.8-REAL TIME TRANSPORT PROTOCOL

RTP,[14] as its name implies, supports the transport of real-time media (e.g., audio and video) over packet networks. (It is also used by H.323.)

The process of transport involves taking the bitstream generated by the media encoder, breaking it into packets, sending the packets across the network, and then recovering the bitstream at the receiver. The process is complex because packets can be lost, delayed by variable amounts, and reordered in the network. The transport protocol must allow the receiver to detect these losses. It must also convey timing information so that the receiver can correctly compensate for jitter (variability in delay). To assist in this function, RTP defines the formatting of the packets sent across the network. The packets contain the media information (called the *RTP payload*), along with an RTP header. This header provides information to the receiver that allows it to reconstruct the media. RTP also specifies how the codec bitstreams are broken up into packets.[15] RTP was also "engineered" for multicast, which means that it works in conferencing applications and broadcast environments where multicast is used to distribute the media. It is important to note that RTP does not try to reserve resources in the network to avoid packet loss and jitter; rather, it allows the receiver to recover in the presence of loss and jitter.

RTP plays a key component in any Internet telephony system. It is effectively at the heart of the application, moving the actual voice among participants. The relationship between the signaling protocol and RTP is that signaling protocols are used to establish the parameters for RTP transport.

RTP provides a number of specific functions:

- Sequencing. Each RTP packet contains a sequence number. This can be used for loss detection and compensation for reordering.

- Intramedia synchronization. Packets within the same stream may suffer different delays (jitter). Applications use playout buffers[16-18] to compensate for delay jitter. They need the timestamps provided by RTP to measure it.

- Payload identification. In the Internet, network conditions such as packet loss and delay vary, even during the duration of a single call. Speech and video codecs differ in their ability to work properly under various loss conditions. Therefore, it is desirable to be able to change the encoding for the media (the "payload" of RTP) dynamically as network conditions vary. To support this, RTP contains a payload type identifier in each packet to describe the encoding of the media.

- Frame indication. Video and audio are sent in logical units called *frames*. It is necessary to indicate to a receiver where the beginning or end of a frame is, in order to aid in synchronized delivery to higher layers. A frame marker bit is provided for this purpose.

- Source identification. In a multicast session, many users are participating. There must be a way for a packet to contain an indicator of which participant sent it. An identifier called the *synchronization source* (SSRC) is provided for this purpose.

RTP also has a companion control protocol, called Real Time Control Protocol (RTCP). RTCP provides additional information to session participants. In particular, it provides:

- QoS feedback. Receivers in a session use RTCP to report back the quality of their reception from each sender. This information includes the number of lost packets, jitter, and round-trip delays. This information can be used by senders for adaptive applications[19, 20, 21] which adjust encoding rates and other parameters based on feedback.

- Intermedia synchronization. For flexibility, audio and video are often carried in separate packet streams, but they need to be synchronized at the receiver to provide "lip sync." The necessary information for the synchronization of sources, even if originating from different servers, is provided by RTCP.

- Identification. RTCP packets contain information such as the e-mail address, phone number, and full name of the participant. This allows session participants to learn the identities of the other participants in the session.

- Session Control. RTCP allows participants to indicate that they are leaving a session (through a BYE RTCP packet). Participants can also send small notes to each other, such as "stepping out of the office."

RTCP mandates that all session participants (including those who send media and those who just listen) send a packet periodically which contains the information described above. These packets are sent to the same address (multicast or unicast) as the RTP media, but on a different port. The information is sent periodically since it contains time-sensitive information, such as reception quality, which becomes stale after some amount of time. However, a participant cannot just send an RTCP packet with a fixed period. Since RTP is used in multicast groups, there could be sessions (like a large lecture) with hundreds or thousands of participants. If each one were to send a packet with a fixed period, the network would become swamped with RTCP packets. To fix this, RTCP specifies an algorithm that allows the period to increase in larger groups.[22]

3.9-REAL-TIME STREAMING PROTOCOL

RTSP[23] is used to control a stored media server. A stored media server is a device capable of playing prerecorded media from disk to the network, and recording multimedia content to disk. RTSP offers controls similar to those in a VCR remote control. A client can instruct the server to play, record, fast forward, rewind, and pause. It can also configure the server with the IP addresses, UDP ports, and speech codecs to use to deliver (or record) the media. Typically, media is sent from the media server using RTP.

Stored media has a number of applications in Internet telephony:

• A media server can record the content of a conference for future reference.

• Media servers can play media into an existing conference. For example, if participants in a multiparty conference are discussing a movie, it would be useful to be able to bring a media server into the conference, and have it play portions of the movie into the conference. (This is done in the example in Figure 1, where Joe has the media server play the IEEE tutorial into the conference.)

• Media servers can record voice mail. RTSP clients can use the protocol to control playback of the message. This would allow users to listen to their voice mail, and rewind to a critical part (e.g., a phone number in the message). RTSP can also be used to record the incoming or outgoing voice mail message.

A client executes the following steps to cause a media server to play back content.

• Obtain the presentation description. The first step is to obtain the presentation description. This description enumerates the various components of the session. As an example, a classroom presentation might contain three components: a document camera, a video view of the professor, and the audio. For each component, the description defines the media parameters needed to decode the component, such as the codec type and frame rate. The presentation description can be obtained in several ways. The client can issue a DESCRIBE request to the server, which causes the server to return a description. It is also possible to obtain a description through other means, such as a Web page.

• Set up the server. Once it has obtained the description, the client can issue a SETUP request to the server. This request establishes the destination to which the media should be delivered. The destination includes the IP address (unicast or multicast), port numbers, protocols (usually, but not necessarily, RTP over UDP), TTL, and number of multicast layers. In response to the SETUP message, the server returns a session id. This id is used in further requests.

• Issue media requests. After the stream has been set up, the client can issue media requests to the server. These include operations such as PLAY, RECORD, and PAUSE. The PLAY method encompasses seek, fast forward, and reverse, in addition to straight play. This is accomplished by including a Scale header, which indicates the time speedup (or speeddown) at which the media should be played. The Range header specifies where in the stream playback should start.

• Teardown. Once interaction with the server is complete, the client issues a TEARDOWN request. This request destroys any state associated with the session. Further requests for the given session id will no longer be valid.

3.10-CALL PROCESSING LANGUAGE

CPL[24, 25] is a scripting language that allows end users to specify the behavior of call agents that execute on their behalf. The call agents are invoked when a call arrives at a SIP server. These agents execute the instructions contained in the CPL. This allows end users to specify their own call services. For example, a user can instruct the agent to connect a call by trying a cell phone, home PC, and work PC all at once.

3.11-GATEWAY LOCATION PROTOCOL

SIP allows a user on the Internet to call another user, also on the Internet. What if an Internet user wishes to call someone not on the Internet, but rather, on the telephone network? In this case, an Internet telephony gateway is needed. This device is capable of converting signaling and media between a packet network and the telephone network (PSTN). We anticipate that many gateways will be deployed in the future by Internet service providers (ISPs) and telephone companies.

To complete a call to a PSTN endpoint, an IP host must send a SIP invitation to the gateway. However, given a phone number to call, how does the caller find and select one of the many gateways to complete the call? In theory, each gateway could dial (almost) any phone number, but the caller may want to minimize the distance of the PSTN leg of the call, for example. This function is supported by the Gateway Location Protocol (GLP).

An overall architecture for GLP is shown in Figure 3.3

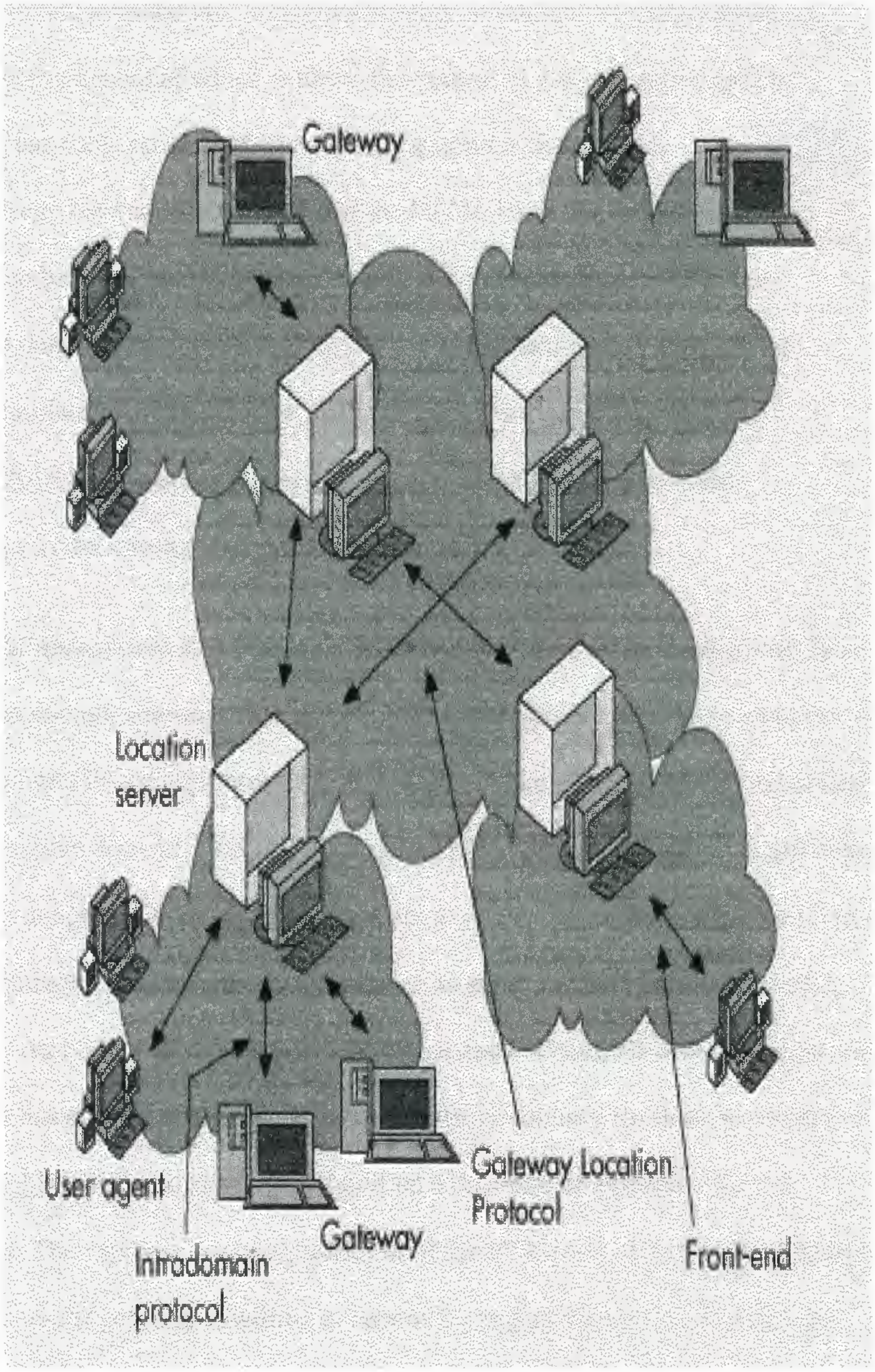


figure3.5: architecture for GLP

In that architecture there are a number of Internet telephony domains in the Internet, each of which is under the control of a single authority. Each domain has some number of IP telephony gateways that provide connectivity between the Internet and the PSTN. Each domain also has some number of IP users and some number of location servers (LS). The LSs in a domain know about the gateways in their own domains, by means of an intradomain protocol, such as the Service Location Protocol (SLP).[26] The intradomain protocol propagates information from the gateways to the location servers within a domain.

Unfortunately, it is unlikely that a single administrative domain will have access to enough gateways to complete calls to all possible telephone numbers. As a result, users in one administrative domain can make use of gateways in another administrative domain. This usually requires pre-established business relationships between domains. Once the agreements are in place, it is necessary for the LS in one domain to exchange information about its gateways with the LS in another domain. An LS can then take this information and exchange it with other LSs with which it has an established relationship. The protocol used for these exchanges is the GLP.[27] GLP allows an LS to build up a database of gateways in other domains. The database contains entries with the IP address of the gateway, a range of numbers the gateway is willing to terminate, and attributes that describe the gateway. These attributes include signaling protocols, cost information, and provider identifiers, among others. An LS can use the attributes to decide which gateways to use to terminate a call to a particular number. The information can also be used to determine which gateways to further advertise to other LSs. Both of these decisions are embodied in the *policy* that directs the behavior of the LS.

When a client within the domain wishes to make a call to a number in the phone network, it can proceed in several ways:

- Lightweight Directory Access Protocol. The database of the LS is made available through LDAP.[1] The calling client queries this database with the destination phone number, and the LS returns the IP address of the gateway. The client can then send a SIP invitation to that address.

- *Session Initiation Protocol*. Rather then sending a SIP invitation directly to the gateway, the caller sends it to the LS. The invitation contains the desired destination phone number. The LS consults its database, finds the right gateway, and proxies the call to it. In this case, the LS also acts as a SIP proxy. By acting as a proxy, the LS hides the gateway selection process from the caller. The caller application does not need to know whether the address being called is a phone number or a SIP universal resource locator (URL). In either case, the invitation is sent to the local proxy.

- Web pages. The LS can make its database available through Web pages. A user that wishes to make a call browses the Web page and finds the gateway it likes (perhaps this can be done through a Web form), and the

LS returns the address of the gateway on the Web page. The user copies this address to their SIP software, and completes a call to the gateway.

GLP is just beginning the process of specification. Because it is similar to existing interdomain IP routing protocols, such as BGP-4,[28] it is likely to borrow heavily from them.[29, 30]

3.12- QoS for IP:

The existing Internet service (i.e., the *best-effort service* of IP) cannot satisfy the QoS requirements of emerging multimedia applications, primarily caused by the variable queuing delays and packet loss during network congestion. There has been a significant amount of work in the past decade to extend the Internet architecture ;md protocols to provide QoS support for multimedia applications. This has led to the development of a number of service models and mechanisms..

3.12.1- The integrated service (Intserv) model:

The Intserv model was proposed as an extension to support real-time applications. The key is to provide some control over the end-to-end packet delays in order to meet the real-time QoS. specifically, the Intserv model proposes two service classes in addition to best-effort service. They arc:

- Guaranteed service for applications requiring a fixed delay bound
- Controlled-load service for application requiring reliable and enhanced best-effort service

NEAR EAST UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

COM400

VOICE AND VIDEO ON INTERNET

SUBMITTED BY: YASIR ALI

SUBMITTED TO: EKREM VAROGLU

JUNE2000

Acknowledgment

First of all I am thankful to gracious Allah, the all Almighty, who enable me to complete this project.

I would like to thanks all of my teacher because of whom I able to complete my graduation especially to Mr. Ekrem Varoglu who help me lot and encouraging me to complete this project

Also I would like to thank my parents, for their encouragement support and prayer for me.

I am also thankful to all of my friend who help a lot to complete this project especially to M.Nauman, Adnan Rizvi, Shahid Butt, Baber Rahman, Hafiz Zullifqar Ali, Syed Jawad Ali, Naveed Mustafa and Rizwan Ahmed ch.

Yasir Ali

ABSTRACT

The Internet is under rapid growth and continuous evolution in order to accommodate an increasingly large number of applications with diverse service requirements. In particular, voice and video on IP is one of the most promising services currently being deployed. Besides the potentially significant cost reduction, multimedia can offer many new features and easier integration with widely adopted Web-based services. Despite these advantages, there still exist a number of barriers to the widespread deployment of multimedia application such as the lack of control architectures and associated protocols for managing calls and video, a security mechanism for user authentication, and proper charging schemes. The most prominent one, however, is how to ensure the QOS needed for voice conversation. The purpose of this project is to survey the state-of-the-art technologies in enabling the QoS and protocols support for voice and video communications in the next-generation Internet. In this project, we first review the existing technologies in supporting voice and video over IP networks, including the basic mechanisms in the IETF multimedia application architecture, and ITU-T H.323-related Recommendations.

Table of contents

Chapter number **Page number**

4
Chap four

video over Internet

5
Chap five

Chapter one

1.1-INTRODUCTION

The project is about voice and video on Internet. In this project first of all I will discuss about the "Internet "

What is Internet? When it was established and who established it. What was the main purpose of Internet at that time and why they needed to establish it. The Internet is under rapid growth and continuous evolution in order to accommodate an increasingly large number of applications with diverse service requirements. While the Internet has served as a research and education vehicle for more than two decades, the last few years have witnessed its tremendous growth and its great potential for providing a wide variety of services. In particular, using the Internet to carry video and phone conversations, known as Internet telephony or voice over IP (VoIP), is taking the telecommunications industry by storm. Not only does it represent the best opportunity so far for companies and Telco's to facilitate voice and data convergence, but it also promises to deliver a new era in cheap telephone calls. Five years ago. Many to be far too unreliable for mass-market deployment regarded Internet telephony. But over the past few years, reliability and quality have quickly improved, and Internet telephony is now one of the fastest growing industries.

Voice on Internet is beginning to link the worlds of data and voice. The goal is to combine the strengths of telephones, telephone lines, computers and computer networks to achieve the benefits of both worlds.

THE first two projects in this area are:

1- The Internet Connection Phone, which lets computer users hold voice communications over networks, using their PCs instead of telephones. It can work over the Internet, using either a LAN or a telephone line to connect to the Internet.

2-The Voice Data Gateway, which takes this concept one step further: it lets computer users use a single telephone line to connect to the Internet and still continue to use the telephone line for normal voice calls.

Eventually, goal is to make the Internet a full function platform for voice communications, in addition to text, graphics and multimedia - and to integrate voice and data communications seamlessly within the same network.

Unfortunately, Internet telephony technology is also relatively immature, with quality and latency still being major issues. They are, however, both being addressed. Voice quality has improved greatly from early versions of the technology, which was characterized by distortions and disruptions in speech. Improved technologies for voice coding and lost packet reconstruction have also yielded products where speech is easy to understand. Latency, a factor that affects the pace of a conversation, is also being addressed. Humans can tolerate about 250 ms of latency before it has a noticeable effect, and voice services over the public Internet today typically exceed this figure. Latency will, however, continue to improve, driven by three factors: improved gateways (developers are just beginning to squeeze latency out of the first generation of products); deployment over private networks — by deploying gateways on private circuits, organizations and service providers can control the bandwidth utilization; and hence latency; Internet development (today's Internet was not designed with real-time communications in mind). The Internet Engineering Task Force (IETF), together with Internet backbone equipment providers, is addressing this with technologies like Resource Reservation Protocol (RSVP), which will let bandwidth be reserved. While it will take some time for the world's routers to be upgraded and operational aspects (e.g., how to bill for high quality of service, QoS) to be resolved

Chapter two

Internet

2.1-The Internet

The Internet, sometimes called simply "the Net," is a worldwide system of computer networks - a network of networks in which users at any one computer can, if they have permission, get information from any other computer (and sometimes talk directly to users at other computers). It was conceived by the Advanced Research Projects Agency (ARPA) of the U.S. government in 1969 and was first known as the ARPANet. The original aim was to create a network that would allow users of a research computer at one university to be able to "talk to" research computers at other universities. A side benefit of ARPANet's design was that, because messages could be routed or rerouted in more than one direction, the network could continue to function even if parts of it were destroyed in the event of a military attack or other disaster.

