SUBMITTED BY

ANSHUL KUNDAJE
GEETALI BHATIA
MANGESH DALVI
MEGHNA HARIDAS
SUMERU NANDI

GUIDE

DR. (MRS.) NANDINI. K. JOG

ELECTRICAL ENGINEERING DEPARTMENT
V.J. TECHNOLOGICAL INSTITUTE
UNIVERSITY OF BOMBAY
2000-2001

VOICE OVER IP

A Project Report submitted to the University of Bombay (Mumbai) in partial fulfillment of the requirements for the degree of

Bachelor of Engineering (Electrical)

By

NAME	*ROLL.NO.*	*UNIV.NO.*	*SIGN.*
ANSHUL KUNDAJE	**GH-97729**		
GEETALI BHATIA	**GH-97705**		
MANGESH DALVI	**GH-97712**		
MEGHNA HARIDAS	**GH-97732**		
SUMERU NANDI	**GH-97736**		

Under the guidance of

Dr. (Mrs.) Nandini. K. Jog

Professor of Electronics
Department of Electrical Engineering
Veermata Jijabai Technological Institute

Department of Electrical Engineering
Veermata Jijabai Technological Institute
Matunga, Mumbai 400 019

VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE
DEPARTMENT OF ELECTRICAL ENGINEERING

CERTIFICATE OF APPROVAL

This is to certify that the dissertation titled " VOICE OVER IP*" is a bonafide record of dissertation work done by*

ANSHUL KUNDAJE	**GH-97729**
GEETALI BHATIA	**GH-97705**
MANGESH DALVI	**GH-97712**
MEGHNA HARIDAS	**GH-97732**
SUMERU NANDI	**GH-97736**

Under the guidance of **Dr. (Mrs.) N. K. Jog** *, Professor of Electronics, Department of Electrical Engineering; in partial fulfillment of the requirement for the degree of Bachelor of Engineering (Electrical) of the University of Bombay (Mumbai)*

GUIDE PRINCIPAL

Dr. (Mrs.) N. K. Jog **Dr. S. D. Varwandkar**

CERTIFICATE

This is to certify that the project titled “ VOICE OVER IP*” is a bonafide record of the project work done by*

ANSHUL KUNDAJE	**GH-97729**
GEETALI BHATIA	**GH-97705**
MANGESH DALVI	**GH-97712**
MEGHNA HARIDAS	**GH-97732**
SUMERU NANDI	**GH-97736**

During the Academic Year 2000-2001 under the guidance of
Dr. (Mrs.) N. K. Jog
Professor of Electronics, Department of Electrical Engineering,
V.J.T.I., University of Bombay (Mumbai)

Dr. B. K. Lande	**Dr. (Mrs.) N. K. Jog**
(Head of Electrical Engg. Dept.)	**(Professor and Guide)**

CERTIFICATE

This is to certify that the project titled " VOICE OVER IP *" is a bonafide record of the project work done by*

ANSHUL KUNDAJE	**GH-97729**
GEETALI BHATIA	**GH-97705**
MANGESH DALVI	**GH-97712**
MEGHNA HARIDAS	**GH-97732**
SUMERU NANDI	**GH-97736**

During the Academic Year 2000-2001 and they have satisfactorily completed the requirements of **PROJECT-II** *as prescribed by the University of Bombay (Mumbai)*

Examiner
(Internal)

Examiner
(External)

ACKNOWLEDGEMENTS

We take this opportunity to offer our gratitude to our guide ***Dr. (Mrs.) N. K. Jog*** *for her support and guidance. She allowed us a great deal of freedom in choosing our topics for study and also provided us encouragement throughout this venture.*

We also wish to thank ***Mr. Suresh Vishwanathan*** *and* ***Mr. Rajesh Jain*** *of SITA (Societe Internationale de Telecommunications Aeronautiques) who took time off from their busy schedules to help us with the project. They allowed us full access to many facilities without which our project would have not been possible.*

Last but not the least we are grateful to our parents for all their support and encouragement.

ANSHUL KUNDAJE
GEETALI BHATIA
MANGESH DALVI
MEGHNA HARIDAS
SUMERU NANDI

ABSTRACT

Ever tried placing a voice call over the Internet. If you have, we are sure you haven't had a pleasant experience. You might have even promised yourself never to try it again.
Stop right there!!
Take some time-off from your busy schedules and have a look at what we have to say.
We guarantee that you will change your mind.

In the near future, if you make a telephone call, it is more than likely that it would be over the Internet or some other packet network. But, what is it that would make this possible? It is a bunch of protocols and standards; and years of research done by organizations all over the world that would bring about this revolution.
They call it 'VOICE OVER IP', 'INTERNET TELEPHONY' & a host of other names.
We like to refer to it as 'A Dawn Of a New Era in Telecommunications'.

The next few chapters of this project report will discuss this phenomenon in detail.

CONTENTS

SECTION I

SECTION II

SECTION III

SUPPLEMENT

REFERENCES

1. PURPOSE AND BACKGROUND:

1.a. OUTLINE OF THE REPORT:

Until today data networks and voice (telephony) networks have always treaded their separate ways. With the dawn of the 21st century they seem to have reached the crossroad. We have entered into an age of 'CONVERGENCE'. The border separating the two has narrowed to a thin red line. Thin red line- because the integration must be done carefully and intelligently. 'VoIP' is a tool that would greatly hasten this process.

This thesis is logically divided into 3 sections.
Section 1 discusses the following points.

- The history of telephony
- The traditional telephony System
- What is VoIP?
- The key applications of VoIP
- VoIP vs. other parallel technologies
- The key issues involved
- The advantages and risk factors involved

Section 2 discusses the following points

- Framework for Internet Telephony
- The key architectures and protocol families
- The supporting technologies

Section 3 discusses the following points

- IP phone design
- The CISCO IOS (Internetwork Operating System)
- Practical Configuration of CISCO 3640 routers for VoIP

Finally we end with a list of vendors, products, services & references

1.b. PURPOSE OF THE REPORT:

This project report is part of the curriculum for the final year of the Electrical Engineering Bachelor's program as prescribed by Bombay (Mumbai) University. The project has involved one and a half year of reading, understanding and thinking. The seventh semester involved a theoretical study. In the eight semester, we used the 'CISCO Voice over IP Network Simulator' to study practical router configuration. We visited the SITA office at Mumbai to have a hands-on-experience with the hardware. We have found the overall experience very enlightening and interesting.

SECTION I

2. INTRODUCTION:

The global communications transformation is in full swing. Packet-switched technology has moved from data-only applications into the heart of the network to take up the functions of traditional circuit-switched equipment. While the lower cost of packet-switched networks initially drove this change, the improving quality and reliability of voice over these networks is speeding integration of voice and data services. Consequently, the overwhelming majority of voice networks in service today will be replaced by packet infrastructure within the next decade. Service providers and corporate organisations, therefore, must develop a plan to migrate their voice services from circuit-switched networks to packet-switched networks to ensure their future success---and survival.

2.a. WHAT IS VOICE OVER IP (VoIP)?

Voice over IP (VOIP) uses the Internet Protocol (IP) to transmit voice as packets over an IP network. So VOIP can be achieved on any data network that uses IP, like Internet, Intranets and Local Area Networks (LAN). Here the voice signal is digitized, compressed and converted to IP packets and then transmitted over the IP network. Signaling protocols are used to set up and tear down calls, carry information required to locate users and negotiate capabilities. One of the main motivations for Internet telephony is the very low cost involved.

There are four different types of connections for setting up the call. In all of the cases of VoIP, the Internet Protocol (IP) is used. This means that the service is best effort i.e. the application handles the end-to-end communication without any guarantees from the net. The four different types are

1. PC to PC
2. PC to Telephone
3. Telephone to PC
4. Telephone to Telephone

    4.1 Regular phones connected to PSTN

    4.2 IP-telephones connected to a data net

Most of the focus on VoIP is currently centered on two key applications.

The first is *private business network applications*. Businesses with remotely located branch offices which are already connected together via a corporate intranet for data services can take advantage of the existing intranet by adding voice and fax services using VoIP technologies. Businesses are driving the demand for VoIP solutions primarily because of the incredible cost savings that can be realized by reducing the operating costs of managing one network for both voice and data and by avoiding access charges and settlement fees, which are particularly expensive for corporations with multi-international sites. Managed corporate intranets do not have the QOS issues which currently plague the Internet; thus voice quality approaches toll quality.

The second key application is *VoIP over public networks*. This application involves the use of voice gateway devices designed to carry voice to Internet Service Providers, now known as *Internet Telephony Service Providers*, or to the emerging Next Generation Carriers which are developing IP networks specifically to carry multimedia traffic such as VoIP. ISPs are interested in VoIP as a way of offering new value-added services to increase their revenue stream and break out of the low monthly fixed fee structure currently in place for data services. VoIP also allows them to improve their network utilization. These new services include voice and fax on a per-minute usage basis at rates significantly less than prevailing voice and fax rates for service through the PSTN. The sustainability of this price advantage may be short term, and is dependent on whether the FCC and foreign regulatory agencies will require ISPs to pay the same access charges and settlement fees PSTN carriers are obligated to pay. In the long term, IP networks are more efficient for a wide range of new applications, particularly multimedia applications enabling convergence of voice, video, data, and fax. *Carriers*, too, are interested in VoIP, primarily for competitive reasons. Although VoIP will cannibalize some of their POTS services, they have wisely determined that they too must compete in this rapidly growing marketplace. The market projections are too staggering to be ignored: according to a survey, 10% of the world's fax market could be on the internet in 2 - 3 years, and by 2002, the Internet could carry 11% of US and international long distance traffic. [*Telogy Networks*]

In this project report we discuss VoIP in both contexts.

We now discuss the switching technology that differentiates Voice over an IP network from the traditional circuit switched technology.

3. CIRCUIT SWITCHING vs. PACKET SWITCHING:

For transmission of data beyond a local area, communication is typically achieved by transmitting data from source to destination through a network of intermediate switching nodes; this switched-network design is sometimes used to implement LANs and MANs as well. The switching nodes are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination. The end devices that wish to communicate may be referred to as stations. The stations may be computers, terminals, telephones, or other communicating devices. Each station attaches to a node, and the collection of nodes is referred to as a communications network.

The types of networks that are discussed in this are referred to as switched communication networks. Data entering the network from a station are routed to the destination by being switched from node to node. Several observations are in order:

1. Some nodes connect only to other nodes. Their sole task is the internal (to the network) switching of data. Other nodes have one or more stations attached as well; in addition to their switching functions, such nodes accept data from and deliver data to the attached stations.
2. Node-node links are usually multiplexed, using either *frequency-division multiplexing* or *time-division multiplexing*.
3. Usually, the network is not fully connected; that is, there is not a direct link between every possible pair of nodes. However, it is always desirable to have more than one possible path through the network for each pair of stations; this enhances the reliability of the network.

Two quite different technologies are used in wide-area switched networks: circuit switching and packet switching. These two technologies differ in the way the nodes switch information from one link to another on the way from source to destination.

3.a. CIRCUIT-SWITCHING NETWORKS

Communication via circuit switching implies that there is a dedicated communication path between two stations. That path is a connected sequence of links between network nodes. On each physical link, a logical channel is dedicated to the connection. Communication via circuit switching involves three phases

1. **Circuit establishment**: Before any signals can be transmitted, an end-to-end station- to-station) circuit must be established. In completing the connection, a test is made to determine if the receiver is busy or is prepared to accept the connection.
2. **Data transfer:** Information can now be transmitted from the sender through the network to the receiver. The data may be analog or digital, depending on the nature of the network. As the carriers evolve to fully integrated digital networks, the use of digital (binary) transmission for both voice and data is becoming the dominant method. Generally, the connection is full-duplex.

3. **Circuit disconnect:** After some period of data transfer, the connection is terminated, usually by the action of one of the two stations. Signals must be propagated to the respective nodes to deallocate the dedicated resources.

The connection path is established before data transmission begins. Thus, channel capacity must be reserved between each pair of nodes in the path, and each node must have available internal switching capacity to handle the requested connection. The switches must have the intelligence to make these allocations and to device a route through the network.

Circuit switching can be rather inefficient. Channel capacity is dedicated for the duration of a connection, even if no data are being transferred. For a voice connection, utilization may be rather high, but it still doesn't approach 100 percent. For a terminal-to-computer connection, the capacity may be idle during most of the time of the connection. In terms of performance, there is a delay prior to signal transfer for call establishment. However, once the circuit is established, the network is effectively transparent to the users. Information is transmitted at a *fixed data rate* with no delay other than that required for propagation through the transmission links. The delay at each node is negligible.

Circuit switching was developed to handle voice traffic but is now also used for data traffic. The best-known example of a circuit-switching network is the public telephone network. *One of the key requirements for voice traffic is that there must be virtually no transmission delay and certainly no variation in delay*. A constant signal transmission rate must be maintained, as transmission and reception occur at the same signal rate. These requirements are necessary to allow normal human conversation. Further, the quality of the received signal must be sufficiently high to provide, at a minimum, intelligibility.

Circuit switching achieved its widespread, dominant position because it is well suited to the analog transmission of voice signals; in today's digital world, its inefficiencies are more apparent. However, despite the inefficiency, circuit switching will remain an attractive choice for both local-area and wide-area networking. One of its key strengths is that it is transparent. Once a circuit is established, it appears as a direct connection to the two attached stations; no special networking logic is needed at either point.

3.b. PACKET SWITCHING PRINCIPLES:

Data are transmitted in short packets. A typical upper bound on packet length is 1000 octets (bytes). If a source has a longer message to send, the message is broken up into a series of packets. Each packet contains a portion (or all for a short message) of the user's data plus some control information. The control information, at a minimum, includes the information that the network requires in order to be able to route the packet through the network and deliver it to the intended destination. At each node en route, the packet is received, stored briefly, and passed on to the next node. A question arises as to how the network will handle this stream of packets as it attempts to route them through the network and deliver them to the intended destination; there are two approaches that are used in contemporary networks: datagram and virtual circuit.

- In the *datagram* approach, each packet is treated independently, with no reference to packets that have gone before. So the packets, each with the same destination address, do not all follow the same route. As a result, it is possible that ‘packet 2’ will beat ‘packet 1’ to a node. Thus, it is also possible that the packets will be delivered to the destination in a different sequence from the one in which they were sent. It is up to the destination to figure out how to reorder them. Also, it is possible for a packet to be destroyed in the network. For example, if a packet-switching node crashes momentarily, all of its queued packets may be lost. In this technique, each packet, treated independently, is referred to as a datagram.
- In the *virtual-circuit* approach, a preplanned route is established before any packets are sent. The sender node (S) first sends a special control packet, referred to as a Call-Request packet, requesting a logical connection to the destination node (D). If D is prepared to accept the connection, it sends back a Call-Accept packet. The two stations may now exchange data over the route that has been established. Because the route is fixed for the duration of the logical connection, it is somewhat similar to a circuit in a circuit-switching network, and is referred to as a virtual circuit. Each packet now contains a virtual-circuit identifier as well as data. Each node on the preestablished route knows where to direct such packets; no routing decisions are required. Eventually, one of the stations terminates the connection with a Clear-Request packet. At any time, each station can have more than one virtual circuit to any other station and can have virtual circuits to more than one station. So, the main characteristic of the virtual circuit technique is that a route between stations is set up prior to data transfer. *Note that this does not mean that this is a dedicated path, as in circuit-switching*. A packet is still buffered at each node, and queued for output over a line. The difference from the datagram approach is that, with virtual circuits, the node need not make a routing decision for each packet; it is made only once for all packets using that virtual circuit.

If two stations wish to exchange data over an extended period of time, there are certain advantages to virtual circuits. First, the network may provide services related to the virtual circuit, including sequencing and error control. Another advantage is that packets should transit the network more rapidly with a virtual circuit; it is not necessary to make a routing decision for each packet at each node.

One advantage of the datagram approach is that the call setup phase is avoided. Thus, if a station wishes to send only one or a few packets, datagram delivery will be quicker. Another advantage of the datagram service is that, because it is more primitive, it is more flexible. For example, if congestion develops in one part of the network, incoming datagrams can be routed away from the congestion. With the use of virtual circuits, packets follow a predefined route, and it is thus more difficult for the network to adapt to congestion.

A third advantage is that datagram delivery is inherently more reliable. With the use of virtual circuits, if a node fails, all virtual circuits that pass through that node are lost. With datagram delivery, if a node fails, subsequent packets may find an alternate route that bypasses that node. Voice packets generally employ the datagram service.

3.c. CIRCUIT SWITCHING vs. PACKET SWITCHING:

Having looked at the operation of packet-switching, we now return to a comparison of this technique with circuit-switching.

Circuit switching	***Datagram packet switching***	***Virtual-circuit packet switching***
Dedicated transmission path	No dedicated path	No dedicated path
Continuous transmission of data	Transmission of packets	Transmission of packets
Fast enough for Interactive	Fast enough for Interactive	Fast enough for Interactive
Messages are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire connection	Route established for each packet	Route established for entire conversation
Call setup delay; negligible transmission delay	Packet transmission delay	Call setup delay; packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender informed of connection denial
Overload may block call setup; no delay for established calls	Overload increases packet delay	Overload may block call setup; increases packet delay
Electromechanical or computerized switching nodes	Small switching nodes	Small switching nodes
User responsible for message loss protection	Network may be responsible for individual packets	Network may be responsible for packet sequences
Usually no speed or code conversion	Speed & code conversion	Speed & code conversion
Fixed bandwidth transmission	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each message	Overhead bits in each packet
Charged per minute	Charged per packet	Charged per packet

Most of the voice traffic today is transported over the PSTN, which is a circuit switched network in analog as well as digital form. However, there is a gradual shift towards the packet-switched networks i.e. networks based on label switching and routing; such as IP, frame relay and ATM networks. We now discuss the traditional telephony system in detail.

4. HISTORY OF TELEPHONY:

This module explains the general history of telephony. The telephone network, as we know it today, including many of the terms and acronyms and much of the network architecture and existing paradigms, has a foundation in the early days of the telephone and the first telephone systems.

4.a. VOICE COMMUNICATIONS:

When humans talk, we expel air from the lungs and move the lips, tongue, and larynx to generate sound waves. The sound waves traverse the air and reach the inner ear of the other party, stimulating the sense of hearing. There is a limitation, however, in the distance and energy level of the sound wave. The distance at which a sound can be heard depends upon the intensity (decibels) and volume (amplitude) of the sound. A walkie-talkie, a microphone, and a residential telephone are all examples of devices that can be used to carry and amplify the sound so it can be heard over a distance. Basically, these and other electronic devices convert voice activity into electrical current or voltage, helping to overcome the problems associated with the nature of the sound wave. The most common example of a telecommunications device is your telephone at home. When you plug an ordinary analog phone into the phone jack installed by the telephone service provider, you are then able to place phone calls. (Note that the voice still could be digitized at some point along the transmission path from source to destination.)

4.b. THE FIRST TELEPHONE NETWORK:

The basic residential analog phone system that exists today is a direct descendent of the company started by the inventor of the telephone, Alexander Graham Bell.

Fig1: Sir Alexander Graham Bell

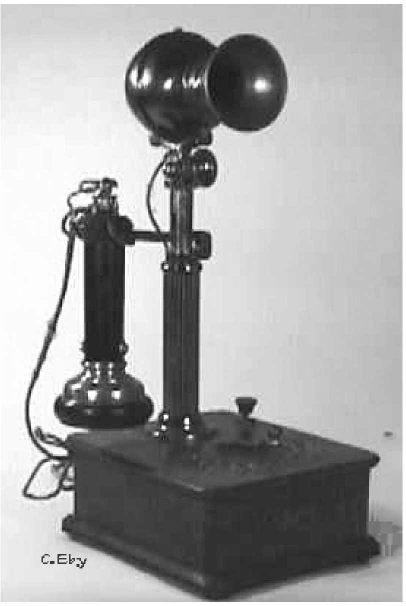


Fig2:The first candlestick telephone

In the late nineteenth century if one wanted phone service, one would have to contact the Bell Company to request service and tell the company exactly whom you wanted to be able to call. The phone company would then install wiring between your home or business and that of the party you wanted to call. You, as the customer, would have to provide the telephone itself (it was leased from the Bell company), along with the wiring; you also would be responsible for all installation and maintenance costs. There was no real amplification, so a call would have traveled only as far as the physical wire would carry it.

Within about two years after its founding, the Bell Company realized that it needed some kind of a switching system to be able to service a greater number of customers. The early switches were literally cord boards, which would alert an operator at a central office (CO) to an incoming call, typically by ringing a bell or lighting some type of lamp at the operator's station. This system was user friendly and highly intelligent, but the system did not offer very good performance. Even a fast operator worked at the pace of a human and was capable of handling only one call at a time.

4.c. TELEPHONE SERVICE TODAY:

Today private branch exchanges (PBXs) provide the service that the operator performed in the early telephone company. *A PBX is a small telephone switch owned by a company or organization.* Without a PBX, a company would need to lease one telephone line for every employee who has a telephone. With a PBX, the company only needs to lease as many lines from the telephone company as the maximum number of employees that will be making outside calls at one time. This is usually around 10% of the number of extensions. A telephone switching system must perform the following functions:

- Recognize a request for service.
- Notify the station of an incoming call.
- Detect on-hook or off-hook status.
- Provide status information to the originator of the call, such as when the called telephone goes off hook, or the network is busy.
- Establish a connection—really just a path across the network from one endpoint to another.

Switches also act as concentrators because the number of telephones in use is usually greater than the number of simultaneous calls that can be made. E.g. a company may have 600 telephone sets connected to a PBX, but may have only 15 trunks connecting the PBX to the CO switch.

CO versus PBX

A PBX is like a miniature CO switching system designed for a business or a private institution. Although a PBX and a CO are closely related, there are differences between them.

- A PBX is intended for private operation within a company. A CO is intended for public service.
- A PBX usually has some type of arrangement to greet outside callers and connect them to internal extensions.
- Most PBXs do not maintain the high level of service protection that must be maintained in a CO. Assurance features such as processor redundancy (in the event of processor failure) and battery backup power, which are standard in a CO, may not be a part of a PBX.
- COs require a seven-digit local telephone number, while PBXs can be more flexible and create dialing plans to best serve their users.
- A PBX can restrict individual stations or groups of stations from certain features and services, such as access to outside lines. A CO usually has no interest in restricting usage because these features and services are billed to the customer. COs normally provide unlimited access to every member on the network.

LOOPS; LINES; TRUNKS; PBX TRUNKS:

The key components of a telecommunications network are loops, lines, trunks, and PBX trunks.

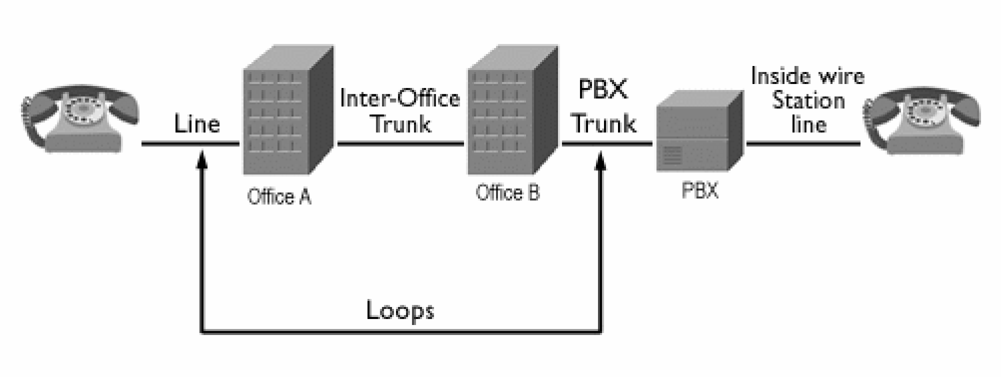


Fig3: Loops, Lines & Trunks

LOOPS: A *loop* is literally the physical pair of wires that connects a telephone directly to a CO's switch or a PBX. It consists of a two- or four-wire pair that is twisted to minimize the electromagnetic radiation created by the current flowing through the wire. When a phone handset is off hook, the current flow is detected by a current detector in the CO. When a phone is on hook, the loop is open and no current flows. A *local loop* is the connection between your house and the phone company.

LINES: The terms *line* and *loop* often get confused. A *line* is a communications path between a customer's telephone and a telephone switch, such as a PBX or a CO switch. Although a line is typically based on a physical loop, it is not necessarily a physical connection. A line could be a logical connection, such as a channel on a multiplex system. A *leased line* is a dedicated line reserved by a carrier for the private use of a leasing customer. A *tie line* is a private leased line that directly links two telephones or two PBXs.

TRUNKS: Whereas a line connects the telephone with the switch, a *trunk* is a shared communications channel that connects switches and has the capacity to carry a phone call. A trunk is assigned to connections on a call-by-call basis. Over time, the trunk is shared by all the users who call between this pair of switches. In the early days of telephone networks, a trunk was actually a pair of wires that carried only one telephone call at a time. An example of a trunk is a tie line that connects two PBXs.
Common types of trunks are as follows:

- Private trunk line—a line connecting a PBX to another PBX
- CO trunk—a direct connection between a CO and a PBX or another CO
- Foreign Exchange trunk—a trunk interface used to directly connect a remote phone to a PBX via *FXS/FXO* interfaces
- Direct Inward Dial/Direct Outward Dial (DID/DOD) trunk—a one-way trunk allowing a user to dial into a PBX (DID) or the CO (DOD) without operator intervention

PBX TRUNKS: In a business using a PBX, every employee's telephone line is connected to the PBX. When an employee picks up the receiver and dials the outside access code (typically 9), the PBX connects the employee to an outside line—the Public Switched Telephone Network, or PSTN. PBXs combine with trunks between them. *A PBX trunk* is a shared communications path specifically between the customer's switch and the edge of the PSTN. PBX trunks use out-of-band signaling as opposed to in-band signaling.

TRANSMISSION MEDIA:

Media	User to Network	Network to Network	Speed
Twisted Pair	Analog voice ISDN T1 / E1 Digital subscriber line (xDSL)	T1 E1	1.544 Mbps / 2.048 Mbps
Coaxial Cable	Cable TV	T3, T4 E3	44.736 / 34.368 Mbps
Radio	Cellular WLL, LMDS, and MMDS	T3, T4 E3	44.73 Mbps 34.368 Mbps
Fiber	SONET Cable Television	SONET	2.5 Gbps

4.d. TELEPHONY BASICS:

HOW DOES A TELEPHONE WORK?

A telephone typically consists of the following components:

- Handset containing a transmitter and receiver
- Switch hook, which is a lever that is depressed when the handset is resting in its cradle
- Two-wire to four-wire converter to provide conversion between the four-wire handset and the two-wire local-loop
- Dialer (either rotary or touch-tone)
- Ringer

A typical telephone does the following:

- Requests service from the network
- Performs dialing functions
- Performs a notification function (it rings)
- Provides answer and disconnect supervision
- Converts outgoing speech to electrical signals, and vice versa
- Automatically adjusts to the supplied power

TELEPHONY SIGNALING:

For a telephone call to be completed, several forms of signaling must occur:

- Access signaling
- Station loop signaling
- Address signaling

The purpose of signaling in a voice network is to establish a connection. You typically begin a phone call by taking a phone off hook, which sends an access signal. The line is seized, a path established across the network, and on the other end, the call is acknowledged.

ACCESS SIGNALING: determines when a line is off hook or on hook. When the handset is on its cradle, the phone is referred to as being *on hook*. When a telephone is on hook, the two wires do not touch, so the circuit (loop) is open and no current flows. When the handset is out of its cradle, it goes *off hook*. The wires touch, closing the loop, allowing current to flow through the two conductors that connect the phone to the network, sending an ‘off-hook’ signal to the switch. To place a call, the phone must be off hook. To receive a call it must be on hook.

There are two common methods of providing the basic access signal: *loop start* and *ground start*.

- **Loop start.** This is the most common technique for access signaling in a standard Public Switched Telephone Network (PSTN). Most residential telephones are analog loop-start telephones. The loop is an electrical communications path consisting of two wires, one for transmitting and one for receiving voice signals. The two-wire circuit is still referred to as the *tip and ring*, with the tip being tied to the ground and the ring tied to the negative side of the battery. When the phone handset is picked up, this action closes the circuit, establishing a loop between the PBX and the phone. Current is drawn from the battery of the PBX, indicating a change in status. This change in status signals the current detector in the PBX to provide dial tone. An incoming call is signaled to the handset by a standard on/off pattern, which causes the telephone to ring.
- **Ground start.** Ground start is another access signaling method used on trunk lines or tie lines between PBXs to indicate on-hook/off-hook status to the CO. In ground-start signaling, one side of the two-wire trunk (typically the *ring* in the tip and ring configuration) is momentarily grounded to create dial tone. When a user tries to place a call by grounding the ring lead, the PBX at the telephone company detects the flow of current and grounds the tip lead to indicate the PBX is ready. The user's telephone perceives the flow of current on its "tip" and knows that the PBX is ready. The seizure of the line requires the cooperation of both parties to the call. A failure on either side stops the progress of the call. Therefore, both parties terminate service upon disconnecting. This setup averts the disconnect supervision problem that might occur on the loop-start circuit, when a given line can be released only by the party who originated the phone call. For this reason, PBXs work best on ground-start trunks.