Today, the Internet is a public, cooperative, and self-sustaining facility accessible to hundreds of millions of people worldwide. Physically, the Internet uses a portion of the total resources of the currently existing public telecommunication networks. Technically, what distinguishes the Internet is its use of a set of protocols called TCP/IP (Transmission Control Protocol/Internet Protocol). Two recent adaptations of Internet technology, the intranet and the extranet, also make use of the TCP/IP protocol.

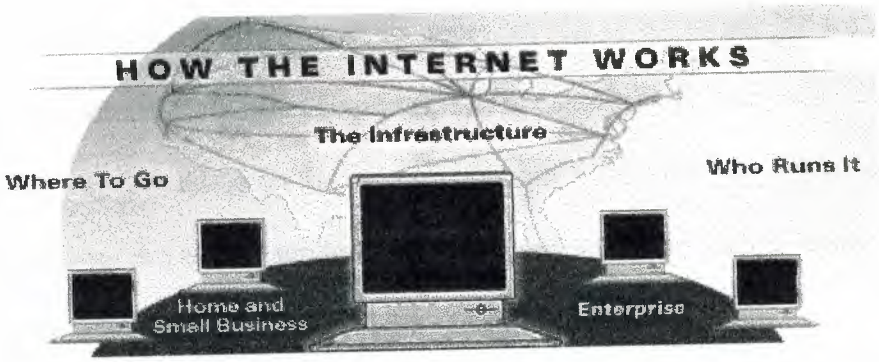


Figure2.1how the internet work

For many Internet users, electronic mail (e-mail) has practically replaced the Postal Service for short written transactions. Electronic mail is the most widely used application on the Net. You can also carry on live "conversations" with other computer users, using IRC (Internet Relay Chat). More recently, Internet telephony hardware and software allows real-time voice conversations.

The most widely used part of the Internet is the World Wide Web (often abbreviated "WWW" or called "the Web"). Its outstanding feature is hypertext, a method of instant cross-referencing. In most Web sites, certain words or phrases appear in text of a different color than the rest; often this text is also underlined. When you select one of these words or phrases, you will be transferred to the site or page that is relevant to this word or phrase. Sometimes there are buttons, images, or portions of images that are "clickable." If you move the pointer over a spot on a

Web site and the pointer changes into a hand, this indicates that you can click and be transferred to another site.

Using the Web, you have access to millions of pages of information. Web "surfing" is done with a Web browser, the most popular of which are Netscape Navigator and Microsoft Internet Explorer. The appearance of a particular Web site may vary slightly depending on the browser you use. Also, later versions of a particular browser are able to render more "bells and whistles" such as animation, virtual reality, sound, and music files, than earlier versions.

2.2- IP (Internet Protocol)

The Internet Protocol (IP) is the method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one address that uniquely identifies it from all other computers on the Internet. When you send or receive data (for example, an e-mail note or a Web page), the message gets divided into little chunks called packets. Each of these packets contains both the sender's Internet address and the receiver's address. Any packet is sent first to a gateway computer that understands a small part of the Internet. The gateway computer reads the destination address and forwards the packet to an adjacent gateway that in turn reads the destination address and so forth across the Internet until one gateway recognizes the packet as belonging to a computer within its immediate neighborhood or domain. That gateway then forwards the packet directly to the computer whose address is specified.

Because a message is divided into a number of packets, each packet can, if necessary, be sent by a different route across the Internet. Packets can arrive in a different order than the order they were sent in. The Internet Protocol just delivers them. It's up to another protocol, the Transmission Control Protocol (TCP) to put them back in the right order.

IP is a connectionless protocol, which means that there is no established connection between the end points that are communicating. Each packet that travels through the Internet is treated as an independent unit of data without any relation to any other unit of data. (The reason the packets do get put in the right order is because of TCP, the connection-oriented protocol that keeps track of the packet sequence in a message.) In the Open Systems Interconnection (OSI) communication model, IP is in layer 3, the Networking Layer.

The most widely used version of IP today is Internet Protocol Version 4 (IPv4). However, IP Version 6 (IPv6) is also beginning to be supported. IPv6 provides for much longer addresses and therefore for the possibility of many more Internet users. IPv6 includes the capabilities of IPv4 and any server that can support IPv6 packets can also support IPv4 packets.

IP networks run over a variety of devices, including simple hubs, high-function routers, and sophisticated policy-enabled switches. These platforms offer a rich set of functions in their hardware and software to perform encryption, sequencing, classification, prioritization, compression, and so on. Driven by the incredible performance of these platforms, IP networks are able to offer an increasingly rich set of functions to the applications that run on them.

Many of these functions---in particular, those related to the delivery of traffic and the broad reach of IP---are considered "fundamental" services today. Functions such as forwarding, sequencing, service location, and so on might have been "exotic" a few short years ago, but today, you will find them in nearly all good quality networking equipment.

A new set of functions, known as Intelligent Network Services, builds on the functions these platforms offer to deliver high-level network intelligence to e-business applications. Intelligent Network Services include voice, video, legacy integration, load balancing, caching, and more.

With Intelligent Network Services in place, information technology professionals can deploy Internet Application Technologies like real-time trading, distance learning, and unified messaging.

The forwarding functions of IP networks are

Security services might use a *specialized security application-specific integrated circuit (ASIC)* to factor large prime numbers or encrypt data securely at sustained data rates.

- Voice and video services use coders-decoders (CODECs), switching, and high-speed cable plants to make efficient use of capacity.

- SNA services rely on link spoofing, tunnels, and other prioritization facilities in devices in order to simulate legacy environments or to present legacy information to enterprise applications (data mining, Web front-ends).
- QoS services leverage differentiated switching hardware that can classify and queue traffic at wire speed.
- Intelligent Network Classification services use technologies such as Network Based Application Recognition (NBAR), Context-Based Access Control (CBAC), Multimedia Conference Manager (MCM) proxy's to uniquely identify network traffic and handle it accordingly.

High-availability applications *use* redundant architectures and fault-tolerant equipment to ensure maximum service availability

2.3-TCP (Transmission Control Protocol)

TCP (Transmission Control Protocol) is a method (protocol) used along with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. While IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the individual units of data (called packets) that a message is divided into efficient routing through the Internet.

For example, when an HTML file is sent to you from a Web server, the Transmission Control Protocol (TCP) program layer in that server divides the file into one or more packets, numbers the packets, and then forwards them individually to the IP program layer. Although each packet has the same destination

IP address, it may get routed differently through the network. At the other end (the client program in your computer), TCP reassembles the individual packets and waits until they have arrived to forward them to you as a single file.

TCP is known as a connection-oriented protocol, which means that a connection is established and maintained until such time as the message or messages to be exchanged by the application programs at each end have been exchanged. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end. In the Open Systems Interconnection (OSI) communication model, TCP is in layer 4, the Transport Layer.

2.4-OSI (Open System interconnection)

OSI (Open Systems Interconnection) is a standard description or "reference model" for how messages should be transmitted between any two points in a telecommunication network. Its purpose is to guide product implementers so that their products will consistently work with other products. The reference model defines seven layers of functions that take place at each end of a communication. Although OSI is not always strictly adhered to in terms of keeping related functions together in a well-defined layer, many if not most products involved in telecommunication make an attempt to describe them in relation to the OSI model. It is also valuable as a single reference view of communication that furnishes everyone a common ground for education and discussion.

Developed by representatives of major computer and telecommunication companies beginning in 1983, OSI was originally intended to be a detailed specification of interfaces. Instead, the committee decided to establish a common reference model for which others could develop detailed interfaces that in turn could become standards. OSI was officially adopted as an international standard by the International Organization of Standards (ISO). Currently, it is Recommendation X.200 of the International Telecommunication Union.

The main idea in OSI is that the process of communication between two end users in a telecommunication network can be divided into layers, with each layer adding its own set of special, related functions. Each communicating user is at a computer equipped with these seven layers of function. So, in a given message between users, there will be a flow of data through each layer at one end down through the layers in that computer and, at the other end, when the message arrives, another flow of data up through the layers in the receiving computer and ultimately to the end user. The actual programming and hardware that furnishes these seven layers of function is usually a combination of the computer operating system, applications (such as your Web browser), TCP/IP or alternative transport and network protocols, and the software and hardware that enable you to put a signal on one of the lines attached to your computer.

OSI divides telecommunication into seven layers. The layers are in two groups. The upper four layers are used whenever a message passes from or to a user. The lower three layers (up to the network layer) are used when any message passes through the host computer. Messages intended for this computer pass to the upper layers. Messages destined for some other host are not passed up to the upper layers but are forwarded to another host. The seven layers are:

Layer 7: The application layer...This is the layer at which communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. (This layer is *not* the application itself, although some applications may perform application layer functions.)

Layer 6: The presentation layer...This is a layer, usually part of an operating system, that converts incoming and outgoing data from one presentation format to another (for example, from a text stream into a popup window with the newly arrived text). Sometimes called the syntax layer.

Layer 5: The session layer...This layer sets up, coordinates, and terminates conversations, exchanges, and dialogs between the applications at each end. It deals with session and connection coordination.

Layer 4: The transport layer...This layer manages the end-to-end control (for example, determining whether all packets have arrived) and error-checking. It ensures **complete data transfer**.

Layer 3: The network layer...This layer handles the routing of the data (sending it in the right direction to the right destination on outgoing transmissions and receiving incoming transmissions at the packet level). The network layer does **routing and forwarding**.

Layer 2: The data link layer...This layer provides error control and synchronization for the physical level and does bit-stuffing for strings of 1's in excess of 5. It furnishes transmission protocol knowledge and management.

Layer 1: The physical layer...This layer conveys the bit stream through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier

2.5-E-mail explanation:

This document describes how electronic mail (e-mail) works. It begins by defining some terms and concepts, which are a vital part of e-mail. It then goes a layer deeper, explaining some lower-level concepts. Several specific applications are then discussed: some briefly, some in great detail.

2.6-High-level Concepts

2.6.1Mail-boxes

A mail-box is a file, or possibly a directory of files, where incoming messages are stored.

2.6.2User Agents

A mail user agent, or MUA, is an application run directly by a user. User agents are used to compose and send out-going messages as well as to display, file and print messages which have arrived in a user's mail-box. Examples of user agents are elm, mailx, mh, zmail, Netscape, ...; more information is provided about these in the Specific Applications section below.

2.6.3-Transfer Agents

Mail transfer agents (MTAs) are used to transfer messages between machines. User agents give the message to the transfer agent, who may pass it onto another transfer agent, or possibly many other transfer agents. Users may give messages to transfer agents directly, but this requires some expertise on the part of the user and is only recommended for experts.

Transfer agents are responsible for properly routing messages to their destination. While their function is hidden from the average user, theirs is by far the most complex part of getting messages from their source to their destination. The most common transfer agent is send mail (1m).

2.6.4-Delivery Agents

Delivery agents are used to place a message into a user's mail-box. When the message arrives at its destination, the final transfer agent will give the message to the appropriate delivery agent, who will add the message to the user's mailbox. The standard delivery agent for Solaris, starting with 2.5, is mail. Local (1m).

2.7-Low-level Concepts

2.7.1-Character Sets

A character set is simply a mapping of byte values to characters.

The most common character set is US-ASCII, which has 32 non-printable control characters and 96 printable characters, for a total of 128. These 128 characters can be encoded in 7 bits of data, so each 8-bit byte representing one of these characters has the lower 7 bits set to the appropriate value for the given character and the 8th (high) bit set to zero. US-ASCII is therefore considered a single-byte 7-bit character set.

Many European languages have accentuated characters (like the German ü, the French ç and é, the (Swedish?) ø and the Spanish ñ). Such languages are commonly represented by characters sets whose lower half (*i.e.*, values 0 - 127) are identical to those of US-ASCII, and whose upper half (*i.e.*, values 128 - 255) represent these accentuated characters. These are therefore considered single-byte 8-bit characters sets; an example is ISO-8859-1.

Many Asian languages have so many characters that they need multiple bytes to represent them all. They are therefore considered multiple-byte character sets.

2.7.2-Headers & Bodies

Each message consists of two parts. The headers contain information about who authored the message, the intended recipients, the time of creation, the subject of the message, delivery stamps, ... Each header is of the form "*keyword*: *value*", where *keyword* is a special word (like **From** or **Date**) identifying the type of information contained in that header, and *value* is the information itself. More information about message headers can be found in RFC 822 and RFC 1123, section 5.

A blank line always separates the headers from the body.

The body contains the information the sender is trying to communicate. The "message" as most people think of it is really the body of the message.

2.7.3-MIME

For many years, most messages were plain text in the US-ASCII character set, so no structure was needed for message bodies. The recent explosion of messaging in Europe and Asia and the transmission of multi-media messages have brought about such a need.

MIME (Multipurpose Internet Mail Extensions, specified in RFCs 2045 - 2049, especially RFC 2045 and RFC 2046, defines such a body structure. It specifies how a Content-Type header can be used to specify a particular character set or other non-textual data type for a message. For example, the header:

```
Content-Type: text/plain; charset=us-ascii
```

indicates that the message consists of plain text in the US-ASCII character set. MIME also specifies how to encode data when necessary (more on this below). It is the responsibility of the receiving user agent to use this information to display the message in a form that will be understood by the user.

2.8-Transfer Protocols

The language spoken between transfer agents is known as a transfer protocol. There are many in existence; the most common is *SMTP* (Simple Mail Transfer Protocol); also well-known are UUCP (Unix-to-Unix copy) and X.400. This document studies SMTP at length. For further information about SMTP, refer to RFC 821 and RFC 1123, section 5.

2.9-Envelopes and Bodies

SMTP uses the concept of an envelope to transfer messages; this merely contains information about from whom the message originated and to whom it is destined. The originator address is important: in case there is a problem transferring or delivering the message, the originator can be notified.

The SMTP body is the entire message as defined above in Headers & Bodies. So the message headers plus the message body equals the SMTP body. The term *SMTP body* is not used that commonly, but it is important to distinguish it from the message body.

2.10-7-bit data *vs.* 8-bit data

For historical reasons relating to the US-ASCII character set, SMTP is a 7-bit protocol, which means it limits bytes of data sent to use only the low-order 7-bits. If the 8th (high) bit of a byte is set, SMTP dictates that the bit must be zeroed out. In order for a message containing 8-bit data to be transferred without data loss, the message must first be encoded into 7-bit data. As most early e-mail users spoke English, however, and most computers used the 7-bit US-ASCII character set, this was not a problem.

In recent years, however, several factors have increased the need for 8-bit message transfer. As mentioned above, European languages often use 8-bit character sets, and Asian language character sets often require multiple bytes; their transmission is greatly simplified if all 8 bits can be transferred unaltered. Finally, the explosion of multi-media messages like audio and video clips have brought about a two-fold need for 8-bit message transfer: encoding messages into 7-bit data is not only cumbersome, but the resultant encoded message is significantly (typically 33%) larger than the original message.

To meet this need, SMTP has been extended to allow 8-bit data to be properly transferred between consenting transfer agents. The negotiating process used to verify consent is specified in RFC 1869, which describes the general extension mechanism to SMTP (called *ESMTP*), and RFC 1652, which describes the specific extension to allow 8-bit data transfer, called *8BITMIME*. If a transfer agent has a message containing 8-bit data and it cannot negotiate the proper transfer of that data, it must either encode the message into 7-bit data using MIME, or return the message to the sender indicating the reason for the return.

It is no coincidence that MIME and ESMTP have common rationales and goals; they were developed in conjunction with each other towards the same end.

2.11-Routing

RFC 974 describes Mail Routing and the Domain Name System; a brief overview of how send mail implements this is given here.

Mail exchangers (**MX**) records are maintained by domain name servers (DNS) to tell MTAs where to send mail messages. An MX record can be specified for a specific host, or a wild-card MX record can specify the default for a specific domain. The MX record tells an MTA where a message, whose ultimate target is a given host in a given domain, should be sent to next, i.e., which intermediate hosts should be used to ultimately deliver a message to the target host. These MX records vary depending on the domain. To illustrate, here is an an example of how a message from a.eng.sun.com destined for b.ucsb.edu might be routed:

The MTA on a.eng.sun.com looks up the MX record for b.ucsb.edu, which tells it to route the message to venus.sun.com. The MTA on venus.sun.com looks up the MX record for b.ucsb.edu, which tells it to route the message to hub.ucsb.edu. The MTA on hub.ucsb.edu looks up the MX record for b.ucsb.edu, which tells it to route the message directly to b.ucsb.edu. The MTA on b.ucsb.edu recognizes that the message has arrived at its intended destination and processes the message for local delivery.

2.12-send mail specifics

MX records are maintained by DNS only (i.e., not hosts files or NIS). If no MX records are available for a given host, sendmail will try to send to that host directly. Once sendmail determines which host to attempt to send the message to: an intermediate host as indicated by an MX record, or a direct connection to the target host, it uses gethostbyname() to determine the IP-address of the target machine in order to make a connection.

The gethostbyname library routine may use DNS, an /etc/hosts file, or NIS to perform its name-to-IP-address look-up, as configured by the /etc/nsswitch.conf file. N.B.: the host name passed to gethostbyname may have been derived from an MX record if a domain name server is running, even though gethostbyname() may not use DNS to resolve this name's address. Remember that MX records are only available from DNS, and the name service switch does not affect a search for MX records. This is as required by RFC 1123, section 5.3.5. This situation may be most noticeable when DNS is not first in the /etc/nsswitch.conf file. It may then be possible that a host name only in /etc/hosts or NIS be redirected by a wild-card MX record to another host

Chapter Three

Voice on Internet

3.1- Introduction:

Voice on Internet is beginning to link the worlds of data and voice. The goal is to combine the strengths of telephones, telephone lines, computers and computer networks to achieve the benefits of both worlds.

THE first two projects in this area are:

1- The Internet Connection Phone, which lets computer users hold voice communications over networks, using their PCs instead of telephones. It can work over the Internet, using either a LAN or a telephone line to connect to the Internet.

2-The Voice Data Gateway, which takes this concept one step further: it lets computer users use a single telephone line to connect to the Internet and still continue to use the telephone line for normal voice calls.

Eventually, our goal is to make the Internet a full function platform for voice communications, in addition to text, graphics and multimedia - and to integrate voice and data communications seamlessly within the same network.

3.2- Voice Mail

Voice mail provides the basic ability to record, store, and manipulate spoken messages. Callers can leave messages for others that can be retrieved at a later time, so problems arising from time zone differences are reduced. Call recipients (subscribers) can leave detailed greetings that tell callers when they will be available. Businesses can receive orders and deliver information during non-business hours or when no one is available. Organizations can use voice mail to distribute general information efficiently to large numbers of employees or customers.

When implemented with screening options, voice mail systems can give busy users the freedom to choose between answering any call immediately and deferring response to a more convenient time.

The automated attendant function commonly packaged with voice mail performs the duties of an operator/receptionist: supervising transfers, screening calls and offering the caller directory assistance to the proper extension. On the extension side, voice mail systems commonly offer employees the ability to program their voice mail boxes with call-forwarding and paging options. They also allow consultants and visitors to maintain mailboxes which are not linked to any particular extension, but may be used for leaving and retrieving messages from any phone, at any time.

In countries where rotary dialing is still prevalent, the addition of a speech recognition resource makes automated attendant features available to all callers. It does this by augmenting touchtone input, allowing callers to speak the extension numbers of persons they wish to reach. Dial pulse detection (DPD), available on some boards, is another way to accept telephone responses when touchtone is not available.