In a normal loop-start circuit, when you pick up the handset, you hear a dial tone indicating that a circuit is ready. On a ground-start circuit, however, the equipment at the user's end should sense the flow of electrical current on the "tip" lead and interpret that the PBX is ready, so a dial tone from a PBX is not necessary, and its presence is optional. This setup allows the network to indicate off-hook status, or seizure of an incoming call independent of the ringing signal.

STATION LOOP SIGNALING: When the PBX receives the off-hook signal, it responds with an audible signal indicating that it is ready for a call—the *dial tone*. This two-way exchange between the PBX and the telephone is known as *station loop signaling*.

ADDRESS SIGNALING: In response to the audible prompt of the dial tone, the caller can request connection to another telephone by transmitting the address (telephone number) of the requested telephone (sometimes referred to as the called party identification number) to the PBX. This is known as *address signaling*.

Telephones generally use two basic types of address signaling: pulse and tone.

- **Pulse dial** (rotary dialing): Rotary dial phones represent the digit being dialed by momentarily stopping the current flow when the user turns the circular dial. For example, the circuit is broken three times, which creates three pulses in the current, to dial the digit "3."

- **Tone dial** (dual tone multifrequency, or DTMF; the method used by pushbutton telephones). DTMF is the most commonly used signaling system today. The keypad on a pushbutton phone has 12 keys. Each key press generates both a low frequency and a high-frequency tone (the *Dual Tone*) that is specific to each individual key. The tones are then picked up and interpreted by telephone switches. The tones were selected to easily pass through the phone network with minimum interaction with each other and little attenuation.

CALL PROGRESS INDICATORS: While you are placing a call, you also hear a variety of audible signals that indicate the status of the call at various points along its path—for example, the dial tone, a busy signal, a signal indicating no circuit is available (fast busy), and ringing of the called party's phone. These tones are called *call progress indicators*.

ANSWER SUPERVISION SIGNALING: Assuming the call can be established, signaling would then occur at the remote end of the network. The CO seizes a line to the PBX and forwards the digits. The PBX selects the appropriate station, and signals an alert. The call proceeds, and the switch generates *ring voltage* to the phone. When the phone detects the voltage, it rings. The caller also hears an audible ring through the receiver; this signal is the ringback signal that is generated by the switch.

FXO & FXS SIGNALING: A *foreign exchange (FX)* is a term applied to a trunk that has access to a distant CO. FX trunk signaling can be provided over either analog or T1 links and which utilize either loop-start or ground-start off-hook signaling techniques.

- **Foreign eXchange Station (FXS):** Standard residential phone lines are configured for FXS signaling. An FXS interface can be used to connect basic devices such as phones, modems, and faxes and must provide voltage, ring generation, off-hook detection, and call progress indicators.
- **Foreign eXchange Office (FXO):** FXO signaling is used primarily to communicate with CO switching equipment or PBXs. Because an FXO port on a router communicates directly with the PSTN or a PBX, it requires that a dialtone, ring indication, and call progress indicators be provided to it.

E&M SIGNALING: Another analog signaling technique, used mainly between PBXs or other network-to-network telephony switches, is known as E&M, which stands for "ear and mouth" (or for "recEive and transMit"). There are five E&M signaling types, as well as two different wiring methods.

We now discuss the paradigm shift toward packet telephony.

5. VOICE OVER PACKET NETWORKS:

IP, ATM and frame relay networks are considered candidates for a backbone supporting integrated voice and data applications. However, the first two are seen to be the hot favourites for wide area networks and internetworks. Frame relay is a better candidate for smaller networks. First we describe the technologies in brief. Then we compare voice over ATM and IP and state why we feel the latter is the preferred technology.

5.a FRAME RELAY:

Frame relay is a protocol standard for LAN internetworking which provides a fast and efficient means of transmitting information from a user device to the LAM bridges and routers. It is a service for people who want an absolute bare-bones connection-oriented way to move bits at a reasonable speed and low cost. Its existence is due to changes in technology over the years. It is a direct evolution of the traditional packet-switching technology, which used complex protocols and a great deal of overhead for error detection, correction and flow control, as the user terminals were not able to do so themselves. The situation has changed radically. Leased lines and now fast, digital and considerably reliable. The terminals now boast of superior processing power, capabilities at much lower costs. This suggests the use of simple protocols for data link, with most of the work done by the terminals rather than the network.

Frame relay could be understood as a basic virtual leased line. The customer leases a PVC (permanent virtual circuit) between two points and can then send frames of upto 1600 bytes each between them. It is also possible to lease PVCs between a given site and multiple other sites. Each frame carries a 10-bit address called the DLCI (Data Link Connection Identifier), which identifies the virtual circuit to be used. The difference between frame relay service and a permanent leased line is that the latter permits the customer to send data all day long at maximum data rate. For the frame relay virtual line, data bursts may be sent at the maximum rate with the long-term average data rate being within a permissible value. That is exactly the reason why frame relay is cheaper than leased lines. The standard frame relay speed is around 1.5 Mbps.

Frame relay provides a minimum service. It is primarily a way to determine the start and end of a frame; source and destination address and detect errors. It does not provide flow control or acknowledgement service. If a frame is corrupted it is discarded. How the data is recovered is up to the higher service layers at the end terminals. This is exactly the kind of service real-time traffic, such as voice, needs. However, the success of frame relay is incumbent on an inherent reliability of the transport network.

As can be deduced for an Internetwork, frame relay service proves inadequate for voice transport. However, it is ideal for LANs (Local Area Networks) and medium sized MANs (Metropolitan Area Networks).

5.b. ASYNCHRONOUS TRANSFER MODE (ATM):

The phrase 'Voice over ATM' has two aspects to it. We can either transport voice packets (cells) directly over the ATM architecture or tunneling IP packets over base ATM transport layers. The latter option is out of the question since it goes leads to inefficiency due to excess protocol overhead. This goes against the very reason why ATM was developed: An efficient, integrating, multiservice architecture. Unlike frame relay or IP, ATM is not just a protocol. It is not confined to a single layer in some architecture. It is an architecture itself. It is part of the envisioned B-ISDN (Broadband ISDN).

ATM is so called because it is not synchronous i.e. tied to a master clock. The basic idea behind ATM is to transmit all information in small, fixed size packets called *cells* which are 53 bytes long: 5 bytes of header and 48 bytes of payload, in our case voice. ATM service is also called cell relay.

Cell switching is highly flexible and can handle both constant rate traffic (audio, video) and variable rate traffic (data) easily. Second, at the very high speeds envisioned digital switching of cells is easier than using traditional multiplexing techniques, especially using fibre optics. ATM networks are *connection-oriented.* Making a call requires first, sending a message to setup the connection. After that, subsequent cells follow the same path to the destination. Similar to frame relay, cell delivery is not guaranteed but their order is.

The intended speeds for ATM networks are 155.52 MB/s (for compatibility with SONET) and 622 MB/s (for four 155 MB/s channels).

The ATM architecture consists of two main layers. Just above the physical layer is the *ATM layer*, which deals with cells and cell transport. It defines the layout of the cell and what the header means. It also deals with establishment and release of virtual circuits. Congestion control is also located here. Because most applications do not work differently with cells, a layer above the ATM layer has been defined that allows users to send packets larger than a cell. This is called the *ATM Adaptation Layer (AAL).* There are five types of AALs defined for different types of services. AAL 1 and AAL 2 are used for transporting voice directly over ATM whereas, if we want to use voice over IP over ATM, then we would need to use AAL 5.

5.c. IP NETWORKS:

IP networks are networks that use the ubiquitous internet protocol i.e. IP. They are generally based on the TCP/IP stack, which is described in more detail in the next section. IP is the network level protocol that encapsulates the higher layer PDU (protocol data unit) into IP datagrams. The most significant feature of IP is its 32-bit IP address: a virtual address given to each host and router in the network. The actual physical address of the device is obtained using some address resolution protocol (ARP). The IP address was developed as a reference format for an address field understood by all devices as they use different formats for physical addresses based on the standards used. A characteristic of all IP networks is that they are best-effort networks i.e. the network does not implicitly provide a guaranteed or differentiated quality of service (QoS) or class of service (CoS). Higher layers such as TCP need to compensate for this. This is a major drawback for real-time traffic over IP networks, which require a certain QoS for the service to be acceptable. Thus separate protocols need to be developed and implemented in order to transfer voice over IP networks with an acceptable quality. These are discussed in detail in the next section.

5.d. IP vs ATM:

IP networks have been mainly designed to support a best-effort service, which is suitable for data. Thus as mentioned before, they still face many challenges to supporting high-quality LD voice in a multiservice environment. The new architectures and protocols developed will go a long way to help meet these challenges.

ATM protocols and standards for multiservice networks are largely defined. The main challenges remaining for LD voice services are related to the selection of speech processing algorithms, a transport protocol above the ATM layer i.e. AAL1 or AAL2 and the right network architecture. ATM interoperates with PSTN networks rather well as they can use similar signaling standards. ATM also beautifully integrates different types of traffic encompassing all type of CBR (Constant Bit Rate) and VBR (Variable Bit Rate) traffic. It supports QoS guarantee and inbuilt QoS differentiation. ATM also has tremendous support from physical layer standards such as SONET. Thus, it appears to the perfect technology for voice. However, being multifaceted has its drawbacks. Congestion control, flow control and management issues are yet to be solved in their entirety.

While an IP network incurs significantly higher delay and jitter at lower link speeds, the difference diminishes as the link speed increases. The continuing growth of data traffic will justify the higher-speed links needed to support IP data traffic. Carrying voice at high priority on these IP networks will then be very attractive. While IP packets are allowed to be as large as 64 KB, the maximum size for packets carried on today’s Internet is 1,536 bytes, with an average of about 350 bytes. If this maximum packet size does not change, then excellent delay jitter performance is possible with link speeds of OC-3 and higher. For ATM, we do not need to limit the size of IP packets to allow control of delay jitter for voice.

With the current protocol stack, voice over IP is less bandwidth efficient than voice over ATM using either AAL-1 or AAL-2. Innovations discussed in this report will help narrow or eliminate the gap. As mentioned before, for data traffic originating from IP endpoints, IP is more bandwidth efficient than IP over ATM, owing to the ATM and AAL-5 overheads, as well as the partial fill of ATM cells using AAL-5. As the proportion of IP data traffic grows, the overall bandwidth efficiency of an integrated IP network becomes more favorable than that of an integrated ATM network.

Both ATM and IP can offer multi-application VPN (Virtual Private Network) services. The ability to offer such services is built into ATM standards and products. The currently emerging Layer 3/4 IP switches have flexible, hierarchical bandwidth management that will help make IP networks increasingly more suitable to offer such services. The emerging DiffServ standardization using the DS field will further add to the flexibility of IP networks that provide LD voice and multi-application VPN services.

The biggest hindrance in the widespread deployment of ATM for integrated voice-data services is the cost. ATM equipment is very expensive currently. Also, IP networks are he most prevalent in the world today. To convert them to ATM requires a complete revamping which is too expensive for most. The success of voice over packet networks is based on the lower cost factor. If we take that away there is no reason for going ahead with the idea. Thus, we feel it is better to stick to IP networks and enhance their capabilities. Although a daunting task it would be worth it. Time is the essence; since if the current PSTN carriers do provide better services at lower costs sometime soon, the market for VoIP services may die out.

We now discuss some of the benefits, potential pitfalls and applications of VoIP.

6. BENEFITS OF VOIP:

Below we discuss the benefits of using VoIP. We deal with the corporate as well as public (ISP) aspects.

6.a. CORPORATE ADVANTAGES:

A) **Lower Recurring Transmission Charges:** By directing voice calls over the corporate data network, rather than through a carrier, companies can significantly reduce their monthly phone bills. These savings are obviously dependent on several factors, including the volume of intracompany calls and the distances between company offices. Companies with overseas offices, obviously, can experience the greatest savings, since they can eliminate a great deal of international long-distance charges. These charges are often particularly high when the call originates in a foreign country that still has a highly monopolistic telecom market. In some configurations, these savings can be extended to calls outside of the company as well using PSTN gateways.

B) **Economic Factors:** The economic appeal of transmitting voice calls over the data network arises from two technical factors. First, data networks almost always have spare capacity. Network managers typically over-provision IP networks to allow room for growth and to avoid congestion during periods of peak utilization. At the same time, voice calls consume relatively little bandwidth. The characteristics of human speech, especially the comparatively large amount of silence that takes place during conversations, allows for a great deal of compression in the digitized transport of the call. This makes it possible for voice to ``piggyback" on existing data network connections without requiring investments in adding to the capacity of those connections. Even when such additions have to be made to the existing network because of call volumes, those costs are insignificant compared to the recurring costs charged by carriers to carry that same calling volume.

C) **Reduced Long-term Network Ownership Costs**: In addition to reducing a company's monthly phone bills, converged network architecture also reduces the ongoing costs of owning two separate networks --- one for voice and one for data. These costs include the need to buy two separate sets of equipment, the staff time dedicated to the operation and maintenance of that equipment, the licensing of any software relating to the management of that equipment, and the monitoring of traffic on the two networks. With the Internet revolution in full swing, the demand for skilled, experienced technicians far outstrips supply. This has driven salaries for voice and data network staff through the roof, and has also made it difficult to recruit and retain such engineering talent. Companies that are able to reduce their need for technical staff by streamlining their network operations can therefore eliminate many of the human resource management headaches that plague their competitors.

D) **Advanced Applications:** The most compelling aspect of converged voice/data networking may well be the new generation of applications it enables. These applications include Web enabled call centers, unified messaging and real-time collaboration. Other examples include real-time multimedia video/audio conferencing, distance learning, and the embedding of voice links into electronic documents. Three or four years ago, the Internet was not ready for prime time as a medium for commerce. But now it is. VoIP will be a valuable enhancement.

6.b. WHY ISPs SHOULD MIGRATE TO IP TELEPHONY:

The tough competition in telecommunications has slashed revenue margins on basic data and voice services. In order to stay profitable, service providers must introduce new revenue-producing services to retain their customers, and cut costs simultaneously. Only through integrating their circuit and packet networks can service providers take advantage of the universality of the circuit switch telephone network (current PSTN) and leverage the inherent flexibility in the software-based and open architecture of packet networks. Packet-based network equipment is easier to upgrade and configure, enabling quick time to market. Service providers need to quickly rollout new revenue generating services

- The software-based architecture of packet-based devices can be leveraged to quickly develop new services.
- The standards-based interfaces of packet-based devices can be used to quickly configure or upgrade devices to support new services

7. RISK FACTORS:

A) **Loss of Voice Quality:** Technologists understand that data networks are very different from voice networks. On the data network, especially on those Ethernet transports that dominate corporate computing environments, packets bounce around somewhat indeterminately. They can collide and get distorted or even lost. Error correction mechanisms in Ethernet hardware and the IP protocol itself can readily compensate for these phenomena on the data side. But such problems can adversely affect voice calls, which require a good quality, real-time flow of packets from one end of the network to the other. And, while the human brain can comprehend human speech even when there is a lot of distortion, users have become accustomed to a certain level of call quality.

B) **Loss of Reliability:** Data networks are not yet as reliable as voice networks. We all know what it is like to have our computer freeze or to be told that the network is ``down." But this rarely happens with our phones or our telephone carriers. Immediate and uninterrupted access to others over the phone is such an essential aspect of conducting business that few executives want to put voice communications at risk, regardless of how attractive the potential savings may be.

C) **Vendor Architecture Dependence**: The pace of change in computing and communications technology today makes vendor ``lock-in" a major concern for any potential buyer. There are really two aspects of lock-in that trouble most decision makers. One is the possibility that another, superior solution for VoIP will come along shortly after a commitment has been made to a particular vendor's product. If the investment in that product is substantial, it's usually impractical to scrap it and switch to the better approach. Of even greater concern for technology managers, however, is the fact that selection of one vendor's approach to voice/data convergence may cause a lock-in that extends far beyond the VoIP solution itself, forcing a long-term commitment to that vendor's overall networking architecture. This concern is exacerbated by the lack of clear standards in the VoIP market. In the absence of such standards, technology managers have legitimate concerns about committing their companies to any proprietary architecture.

D) **Lack of Expertise and Experience**: VoIP technology is new. Every new tool must be tested and mastered. This takes time. Haste makes waste. Without the proper expertise and careful planning the technology can work against the customer.

8. APPLICATIONS OF VOICE OVER IP:

VoIP can be defined as the ability to make telephone calls (i.e., to do everything we can do today with the PSTN) and to send facsimiles over IP-based data networks with a suitable quality of service (QoS) and a much superior cost/benefit. VoIP could be applied to almost any voice communications requirement, ranging from a simple inter-office intercom to complex multi-point teleconferencing/shared screen environments. The quality of voice reproduction to be provided could also be tailored according to the application. Customer calls may need to be of higher quality than internal corporate calls, for example. Hence, VoIP equipment must have the flexibility to cater to a wide range of configurations and environments and the ability to blend traditional telephony with VoIP.

Some examples of VoIP applications that are likely to be useful would be:

a) **PSTN gateways**: Interconnection of the Internet to the PSTN can be accomplished using a gateway, either integrated into a PBX or provided as a separate device. A PC-based telephone, for example, would have access to the public network by calling a gateway at a point close to the destination (thereby minimizing long distance charges). This will be discussed in detail later.

b) **Internet-aware telephones**: Ordinary telephones (wired or wireless) can be enhanced to serve as an Internet access device as well as providing normal telephony. Directory services, for example, could be accessed over the Internet by submitting a name and receiving a voice (or text) reply.

c) **Inter-office trunking over the corporate intranet**: Replacement of tie trunks between company-owned PBXs using an Intranet link would provide economies of scale and help to consolidate network facilities.

d) **Remote access from a branch (or home) office:** A small office (or a home office) could gain access to corporate voice, data, and facsimile services using the company's Intranet (emulating a remote extension for a PBX, for example). This may be useful for home-based agents working in a call center, for example.

e) **Voice calls from a mobile PC via the Internet:** Calls to the office can be achieved using a multimedia PC that is connected via the Internet. One example would be using the Internet to call from a hotel instead of using expensive hotel telephones. This could be ideal for submitting or retrieving voice messages.

f) **Internet call center access:** Access to call center facilities via the Internet is emerging as a valuable adjunct to electronic commerce applications. Internet call center access would enable a customer who has questions over the Internet to access customer service agents online. Another VoIP application for call centers is the interconnection of multiple call centers. Take the example of a Web-enabled call center. One of the biggest obstacles that companies face in converting Web site visitors into Web site buyers is poor online interaction. In a store, customers can ask a nearby salesperson a question that may end up determining whether or not they head for the checkout line. On a Web site, that kind of interaction is more problematic. But using VoIP, site visitors can click a button and open up a voice conversation with a real, live call center agent who can quickly address any question or problem the customer might have.

One of the parallel applications for IP telephony is *real-time facsimile transmission* also called *FoIP (Fax over IP).* Facsimile services normally use dial-up PSTN connections, at speeds up to 14.4 Kbps, between pairs of compatible fax machines. Transmission quality is affected by network delays, machine compatibility, and analog signal quality. To operate over packet networks, a fax interface unit must convert the data to packet form, handle the conversion of signaling and control protocols, and ensure complete delivery of the scan data in the correct order.

Most VoIP applications that have been defined are considered to be real-time activities. *Store-and-forward voice services* will also be implemented using VoIP. For example, voice messages could be prepared locally using a telephone and delivered to an integrated voice/data mailbox using Internet or intranet services. Voice annotated documents; multimedia files etc. will also become standard within office suites in the near future. The real-time and store-and-forward modes of operation will need to be compatible and interoperable.

9. MARKET SIZE & POTENTIAL CUSTOMERS:

9.a. MARKET SIZE:

There is a wide range of numbers describing both the current size of the IP telephony market, and the growth of the market over the next three to five years. While the specific projections vary, even the most conservative analysts are predicting phenomenal growth. The numbers for the IP telephony equipment market are summarized below.

Frost & Sullivan:

- 1997 market size $47.3 million
- Predicted compound annual growth rate: 132%
- Predicted market size in 2002: $3.16 billion

International Data Corp.:

- International and US long distance on IP in 1997: 0.2%
- Predicted international and US long distance on IP in 2002: 11%

9.b. POTENTIAL CUSTOMERS:

Potential customers for IP telephony technology and applications include not only companies directly involved in the Internet business but also companies with private networks or intranets, wide area networks (WANs)/extranets, and enterprise networks. Examples include:

IXCs (InterExchange Carriers): Examples include AT&T, MCI, Sprint, GTE, Deutsche TeleKom, Telecom Finland, Telecom New Zealand, and Daewoo International. AT&T's Globalnet division is currently offering IP telephony service, MCI has built PC-based web servers which support IP telephony, and Sprint is developing its Global One service that offers dedicated bandwidth on demand. Deutsche TeleKom purchased an equity stake in VocalTec, a leading supplier of IP telephony gateways, and launched a service in 1998, Telecom New Zealand completed a trial of IP telephony in early 1998, and the other international carriers have similar plans for deployment.

Long distance carriers: such as AT&T, MCI, Sprint, Cable & Wireless, NTT, France Telecom, and Deutsche Telekom. Even telephone carriers not directly involved in the Internet business are offering services that take advantage of IP telephony, such as reduced-rate voice and fax, and store-and-forward fax and voice messaging.

RBOCs (Regional Bell Operating Companies): such as Bell Atlantic, US West, and Pacific Bell. While the recent ruling allowing RBOCs to offer long distance services will be in dispute for quite some time, it is natural to assume that RBOCs will look at packet-based technology as they begin to roll out their inevitable long distance applications. Additionally, through their alliances with ISPs, RBOCs are a natural convergence point for traditional voice communications and data networks.

Internet Service Providers (ISPs): There are arguably between one and two dozen large ISPs in the USA, notably Alternet, BBN Planet, Digital Express, NetCom, PSI, and UUNET/MFS / Worldcom. There are also hundreds to thousands of smaller and single-region ISPs such as PANIX and TIAC, as well as tens of thousands of bulletin board services (BBSs). Many international ISPs are also offering IP telephony services, such as Rimnet of Japan and OzEmail of Australia. VSNL India has promised VoIP services by 2002.

Internet Telephony Service Providers (ITSPs): Representing a new class of service providers, these companies are building global IP networks specifically designed for low-latency traffic, including voice and fax. This emerging market already has established leaders including Level 3, Qwest, Williams, ITXC, Delta Three, and VIP Calling, and will undoubtedly spawn new competitors in the coming months.

Call centers and larger service bureaus: These organizations will benefit from new business processes that will capture revenue from web-related sources. This is one of the key applications for IP telephony and would serve organizations like SITA (Societe Internationale de Telecommunications Aeronautiques).

Businesses and organizations: Any company with an Internet connection or a corporate intranet is a potential IP telephony customer. This includes Fortune 100--2000 companies and countless smaller businesses, as well as colleges, universities, government agencies, and non-profit organizations, all of whom have significant needs for long distance and international voice and fax service or other IP telephony applications.

For Voice over IP to become popular some key issues need to be resolved. Below, we discuss them in brief.

10. KEY ISSUES:

The goal for developers is relatively simple: add telephone calling capabilities (both voice transfer and signaling) to IP-based networks and interconnect these to the public telephone network and to private voice networks in such as way as to maintain current voice quality standards and preserve the features everyone expects from the telephone. The key issues are discussed below

1. Voice quality should be comparable to what is available using the PSTN, even over networks having variable levels of QoS.
2. The underlying IP network must meet strict performance criteria including minimizing call refusals, network latency, packet loss, and disconnects. This is required even during congestion conditions or when multiple users must share network resources.
3. Call control (signaling) must make the telephone calling process transparent so that the callers need not know what technology is actually implementing the service.
4. PSTN/VoIP service interworking (and equipment interoperability) involves gateways between the voice and data network environments.
5. System management, security, addressing (directories, dial plans) and accounting must be provided, preferably consolidated with the PSTN operation support systems (OSSs).

10.a. SPEECH QUALITY AND CHARACTERISTICS:

VOICE ENCODING:

The analog data is converted to digital form with a codec. If the digital samples are to represent an analog waveform, the sampling rate must be at least twice the highest frequency present in the analog signal (Nyquist theorem). Since the PSTN telephone networks eliminate frequencies over 3.3kHz, the analog speech signals are sampled at 8 kHz for Pulse Code Modulation (PCM) applications. G.711 is the international standard for encoding telephone audio on a 64 kbps channel. It is a PCM scheme operating at 8 kHz sample rate, with 8 bits per sample. There are two different variants of G.711: A-law and mμ-law. A-law is the standard for international circuits; mμ-law is the US standard. The benefit of this kind of techniques is that while maintaining a high quality, the amount of bandwidth is significantly reduced. More advanced compression algorithms can be used in cases of bandwidth starvation. But this does reduce the quality of the speech.

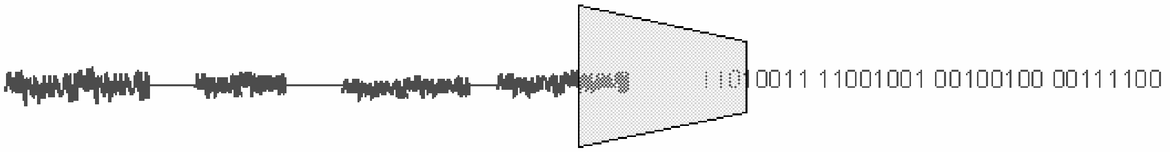


Fig4: Representation of a Codec

Providing a level of quality that at least equals the PSTN (this is usually referred to as ``toll quality voice") is viewed as a basic requirement, although some experts argue that a cost versus function versus quality trade-off should be applied. Although QoS usually refers to the fidelity of the transmitted voice and facsimile documents, it can also be applied to network availability (i.e., call capacity, or level of call blocking), telephone feature availability (conferencing, calling number display, etc.), and scalability (any-to-any, universal, expandable). The quality of sound reproduction over a telephone network is fundamentally subjective, although standardized measures have been developed by the ITU. It has been found that there are three factors that can profoundly impact the quality of the service.

<u>DELAY:</u>

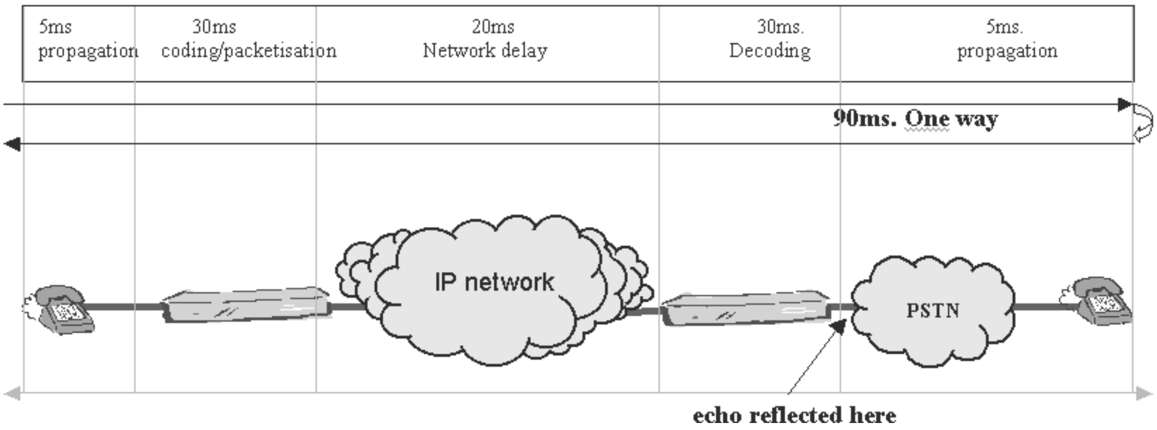
Following are sources of delay in an end-to-end Voice over Packet call:

1. Accumulation Delay (sometimes called algorithmic delay): This delay is caused by the need to collect a frame of voice samples to be processed by the voice coder. It is related to the type of voice coder used and varies from a single sample time (.125 microseconds) to many milliseconds.
A representative list of standard voice coders and their frame times follows:

- G.726 ADPCM (16, 24, 32, 40 Kbps) - .125 microseconds
- G.728 - LD-CELP (16 Kbps) - 2.5 milliseconds
- G.729 - CS- ACELP (8 Kbps) - 10 milliseconds
- G.723.1 - Multi Rate Coder (5.3, 6.3 Kbps) - 30 milliseconds

2. Processing Delay: This delay is caused by the actual process of encoding and collecting the encoded samples into a packet for transmission over the packet network. The encoding delay is a function of both the processor execution time and the type of algorithm used. Often, multiple voice coder frames will be collected in a single packet to reduce the packet network overhead. For example, three frames of G.729 codewords, equaling 30 milliseconds of speech, may be collected and packed into a single packet.