For low-density systems, the D/41™ family provides four channels of voice processing per slot. The D/41ESC is the correct choice for four ports of voice in countries with high-voltage protection requirements, and for incorporating calling line ID into voice mail applications.

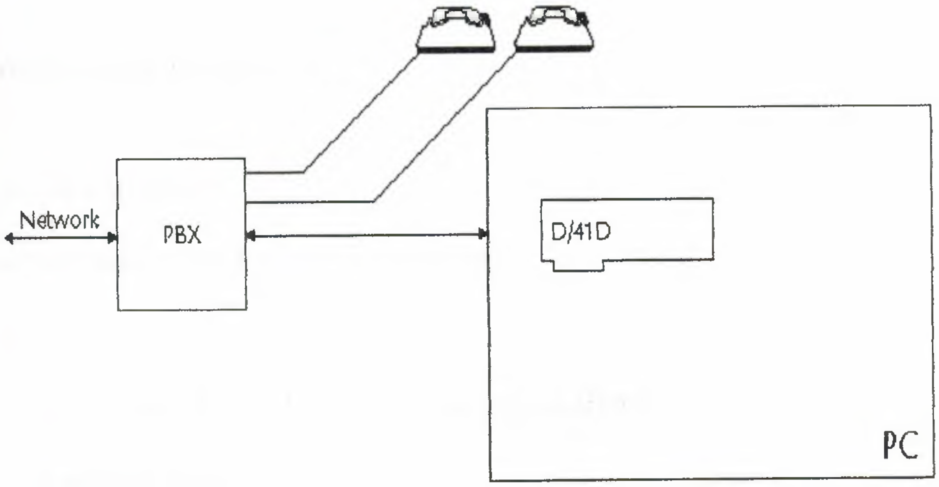


Fig. 3.1 For higher densities, the D/160SC-LS™ provides 16 channels of voice per slot.

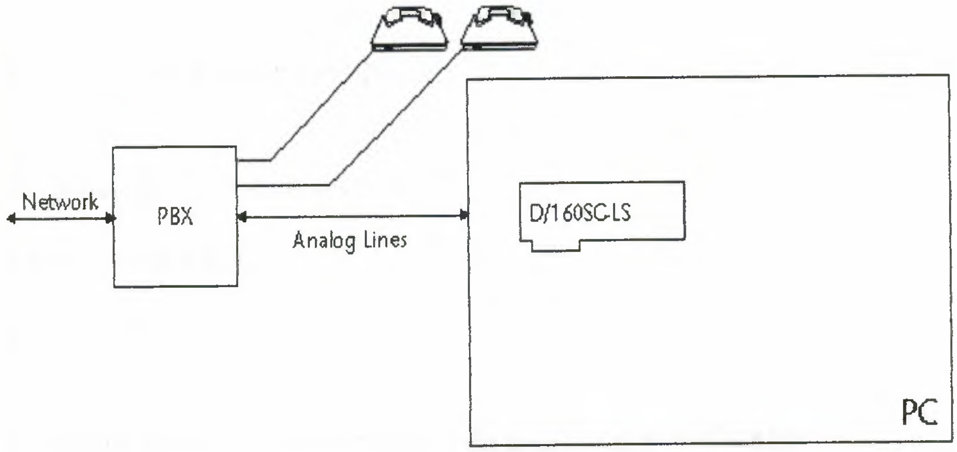


Fig. 3.2: Boards such as the D/42-NS (See p. xx, "Dialogic PBX Integration

Boards") exploit the call control signals of particular switches, allowing application control of such PBX features as

message waiting lights and call transferring.

3.2.1 Component Choices

- Voice boards
- Automatic speech recognition boards
- Fax boards
- Antares board with ASR or text-to-speech algorithm
- Dial pulse detection

Software

- Development packages

Development Tools And Utilities

- Application Generation ToolKits
- Voice Starter Kit
- PromptMaster

- System Density2 to 96 (typical) with additional voice boards

3.2.2 Application Enhancements

- Automated attendant
- Unified messaging
- Preview/predictive dialing
- PBX integration

3.2.3Application Tasks

- Receive a call
- Transfer a call automatically
- Define a menu of choices callers will hear
- Recognize DTMF tones
- Play message to callers
- Screen a call
- Route call to a live operator
- Record, play, forward and delete messages from callers
- Play message to answerer
- Allow users to adjust speed and volume of voice playback
- Copy messages to third parties
- Generate DTMF tones (outdial)
- Establish a call

3.3- Internet telephony

3.3.1- History:

The Multimedia Networking Applications group has been working on IP Telephony Technology since 1995.Our main areas of interest are Internet telephony

and telephony-based applications. The group delivered its first product - IBM Internet Connection Phone (known as ICPhone), an Internet Phone, built on technologies from the Audio\Video group. IP Telephony was identified by HRL as an emerging technology with very vast opportunities, as early as 1995.

The ICPhone used proprietary protocols, since a standard did not exist at the time, and we developed a global directory based on Light Directory Access Protocol (LDAP), which has since, became a standard. End users on one of the IGN/Advantis servers have used this directory without any support since 1996.

At the end of 1996, the IP Telephony development community accepted the ITU H.323 standard. The Haifa Research Lab, in cooperation with Zurich

Research Lab, moved all related activities, such as Internet Phone clients as well as IP/PSTN Gateways and Gatekeepers, to this standard. Voice over IP Technology To achieve high quality voice over IP, several audio technologies were developed in HRL and integrated within HRL IP Telephony components. Below are some of these components:

GSM and G.723.1 codecs

Echo Cancellation

Echo Suppression

Voice Activity Detection

Silence Suppression

Automatic Gain Control

Comfort Noise Generation

Jitter Control

Low Latency

Packet Trucking

DTMF Detection, Generation and Transfer within RTP

Packets

Simultaneous Voice and Data Protocol (SDP) over PPP

3.3.2 Current Activities IP/PSTN Gateway

Haifa Research Lab developed a prototype-based IP/PSTN Gateway during 1997, and a product-based IP/PSTN Scaleable Gateway during 1998.The IP/PSTN Scaleable Gateway is a PRPQ with Telecommunication and Media ISU. The gateway is based on the Direct Talk (DT/6000) product and on the Multi Service Platform (MSP/6000) product. The main features of the gateway are as follows:

Service creation environment based on Direct Talk Programmability tools providing high level Programming as well as APIs to create new Services.

1xT1/E1 to 4xT1/E1 ports in the first version. Both GSM and G.723.1 are deployed on the DSP (IBM SPN256 card).

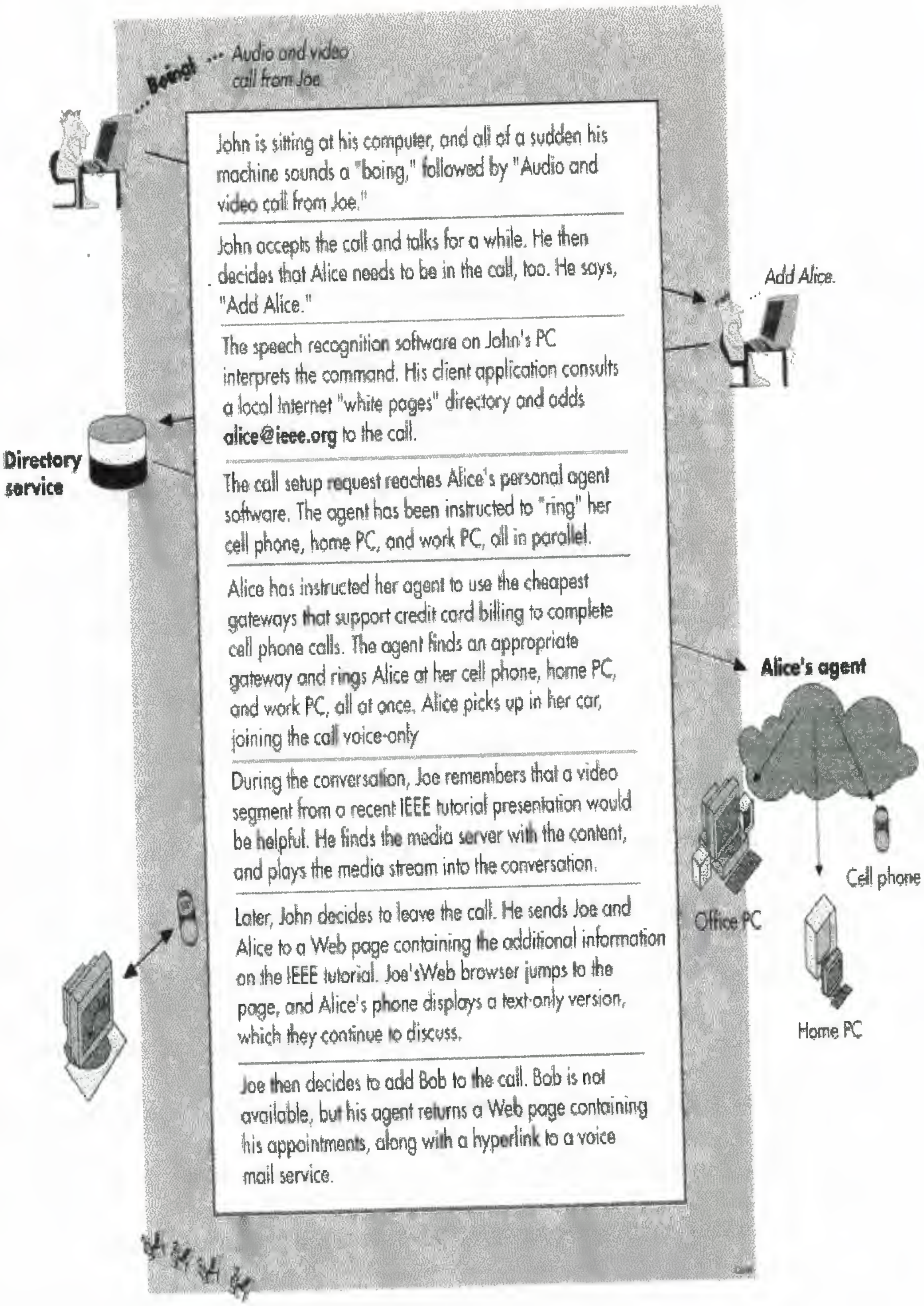
All audio technology features as listed above. Packet trucking which gains up to 70% on Bandwidth traffic as well as TCP/UDP I/O Overhead. The IP/PSTN Scaleable Gateway was developed using Distributed architecture concepts and function

decomposition concepts, i.e. all its components communicate via message on sockets preceding even new emerging standards, such as MGCP.

3.3.3 Internet telephony

Internet telephony was first used as a simple way to provide point-to-point voice transport between two IP hosts, primarily to replace expensive international phone calls. However, the growing interest in providing integrated voice, data, and video services has caused its scope to be expanded. Internet telephony now encompasses a range of services. These services include not only traditional conferencing, call control supplementary services, multimedia transport, and mobility, but also new services that integrate Web, e-mail, presence, and instant messaging applications with telephony. Furthermore, it is generally accepted that Internet telephony and traditional circuit-switched telephony will coexist for quite some time, requiring gateways between the two worlds.

Figure 3.3gives an example scenario of an integrated IP telephony call. The services contained in the call scenario require many protocol components in order to work. In this article we examine the various protocols and discuss how they fit into the broader picture.



John is sitting at his computer, and all of a sudden his machine sounds a "boing," followed by "Audio and video call from Joe."

John accepts the call and talks for a while. He then decides that Alice needs to be in the call, too. He says, "Add Alice."

The speech recognition software on John's PC interprets the command. His client application consults a local Internet "white pages" directory and adds **alice@ieee.org** to the call.

The call setup request reaches Alice's personal agent software. The agent has been instructed to "ring" her cell phone, home PC, and work PC, all in parallel.

Alice has instructed her agent to use the cheapest gateways that support credit card billing to complete cell phone calls. The agent finds an appropriate gateway and rings Alice at her cell phone, home PC, and work PC, all at once. Alice picks up in her car, joining the call voice-only

During the conversation, Joe remembers that a video segment from a recent IEEE tutorial presentation would be helpful. He finds the media server with the content, and plays the media stream into the conversation.

Later, John decides to leave the call. He sends Joe and Alice to a Web page containing the additional information on the IEEE tutorial. Joe'sWeb browser jumps to the page, and Alice's phone displays a text-only version, which they continue to discuss.

Joe then decides to add Bob to the call. Bob is not available, but his agent returns a Web page containing his appointments, along with a hyperlink to a voice mail service.

Figure3.3:integerated internet telephony

First, a signaling protocol is needed to allow calls between participants, and to establish a multimedia session so that the participants can exchange audio and video. We discuss signaling protocols in the next section. The actual audio and video are exchanged between session participants using a transport protocol called RTP, which we discuss after that. Directory access protocols, used to access white pages services, for example, are also important, but we do not discuss them further here, since they are the same as for e-mail service, for example.[1]

During the call illustrated in Figure 1, Joe brings in a media server and instructs it to play a video segment. This is accomplished using a streaming media control protocol, called the Real Time Streaming Protocol (RTSP), which we describe. The intelligent agent concept described in the figure interacts with the signaling protocol to provide advanced services. To realize these agents, we briefly describe a call processing language. We then present the Gateway Location Protocol (GLP), which helps the agent in selecting a gateway for terminating the call from the Internet on the telephone network.

This tutorial does not cover the protocols that allow controllers and signaling gateways -- gateways connecting to the public switched telephone network (PSTN) at the signaling layer -- to communicate. Proposals for such protocols are being discussed within the Internet Engineering Task Force (IETF) at this time, including Media Gateway Control Protocol (MGCP)[2] and Media Description Control Protocol (MDCP).[3] Other protocols, such as those for billing and authentication, are also beyond the scope of this survey.

3.4-SIGNALING PROTOCOLS

Signaling protocols are at the heart of Internet telephony and distinguish it from other services. They play several roles,[4] discussed below.

- User location. If user *A* wishes to communicate with user *B*, *A* first needs to find out where *B* is currently located on the network, so that the session establishment request (below) can reach him. This function is known as *user location*. Users can be in different places at different times, and even reachable by several means at the same time (by work PC or traditional phone). This function is particularly important for users whose PCs do not have a permanent IP address. (Almost all modem connections, including asynchronous digital subscriber line (ADSL) and cable modems, assign addresses to PCs dynamically using the Dynamic Host Configuration Protocol (DHCP).[5]

- Session establishment. The signaling protocol allows the called party to accept the call, reject it, or redirect it to another person, voice mail, or a Web page. (Generally, the terms *call* and *session* are used interchangeably in this article, although session has a somewhat wider meaning, including, for example, a group of hosts listening to an "Internet radio" multicast.)

- Session negotiation. The multimedia session being set up can comprise different media streams, including audio, video, and shared applications. Each of these media streams may use a variety of different speech and video compression algorithms, and take place on different multicast or unicast addresses and ports. The process of session negotiation allows the parties involved to settle on a set of session parameters. This process is also sometimes known as *capabilities exchange*.

- Call participant management. New members can be added to a session, and existing members can leave a session.

- Feature invocation. Call features, such as hold, transfer, and mute, require communication between parties.

Several protocols exist to fill this need. One is International Telecommunications Union (ITU) Recommendation H.323,[6] which describes a set of protocols. The IETF has defined two protocols to perform many of the above tasks: the Session Initiation Protocol (SIP)[7] and the Session Description Protocol (SDP).[8]

3.5 BASIC MECHANISMS IN H.323 protocol:

International Telecommunication Union —Telecommunication Standardization Sector (II'U-T) H.323-related Recommendations for enabling multimedia communications in packet-based networks. We then discuss the IETF QoS framework, specifically the integrated services model (Intserv) and differentiated services (Diff-serv) architecture

H.323 are a series of Recommendations of the ITU-T to enable multimedia communications in packet-switched networks [4]. H.323 is designed to extend the traditionally circuit-based services including audiovisual and multimedia conferencing services into packet-based networks. The Internet Telephony can be based on a subset function of the H.323 for voice only support. Therefore, one of the primary objectives of H.323 is the interoperability with the existing circuit-switching systems (PSTN and ISDN).

The basic elements defined in H.323 architecture are: terminals, gateways, gatekeepers, and multipoint control units (MCUs), in which the terminals, gateways, and MCUs are collectively referred as endpoints.

A terminal is an end user device, which can be a simple telephone or PC/workstation. Its main responsibility is to participate in H.323-defined communications, including both point-to-point calls and multipoint conferences.

A gateway, as the name suggests, is an intermediate device to provide interoperation between H.323 compliant devices and non-H.323 device in particular PSTN and ISDN devices. The main functionalities contain the translation of signaling media encoding, and packetization. There exist number of different types of gateways, for example gateways for PSTN devices and gateways for ISDN (H.320) videoconferencing devices.

A gatekeeper manages a set of register endpoints, collectively referred as a zone in main functions include call admission (or call authorization), address resolution, and other management-related functions (e.g., bandwidth allocation). Each endpoint before initiating call or conference has to register with the designated gatekeeper within the zone. The gatekeeper provides the address resolution to a special transport address of the target recipient. It also determines whether to accept or reject the call connection request based on the available bandwidth or other system parameters.

An MCU provides the necessary control needed for multiparty videoconferences. It contains two logical components: a multipoint controller (MC) for call control coordination and multipoint processor (MP) to handle audio ' video mixing.

The key protocols used in the call setup are the Registration Admission Status (RAS) protocol. Q.931-based signaling protocol, and an H.245 media and conference control protocol.

RAS protocol is responsible for registration of endpoints (terminals, gateways, and MCUs) to the correspondent gatekeeper. RAS message carried in User Datagram Protocol (UDP) packets contain a number of request/reply message exchanged between the endpoints and gatekeeper. Besides the registration, RAS protocol also provide a means for the gatekeeper to monitor the endpoints within the zone and manage tl' associated resources.

The Q.931-based signaling protocol is derived from the integrated services digital network (ISDN) 0.931 signaling protocol tailored for n in the H.323 environment. The signaling messages are carried in reliable TCP packets. It provides the logical connection between the calling and called parties.

The H.245 media and conference control protocol is used for the two connected parties (after 0.931 establishment) to exchange various information related to their communications; for instance, type of messages (audio, video, or data) and format. In addition, it provides a set of control functions for multiparty videoconferences.RTF and RTCP, described earlier, are used for actual message transmission.

A summary of the H.323 protocol phases is given in. fig The RAS protocol is used in phases 0, 1, and 6 for registration and shutdown process. Signaling protocol is involved in phases 2, 5, and 6. The H.245 media and conference control protocol is active during phases 3 and 5, and media exchanged based on RTP/RTCP is carried out in phase 4 [4].

3.6-SESSION INITIATION PROTOCOL

As its name implies, SIP is used to initiate a session between users. It provides for user location services (this is its greatest strength, in fact), call establishment, call participant management (using a SIP extension [10]), and limited feature invocation. Interestingly, SIP does not define the type of session that is established.

SIP can just as easily establish an interactive gaming session as an audio/video conference.

Each SIP request consists of a set of header fields that describe the call as a whole followed by a message body that describes the individual media sessions that make up the call. Currently, SDP (described below) is used, but consenting parties may agree on another capability exchange protocol.

SIP is a client-server protocol, similar in both syntax and semantics to Hypertext Transport Protocol (HTTP). [11] Requests are generated by one entity (the client) and sent to a receiving entity (the server). The server processes the requests, and then sends a response to the client. A request and the responses that follow it are called a *transaction*. The software on an end system that interacts with a human user is known as a *user agent*. A user agent contains two components, a user agent client (UAC) and a user agent server (UAS). The UAC is responsible for initiating calls (sending requests), and the UAS for answering calls (sending responses). A typical Internet telephony appliance or application contains both a UAS and a UAC. (Note that this differs from a Web browser, which acts only as a client.)

Within the network, there are three types of servers. A registration server receives updates on the current locations of users. A *proxy server* receives requests and forwards them to another server (called a *next-hop* *server)*, which has more precise location information about the callee. The next-hop server might be another proxy server, a UAS, or a redirect server. A *redirect server* also receives requests, and determines a next-hop server. However, instead of forwarding the request there, it returns the address of the next-hop server to the client. The primary function of proxy and redirect servers is *call routing* -- the determination of the set of servers to traverse in order to complete the call. A proxy or redirect server can use any means at its disposal to determine the next-hop server, including executing programs and consulting databases. A SIP proxy server can also *fork* a request, sending copies to multiple next-hop servers at once. This allows a call setup request to try many different locations at once. The first location to answer is connected with the calling party.

As in HTTP, the client requests invoke *methods* (commands) on the server. SIP defines several methods. INVITE invites a user to a call. BYE terminates a connection between two users in a call. OPTIONS solicits information about capabilities, but does not set up a call. ACK is used for reliable message exchanges for invitations. CANCEL terminates a search for a user. Finally, REGISTER conveys information about a user's location to a SIP registration server.

A client sets up a call by issuing an INVITE request. This request contains header fields used to convey information about the call. The most important header fields are To and From, which contain the callee's and caller's address, respectively. The Subject header field identifies the subject of the call. The Call-ID header field contains a unique call identifier, and the CSeq header field contains a sequence number. The Contact header field lists addresses where a user can be contacted. It is placed in responses from a redirect server, for example. The Require header field is used for negotiation of protocol features, providing extensibility. The Content-Length and Content-Type header fields are used to convey information about the body of the message. The body contains a description of the session which is to be established.

Extensions can be defined with new header fields. One such extension, used for call control,[10] defines several new headers used for feature invocation (such as call transfer) and multiparty conferencing.[12, 13]

A typical SIP transaction is depicted in Figure 3.4

3.7-SESSION DESCRIPTION PROTOCOL

SDP is used to describe multimedia sessions, for both telephony and distribution applications like Internet radio. The protocol includes information about:

- Media streams. A multimedia session can contain many media streams; for example, two audio streams, a video stream, and a whiteboard session. SDP conveys the number and type of each media stream. It currently defines audio, video, data, control, and application as stream types, similar to MIME types used for Internet mail.

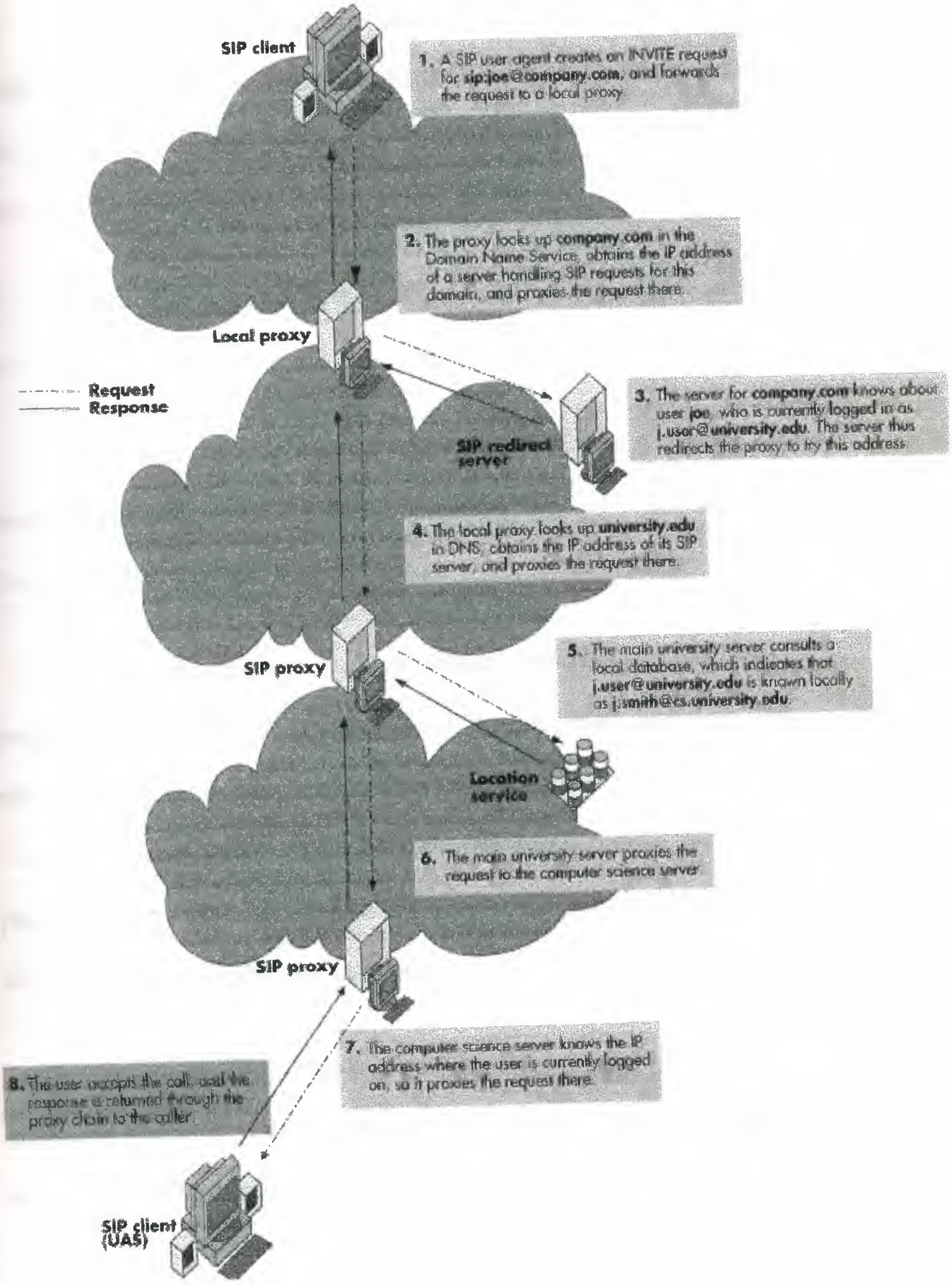

SIP client
1. A SIP user agent creates an INVITE request for sip:joe@company.com, and forwards the request to a local proxy.
2. The proxy looks up company.com in the Domain Name Service, obtains the IP address of a server handling SIP requests for this domain, and proxies the request there.
Local proxy
Request
Response
3. The server for company.com knows about user joe, who is currently logged in as j.user@university.edu. The server thus redirects the proxy to try this address.
SIP redirect server
4. The local proxy looks up university.edu in DNS, obtains the IP address of its SIP server, and proxies the request there.
SIP proxy
5. The main university server consults a local database, which indicates that j.user@university.edu is known locally as j.smith@cs.university.edu.
Location service
6. The main university server proxies the request to the computer science server.
SIP proxy
7. The computer science server knows the IP address where the user is currently logged on, so it proxies the request there.
8. The user accepts the call, and the response is returned through the proxy chain to the caller.
SIP client (UAS)

figure3.4:SIP transaction

- Addresses. For each stream, the destination address (unicast or multicast) is indicated. Note that the addresses for different media streams may differ, so a user may, for example, receive audio on a low-delay Internet telephone appliance and video on a workstation.

- Ports. For each stream, the UDP port numbers for sending and/or receiving are indicated.

- Payload types. The media formats that can be used during the session are also conveyed. For unicast sessions ("traditional" IP telephony), this list is called a *capability set*.

- Start and stop times. For broadcast-style sessions like a television program, the start, stop, and repeat times of the session are conveyed. Thus, one can announce or invite others to a weekly TV show or a Tuesday/Thursday lecture. (SIP can be used not just to make a traditional phone call, but a caller can also invite others to, say, a TV program, without the caller and callee talking to each other.)

- Originator. For broadcast-style sessions, the session description names the originator of the session and how that person can be contacted (e.g., in case of technical difficulties).

SDP conveys this information in a simple textual format. In fact, the acronym for SDP is a misnomer, since SDP is more of a description format. When a call is set up using SIP, the INVITE message contains an SDP body describing the session parameters acceptable to the caller. The response from the called party contains a modified version of this description, incorporating the capabilities of the called party.

The v line is a version identifier for the session. The o line is a set of values that form a unique identifier for the session. The u and e lines given the URL and e-mail addresses for further information about the session. The c line indicates the address for the session, the b line indicates the bandwidth (64 kbps in this case), and the t line the start and stop times (where 0 means that the session continues indefinitely). The k line conveys the encryption key for the session. There are three m lines, each of which identifies a media stream type (audio, video, and whiteboard application), the port number for that stream, the protocol, and a list of payload types. The a line specifies an attribute. For example, the line below the audio stream definition defines the codec parameters for RTP payload type 96.

3.8-REAL TIME TRANSPORT PROTOCOL

RTP,[14] as its name implies, supports the transport of real-time media (e.g., audio and video) over packet networks. (It is also used by H.323.)

The process of transport involves taking the bitstream generated by the media encoder, breaking it into packets, sending the packets across the network, and then recovering the bitstream at the receiver. The process is complex because packets can be lost, delayed by variable amounts, and reordered in the network. The transport protocol must allow the receiver to detect these losses. It must also convey timing information so that the receiver can correctly compensate for jitter (variability in delay). To assist in this function, RTP defines the formatting of the packets sent across the network. The packets contain the media information (called the *RTP payload*), along with an RTP header. This header provides information to the receiver that allows it to reconstruct the media. RTP also specifies how the codec bitstreams are broken up into packets.[15] RTP was also "engineered" for multicast, which means that it works in conferencing applications and broadcast environments where multicast is used to distribute the media. It is important to note that RTP does not try to reserve resources in the network to avoid packet loss and jitter; rather, it allows the receiver to recover in the presence of loss and jitter.

RTP plays a key component in any Internet telephony system. It is effectively at the heart of the application, moving the actual voice among participants. The relationship between the signaling protocol and RTP is that signaling protocols are used to establish the parameters for RTP transport.

RTP provides a number of specific functions:

- Sequencing. Each RTP packet contains a sequence number. This can be used for loss detection and compensation for reordering.

- Intramedia synchronization. Packets within the same stream may suffer different delays (jitter). Applications use playout buffers[16-18] to compensate for delay jitter. They need the timestamps provided by RTP to measure it.

- Payload identification. In the Internet, network conditions such as packet loss and delay vary, even during the duration of a single call. Speech and video codecs differ in their ability to work properly under various loss conditions. Therefore, it is desirable to be able to change the encoding for the media (the "payload" of RTP) dynamically as network conditions vary. To support this, RTP contains a payload type identifier in each packet to describe the encoding of the media.

- Frame indication. Video and audio are sent in logical units called *frames*. It is necessary to indicate to a receiver where the beginning or end of a frame is, in order to aid in synchronized delivery to higher layers. A frame marker bit is provided for this purpose.

- Source identification. In a multicast session, many users are participating. There must be a way for a packet to contain an indicator of which participant sent it. An identifier called the *synchronization source* (SSRC) is provided for this purpose.

RTP also has a companion control protocol, called Real Time Control Protocol (RTCP). RTCP provides additional information to session participants. In particular, it provides:

- QoS feedback. Receivers in a session use RTCP to report back the quality of their reception from each sender. This information includes the number of lost packets, jitter, and round-trip delays. This information can be used by senders for adaptive applications[19, 20, 21] which adjust encoding rates and other parameters based on feedback.

- Intermedia synchronization. For flexibility, audio and video are often carried in separate packet streams, but they need to be synchronized at the receiver to provide "lip sync." The necessary information for the synchronization of sources, even if originating from different servers, is provided by RTCP.

- Identification. RTCP packets contain information such as the e-mail address, phone number, and full name of the participant. This allows session participants to learn the identities of the other participants in the session.

- Session Control. RTCP allows participants to indicate that they are leaving a session (through a BYE RTCP packet). Participants can also send small notes to each other, such as "stepping out of the office."

RTCP mandates that all session participants (including those who send media and those who just listen) send a packet periodically which contains the information described above. These packets are sent to the same address (multicast or unicast) as the RTP media, but on a different port. The information is sent periodically since it contains time-sensitive information, such as reception quality, which becomes stale after some amount of time. However, a participant cannot just send an RTCP packet with a fixed period. Since RTP is used in multicast groups, there could be

sessions (like a large lecture) with hundreds or thousands of participants. If each one were to send a packet with a fixed period, the network would become swamped with RTCP packets. To fix this, RTCP specifies an algorithm that allows the period to increase in larger groups.[22]

3.9-REAL-TIME STREAMING PROTOCOL

RTSP[23] is used to control a stored media server. A stored media server is a device capable of playing prerecorded media from disk to the network, and recording multimedia content to disk. RTSP offers controls similar to those in a VCR remote control. A client can instruct the server to play, record, fast forward, rewind, and pause. It can also configure the server with the IP addresses, UDP ports, and speech codecs to use to deliver (or record) the media. Typically, media is sent from the media server using RTP.

Stored media has a number of applications in Internet telephony:

- A media server can record the content of a conference for future reference.

- Media servers can play media into an existing conference. For example, if participants in a multiparty conference are discussing a movie, it would be useful to be able to bring a media server into the conference, and have it play portions of the movie into the conference. (This is done in the example in Figure 1, where Joe has the media server play the IEEE tutorial into the conference.)

- Media servers can record voice mail. RTSP clients can use the protocol to control playback of the message. This would allow users to listen to their voice mail, and rewind to a critical part (e.g., a phone number in the message). RTSP can also be used to record the incoming or outgoing voice mail message.

A client executes the following steps to cause a media server to play back content.

- Obtain the presentation description. The first step is to obtain the presentation description. This description enumerates the various components of the session. As an example, a classroom presentation might contain three components: a document camera, a video view of the professor, and the audio. For each component, the description defines the media parameters needed to decode the component, such as the codec type and frame rate. The presentation description can be obtained in several ways. The client can issue a DESCRIBE request to the server, which causes the server to return a description. It is also possible to obtain a description through other means, such as a Web page.

- Set up the server. Once it has obtained the description, the client can issue a SETUP request to the server. This request establishes the destination to which the media should be delivered. The destination includes the IP address (unicast or multicast), port numbers, protocols (usually, but not necessarily, RTP over UDP), TTL, and number of multicast layers. In response to the SETUP message, the server returns a session id. This id is used in further requests.

- Issue media requests. After the stream has been set up, the client can issue media requests to the server. These include operations such as PLAY, RECORD, and PAUSE. The PLAY method encompasses seek, fast forward, and reverse, in addition to straight play. This is accomplished by including a Scale header, which indicates the time speedup (or speeddown) at which the media should be played. The Range header specifies where in the stream playback should start.

- Teardown. Once interaction with the server is complete, the client issues a TEARDOWN request. This request destroys any state associated with the session. Further requests for the given session id will no longer be valid.

3.10-CALL PROCESSING LANGUAGE

CPL[24, 25] is a scripting language that allows end users to specify the behavior of call agents that execute on their behalf. The call agents are invoked when a call arrives at a SIP server. These agents execute the instructions contained in the CPL. This allows end users to specify their own call services. For example, a user can instruct the agent to connect a call by trying a cell phone, home PC, and work PC all at once.

3.11-GATEWAY LOCATION PROTOCOL

SIP allows a user on the Internet to call another user, also on the Internet. What if an Internet user wishes to call someone not on the Internet, but rather, on the telephone network? In this case, an Internet telephony gateway is needed. This device is capable of converting signaling and media between a packet network and the telephone network (PSTN). We anticipate that many gateways will be deployed in the future by Internet service providers (ISPs) and telephone companies.

To complete a call to a PSTN endpoint, an IP host must send a SIP invitation to the gateway. However, given a phone number to call, how does the caller find and select one of the many gateways to complete the call? In theory, each gateway could dial (almost) any phone number, but the caller may want to minimize the distance of the PSTN leg of the call, for example. This function is supported by the Gateway Location Protocol (GLP).

An overall architecture for GLP is shown in Figure 3.3

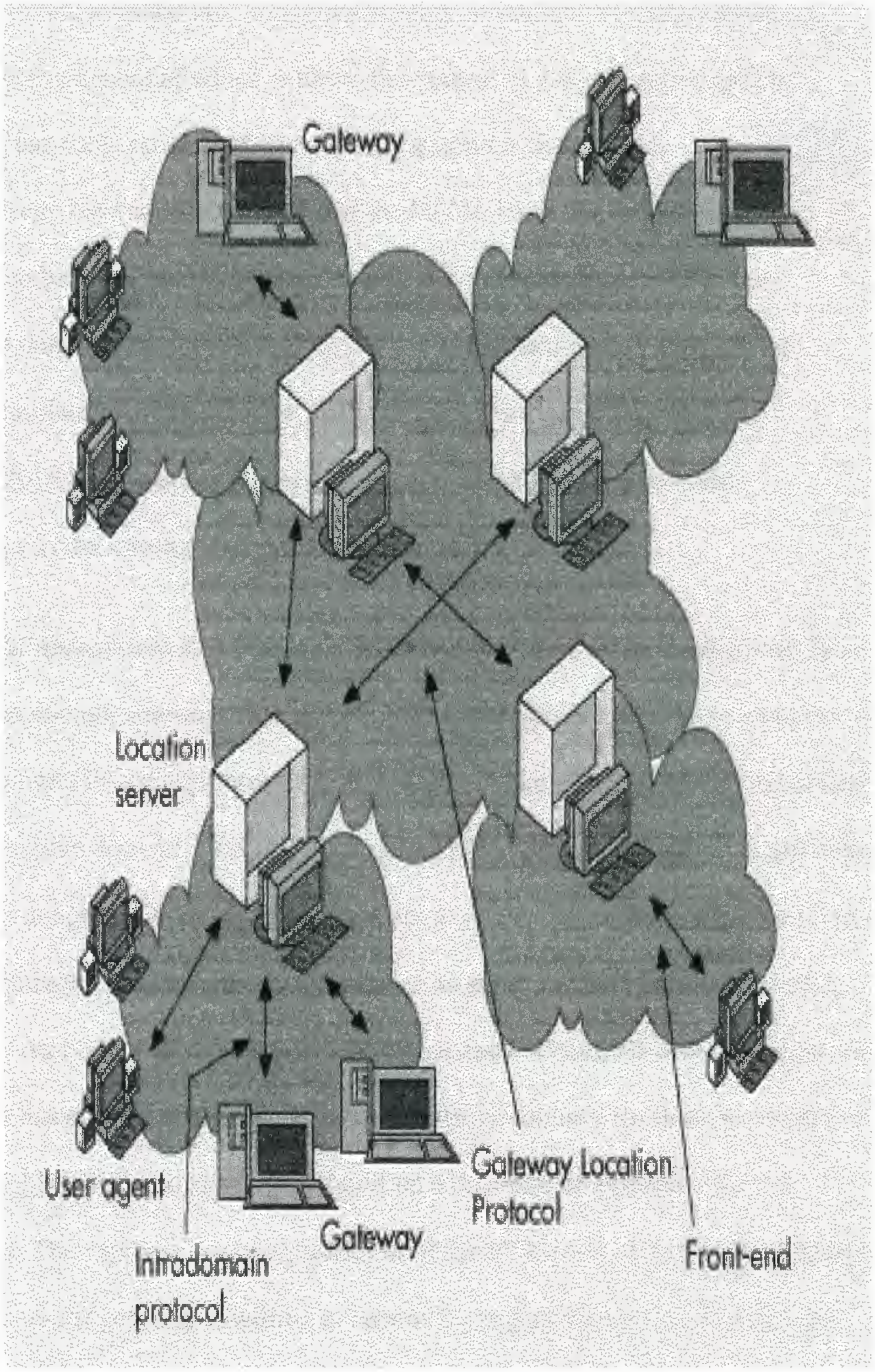


figure3.5: architecture for GLP

In that architecture there are a number of Internet telephony domains in the Internet, each of which is under the control of a single authority. Each domain has some number of IP telephony gateways that provide connectivity between the Internet and the PSTN. Each domain also has some number of IP users and some number of location servers (LS). The LSs in a domain know about the gateways in their own domains, by means of an intradomain protocol, such as the Service Location Protocol (SLP).[26] The intradomain protocol propagates information from the gateways to the location servers within a domain.