3. Network Delay: This delay is caused by the physical medium and protocols used to transmit the voice data, and by the buffers used to remove packet jitter on the receive side. Network delay is a function of the capacity of the links in the network and the processing that occurs as the packets transit the network. The jitter buffers add delay, which is used to remove the packet delay variation that each packet is subjected to as it transits the packet network. This delay can be a significant part of the overall delay since packet delay variations can be as high as 70-100 msec. in some IP networks.



Echo: Two problems that result from high end-to-end delay in a voice network are echo and talker overlap. *Echo* becomes a problem when the round-trip delay is more than 50 milliseconds. In circuit switched systems, echo is caused by signal reflections generated by the hybrid connection that converts between a 4-wire circuit (2 separate transmit and receiver pair) and the 2-wire circuit (1 transmit and receiver pair). When the signals pass from the 4- wire to the 2-wire, some of the energy in the 4-wire circuit is reflected back towards the speaker. The 4-wire conversion is required because all calls, except for local, require the amplification provided by a 4-wire circuit. However, the investment cost of upgrading 2-wire circuits to 4-wire circuits from the local loop to the local subscriber is too high. Echo is not always bad. A small amount of echo is called sidetone, which is positive. The sidetone echo reinforces that your voice is being carried towards the conversation partner. As long as the round-trip delay is less than 50 ms and is not too loud, it is acceptable. However as the delay between your voice and the reflected signal increases, the echo becomes intrusive. In circuit switch networks, the round-trip delay of echo is less then 50 ms because circuit switched networks are configured to cancel out any echo with delay above 45 to 50 ms depending on the network. This standard is defined in ITU specification G.131. The 50 ms standard was developed because it is the maximum length of delay that is unnoticeable to the speaker. Unfortunately, due to network delay, the delay in IP networks can be much higher.

Talker overlap: (the problem of one caller stepping on the other talker's speech) becomes significant if the one-way delay becomes greater than 250 milliseconds. The end-to-end delay budget is therefore the major constraint and driving requirement for reducing delay through a packet network.

JITTER (DELAY VARIABILITY):

The delay problem is compounded by the need to remove jitter; a variable inter-packet timing caused by the network a packet traverses. Removing jitter requires collecting packets and holding them long enough to allow the slowest packets to arrive in time to be played in the correct sequence. This causes additional delay. The two conflicting goals of minimizing delay and removing jitter have engendered various schemes to adapt the jitter buffer size to match the time varying requirements of network jitter removal. This adaptation has the explicit goal of minimizing the size and delay of the jitter buffer, while at the same time preventing buffer underflow caused by jitter.

LOST PACKETS:

IP networks cannot provide a guarantee that packets will be delivered at all, much less in order. Packets will be dropped under peak loads and during periods of congestion (caused, for example, by link failures or inadequate capacity). Due to the time sensitivity of voice transmissions, however, the normal TCP-based retransmission schemes are not suitable. Approaches used to compensate for packet loss include interpolation of speech by re-playing the last packet, and sending of redundant information. Packet losses greater than 10% are generally not tolerable.

10.b. HOW TO IMPROVE THE QUALITY OF VOICE:

ECHO CANCELLATION:

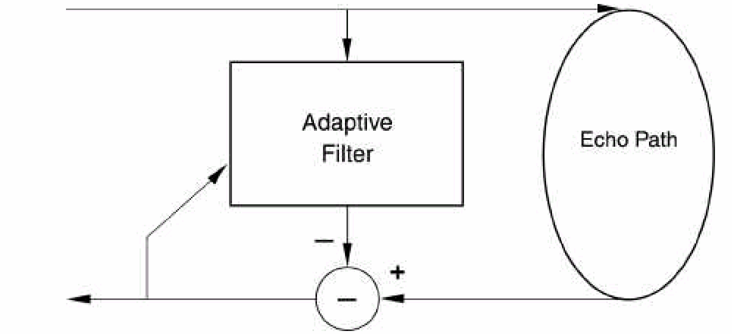


Fig5:Principle of Echo Cancellation

Echo canceller processing can be divided into two processes, storing and comparing/filtering:

Storing: In order to identify a reflected signal, the echo canceller must first store the incoming signal. Therefore, all of the voice traffic transiting the IP network is stored in a First In First Out (FIFO) buffer. The size of the buffer is determined by the expected echo path delay. The longer the expected echo path delay the larger the required buffer. There are two types of echo cancellation. Near End Echo Cancellation is so called because the echo is cancelled nearest to the echo source. It is possible to perform echo cancellation at the far end, but the echo path will be much longer, normally requiring more processing power.

Comparing and Filtering: The echo canceller compares the signal similarity and power level coming from the hybrid point to the previously stored signals passed from the packet network. The comparison is made using a model of the hybrid circuit, which was created by an adaptive algorithm inside the echo canceller. However, identifying echo is a difficult task for the adaptive filter because of ``double-talk" issues. Double talk occurs when both sides of a conversation attempt to speak simultaneously. The near end speaker may confuse the echo identification process, thus limiting the effectiveness of the echo canceller. Also, The echo canceller must not clip the beginning of the double talk session. To alleviate the double talk issue, a double talk detector (DTD) is implemented. The double-talk detector works in most cases, but when it fails to detect double-talk, some echo is still noticeable. The echo canceller also needs to identify and remove background noise. Once an echo is identified, the echo canceller subtracts the echo from the returning signal. Echo cancellation can be very resource intensive. The amount of processing power necessary to accurately compare and filter out echo can be high. This need for high processing power can be detrimental to the vendor's overall solution. Since processing power is a finite resource, as more resources are diverted to echo cancellation, fewer resources are available for processing additional voice channels. In summary, high processing power requirements for echo cancellation mean more power consumption, higher cost, and lower port density.

JITTER COMPENSATION (ADAPTIVE BUFFERS):

Two approaches to adapting the jitter buffer size are detailed below. The approach selected will depend on the type of network the packets are traversing.

1. The first approach is to measure the variation of packet level in the jitter buffer over a period of time, and incrementally adapt the buffer size to match the calculated jitter. This approach works best with networks that provide a consistent jitter performance over time.
2. The second approach is to count the number of packets that arrive late and create a ratio of these packets to the number of packets that are successfully processed. This ratio is then used to adjust the jitter buffer to target a predetermined allowable late packet ratio. This approach works best with the networks with highly variable packet inter-arrival intervals, such as most IP networks.

In addition to the techniques described above, the network must be configured and managed to provide minimal delay and jitter, enabling a consistent quality of service.

LOST PACKET COMPENSATION:

Lost packets can be an even more severe problem, depending on the type of packet network that is being used. Because IP networks do not guarantee service, they will usually exhibit a much higher incidence of lost voice packets than ATM networks. In current IP networks, all voice frames are treated like data. Under peak loads and congestion, voice frames will be dropped equally with data frames. The data frames, however, are not time sensitive and dropped packets can be appropriately corrected through the process of retransmission. Lost voice packets, however, cannot be dealt with in this manner.

Some schemes used by Voice over Packet software to address the problem of lost frames are:

1. Interpolate for lost speech packets by replaying the last packet received during the interval when the lost packet was supposed to be played out. This scheme is a simple method that fills the time between non-contiguous speech frames. It works well when the incidence of lost frames is infrequent. It does not work very well if there are a number of lost packets in a row or a burst of lost packets.
2. Send redundant information at the expense of bandwidth utilization. The basic approach replicates and sends the nth packet of voice information along with the (n+1)th packet. This method has the advantage of being able to exactly correct for the lost packet. However, this approach uses more bandwidth and also creates greater delay.
3. A hybrid approach uses a much lower bandwidth voice coder to provide redundant information carried along in the (n+1)th packet. This reduces the problem of the extra bandwidth required, but fails to solve the problem of delay.

VOICE ACTIVITY DETECTION:

To reduce the effective bandwidth used by the voice signal we can use voice activity detector, which suppresses packet transmission when voice signals are not present (and hence saves additional bandwidth). If no activity is detected for a period of time, the voice encoder output will not be transported across the network. Idle noise levels are also measured and reported to the destination so that ``comfort noise" can be inserted into the call (so that the listener does not get dead air on their telephone).

We now move over to the second section of this report where we discuss in detail some of the prevalent architectures and protocols developed to support voice over IP networks.

SECTION II

11. BASICS OF PROTOCOLS & PROTOCOL ARCHITECTURES:

A protocol is often defined as a set of agreements necessary to exchange data between two nodes connected directly to each other or connected to a network. A protocol is only one module in a hierarchy of modules, together the modules set up the protocol architecture. An explanation of the two most common protocol architectures (OSI and TCP/IP) will be described below. TCP/IP is the most widely used interoperable architecture, and OSI has become the standard model for classifying communications functions.

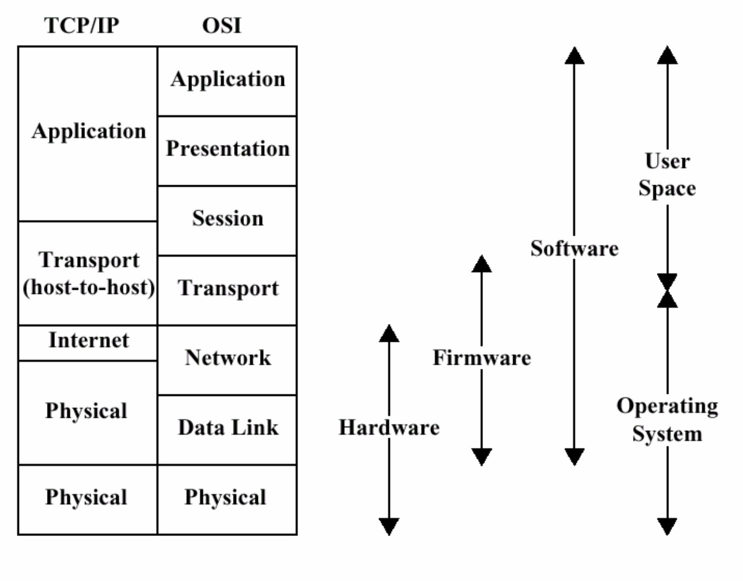


Fig6: Protocol Architectures

11.a. OSI REFERENCE MODEL:

Physical layer: This layer is responsible for the mechanical, electrical, functional and procedural mechanism required for the transmission of data. It can be considered to represent the physical connection of a device to a transmission media.

Data link layer: The data link layer is responsible for the manner in which data is formatted into defined fields and the correction of any errors occurring during a transmission session. It is responsible for framing as well as flow control, error detection and correction.

Network layer: Provides upper layer with independence from the data transmission and switching technologies used to connect system. Responsible for establishing, maintaining and terminate connections.

Transport layer: Provides reliable and transparent transfer of data between end points. The transport layer also provides end-to-end error recovery and flow control.

Session layer: This layer is responsible for establishing and terminating data streams between network nodes. Since each data stream can represent an independent application, the session layer is also responsible for coordinating communications between different applications that require communications.

Presentation layer: Provides independence to the application processes from differences in data representation.

Application layer: Provides access to the OSI environment for users and also provides distributed information services.

11.b. TCP/IP PROTOCOL SUITE:

TCP/IP is a result of protocol research and development conducted on the experimental packet-switched network, ARPANET, funded by the Defense Advanced Research Projects Agency (DARPA), and is generally referred to as the TCP/IP protocol suite. This protocol suite consists of a large collection of protocols that have been issued as Internet standards by the Internet Architecture Board (IAB). There is no official TCP/IP protocol model as there is in the case of OSI. However, based on the protocol standards that have been developed, the communication task for TCP/IP can be organized into five relatively independent layers. The most common protocols are presented.

Physical layer: This Layer is responsible for the mechanical, electrical, functional and procedural mechanism required for the transmission of data. It can be considered to represent the physical connection of a device to a transmission media.

Network access layer: This layer is responsible for accepting and transmitting IP datagrams. It is also concerned with the exchange of data between an end system and the network to it is attached. The sending computer must provide the network with the physical address of the destination computer, so that the network may route the data to the appropriate destination. Ethernet, IBM token ring, PPP (Point to Point Protocol), LAPD, LAPB etc are used at this level. It is analogous to the data link layer of OSI.

Internet layer: This layer handles communication from one machine to the other. It accepts a request to send data from the transport layer, along with the identification of the destination. It encapsulates the transport layer data unit in an IP datagram and uses the datagram routing algorithm to determine whether to send the datagram directly onto a router. The Internet layer also handles the incoming datagrams and uses the routing algorithm to determine whether the datagram is to be processed locally or to be forwarded.

- **Internet protocol IP**: IP provides a connectionless service between end systems. All the intermediate systems between the two ends just have to implement IP and the layers below IP. It is not necessary to implement higher layers. IP receives data from higher layers, and adds a header containing information related to the data received and passes it to the layer below. These packets are called IP datagrams. Each packet that travels through the Internet is treated as an independent unit of data without any relation to any other unit of data. It is for the protocol above to keep track of the order between packets and for guaranteeing that the packet arrives. The IP header contains all routing information necessary about sending the packet to the right next hop. If the packets are bigger than the maximum size that the link can handle, the IP protocol also takes care of fragmentation and reassembling of packets.
- **Internet Control Message Protocol ICMP**: The IP standard specifies that a compliant implementation must also implement ICMP. ICMP provides a means for transferring messages from routers and other hosts to a host. In essence, ICMP provides feedback about problems in the communication environment. Although ICMP is, in effect, at the same level as IP in the TCP/IP architecture, it is a user of IP. An ICMP message is constructed and then passed down to IP, which encapsulate the message with an IP header and then transmits the resulting datagram in the usual fashion.

Transport layer: In this layer the stream of data is segmented into small data units and each packet is passed along with a destination addresses, to the layer below for transmission. The software also adds information to the packets, including port number that identify which application program sent it, as well as a checksum. This layer also regulates the flow of information and provides reliable transport, ensuring that data arrives in sequence and with no errors.

- **Transmission Control Protocol TCP**: TCP was developed to provide a reliable, connection-oriented service that supports end-to-end transmission reliability. To accomplish this task, TCP supports error detection and correction as well as flow control to regulate the flow of packets through a network. Error checking requires the computation of a Cyclic Redundancy Check (CRC) algorithm at a network node based on the contents of a packet and a comparison of the computed CRC against the CRC carried in the packet. If an error occurs TCP requests a retransmission of the faulty packets. This can result in unacceptable delays when transporting digitized voice and is rarely, if ever, used for voice transmission.
- **User Datagram Protocol UDP**: UDP was developed to provide an unreliable, connectionless transport service. We should note that this is not necessary bad and adds a degree of flexibility to the protocol family. That is, if reliability is required, a higher layer, such as the application layer, can be used to ensure that messages

are properly delivered. A second feature of UDP is the fact that it is a connectionless protocol. This means that instead of requiring a session to be established between two devices, transmission occurs on a best-effort basis. That is, function associated with connection setup and the exchange of status information as well as flow control procedures are avoided.

Application layer: At this level, users invoke application programs to access available services across the TCP/IP Internet. The application program chooses the kind of transport needed, which can be either messages or streams of bytes, and passes it to the transport level. Some of the common application layer protocols are discussed below:

- **File Transfer Protocol FTP**: The FTP is a mechanism for moving data files between hosts via a TCP/IP network. FTP operates as a client-server process, with the client issuing predefined commands to the server to navigate its directory structure and to upload and download files to and from the server.
- **Telnet**: Telnet represents another TCP/IP client-server application. This application is designed to enable a client to access a remote computer as though the client was a terminal directly connected to the remote computer.
- **Simple Mail Transfer Protocol SMTP**: The SMTP provides the data transportation mechanism for electronic messages to be routed over a TCP/IP network. This protocol is completely transparent to the user since there are no commands that govern the transfer of electronic mail via SMTP.
- **Hypertext Transmission Protocol HTTP**: HTTP is the foundation protocol of the World Wide Web and can be used in any client/server application involving hypertext. The data transferred by the protocol can be plain text, hypertext, audio, images, or any Internet accessible information.
- **Simple Network Management Protocol SNMP**: The SNMP provides the mechanism to transport status messages and statistical information about the operation and utilization of TCP/IP devices. In addition, with SNMP, devices can generate alarms when certain predefined thresholds are reached.

The next section will describe the potential architectures for Internet Telephony.

12. FRAMEWORK FOR INTERNET TELEPHONY:

Users wishing to employ IP telephony today have a wide choice of protocols. The IP service architecture for voice support has a number of requirements that are not available in the IP architectures used in the field today.

12.a. REQUIREMENTS:

- **Billability**. Service providers want to make a profit. Therefore, the architecture must support billable services.
- **Privacy**. The architecture should provide privacy to its users. A user may, for example, not wish to reveal the IP address of the terminal he/she is currently using because it can give information about the user's location.
- **Security**. The architecture should be able to provide user authentication, authorization, confidentiality, non-repudiation, and integrity.
- **Scalability**. The architecture should be able to scale to deployments as large as those of telephony operators today.
- **Flexibility for terminals and deployment**. The current growth of computerized equipment and corresponding applications gives the user a choice of possible communications tools. Users should not be constrained to use a small subset of the possibilities because of limitations in the architecture.
- **Sufficient detail**. The architecture should provide enough detail to define clear interfaces that allow proper protocol development.

In addition to meeting these requirements, the architecture should support the following services:

- **Basic services**, such as the ability to place and receive basic voice calls, hold conference calls, block caller ID, and use call waiting, as well as reliability with regard to toll-quality voice and caller ID.
- **Billing options**, such as free phone, calling card, credit card, and subscription.
- **Lifeline service**, or the ability to continue to function at a minimal level in cases of power failure.

Other services that could be provided when the user operates in the IP domain include distributed computing, shared whiteboard, video, click-and-call, and CD-quality audio. Finally, since IP users are often very mobile, nomadic roaming and terminal mobility should be supported.

An ideal architecture should work as follows:

1. A user boots a terminal and has the terminal register both itself and the user's identity with the service provider (SP).
2. The registration of the user should be checked with some kind of authentication manager and if the outcome is positive the registration is accepted.
3. The SP should now update the user directory and/or the user profile allowing the user to place and receive calls at his terminal.

4. A registered user wishing to place a call locates some kind of server, which acts as a session manager an initiates a session specifying the intended recipient and the media types.
5. This session manager queries the directory services and subsequently the user profile, locating the appropriate terminal that can accept the call for the recipient.
6. If and when the recipient accepts the call, the session manager should activate some kind of media manager that will aid in media negotiation between the terminals.
7. The media manager should communicate with the network management to ensure adequate network resources.
8. Finally, the media are streamed at the appropriate media and connection quality between the terminals.
9. A billing manager is informed of the session and its attributes so that billing can take place.
10. Interworking with say the PSTN can be accomplished by replacing the terminal with a media gateway and its service interface.

12.b. CURRENT ARCHITECTURES:

We consider several existing approaches: H.323, telecommunications and IP harmonization over networks (TIPHON), SIP, and telecommunications information networking architecture (TINA).

H.323. The H.323 standard developed by the ITU-T is actually a suite of protocols intended for audio and video communication between two or more parties over a non-QoS data network such as IP. H.323 is currently the most popular protocol for IP telephony and video conferencing applications. The ITU-T does not specify a functional architecture with specific details. This has resulted in an environment where several functions necessary for the working of the protocol may be implemented in more than one of the architecture elements. Thus, H.323 has become a fairly extensive protocol. The H.323 family will be described in more detail in the later sections.

TIPHON. The European Telecommunications Standards Institute (ETSI)/TIPHON standardization effort produces specifications for interworking between switched telephone networks and IP telephony. TIPHON uses H.323 as its basis and creates an architecture that allows this interworking. TIPHON delivers an architecture with more detail than that offered by H.323, as shown in the figure.

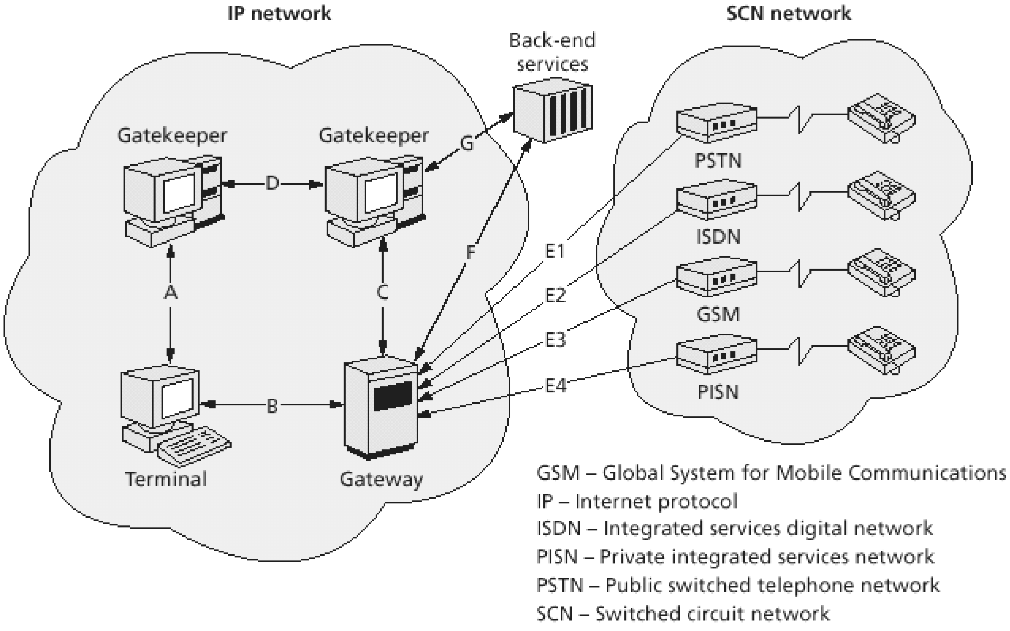

IP network
SCN network
Back-end services
Gatekeeper
Gatekeeper
D
G
A
C
F
E1
E2
E3
E4
B
Terminal
Gateway
PSTN
ISDN
GSM
PISN
GSM – Global System for Mobile Communications
IP – Internet protocol
ISDN – Integrated services digital network
PISN – Private integrated services network
PSTN – Public switched telephone network
SCN – Switched circuit network

Fig7: TIPHON architecture

It depicts a TIPHON reference architecture, whose functions have been organized partly by protocol and partly by function. TIPHON has also defined an architecture for interdomain communication.

SIP. SIP is under development within the multiparty multimedia session control (MMUSIC) group of the IETF (Internet Engineering Task Force). This group is developing a suite of protocols for multimedia conferencing. The authors of SIP took a different approach from H.323; they started out small and extended their protocol to suit their needs. The basic SIP architecture is discussed in more detail later. A call is initiated and directed to a host serving the address of the user (a process similar to that which occurs in the e-mail system). The receiving host can reject the call for a variety of reasons and can provide a forwarding address. Eventually, either a host is found to complete the call or there is no longer a forwarding address and the call fails.

The SIP approach offers great flexibility for the simple task of setting up a telephone call. Servers can be placed both in a service network and on the user's home terminal, and intelligence in the server can provide customized responses depending on the caller's identity. Because of its simplicity, SIP terminals must depend on other protocols to engage in more complex services.

TINA. TINA is more comprehensive and complex than the three architectures discussed above. The TINA architecture logically separates application services and network infrastructure so that services can be generic and independent of the (access) technology used. The TINA framework consists of many well-defined components, called computational objects, among which clear interfaces have been defined.

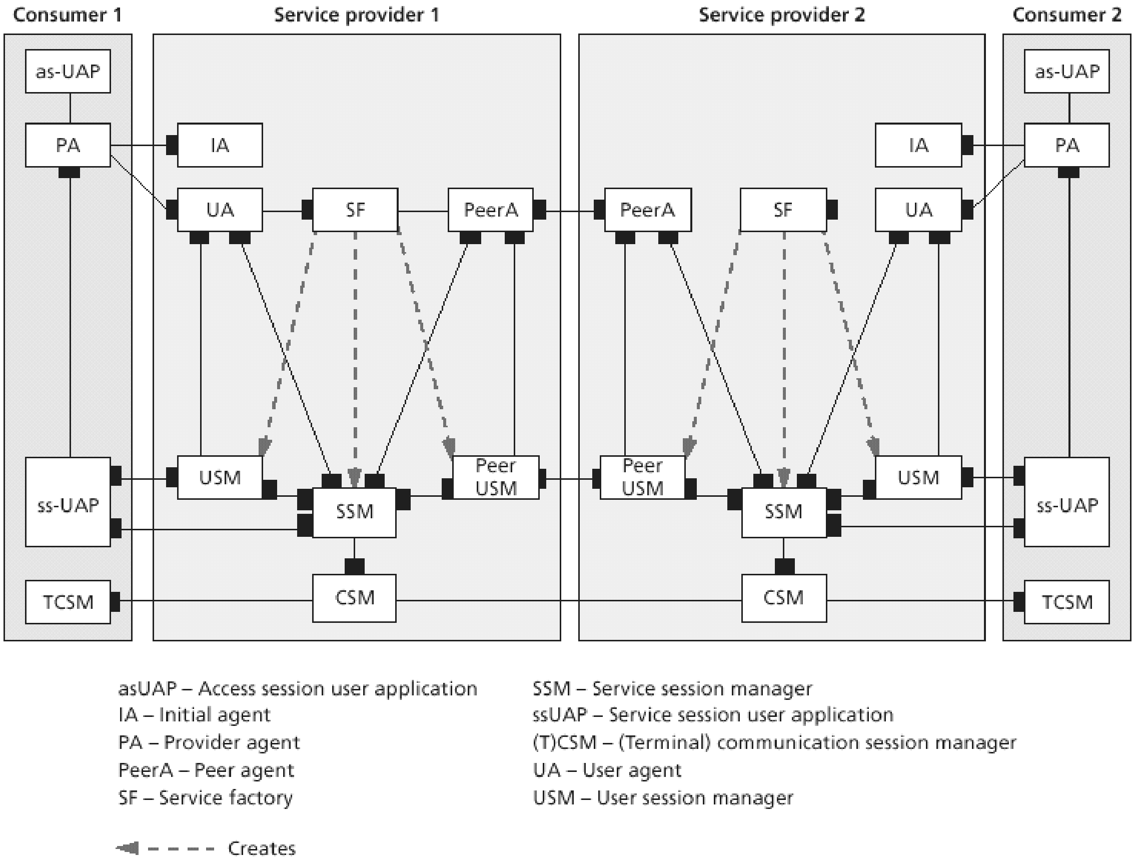


Fig8: TINA service architecture

The figure above gives an overview of the service architecture. The figure shows four domains: two consumer domains and two service domains. These domains consist of a number of computational objects (represented by boxes) with interfaces (represented by black boxes). Interaction between the access session user application (asUAP), provider agent (PA), initial agent (IA), and user agent (UA) is called the access session, during which the user connects to the service network and identifies himself/herself. After this access session, the user can select a service without further identification. The service selected by the user results in a service session consisting of the service factory (SF), service session user application (ssUAP), user session manager (USM), and service session manager (SSM). A service session that spans multiple service domains---for example, a long distance phone call---invokes an interface between two service providers created by the peer agent and the peer USM. Media negotiation and transcoding is handled in the communication session, which consists of the (terminal) communication session managers (TCSMs).

Despite all its sophistication, the TINA architecture assumes a rather rigid location of the service logic in the service network. For example, TINA does not explicitly disallow the placement of certain parts of the user profile in the terminal, but neither does it provide a suitable interface to establish this while other functions have been defined in much detail.

Evaluation: None of the architectures or protocols fully meets the requirements of the IP service architecture. This does not mean they cannot be modified to do so.

Current versions of H.323 lack the level of detail that is needed. For example, the only available service element in the H.323 architecture is the gatekeeper, which can be specialized in several ways that are undefined by H.323. H.323 is often accused of being heavyweight, but recent developments in the area of light H.323 deployments seem to address this issue.