Unfortunately, it is unlikely that a single administrative domain will have access to enough gateways to complete calls to all possible telephone numbers. As a result, users in one administrative domain can make use of gateways in another administrative domain. This usually requires pre-established business relationships between domains. Once the agreements are in place, it is necessary for the LS in one domain to exchange information about its gateways with the LS in another domain. An LS can then take this information and exchange it with other LSs with which it has an established relationship. The protocol used for these exchanges is the GLP.[27] GLP allows an LS to build up a database of gateways in other domains. The database contains entries with the IP address of the gateway, a range of numbers the gateway is willing to terminate, and attributes that describe the gateway. These attributes include signaling protocols, cost information, and provider identifiers, among others. An LS can use the attributes to decide which gateways to use to terminate a call to a particular number. The information can also be used to determine which gateways to further advertise to other LSs. Both of these decisions are embodied in the *policy* that directs the behavior of the LS.

When a client within the domain wishes to make a call to a number in the phone network, it can proceed in several ways:

- Lightweight Directory Access Protocol. The database of the LS is made available through LDAP.[1] The calling client queries this database with the destination phone number, and the LS returns the IP address of the gateway. The client can then send a SIP invitation to that address.

- *Session Initiation Protocol.* Rather then sending a SIP invitation directly to the gateway, the caller sends it to the LS. The invitation contains the desired destination phone number. The LS consults its database, finds the right gateway, and proxies the call to it. In this case, the LS also acts as a SIP proxy. By acting as a proxy, the LS hides the gateway selection process from the caller. The caller application does not need to know whether the address being called is a phone number or a SIP universal resource locator (URL). In either case, the invitation is sent to the local proxy.

- Web pages. The LS can make its database available through Web pages. A user that wishes to make a call browses the Web page and finds the gateway it likes (perhaps this can be done through a Web form), and the

LS returns the address of the gateway on the Web page. The user copies this address to their SIP software, and completes a call to the gateway.

GLP is just beginning the process of specification. Because it is similar to existing interdomain IP routing protocols, such as BGP-4,[28] it is likely to borrow heavily from them.[29, 30]

3.12- QoS for IP:

The existing Internet service (i.e., the *best-effort service* of IP) cannot satisfy the QoS requirements of emerging multimedia applications, primarily caused by the variable queuing delays and packet loss during network congestion. There has been a significant amount of work in the past decade to extend the Internet architecture ;md protocols to provide QoS support for multimedia applications. This has led to the development of a number of service models and mechanisms..

3.12.1- The integrated service (Intserv) model:

The Intserv model was proposed as an extension to support real-time applications. The key is to provide some control over the end-to-end packet delays in order to meet the real-time QoS. specifically, the Intserv model proposes two service classes in addition to best-effort service. They arc:

• Guaranteed service for applications requiring a fixed delay bound

• Controlled-load service for application requiring reliable and enhanced best-effort service

The fundamental assumption of the Intserv model is that resources (e.g., bandwidth and buffer) must be explicitly managed for each real-lime application. This requires a router to reserve resources in order to provide specific QoS for packet streams, *OT flows,* which in turn requires flow-specific state in the router. The challenge is to ensure that this new service model can work seamlessly with the existing best-effort service in one common IP infrastructure.

Intserv is implemented by four components: flow specification, the signaling protocol (e.g., RSVP), admission control routine, and packet classifier and scheduler. Applications requiring guaranteed or controlled-load service must set up path and reserve resources before transmitting their data. *Howspec,* describing the source traffic characteristics, has to he provided to the network. Under the Intserv framework, two separate parts ot the Flowspcc are defined: one describes the flow's traffic characteristics (the *Tspec),* and theother specifies the service requested from the network (the *Rspec).* Admission control routines determine whether a request for resources can be granted. When a router receives a packet, the packet classifier will perform a classification and put the packet in the appropriate queue based on the classification result. The packet scheduler will then schedule the packet accordingly to meet its QoS requirement.

3.13- Existing solutions

In this section I present two leading companies' solutions to offering IP telephony services as examples to illustrate how real systems are implemented. The

Cisco IP telephony system described is targeted at enterprise networks, while the Lucent solution is for carrier networks.

3.13.1-THE Cisco SOLUTION

ENTERPRISE IP TELEPHONY

The Cisco solution for IP telephony in enterprise i networks includes hardware, such as switches. Routers, IP/PSTN gateways, desktop IP phones, and software, such as the call manager. An IP telephony system can be built by utilizing these products in the current IP infrastructure. Figure 5 illustrates a typical scenario of a Cisco IP telephony system.

In this IP telephony system, voice and data can be integrated in the wide area network (WAN) by permitting long distance calls to traverse the existing data infrastructure between remote locations. By using routers and gateways to connect the PBX, voice traffic can be carried over data IP networks. Call management software and IP telephones are deployed in the existing IP networks at each remote site. This will reduce the cost of WAN consolidation while ,it the same time eliminating the cost of installing a second network at each remote location. Using the analog access gateway (at the remote site), local calls can he enabled for remote users. Long distance calls can be routed over the WAN link and consolidated from the central site. With this approach, the transport for IP

.telephony becomes transparent to users, who will be unable to distinguish whether a call is placed over a packet network, a circuit-switched network, or a combination of both. The networks can support multiple classes of services (**CoSs**) and provide guaranteed **OoS** to real-time communications. **OoS** functions and mechanisms are distributed between cooperating edge/aggregation devices and core/backbone switches. Packet classification and user policies arc applied at the edge of the network. Packet classification identifies and categorizes network traffic into multiple classes. The Cisco IP phone can set the **IPv4 ToS** at the ingress to the network.

The **OoS** guarantees are primarily provided by two mechanisms: the call manager equipped with a resource reservation protocol (e.g., RSVP) and a priority queue mechanism. The priority queue mechanism is maintained in the core routers, and is responsible for high-speed switching and transport as well as congestion avoidance. Congestion avoidance uses packet discard mechanisms such as weighted random early detection (WRED) to randomly drop packets on a congested link. WRED ensures that the voice packets will get higher-priority services while no one user monopolizes network resources.

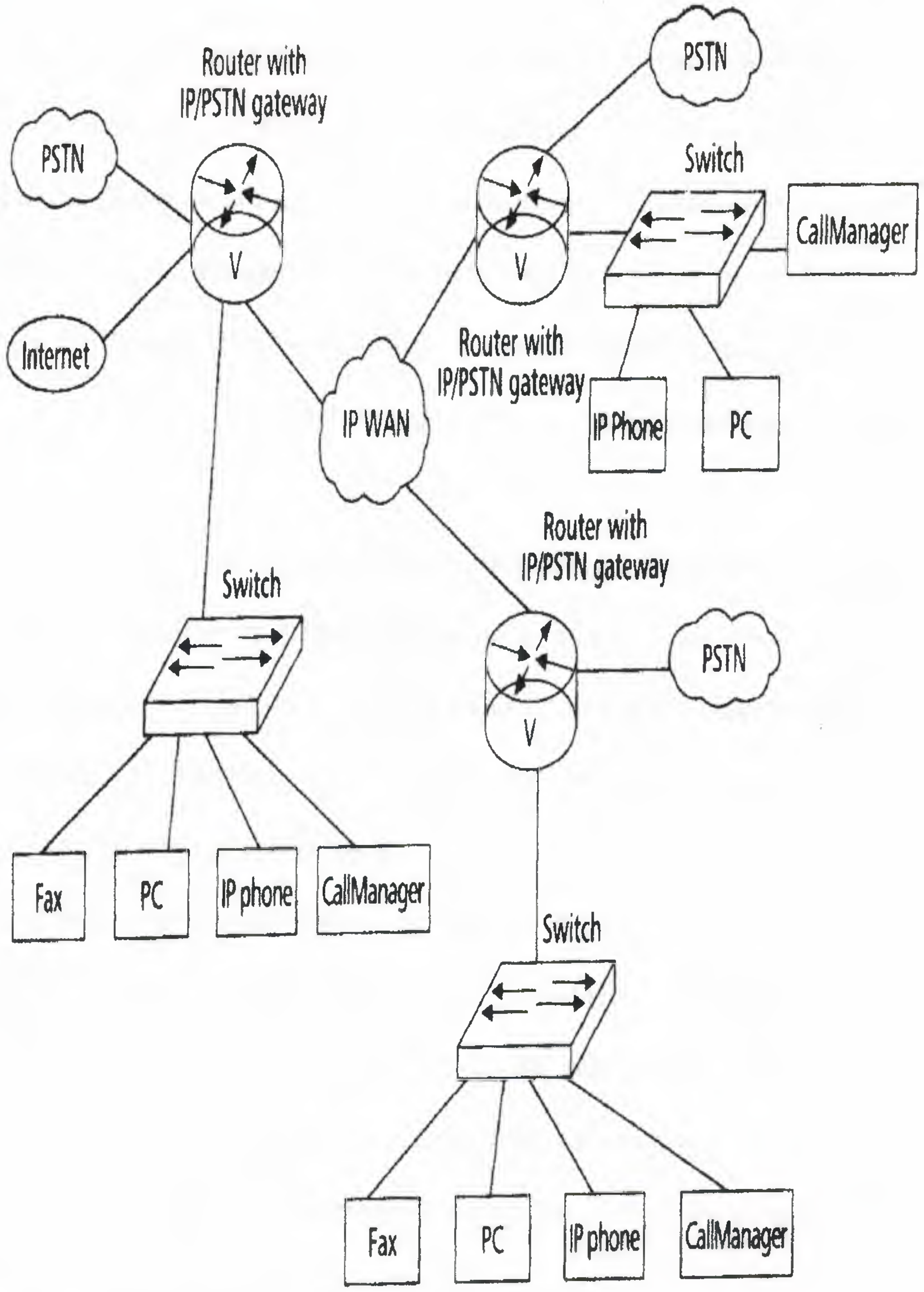


FIGURE3.6 THE CISCO DATA AND IP TELEPHONEY NETWORK CONFIGURATION

3.13.2-LUCENT GATEWAY SOLUTION FOR SERVICE PROVIDER NETWORKS

The Lucent Gateway approach is target for service provider networks [6]. In this architecture an H.323- or SIP-compliant terminal (e.g., an IP phone) is connected to the IP switch or router. The edge switches or routers serve as access points and concentrators **Figure** I'fie Cisco data and IP telephony network configuration.for the core IP network, which comprises higher-capacity IP routers or switches. A directory server is connected to the core network and serves multiple edge nodes.

The core network can be implemented using several different technologies: IP routers, IP switches, IP-over-ATM (asynchronous transfer mode) switches, IP over a synchronous optical network (SONET), and IP over dense wavelength-division multiplexing (DWDM). To the end terminal, the network is an IP network regardless of the underlying technologies.

Two gateways are added to the IP network architecture as interfaces to the public switched telephone network (PSTN). The first added is a connection gateway (CG), which performs signaling interworking between the IP protocol (e.g., H.323 or SIP) and PSTN protocols. The second is a voice gateway (VG), which converts time-division multiplexed (TDM) signals into IP packet and vice versa. The gateways allow a local area network

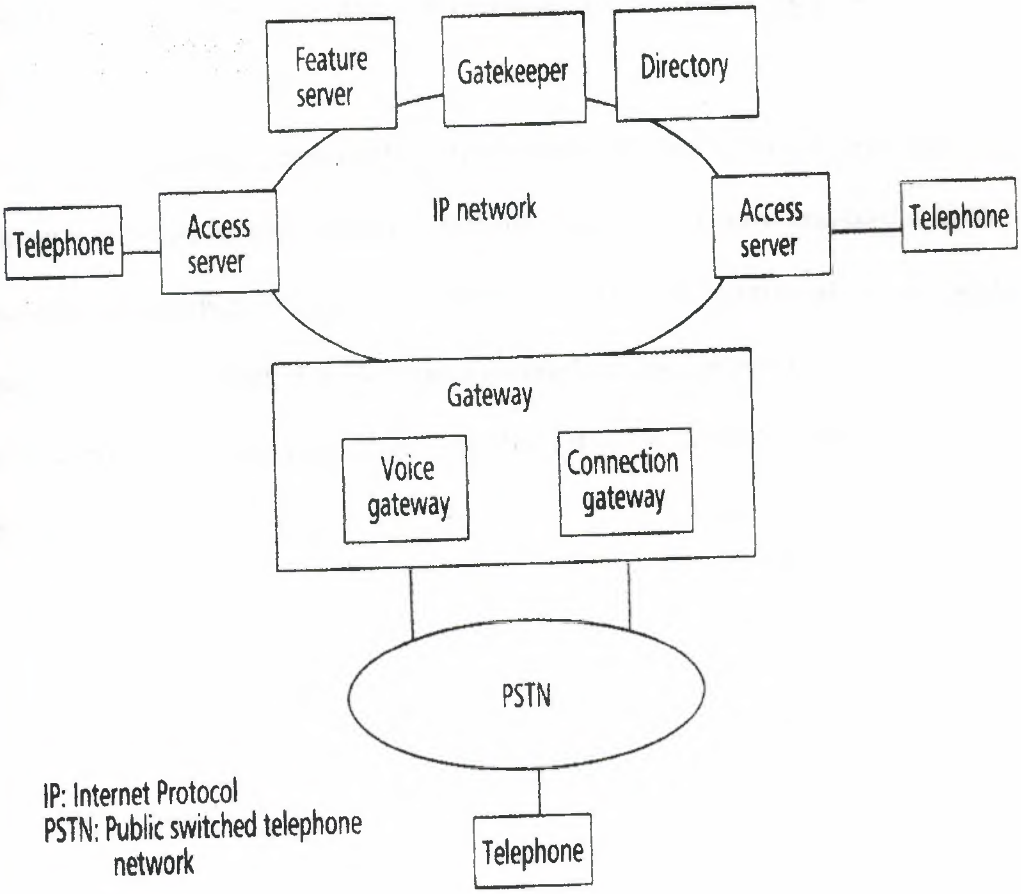


Figure3.7 . Lucent IP and PSTN architecture.

(LAN) telephone to call another LAN phone on the network and eliminates the need for a voice gateway. In addition, a LAN telephone can also call a regular plain old telephone service (POTS) phone through the gateway. Note, however, that within an IP network there is no distinction between local and long distance calls.

The Lucent router implements a straightforward scheme for QoS. It simply extracts **ToS** information from incoming IP packets and sets up a series of prioritized queues. These queues can control packet flow based on the **CoS** value, which allows the router to prioritize voice data and move fax data to a lower priority, thereby minimizing delay on real-time information at the expense of less time-critical information.

The difference between these two approaches lies in the fact that the Cisco system is targeted for the enterprise network, in which per-flow end-to-end **QoS** guarantee is possible. However, the requirement for setting up a path might not be feasible for the Internet, due to its poor scalability. The Lucent approach is used for carrier networks, which is more scalable but relies on the underlying IP network to provide the needed **QoS.**

Chapter four

Video over Internet

4.1-Introduction:

Video data transfer is one of the major types of traffic over the Internet. However, it is impossible for the current Internet to guarantee the quality of service (QoS) for video transfer, and an influx of a large amount of video data into the Internet .

The IETF has developed the notion of an Integrated Services Internet, which envisages a set of enhancements to IP to allow it to support integrated or multimedia services. These enhancements include traffic management mechanisms that closely match the traffic management mechanisms of ATM. For instance, protocols such as Resource Reservation Protocol (RSVP) are being defined to allow for resource reservation across an IP network, much as ATM signaling does within ATM networks.

RSVP is an advanced method for dynamically allocating bandwidth to network-based applications running in traditional packet-based networks. RSVP will be particularly useful for CBR multimedia applications because it will allow a network application to request a specific quality of service from the network. It will be the responsibility of internetworking devices (such as routers) to respond to the RSVP request and to establish a connection through the network that can support the requested quality of service.

The IP Version 6 (IPv6) protocol (formally known as the IP Next Generation (IPng protocol), which the IETF is now developing as a replacement for the current IPv4 protocol, incorporates support for a flow ID within the packet header. The network uses the flow ID to identify flows, much as VPI/VCI (virtual path identifier/virtual channel identifier) are used to identify streams of ATM cells. Protocols such as RSVP will be used to associate with each flow a flow specification that characterizes the traffic parameters of the flow, much as the ATM traffic contract is associated with an ATM connection.

The IETF is also in the process of developing a new transport protocol, the Real-Time Transport Protocol (RTP). RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data (such as audio, video, or simulation data) over multicast or unicast network services. RTP builds on protocols like RSVP for resource reservation and on transport technologies such as ATM for quality of service guarantees. The services provided by RTP to real-time applications include payload type identification, sequence numbering, time stamping, and delivery monitoring.

The concept of a Multicast Address Resolution Server (MARS), which can be considered the analog of the ARP server in RFC 1577, is also in development. A MARS serves a group of nodes known as a *cluster*. All end systems within the cluster are configured with the ATM address of the MARS. The MARS supports multicast through multicast meshes of overlaid point-to-multipoint connections, or through multicast servers.

4.2- Multimedia Applications:

There is a wide range of network multimedia applications to choose from, so it is important to understand why a particular application is being deployed. Additionally, it is important to understand the bandwidth implications of the chosen application.

4.2.1Types of Applications:

Network multimedia applications fall into the following categories:

- Point-to-Point Bi-directional Applications
- Point-to-Multipoint Bi-directional Applications
- Point-to-Point Unidirectional Applications
- Point-to-Multipoint Unidirectional Applications

4.2.2Point-to-Point Bi-directional Applications

Point-to-point bi-directional applications, as shown in Figure deliver real-time, point-to-point communication. The process is bi-directional, meaning that video can be transmitted in both directions in real time.