TINA tends to be rather heavyweight for simple activities due to complicated basic messages and may lack the flexibility demanded by the IP environment. In some respects, TINA is lagging behind the times; for instance, it relies on placing a lot of functionality in the network. However, the powerful terminals in use today suggest that the service network of tomorrow can be designed with significantly less weight by exploiting the power of these terminals.

SIP is more lightweight. It also seems to be the most flexible & scalable of the lot and has great potential.

A typical protocol stack for Internet telephony is as shown below.

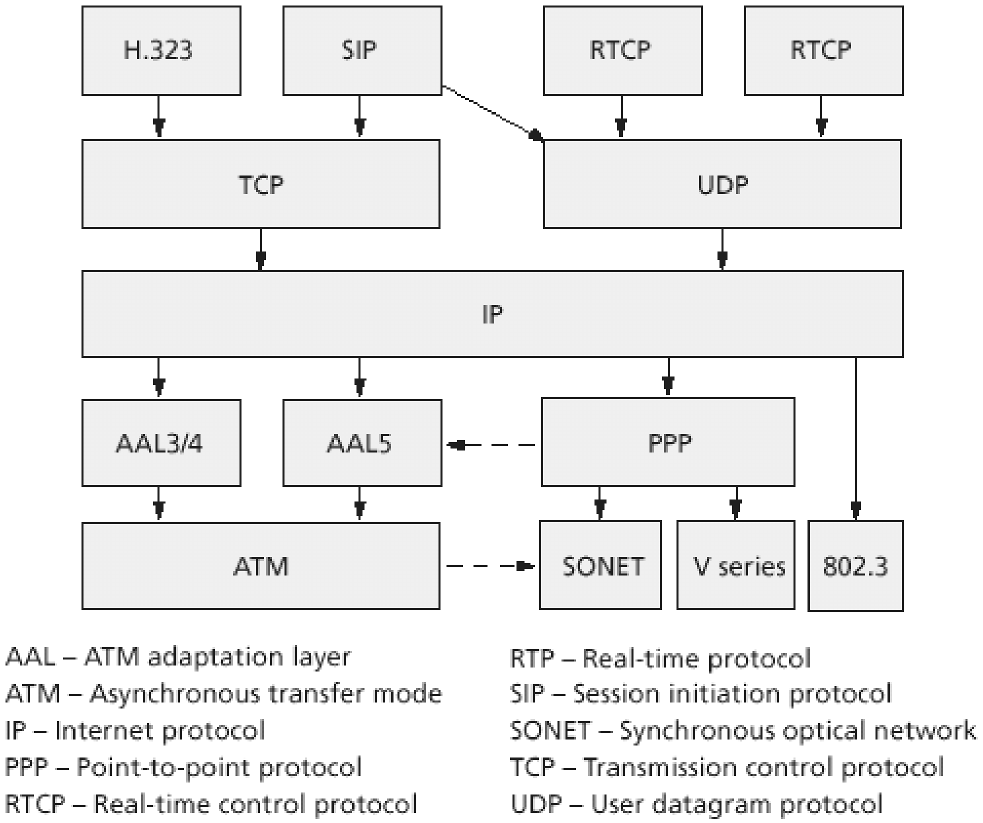

H.323
SIP
RTCP
RTCP
TCP
UDP
IP
AAL3/4
AAL5
PPP
ATM
SONET
V series
802.3
AAL – ATM adaptation layer
ATM – Asynchronous transfer mode
IP – Internet protocol
PPP – Point-to-point protocol
RTCP – Real-time control protocol
RTP – Real-time protocol
SIP – Session initiation protocol
SONET – Synchronous optical network
TCP – Transmission control protocol
UDP – User datagram protocol

Fig9:Internet Telephony Protocol Stack

In this report we will basically discuss the H.323 and SIP families. We also describe some of the supporting protocols.

13. THE H.323 FAMILY:

The H.323 standard provides a foundation for audio, video, and data communications across IP-based networks, including the Internet. By complying to H.323, multimedia products and applications from multiple vendors can interoperate, allowing users to communicate without concern for compatibility. H.323 will be the keystone for LAN-based products for consumer, business, entertainment, and professional applications.

The H.323 standard is specified by the ITU–T Study Group 16. Version 1 of the H.323 recommendation—visual telephone systems and equipment for LANs that provide a nonguaranteed quality of service (QoS)—was accepted in October 1996. It was, as the name suggests, heavily weighted towards multimedia communications in a LAN environment. Version 1 of the H.323 standard does not provide guaranteed QoS.

The emergence of voice-over–IP (VoIP) applications and IP telephony has paved the way for a revision of the H.323 specification. The absence of a standard for voice over IP resulted in products that were incompatible. With the development of VoIP, new requirements emerged, such as providing communication between a PC–based phone and a phone on a traditional switched circuit network (SCN). Such requirements forced the need for a standard for IP telephony. Version 2 of H.323—packet-based multimedia communications systems—was defined to accommodate these additional requirements and was accepted in January 1998.

New features are being added to the H.323 standard, which will evolve to Version 3 shortly. The features being added include fax-over-packet networks, gatekeeper-gatekeeper communications, and fast-connection mechanisms.

H.323 is one of several videoconferencing recommendations issued by ITU-T. The other recommendations in the series include H.310 for conferencing over broadband ISDN (B-ISDN), H.320 for conferencing over narrowband ISDN, H.321 for conferencing over ATM, H.322 for conferencing over LANs that provide a guaranteed quality of service, and H.324 for conferencing over public switched telephone networks (PSTN). The H.323 standard is designed to allow clients on H.323 networks to communicate with clients on the other videoconferencing networks.

13.a. FEATURES OF H.323:

- **Inter-network interoperability**: H.323 clients are interoperable with switched circuit network (SCN) conferencing clients such as those based on Recommendations H.320 (ISDN), H.321 (ATM), and H.324 (PSTN/Wireless).
- **Heterogeneous client capabilities**: A H.323 client must support audio communication; video and data support is optional. This heterogeneity and flexibility does not make the clients incompatible. During call set-up capabilities are exchanged and communication established based on the lowest common denominator.
- **Audio and video codecs**: H.323 specifies a required audio and video codec. However, there is no restriction on the use of other codecs and two clients can agree on any codec, which is supported by both of them.

- **Management and accounting support**: H.323 calls can be restricted on a network based on the number of calls already in progress, bandwidth limitations, or time restrictions. Using these policies the network manager can manage H.323 traffic. Further, H.323 also provides accounting facilities that can be used for billing purposes.
- **Security**: H.323 provides authentication, integrity, privacy, and non-repudiation support.
- **Supplementary services:** Recommendation H.323 recognizes the huge potential for applications based on IP telephony and multimedia. It provides a basic framework for development of such services. In version 2.0 of H.323, two services -- call transfer and call forwarding -- have been specified.
- **Platform and Application Independence:** H.323 is not tied to any hardware or operating system. H.323-compliant platforms will be available in many sizes and shapes, including video-enabled personal computers, dedicated platforms, IP-enabled telephone handsets, cable TV set-top boxes and turnkey boxes.
- **Multipoint Support:** Although H.323 can support conferences of three or more endpoints without requiring a specialized multipoint control unit; MCU's provide a more powerful and flexible architecture for hosting multipoint conferences. Multipoint capabilities can be included in other components of an H.323 system.
- **Multicast Support:** H.323 supports multicast transport in multipoint conferences. Multicast sends a single packet to a subset of destinations on the network without replication. In contrast, unicast sends multiple point-to-point transmissions, while broadcast sends to all destinations. In unicast or broadcast, the network is used inefficiently as packets are replicated throughout the network. Multicast transmission uses bandwidth more efficiently since all stations in the multicast group read a single data stream.

13.b. H.323 ARCHITECTURE-COMPONENTS:

The H.323 standard specifies four kinds of components, which, when networked together, provide the point-to-point and point-to-multipoint multimedia-communication services:

1. terminals
2. gateways
3. gatekeepers
4. multipoint control units (MCUs)

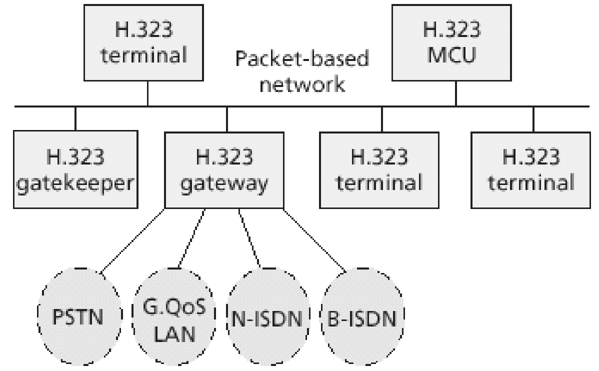


Fig10 : H.323 Components

<u>*TERMINALS:*</u> Used for real-time bi-directional multimedia communications, an H.323 terminal can either be a personal computer (PC) or a stand-alone device, running an H.323 and the multimedia applications. It supports audio communications and can optionally support video or data communications. Because the basic service provided by an H.323 terminal is audio communications, an H.323 terminal plays a key role in IP–telephony services. The primary goal of H.323 is to interwork with other multimedia terminals. H.323 terminals are compatible with H.324 terminals on SCN and wireless networks, H.310 terminals on B–ISDN, H.320 terminals on ISDN, H.321 terminals on B–ISDN, and H.322 terminals on guaranteed QoS LANs. H.323 terminals may be used in multipoint conferences.

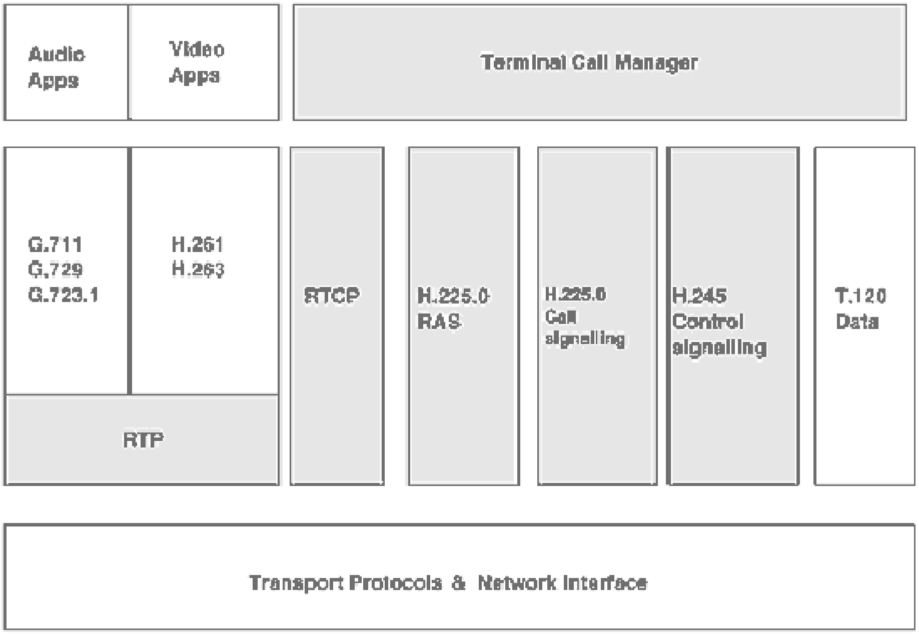


Fig11: H.323 terminal Protocol Stack

H.323 terminals must support the following:

- H.245 for exchanging terminal capabilities and creation of media channels
- H.225 for call signaling and call setup
- RAS for registration and other admission control with a gatekeeper
- RTP/RTCP for sequencing audio and video packets
- H.323 terminals must also support the G.711 audio CODEC. Optional components in an H.323 terminal are video CODECs, T.120 data-conferencing protocols, and MCU capabilities.

*GATEWAY***:** A gateway provides translation of protocols for call setup and release, conversion of media formats between different networks, and the transfer of information between H.323 and non–H.323 networks. An application of the H.323 gateway is in IP telephony, where the H.323 gateway connects an IP network and SCN network.

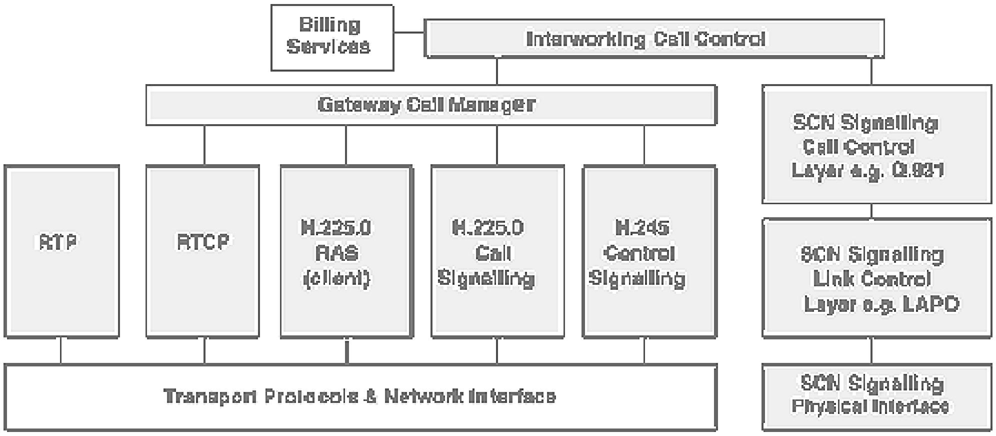

Billing Services
Interworking Call Control
Gateway Call Manager
SCN Signalling Call Control Layer e.g. Q.931
RTP
RTCP
H.225.0 RAS (client)
H.225.0 Call Signalling
H.245 Control Signalling
SCN Signalling Link Control Layer e.g. LAPD
Transport Protocols & Network Interface
SCN Signalling Physical Interface

Fig12: H.323 Gateway Protocol Stack

On the H.323 side, a gateway runs H.245 control signaling for exchanging capabilities, H.225 call signaling for call setup and release, and H.225 RAS for registration with the gatekeeper. On the SCN side, a gateway runs SCN–specific protocols (e.g., ISDN and SS7 protocols).

Terminals communicate with gateways using the H.245 control-signaling protocol and H.225 call-signaling protocol. The gateway translates these protocols in a transparent fashion to the respective counterparts on the non–H.323 network and vice versa. The gateway also performs call setup and clearing on both the H.323–network side and the non–H.323–network side. Translation between audio, video, and data formats may also be performed by the gateway. Audio and video translation may not be required if both terminal types find a common communications mode. The gateway has the characteristics of both an H.323 terminal on the H.323 network and the other terminal on the non–H.323 network it connects.