Examples of point-to-point bi-directional applications include the following:

- Audio and videoconferencing
- Shared whiteboard
- Shared application

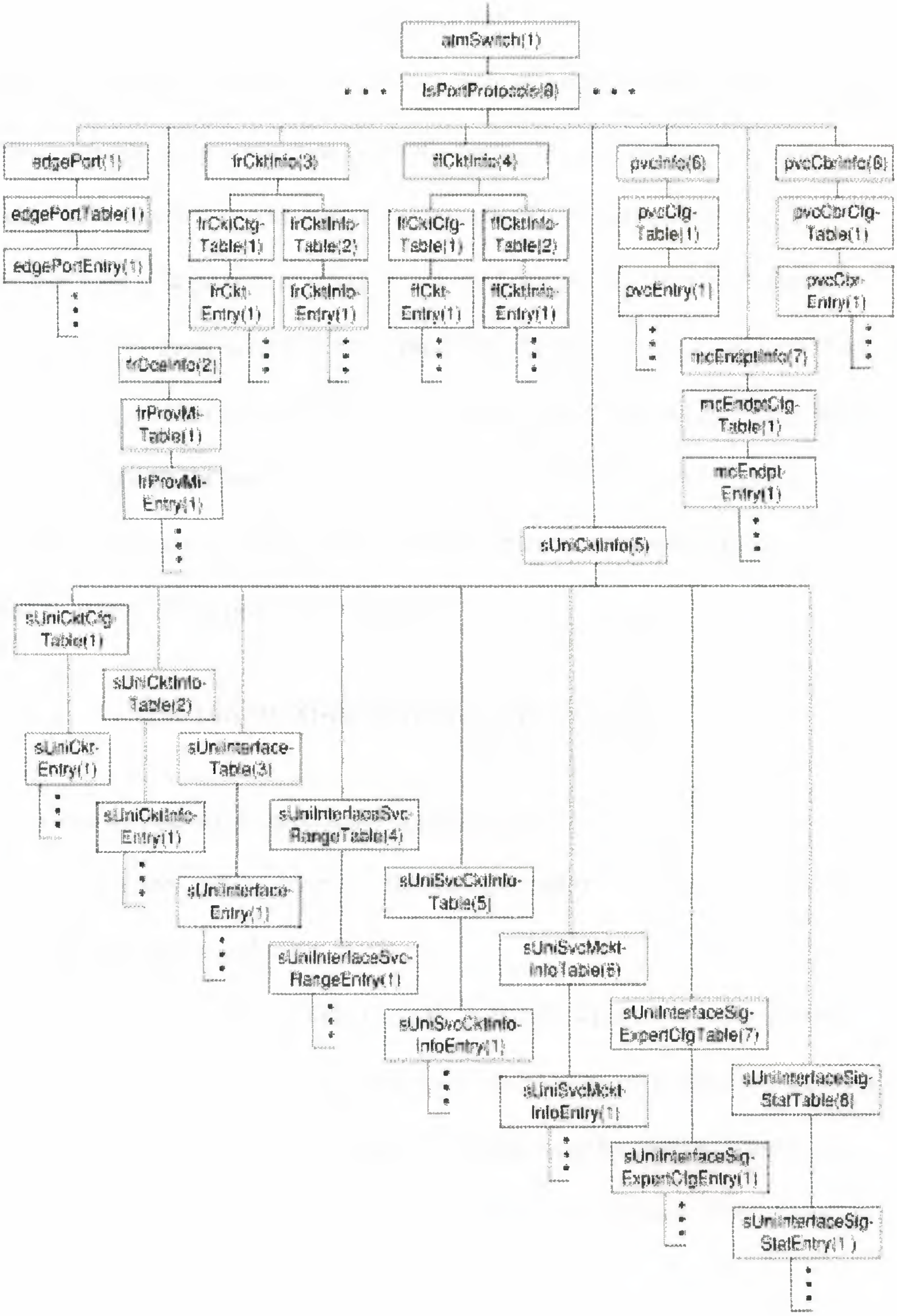


Figure4.1: Point-to-point bi-directional applications.

Audio and videoconferencing applications provide a real-time interactive environment for two users. Often, these applications also include a shared whiteboard application or an application-sharing functionality. Shared whiteboard applications provide a common area that both users can see and draw on. Shared whiteboards (also known as collaborative workspaces) are particularly useful in conversations where "a picture is worth a thousand words." Application sharing is also a useful and productive tool. With application sharing, one user can launch an application, such as Microsoft Access, and the user at the other end can view and work with it as though the application were installed on that user's computer. Coworkers at opposite ends of a network can collaborate in an application regardless of where the application resides.

4.2.3-Point-to-Multipoint Bi-directional Applications

Point-to-multipoint bidirectional applications as shown in Figure use multiple video senders and receivers. In this model, multiple clients can send and receive a video stream in real time.

Interactive video, such as video kiosks, delivers video to multiple recipients. The recipients, however, can interact with the video session by controlling start and stop functions. The video content can also be manipulated by end-user interaction. Some kiosks, for example, have a touch pad that delivers different videos based on the user's selection.

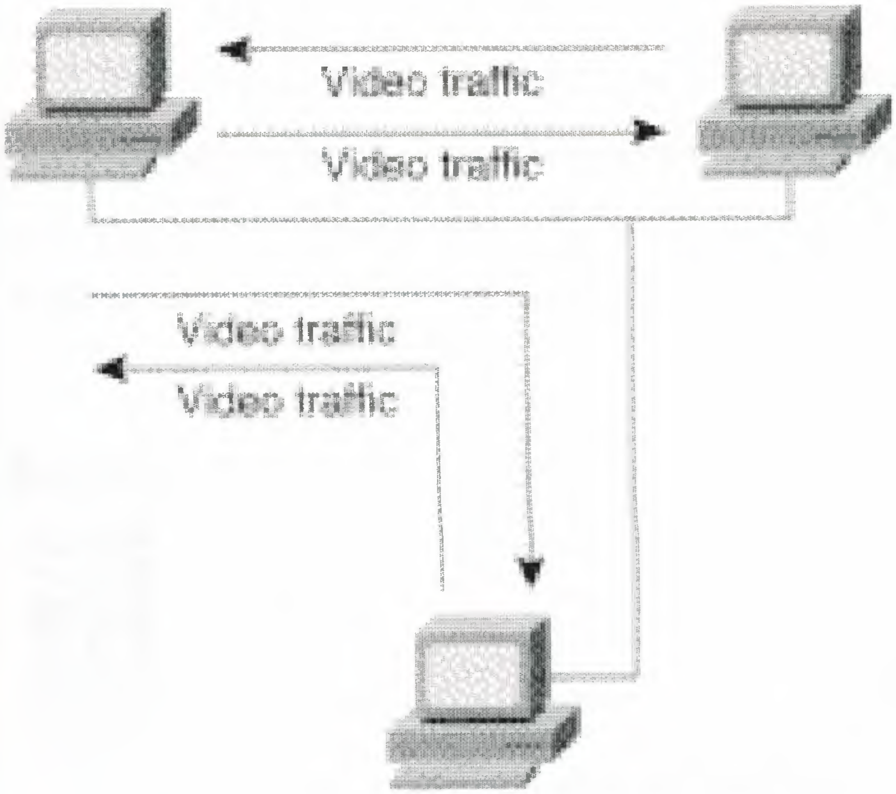


Figure4.2-Point-to-multipoint bidirectional applications.

Examples of point-to-multipoint bidirectional applications include the following:

- Interactive video
- Videoconferencing

Like a telephone calls in which multiple listeners participate, the same can be done with certain videoconferencing applications. For example, a three-way video conference call can occur in which each person can receive video and audio from the other two participants.

4.2.4- Point-to-Point Unidirectional Applications

Point-to-point unidirectional applications, as shown in Figure use point-to-point communications in which video is transmitted in only one direction. The video itself can be a stored video stream or a real-time stream from a video recording source.

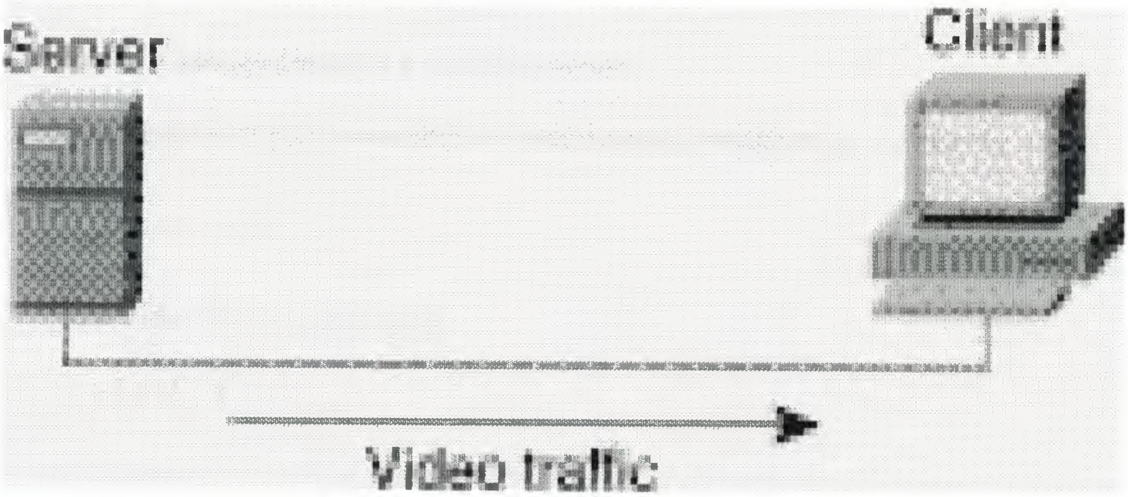


Figure4.3 : Point-to-point unidirectional applications.

Examples of point-to-point unidirectional applications include the following:

- Video server applications
- Multimedia-enabled email applications

In point-to-point unidirectional applications, compressed video clips are stored centrally. The end user initiates the viewing process by downloading the stream across the network to the video decompressor, which decompresses the video clip for viewing.

4.2.4- Point-to-Multipoint Unidirectional Applications

Point-to-multipoint unidirectional applications, as shown in Figure are similar to point-to- point unidirectional applications except that the video is transmitted to a group of clients. The video is still unidirectional. The video can come from a storage device or a recording source.

Examples of point-to-multipoint unidirectional applications include the following:

- Video server applications
- LAN TV

Both of these applications provide unidirectional video services. Video server applications deliver to multiple clients video streams that have already been compressed. LAN TV applications deliver stored video streams or real-time video from a camera source. Distance learning, in which classes are videotaped and then broadcast over the LAN and WAN to remote employees, is a popular example of a point-to-multipoint unidirectional video application.

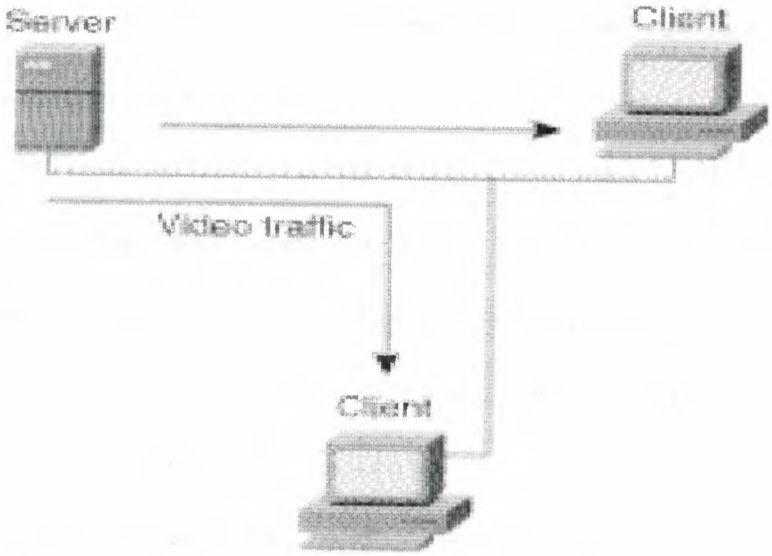


Figure 4.4: Point-to-multipoint unidirectional applications.

4.3-Bandwidth Requirements

Bandwidth requirements for network multimedia applications can range anywhere from 100 Kbps to 70 or 100 Mbps.

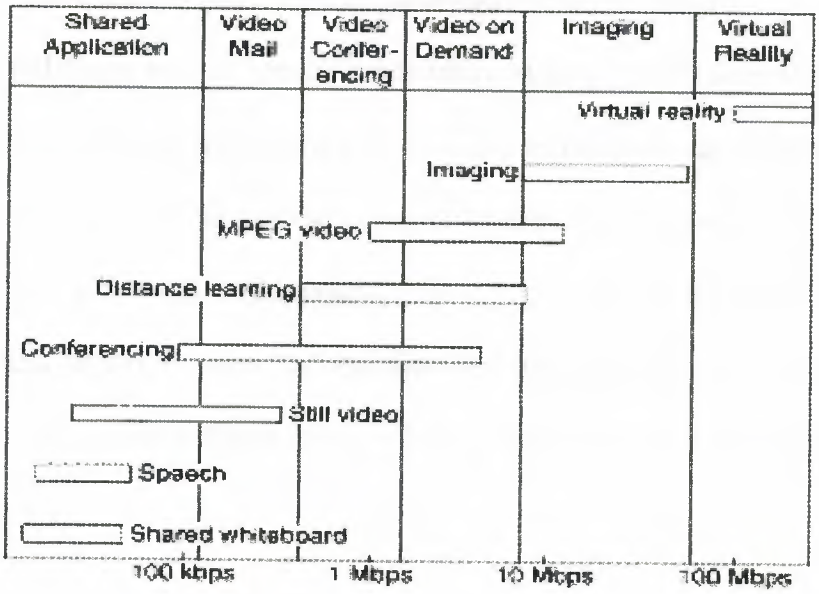


Figure4.5: Network bandwidth usage.

. Figure shows the amount of bandwidth that the various types of network multimedia applications require.

As Figure indicates, the type of application has a direct impact on the amount of LAN or WAN bandwidth needed. Assuming that bandwidth is limited, the choice is either to select a lower quality video application that works within the available bandwidth, or consider modifying the network infrastructure to deliver more overall bandwidth.

Two techniques reduce bandwidth consumption:

- Video Capture Manipulation
- Video Compression

4.4-Video Capture Manipulation

Manipulating video capture parameters involves changing resolution, color depth, and frame rate. To reduce bandwidth consumption, all three variables are often changed. For example, some multimedia applications capture video at 320 ¥ 240 with 8-bit color and at a frame rate of 15 frames per second. With these parameters, bandwidth requirements drop to 9.216 Mbps. Although this level of bandwidth is difficult for a 10-Mbps Ethernet network to achieve, it can be provided by 16-Mbps Token Ring, 100-Mbps Fast Ethernet, and other higher-speed technologies.

4.5-Video Compression

Video compression is a process whereby a collection of algorithms and techniques replace the original pixel-related information with more compact mathematical descriptions. Decompression is the reverse process of decoding the mathematical descriptions back to pixels for display. At its best, video compression is transparent to the end user. The true measure of a video compression scheme is how little the end user notices its presence, or how effectively it can reduce video data rates without adversely affecting video quality. An example of post-digitization video compression is shown in Figure

Video compression is performed using a CODEC (Coder/Decoder or Compressor/Decompressor). The CODEC, which can be implemented either in software or hardware, is responsible for taking a digital video stream and compressing it and for receiving a precompressed video stream and decompressing it. Although most PC, Macintosh, and UNIX video capture cards include the CODEC, capture and compression remain separate processes.

There are two types of compression techniques:

4.5.1- Loss less

A compression technique that creates compressed files that decompress into exactly the same file as the original. Lossless compression is typically used for executables (applications) and data files for which any change in

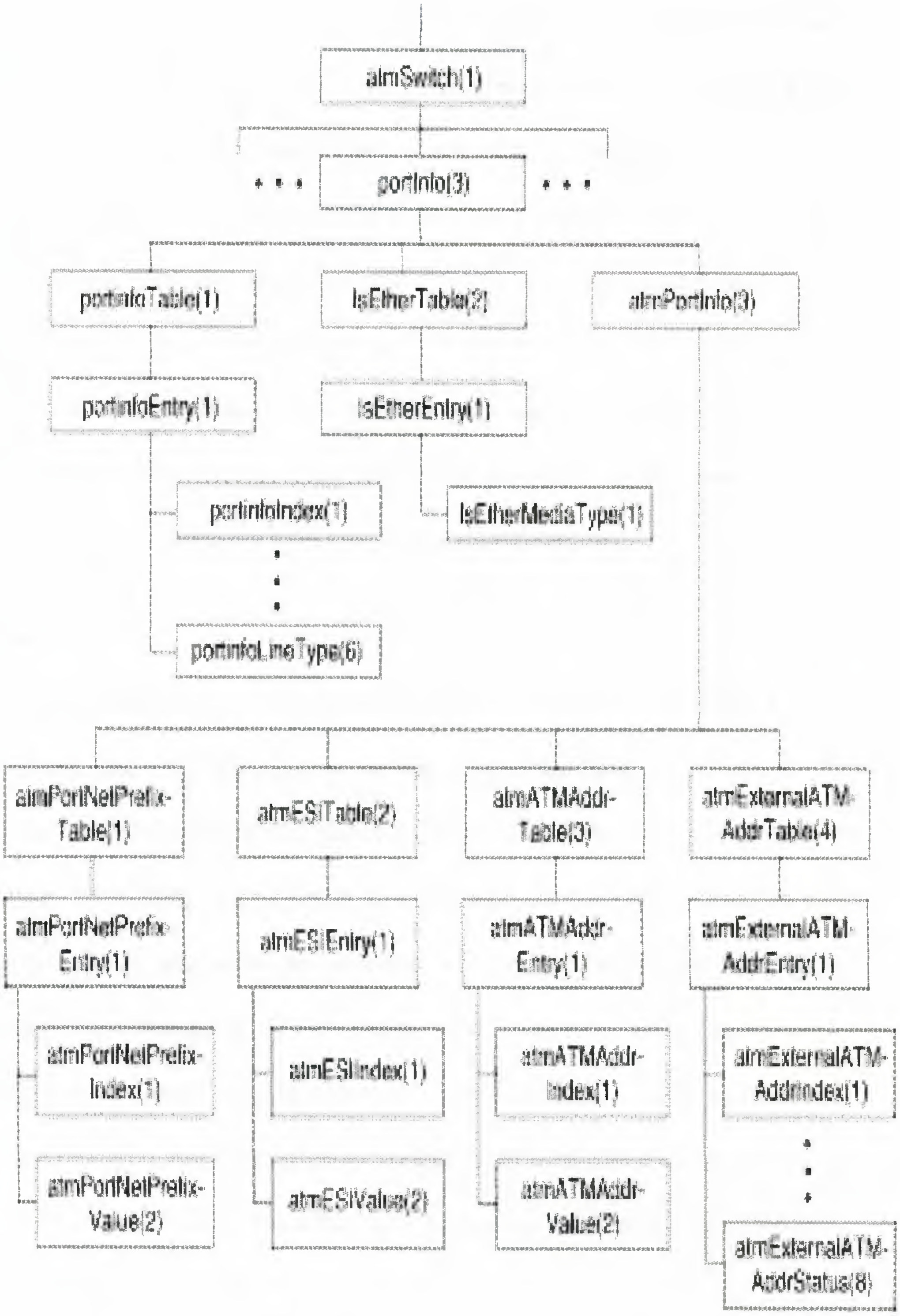


Figure4.6- Post-digitization video compression

digital makeup renders the file useless. In general, lossless techniques identify and utilize patterns within files to describe the content more efficiently. This works well for files with significant redundancy, such as database or spreadsheet files. However, lossless compression typically yields only about 2:1 compression, which barely dents high-resolution uncompressed video files. Lossless compression is used by products such as STAC and Double Space to transparently expand hard drive capacity, and by products like PKZIP to pack more data onto floppy drives. STAC and another algorithm called Predictor are supported in the Cisco IOS software for data compression over analog and digital circuits.

4.5.2-Lossy

Lossy compression, used primarily on still image and video image files, creates compressed files that decompress into images that look similar to the original but are different in digital makeup. This "loss" allows lossy compression to deliver from 2:1 to 300:1 compression. Lossy compression cannot be used on files, such as executables, that when decompressed must match the original file. When lossy compression is used on a 24-bit image, it may decompress with a few changed pixels or altered color shades that cannot be detected by the human eye. When used on video, the effect of lossy compression is further minimized because each image is displayed for only a fraction of a second (1/15 or 1/30 of a second, depending on the frame rate).

A wide range of lossy compression techniques is available for digital video. This simple rule applies to all of them: the higher the compression ratio, the higher the loss. As the loss increases, so does the number of artifacts. (An artifact is a portion of a video image for which there is little or no information.)

In addition to lossy compression techniques, video compression involves the use of two other compression techniques:

4.6-Interframe compression:

Compression between frames (also known as temporal compression because the compression is applied along the time dimension).

4.7-Intraframe compression

Compression within individual frames (also known as spatial compression).

Some video compression algorithms use both interframe and intraframe compression. For example, Motion Picture Experts Group (MPEG) uses Joint Photographic Experts Group (JPEG), which is an intrafame technique, and a separate interframe algorithm. Motion-JPEG (M-JPEG) uses only intraframe compression.

4.6.1- Interframe Compression

Interframe compression uses a system of key and delta frames to eliminate redundant information between frames. Key frames store an entire frame, and delta frames record only changes. Some implementations compress the key frames, and others don't. Either way, the key frames serve as a reference source for delta frames. Delta frames contain only pixels that are different from the key frame or from the immediately preceding delta frame. During decompression, delta frames look back to their respective reference frames to fill in missing information.

Different compression techniques use different sequences of key and delta frames. For example, most video for Windows CODECs calculate interframe differences between sequential delta frames during compression. In this case, only the first delta frame relates to the key frame. Each subsequent delta frame relates to the immediately preceding delta frame. In other compression schemes, such as MPEG, all delta frames relate to the preceding key frame.

All interframe compression techniques derive their effectiveness from interframe redundancy. Low-motion video sequences, such as the head and shoulders of a person, have a high degree of redundancy, which limits the amount of compression required to reduce the video to the target bandwidth.

Until recently, interframe compression has addressed only pixel blocks that remained static between the delta and the key frame. Some new CODECs increase compression by tracking moving blocks of pixels from frame to frame. This technique is called motion compensation (also known as dynamic carry forwards*)* because the data that is carried forward from key frames is dynamic. Consider a video clip in which a person is waving an arm. If only static pixels are tracked between frames, no interframe compression occurs with respect to the moving parts of the person because those parts are not located in the same pixel blocks in both frames. If the CODEC can track the motion of the arm, the delta frame description tells the decompressor to look for particular moving parts in other pixel blocks, essentially tracking the moving part as it moves from one pixel block to another.

Although dynamic carry forwards are helpful, they cannot always be implemented. In many cases, the capture board cannot scale resolution and frame rate, digitize, and hunt for dynamic carry forwards at the same time.

Dynamic carry forwards typically mark the dividing line between hardware and software CODECs. Hardware CODECs, as the name implies, are usually add-on boards that provide additional hardware compression and decompression operations. The benefit of hardware CODECs is that they do not place any additional burden on the host CPU in order to execute video compression and decompression.

Software CODECs rely on the host CPU and require no additional hardware. The benefit of software CODECs is that they are typically cheaper and easier to install. Because they rely on the host's CPU to perform compression and decompression, software CODECs are often limited in their capability to use techniques such as advanced tracking schemes.

4.7.1-Intraframe Compression

Intraframe compression is performed solely with reference to information within a particular frame. It is performed on pixels in delta frames that remain after interframe compression and on key frames. Although intraframe techniques are often given the most attention, overall CODEC performance relates more to interframe efficiency than intraframe efficiency. The following are the principal intraframe compression techniques:

4.7.2Run Length Encoding (RLE)

A simple lossless technique originally designed for data compression and later modified for facsimile. RLE compresses an image based on "runs" of pixels. Although it works well on black-and-white facsimiles, RLE is not very efficient for color video, which have few long runs of identically colored pixels.

4.7.3-JPEG

A standard that has been adopted by two international standards organizations: the ITU (formerly CCITT) and the ISO. JPEG is most often used to compress still images using discrete cosine transform (DCT) analysis. First, DCT divides the image into 8¥8 blocks and then converts the colors and pixels into frequency space by describing each block in terms of the number of color shifts (frequency) and the extent of the change (amplitude). Because most natural images are relatively smooth, the changes that occur most often have low amplitude values, so the change is minor. In other words, images have many subtle shifts among similar colors but few dramatic shifts between very different colors.

Next, quantization and amplitude values are categorized by frequency and averaged. This is the lossy stage because the original values are permanently discarded. However, because most of the picture is categorized in the high-frequency/low-amplitude range, most of the loss occurs among subtle shifts that are largely indistinguishable to the human eye.

After quantization, the values are further compressed through RLE using a special zigzag pattern designed to optimize compression of like regions within the image. At extremely high compression ratios, more high-frequency/low-amplitude changes are averaged, which can cause an entire pixel block to adopt the same color. This causes a blockiness artifact that is characteristic of JPEG-compressed images. JPEG is used as the intraframe technique for MPEG.

4.7.4-Vector quantization (VQ)

A standard that is similar to JPEG in that it divides the image into 8¥8 blocks. The difference between VQ and JPEG has to do with the quantization process. VQ is a recursive, or multistep algorithm with inherently self-correcting features. With VQ, similar blocks are categorized and a reference block is constructed for each category. The original blocks are then discarded. During decompression, the single reference block replaces all of the original blocks in the category.

After the first set of reference blocks is selected, the image is decompressed. Comparing the decompressed image to the original reveals many differences. To address the differences, an additional set of reference blocks is created that fills in the gaps created during the first estimation. This is the self-correcting part of the algorithm. The process is repeated to find a third set of reference blocks to fill in the remaining gaps. These reference blocks are posted in a lookup table to be used during decompression. The final step is to use lossless techniques, such as RLE, to further compress the remaining information.

VQ compression is by its nature computationally intensive. However, decompression, which simply involves pulling values from the lookup table, is simple and fast. VQ is a

public-domain algorithm used as the intraframe technique for both Cinepak and Indeo.

4.8- End-User Video Compression Algorithms

The following are the most popular end-user video compression algorithms. Note that some algorithms require dedicated hardware.

4.8.1MPEG1

A bit stream standard for compressed video and audio optimized to fit into a bandwidth of 1.5 Mbps. This rate is special because it is the data rate of uncompressed audio CDs and DATs. Typically, MPEG1 is compressed in non-real time and decompressed in real time. MPEG1 compression is typically performed in hardware; MPEG1 decompression can be performed in software or in hardware.

4.8.2-MPEG2

A standard intended for higher quality video-on-demand applications for products such as the "set top box." MPEG2 runs at data rates between 4 and 9 Mbps. MPEG2 and variants are being considered for use by regional Bell carriers and cable companies to deliver video-on-demand to the home as well as for delivering HDTV broadcasts. MPEG2 chip sets that perform real-time encoding are available.

Real-time MPEG2 decompression boards are also available. A specification for MPEG2 adaptation over ATM AAL5 has been developed.

4.8.3-MPEG4

A low-bit-rate compression algorithm intended for 64-Kbps connections. MPEG4 can be used for a wide range of applications including mobile audio, visual applications, and electronic newspaper sources.

4.8.4M-JPEG (Motion-JPEG)

The aggregation of a series of JPEG-compressed images. M-JPEG can be implemented in software or in hardware.

4.8.5-Cell B

Part of a family of compression techniques developed by Sun Microsystems. Cell B is designed for real-time applications, such as videoconferencing, that require real-time video transmission. Cell A is a counterpart of Cell B that is intended for non-real time applications where encoding does not need to take place in real time. Both Cell A and Cell B use VQ and RLE techniques.

4.8.6-Indeo

Developed by Intel. Indeo uses VQ as its intraframe engine. Intel has released three versions of Indeo:

4.8.6.1-Indeo 2.1

Focused on Intel's popular capture board, the Smart Video Recorder, using intraframe compression.

4.8.6.2-Indeo 3.1

Introduced in late 1993 and incorporated interframe compression.

4.8.6.3-Indeo 3.2

Requires a hardware add-on for video compression but decompression can take place in software on a high-end 486 or Pentium processor.

4.8.7-Cinepak

Developed by SuperMatch, a division of SuperMac Technologies. Cinepak was first introduced as a Macintosh CODEC and then migrated to the Windows platform in 1993. Like Indeo, Cinepak uses VQ as its intraframe engine. Of all the CODECs, Cinepak offers the widest cross-platform support, with versions for 3D0, Nintendo, and Atari platforms.

4.8.8-Apple Video

A compression technique used by applications such as Apple Computer's quickTime Conferencing.

4.8.9H.261

The compression standard specified under the H.320 videoconferencing standard. H.261 describes the video coding and decoding methods for the moving picture component of audio-visual services at the rate of *p* ¥ 64 Kbps, where *p* is in the range 1 to 30. It describes the video source coder, the video multiplex coder, and the transmission coder. H.261 defines two picture formats:

4.8.9.1-Common Intermediate Format (CIF)

Specifies 288 lines of luminance information (with 360 pixels per line) and 144 lines of chrominance information (with 180 pixels per line).

4.8.9.2-Quarter Common Intermediate Format (QCIF)

Specifies 144 lines of luminance (with 180 pixels per line) and 72 lines of chrominance information (with 90 pixels per line). The choice between CIF and QCIF depends on available channel capacity---that is, QCIF is normally used when *p* is less than 3.

The actual encoding algorithm of H.261 is similar to (but incompatible with) MPEG. Also, H.261 needs substantially less CPU power for real-time encoding than MPEG. The H.261 algorithm includes a mechanism for optimizing bandwidth usage by trading picture quality against motion so that a quickly changing picture has a lower quality than a relatively static picture. When used in o Protocol Independent Multicast (PIM), which is a multicast protocol that can be used with all unicast IP routing protocols, as defined in the two Internet standards-track drafts entitled Protocol Independent Multicast (PIM): Motivation and Architecture *and* Protocol Independent Multicast (PIM): Protocol Specification.

4.10-IP Multicast Group Addressing:

Figure-4.7 shows the format of a Class D IP multicast address.

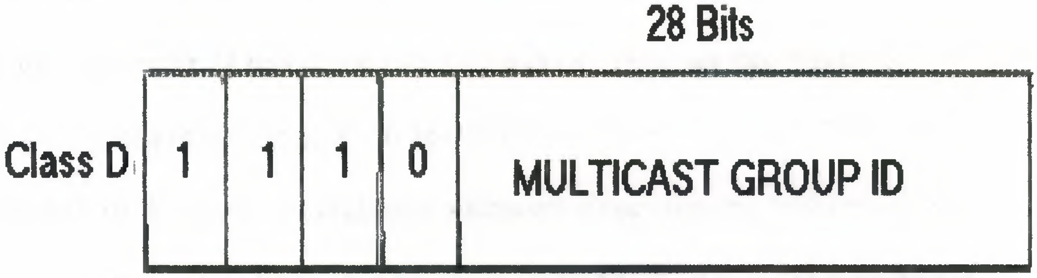


Figure4.7-Class D address format.

Unlike Class A, B, and C IP addresses, the last 28 bits of a Class D address have no structure. The multicast group address is the combination of the high-order 4 bits of 1110 and the multicast group ID. These are typically written as dotted-decimal numbers and are in the range 224.0.0.0 through 239.255.255.255. Note that the high-order bits are 1110. If the bits in the first octet are 0, this yields the 224 portion of the address.

The set of hosts that responds to a particular IP multicast address is called a *host group*. A host group can span multiple networks. Membership in a host group registration, see the section called "Internet Group Management Protocol" later in this chapter.

Some multicast group addresses are assigned as well-known addresses by the Internet Assigned Numbers Authority (IANA). These multicast group addresses are called permanent host groups and are similar in concept to the well-known TCP and UDP port numbers. Address 224.0.0.1 means "all systems on this subnet," and 224.0.0.2 means "all routers on this subnet."

Table lists the multicast address of some permanent host groups.

The IANA owns a block of Ethernet addresses that in hexadecimal is 00:00:5e. This is the high-order 24 bits of the Ethernet address, meaning that this block includes addresses in the range 00:00:5e:00:00:00 to 00:00:5e:ff:ff:ff. The IANA allocates half of this block for multicast addresses. Given that the first byte of any Ethernet address must be 01 to specify a multicast address, the Ethernet addresses

corresponding to IP multicasting are in the range 01:00:5e:00:00:00 through 01:00:5e:7f:ff:ff.

Permanent Host Group	Multicast Address
Network Time Protocol	224.0.1.1
RIP-2	224.0.0.9
Silicon Graphics Dogfight application	224.0.1.2

Table: Example of Multicast Addresses for Permanent Host Groups

This allocation allows for 23 bits in the Ethernet address to correspond to the IP multicast group ID. The mapping places the low-order 23 bits of the multicast group ID into these 23 bits of the Ethernet address, as shown in fig Because the upper five bits of the multicast address are ignored in this mapping, the resulting address is not unique. Thirty-two different multicast group IDs map to each Ethernet address.

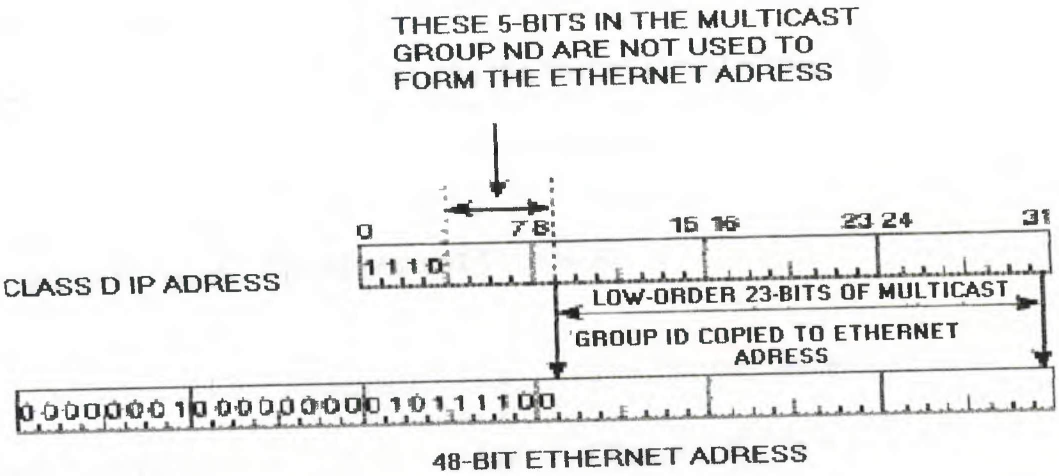

THESE 5-BITS IN THE MULTICAST GROUP ND ARE NOT USED TO FORM THE ETHERNET ADRESS
0
7 8
15 16
23 24
31
1 1 1 0
CLASS D IP ADRESS
LOW-ORDER 23-BITS OF MULTICAST GROUP ID COPIED TO ETHERNET ADRESS
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0
48-BIT ETHERNET ADRESS

Figure4.8-multicast address mapping

Because the mapping is not unique and because the interface card might receive multicast frames in which the host is really not interested, the device driver or IP modules must perform filtering.

Multicasting on a single physical network is simple. The sending process specifies a destination IP address that is a multicast address, and the device driver converts this to the corresponding Ethernet address and sends it. The receiving processes must notify their IP layers that they want to receive datagrams destined for a given multicast address and the device driver must somehow enable reception of these multicast frames. This process is handled by joining a multicast group.

When a multicast datagram is received by a host, it must deliver a copy to all the processes that belong to that group. This is different from UDP where a single process receives an incoming unicast UDP datagram. With multicast, multiple processes on a given host can belong to the same multicast group.

Complications arise when multicasting is extended beyond a single physical network and multicast packets pass through routers. A protocol is needed for routers to know if any hosts on a given physical network belong to a given multicast group. This function is handled by the Internet Group Management Protocol.

4.11-Internet Group Management Protocol:

The Internet Group Management Protocol (IGMP) is part of the IP layer and uses IP datagrams (consisting of a 20-byte IP header and an 8-byte IGRP message) to transmit information about multicast groups. IGMP messages are specified in the IP datagram with a protocol value of 2. Figure shows the format of the 8-byte IGMP message.

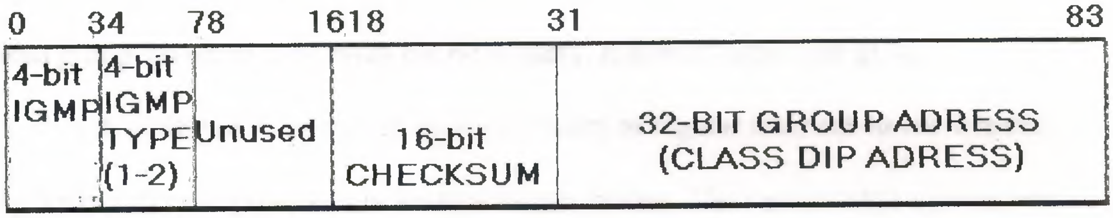


Figure4.9- IGMP message format.

The value of the *version* field is 1. The value of the *type* field is 1 for a query sent by a multicast router and 2 for a report sent by a host. The value of the *checksum* field is calculated in the same way as the ICMP checksum. The group address is a class D IP address. In a query, the group address is set to 0, and in a report, it contains the group address being reported.

The concept of a process joining a multicast group on a given host interface is fundamental to multicasting. Membership in a multicast group on a given interface is dynamic (that is, it changes over time as processes join and leave the group). This means that end users can dynamically join multicast groups based on the applications that they execute.

Multicast routers use IGMP messages to keep track of group membership on each of the networks that are physically attached to the router. The following rules apply:

- A host sends an IGMP report when the first process joins a group. The report is sent out the same interface on which the process joined the group. Note that if other processes on the same host join the same group, the host does *not* send another report.
- A host does not send a report when processes leave a group, even when the last process leaves a group. The host knows that there are no members in a given group, so when it receives the next query, it doesn't report the group.
- A multicast router sends an IGMP query at regular intervals to see whether any hosts still have processes belonging to any groups. The router sends a query out each interface. The group address in the query is 0 because the router expects one response from a host for every group that contains one or more members on a host.

• A host responds to an IGMP query by sending one IGMP report for each group that still contains at least one process.

Using queries and reports, a multicast router keeps a table of its interfaces that have one or more hosts in a multicast group. When the router receives a multicast datagram to forward, it forwards the datagram (using the corresponding multicast OSI Layer 2 address) on only those interfaces that still have hosts with processes belonging to that group.

The Time to Live (TTL) field in the IP header of reports and queries is set to 1. A multicast datagram with a TTL of 0 is restricted to the same host. By default, a multicast datagram with a TTL of 1 is restricted to the same subnet. Higher TTL field values can be forwarded by the router. By increasing the TTL, an application can perform an expanding ring search for a particular server. The first multicast datagram is sent with a TTL of 1. If no response is received, a TTL of 2 is tried, and then 3, and so on. In this way, the application locates the server that is closest in terms of hops.

The special range of addresses 224.0.0.0 through 224.0.0.255 is intended for applications that never need to multicast further than one hop. A multicast router should never forward a datagram with one of these addresses as the destination, regardless of the TTL.

4.12-Multicast Routing Protocols

A critical issue for delivering multicast traffic in a routed network is the choice of multicast routing protocol. Three multicast routing protocols have been defined for this purpose:

- Distance Vector Multicast Routing Protocol
- Multicast OSPF
- Protocol Independent Multicast

The goal in each protocol is to establish paths in the network so that multicast traffic can effectively reach all group members.

4.12.1- Distance Vector Multicast Routing Protocol:

Distance Vector Multicast Routing Protocol (DVMRP) uses a technique known as reverse path forwarding. When a router receives a packet, it floods the packet out all paths except the path that leads back to the packet's source. Reverse path forwarding allows a data stream to reach all LANs (possibly multiple times). If a router is attached to a set of LANs that does not want to receive a particular multicast group, the router sends a "prune" message up the distribution tree to prevent subsequent packets from traveling where there are no members.

New receivers are handled by using grafts. Consequently, only one round-trip time (RTT) from the new receiver to the nearest active branch of the tree is required for the new receiver to start getting traffic.

To determine which interface leads back to the source of the data stream, DVMRP implements its own unicast routing protocol. This unicast routing protocol is similar to RIP and is based on hop counts. As a result, the path that the multicast traffic follows might not be the same as the path that the unicast traffic follows.

The need to flood frequently means that DVMRP has trouble scaling. This limitation is exacerbated by the fact that early implementations of DVMRP did not implement pruning.

DVMRP has been used to build the MBONE---a multicast backbone across the public Internet---by building tunnels between DVMRP-capable machines. The MBONE is used widely in the research community to transmit the proceedings of various conferences and to permit desktop conferencing.

4.12.2-Multicast OSPF:

Multicast OSPF (MOSPF) is an extension of the OSPF unicast routing protocol and works only in internetworks that use OSPF. OSPF works by having each router in a network understand all of the available links in the network. Each OSPF router calculates routes from itself to all possible destinations. MOSPF works by including multicast information in OSPF link-state advertisements so that an MOSPF router learns which multicast groups are active on which LANs.

MOSPF builds a distribution tree for each source-group pair and computes a tree for active sources sending to the group. The tree state is cached and must be recomputed when a link state change occurs or when the cache times out.

MOSPF works well in environments that have relatively few source-group pairs active at any given time. It works less well in environments that have many active sources or in environments that have unstable links.

4.12.3-Protocol Independent Multicast

Unlike MOSPF, which is OSPF dependent, Protocol Independent Multicast (PIM) works with all existing unicast routing protocols. Unlike DVMRP, which has inherent scaling problems, PIM solves potential scalability problems by supporting two different types of multipoint traffic distribution patterns: dense mode and sparse mode. Dense mode is most useful when the following conditions occur:

- Senders and receivers are in close proximity to one another.
- There are few senders and many receivers.
- The volume of multicast traffic is high.
- The stream of multicast traffic is constant.

Dense-mode PIM uses reverse path forwarding and is similar to DVMRP. The most significant difference between DVMRP and dense-mode PIM is that PIM works with whatever unicast protocol is being used---it does not require any particular unicast protocol.

In dense mode, PIM floods the network and prunes back based on multicast group member information. Dense mode is effective, for example, in a LAN TV multicast environment because it is likely that there will be a group member on each subnet. Flooding the network is effective because little pruning is necessary. An example of PIM dense-mode operation is shown in Figure

Sparse-mode PIM is most useful when the following conditions occur:

- There are few receivers in a group.
- Senders and receivers are separated by WAN links.
- The stream of multicast traffic is intermittent.

Sparse-mode PIM is optimized for environments where there are many multipoint data streams. Each data stream goes to a relatively small number of the LANs in the internetwork. For these types of groups, reverse path forwarding would make inefficient use of the network bandwidth.

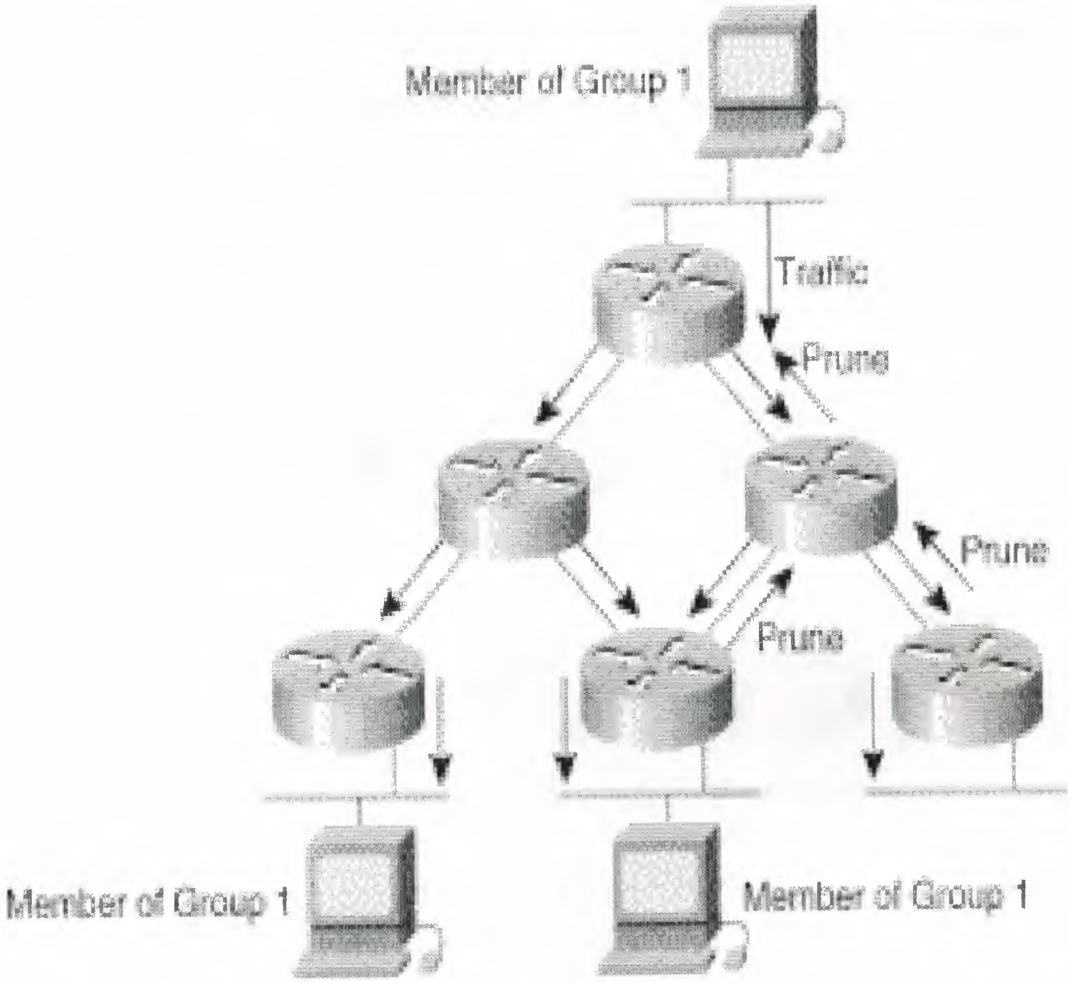


Figure4.10: PIM dense-mode operation.

In sparse-mode, PIM assumes that no hosts want the multicast traffic unless they specifically ask for it. It works by defining a rendezvous point (RP). The RP is used by senders to a multicast group to announce their existence and by receivers of multicast packets to learn about new senders. When a sender wants to send data, it first sends the data to the RP. When a receiver wants to receive data, it registers with the RP. Once the data stream begins to flow from sender to RP to receiver, the routers in the path automatically optimize the path to remove any unnecessary hops. An example of PIM sparse-mode operation is shown in Figure

The administrators of the MBONE plan to adopt PIM because it is more efficient than DVMRP.

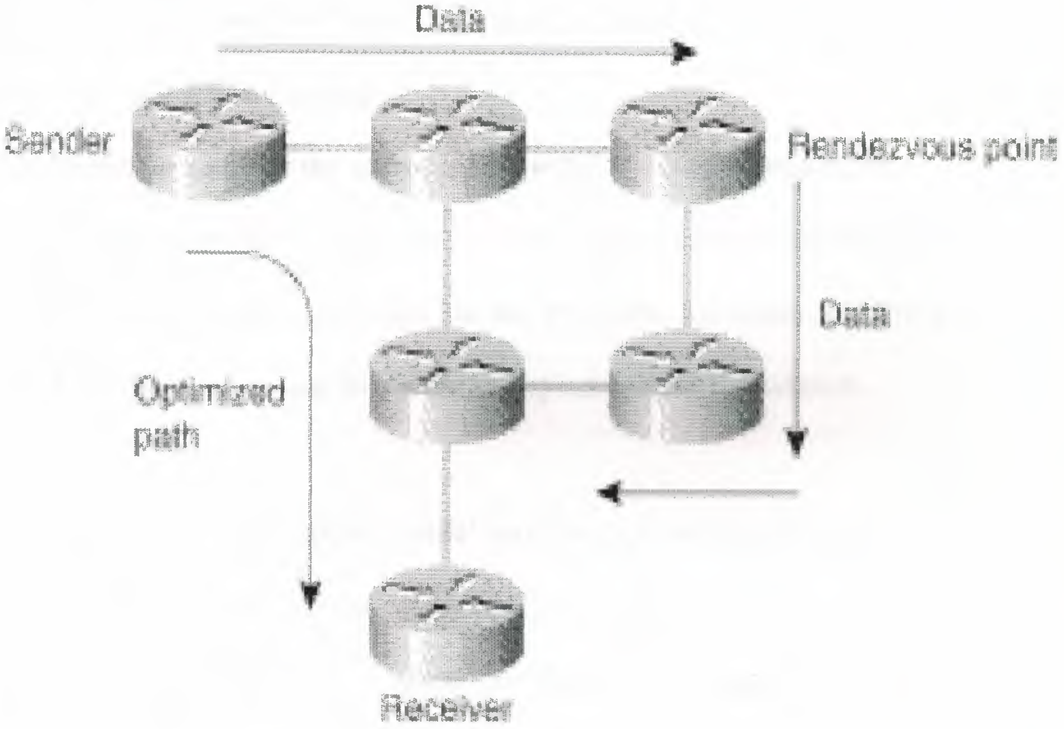


Figure4.11- PIM sparse-mode operation.

4.13- Simple Multicast Routing Protocol

Simple Multicast Routing Protocol (SMRP) is a transport layer multicast protocol standard for multicast AppleTalk and IPX traffic.

With SMRP, a router on each local network segment is elected as the primary node. The primary node handles requests from local devices to create ulticast groups on that segment. When it wants to send multicast data, a device sends a Create Group Request packet to ask the primary node to assign a group address. The primary node responds by sending to the requesting device a Create Group Response packet that contains the assigned group address.

Devices that want to receive multicast data from this group send a Join Request packet to ask their local router to join the group. The local router forwards the Join Request to the primary node that created the group. The primary node responds by sending a Join Response.

Multicast data sent by the source is forwarded by router downstream interfaces toward receivers. Receivers can join and leave a group at any time, and a sender can delete the group at any time. The routers ensure that multicast data is transmitted as efficiently as possible, without duplication, from senders to receivers.

Routers maintain and update SMRP multicast groups by periodically sending Creator Query and Member Query packets to poll the network for the presence of senders and receivers. A router that detects the disappearance of a sender deletes the group. A router that senses the disappearance of a receiver informs its upstream neighbor to stop forwarding multicast data if no other receivers exist on the segment. Each router periodically informs its neighbors of its presence by sending Hello packets.

Chapter five

Conclusion

Voice on internet service is now a small task. It requires signaling protocols, transport protocols, directory protocols, service specification languages, gateway discovery protocols, and a host of other mechanisms. This project, have provided an overview of some of the protocols being developed to solve these problems, and have demonstrated how they work together as building blocks to provide the infrastructure for telephone services in the Internet.

The reason behind Internet telephony's SUCCESS is that it can potentially bring enormous benefits to end users, Telcos, and carriers. There are several compelling reasons that carriers are interested in IP telephony, including:

- It is cheaper for end users to make an Internet telephony call than a circuit-switched call, mainly because operators can avoid paying interconnect charges.
- Internet telephony gives new operators an easy and cost-efficient way to compete with incumbent operators by undercutting their pricing regimes, while avoiding many of the regulatory barriers to standard voice provision.

- Engineering economics favors Internet telephony. While a circuit-switched telephony call takes up to 64 kb/s, an Internet telephony call only takes up to 6-8 kb/s and possibly even less bandwidth.

- In the longer term, it offers exciting new value-added opportunities such as high-fidelity stereo conferencing bridges, Internet multicast conferencing, and telephony distance learning applications, phone directories and screen popping via IP, or even "voice Web browsing," where the caller can interact with a Web page by speaking commands.

- Internet telephony gives carriers the ability to manage a single network handling both voice and data. Internet telephony will also create end user opportunities and demand for new services. VoIP aims to ultimately bring end user benefits in terms of communication management — effectively meaning that people will be able to control different media and different types of terminals: Global System for Mobile Communications (GSM), fixed phone, PC, and so on from their Web browsers. Users will be able to set up conference calls from their homes, route home calls to a GSM phone or centrex voice mail, look at the state of their accounts — even bar their children from accessing certain audiotex services. These are all services people will start to demand from their telephony service providers as the market matures. As a result, the switch to IP as the main delivery mechanism for telecom services in the future is looking increasingly promising.

Video data transfer is one of the major types of traffic over the Internet. However, it is impossible for the current Internet to guarantee the quality of service (QoS) for video transfer, and an influx of a large amount of video data into the Internet may cause serious network congestion. To resolve these problems, a protocol using congestion control based on rate control of video coding level and data transfer level, and evaluated this protocol using software simulation.. This project presents an improved specification of a video transfer protocol with two level rate control. This project also describes the results of implementation and evaluation of an actual video transfer system using improved protocol, and a commercial video tool with H.261 encoding. The results show protocol can transfer video data effectively and fairly .

The Internet world is moving fast and in the right direction.

REFERENCES

[1] W. Yeong, T. Howes, and S. Kille, "Lightweight Directory Access Protocol," RFC 1777, Internet Engineering Task Force, Mar. 1995; available at http://www.ietf.org/rfc/rfc1777.txt.

[2] C. Huitema et al., "An Architecture for Internet Telephony Service for Residential Customers," *IEEE Internet Computing* (co-published in *IEEE Network*), Vol. 3, No. 3, May-June 1999, pp. 73-82.

[3] P. Sijben et al., "Toward the PSTN/Internet Inter-networking MEDIA DEVICE CONTROL PROTOCOL," Internet draft, IETF, Feb. 1999; work in progress.

[4] H. Schulzrinne and J. Rosenberg, "Internet Telephony: Architecture and Protocols -- An IETF Perspective," *Computer Networks and ISDN Systems*, Vol. 31, Feb. 1999, pp. 237-255.

[5] R. Droms, "Dynamic Host Configuration Protocol," RFC 1541, IETF, Oct. 1993; available at http://www.ietf.org/rfc/rfc1541.txt.

[6] ITU-T Rec. H.323, "Visual Telephone Systems and Equipment for Local Area Networks which Provide a Non-guaranteed Quality of Service," Geneva, Switzerland, May 1996.

[7] M. Handley et al., "SIP: Session Initiation Protocol," RFC 2543, IETF, Mar. 1999; available at http://www.ietf.org/rfc/rfc2543.txt.

[8] M. Handley and V. Jacobson, "SDP: Session Description Protocol," RFC 2327, IETF, Apr. 1998; available at http://www.ietf.org/rfc/rfc2327.txt.

[9] H. Schulzrinne and J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony," *Proc. NOSSDAV*, Cambridge, U.K., July 1998.

[10] H. Schulzrinne and J. Rosenberg, "SIP Call Control Services," Internet draft, IETF, Feb. 1998; work in progress.

[11] R. Fielding et al., "Hypertext Transfer Protocol -- HTTP/1.1," RFC 2068, IETF, Jan. 1997; available at http://www.ietf.org/rfc/rfc2068.txt.

[12] H. Schulzrinne and J. Rosenberg, "Signaling for Internet Telephony," Tech. Report CUCS-005-98, Columbia Univ., New York, N.Y., Feb. 1998.

[13] H. Schulzrinne and J. Rosenberg, "Signaling for Internet Telephony," Int'l Conf. Network Protocols, Austin, Tex., Oct. 1998.

[14] H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, IETF, Jan. 1996; available at http://www.ietf.org/rfc/rfc1889.txt.

[15] H. Schulzrinne, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC 1890, IETF, Jan. 1996; available at http://www.ietf.org/rfc/rfc1890.txt.

[16] R. Ramjee et al., "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks," *Proc. IEEE INFOCOM*, Computer Society Press, Los Alamitos, Calif., June 1994, pp. 680-688.

[17] W.A. Montgomery, "Techniques for Packet Voice Synchronization," *IEEE JSAC*, Vol. SAC-1, Dec. 1983, pp. 1,022-1,028.

[18] S.B. Moon, J. Kurose, and D. Towsley, "Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms," *ACM/Springer Multimedia Systems*, Vol. 5, Jan. 1998, pp. 17-28.

[19] J.-C. Bolot and A.V. Garcia, "Control Mechanisms for Packet Audio in the Internet," *Proc. IEEE INFOCOM*, San Francisco, Calif., Mar. 1996.

[20] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS Control of Multimedia Applications Based on RTP," *Computer Comm.*, Vol. 19, Jan. 1996, pp. 49-58.

[21] C. Perkins and O. Hodson, "Options for Repair of Streaming Media," RFC 2354, IETF, June 1998; available at http://www.ietf.org/rfc/rfc2354.txt.

[22] J. Rosenberg and H. Schulzrinne, "Timer Reconsideration for Enhanced RTP Scalability," *Proc. IEEE INFOCOM,* San Francisco, Calif., Mar.-Apr. 1998.

[23] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326, IETF, Apr. 1998; available at http://www.ietf.org/rfc/rfc2326.txt.

[24] J. Lennox and H. Schulzrinne, "CPL: A Language for User Control of Internet Telephony Services," Internet draft, IETF, Mar. 1999; work in progress.

[25] J. Rosenberg, J. Lennox, and H. Schulzrinne, "Programming Internet Telephony Services," *IEEE Internet Computing* (co-published in *IEEE Network*), Vol. 3, No. 3, May-June 1999, pp. 63-72.

[26] J. Veizades et al., "Service Location Protocol," RFC 2165, IETF, June 1997; available at http://www.ietf.org/rfc/rfc2165.txt.

[27] J. Rosenberg and H. Schulzrinne, "A Framework for a Gateway Location Protocol," Internet draft, IETF, Feb. 1999; work in progress.

[28] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, IETF, Mar. 1995; available at http://www.ietf.org/rfc/rfc1771.txt.

[29] M. Squire, "A Gateway Location Protocol," Internet draft, IETF, Feb. 1999; work in progress.

[30] D. Hampton et al., "The IP Telephony Border Gateway Pprotocol Architecture," Internet draft, IETF, Feb. 1999, work in progress.

[31] L. Zhang et al., "RSVP: A New Resource Reservation Protocol," *IEEE Network*, Vol. 7, Sept. 1993, pp. 8-18.

[32] R. Braden et al., "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification," RFC 2205, IETF, Sept. 1997; available at http://www.ietf.org/rfc/rfc2205.txt.

[33] S. Blake et al., "An Architecture for Differentiated Service," RFC 2475, IETF, Dec. 1998; available at http://www.ietf.org/rfc/rfc2475.txt.