Gatekeepers are aware of which endpoints are gateways because this is indicated when the terminals and gateways register with the gatekeeper. A gateway may be able to support several simultaneous calls between the H.323 and non–H.323 networks. In addition, a gateway may connect an H.323 network to a non–H.323 network. A gateway is a logical component of H.323 and can be implemented as part of a gatekeeper or an MCU.

GATEKEEPER:

Gatekeepers provide call-control services for H.323 endpoints, such as address translation and bandwidth management as defined within RAS. Gatekeepers in H.323 networks are optional. If they are present in a network, however, terminals and gateways must use their services. The H.323 standards both define mandatory services that the gatekeeper must provide and specify other optional functionality that it can provide.

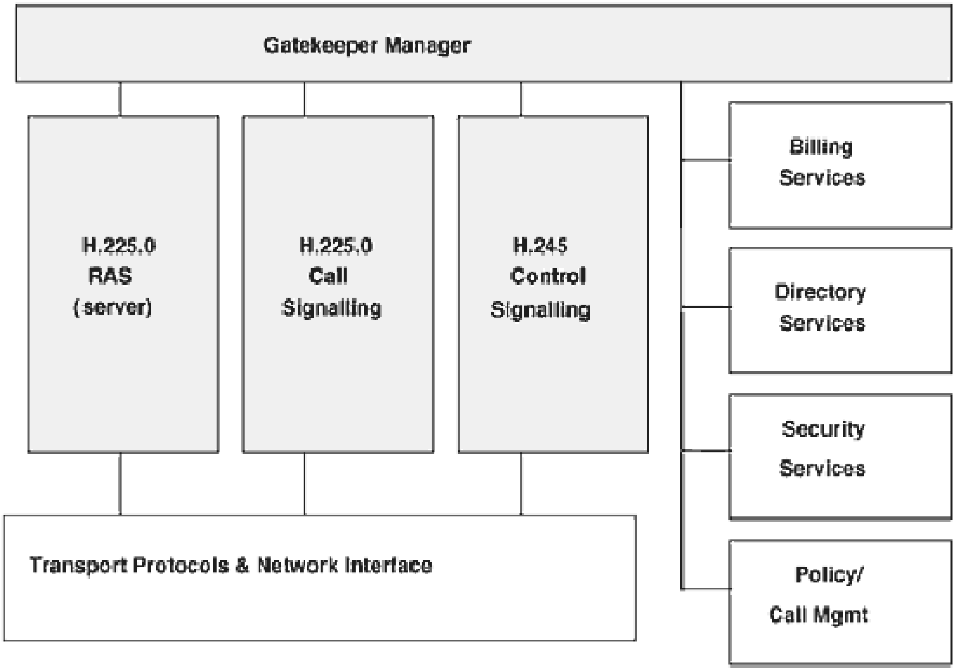


Fig13: H.323 Gatekeeper characteristics

The mandatory functions include:

- Address translation
- Admissions control
- Bandwidth control
- Zone management

The optional features include:

- Call-control Signaling
- Call authorization
- Call management
- Call signaling routing

H.323 networks that do not have gatekeepers may not have these capabilities, but H.323 networks that contain IP–telephony gateways should also contain a gatekeeper to translate incoming E.164 telephone addresses into transport addresses. A gatekeeper is a logical component of H.323 but can be implemented as part of a gateway or MCU.

MULTIPOINT CONTROL UNIT:

A multipoint control unit enables conferencing between three or more endpoints. It consists of a mandatory multipoint controller (MC) and zero or more multipoint processors (MP). Although the MCU is a separate logical unit it may be combined into a terminal, gateway, or gatekeeper. The MCU is an optional component of an H.323-enabled network. The multipoint controller provides a centralized location for multipoint call setup. Call and control signaling are routed through the MC so that endpoints capabilities can be determined and communication parameters negotiated. A MC may also be used in a point-to-point call, which can later be extended into a multipoint conference. Another useful job of the MC is to determine whether to unicast or multicast the audio and video streams depending on the capability of the underlying network and the topology of the multipoint conference. The multipoint processor handles the mixing, switching, and processing of the audio, video, and data streams among the conference endpoints. The MCU is required in a centralized multipoint conference where each terminal establishes a point-to-point connection with the MCU. The MCU determines the capabilities of each terminal and sends each a mixed media stream. In the decentralized model of multipoint conferencing, a MC ensures communication compatibility but the media streams are multicast and the mixing is performed at each terminal.

13.c H.323 ARCHITECTURE-PROTOCOLS:

The protocols specified by H.323 are listed below. H.323 is independent of the packet network and the transport protocols over which it runs and does not specify them.

- Audio CODECs
- Video CODECs
- H.225 registration, admission, and status (RAS)
- H.225 call signaling
- H.245 control signaling
- T.120 for multimedia conferencing
- H.450 Series for supplementary services
- H.235 for Security and encryption for H-Series
- Real-time transfer protocol (RTP)
- Real-time control protocol (RTCP)

AUDIO CODEC:

An audio CODEC encodes the audio signal from the microphone for transmission on the transmitting H.323 terminal and decodes the received audio code that is sent to the speaker on the receiving H.323 terminal. Because audio is the minimum service provided by the H.323 standard, all H.323 terminals must have at least one audio CODEC support, as specified in the ITU–T G.711 recommendation (audio coding at 64 kbps). Additional audio CODEC recommendations such as G.722 (64, 56, and 48 kbps), G.723.1 (5.3 and 6.3 kbps), G.728 (16 kbps), and G.729 (8 kbps) may also be supported.

VIDEO CODECS:

Video communication is bandwidth intensive and bursty in nature. Therefore, efficient compression and decompression techniques are essential for good performance. Recommendation H.323 specifies two video codecs: H.261 and H.263. However, H.323 clients are not limited to these codecs only. Other codecs can be used provided both terminals agree on and support it. Video support in H.323 terminals and MCUs is optional. The H.261 codec produces video transmission for channels with bandwidths p x 64 kb/s where p can range from 1 to 30. The discrete cosine transform (DCT) is used for compression together with quantization and motion compensation. H.261 supports two video formats. The common intermediate format (CIF) has a resolution of 352 x 288 pixels while the quarter common intermediate format (QCIF) has a resolution of 176 x 144 pixels. The CIF format support is optional. The H.263 codec is designed for low bit rate transmission without loss of quality. It uses the same DCT coding with quantization for compression but this is accompanied by both motion estimation and prediction. Additional coding efficiency parameters have also been defined which can be negotiated between the terminals. The result is better quality at a lower bit rate. The video formats supported by H.263 are: sub-QCIF (128 x 96), QCIF (176 x 144), CIF (352 x 244), 4CIF (702 x 576), and 16CIF (1408 x 1152). The first three are required while the remaining two are optional. Through the QCIF format H.263 is compatible with H.261. The quality of video transmission strongly depends on compression techniques. Active work is on-going in the development of more efficient codecs like MPEG-4 and MPEG-7. The architecture of H.323 is designed to allow the incorporation of new codecs as they become available.

H.225 RAS:

The H.225 RAS is used between H.323 endpoints (terminals and gateways) and gatekeepers for the following:

- gatekeeper discovery (GRQ)
- endpoint registration
- endpoint location
- admission control
- access tokens

The RAS messages are carried on a *RAS channel that is unreliable*. Hence, RAS message exchange may be associated with timeouts and retry counts.

- **Gatekeeper discovery:** The gatekeeper discovery process is used by the H.323 endpoints to determine the gatekeeper with which the endpoint must register. The gatekeeper discovery can be done statically or dynamically. In static discovery, the endpoint knows the transport address of its gatekeeper a priori. In the dynamic method of gatekeeper discovery, the endpoint multicasts a GRQ message on the gatekeeper's discovery multicast address. One or more gatekeepers may respond with a GCF message.

- **Endpoint Registration**: Registration is a process used by the endpoints to join a zone and inform the gatekeeper of the zone's transport and alias addresses. All endpoints register with a gatekeeper as part of their configuration.
- **Endpoint Location:** Endpoint location is a process by which the transport address of an endpoint is determined and given its alias name or E.164 address (telephone number).
- **Other Control:** The RAS channel is used for other kinds of control mechanisms, such as admission control, to restrict the entry of an endpoint into a zone, bandwidth control, and disengagement control, where an endpoint is disassociated from a gatekeeper and its zone.

H.225 CALL SIGNALING:

H.225 call signaling is used to set up connections between H.323 endpoints (terminals and gateways), over which the real-time data can be transported. Call signaling involves the exchange of H.225 protocol messages over a *reliable call-signaling channel*. For example, H.225 protocol messages are carried over TCP in an IP–based H.323 network. H.225 messages are exchanged between the endpoints if there is no gatekeeper in the H.323 network. When a gatekeeper exists in the network, the H.225 messages are exchanged either directly between the endpoints or between the endpoints after being routed through the gatekeeper. The first case is direct call signaling. The second case is called gatekeeper-routed call signaling. The method chosen is decided by the gatekeeper during RAS–admission message exchange.

- **Gatekeeper-Routed Call Signaling**: The admission messages are exchanged between endpoints and the gatekeeper on RAS channels. The gatekeeper receives the call-signaling messages on the call-signaling channel from one endpoint and routes them to the other endpoint on the call-signaling channel of the other endpoint.
- **Direct Call Signaling**: During the admission confirmation, the gatekeeper indicates that the endpoints can exchange call-signaling messages directly. The endpoints exchange the call signaling on the call-signaling channel.

H.245 MEDIA CONTROL SIGNALING:

The flexibility of H.323 requires that endpoints negotiate to determine compatible settings before audio, video, and/or data communication links can be established. H.245 uses control messages and commands that are exchanged during the call to inform and instruct. The implementation of H.245 control is mandatory in all endpoints. H.245 provides the following media control functionalities:

- **Capability exchange**: H.323 allows endpoints to have different receive and send capabilities. Each endpoint records its receiving and sending capabilities (e.g. media types, codecs, bit rates, etc) in a message and sends it to the other endpoint(s).
- **Opening and closing of logical channels**: H.323 audio and video logical channels are uni-directional end-to-end links (or multipoint links in the case of multipoint conferencing). Data channels are bi-directional. A separate channel is

needed for audio, video, and data communication. H.245 messages control the opening and closing of such channels. H.245 control messages use logical channel 0 which is always open.

- **Flow control messages**: These messages provide feedback to the endpoints when communication problems are encountered.
- **Other commands and messages**: Several other commands and messages may be used during a call like a command to set the codec at the receiving endpoint when the sending endpoint switches its codec.

H.245 control messages may also be routed through a gatekeeper if one exists.

T.120 DATA CONFERENCING:

Real-time data conferencing capability is required for activities such as application sharing, whiteboard sharing, file transfer, fax transmission, and instant messaging. Recommendation T.120 provides this optional capability to H.323. T.120 is a real-time data communication protocol designed specifically for conferencing needs. Like H.323, Recommendation T.120 is an umbrella for a set of standards that enable the real-time sharing of specific applications data among several clients across different networks. T.120 provides several advantages over regular data transmission such as:

- **Multipoint conferencing support**: T.120 supports multipoint data delivery, which enables group collaboration activities. The MCU handles the mixing and switching of data in a similar manner to that used for video and audio.
- **Network and platform independence**: T.120 operates on top of the transport layer of the underlying network. As such, it is transparent and independent of the network hardware and software.
- **Interoperability**: T.120 is referenced in all the H.32X conferencing standards. This cross-referencing, together with the network and platform independence, ensures a high degree of interoperability at the application level.
- **Multicast support**: T.120 supports multicast of data streams in multicast-capable networks. This support is flexible with mixed unicast and multicast also possible during a conference.
- **Other benefits**: T.120 provides error correction capability on top of the network transport ensuring reliable delivery. In general, T.120 has a scalable and extendible architecture with provisions for the addition of new applications that take advantage of real-time reliable and efficient data delivery among a group of collaborators.

SUPPLEMENTARY SERVICES:

Recommendation H.323 bridges traditional telephone networks with media rich packet-based networks. There is a huge potential for new services and applications that take advantage of the capabilities of both networks. These services can range from value-added traditional telephone services such as call transfer and diversion to new services such as integrated messaging (e-mail, voice mail, fax, instant messaging, etc). H.323 provides a flexible architecture for supplementary services through the H.450.x series of recommendations.

H.450: adopts a hierarchical architecture for the development of new services. A general framework for supplementary services is defined in H.450.1. Several basic services are specified in H.450.2 and above. New services can be developed by end-users by combining zero or more of the basic services. However, all services must use the control mechanisms defined in H.450.1.

H.450.1: provides an essential mechanism for end-to-end control signaling between peer service entities. The H.450.1 protocol is based on the QSIG protocol developed by International Organization for Standardization (ISO) for private ISDN networks. QSIG is the most common services control mechanism employed in call centers and PBXs (private branch exchange). Using QSIG as a basis for H.323 supplementary services provides several advantages such as:

- Interoperability with QSIG based networks
- Existence of several basic services models (from ISDN)
- Extendibility and flexibility of QSIG
- Existence of implementation knowledge base

Two supplementary services have been ratified by ITU-T. These are call transfer (H.450.2) and call diversion (H.450.3). Other services that are being developed include call hold (H.450.4), call park/pickup (H.450.5), call waiting (H.450.6), message waiting (H.450.7), name identification (H.450.8), and call completion on busy subscriber (H.450.9).

H.235 SECURITY:

In development for months, the H.235 standard addresses four general issues when dealing with security, Authentication, Integrity, Privacy, and non-Repudiation. Authentication is a mechanism to make sure that the endpoints participating in the conference are really who they say they are. Integrity provides a means to validate that the data within a packet is indeed an unchanged representation of the data. Privacy/Confidentiality is provided by encryption and decryption mechanisms that hide the data from eavesdroppers so that if it is intercepted, it cannot be viewed. Non-Repudiation is a means of protection against someone denying that they participated in a conference when you know they were there.

Authentication is provided by admission control of endpoints. This is handled by the gatekeeper that administers the zone. Data integrity and privacy is provided by encryption. Non-repudiation ensures that no endpoint can deny that it participated in a call. This is also provided by gatekeeper services.

To implement these security service H.235 can use existing standards such as IP Security (IPSec) and Transport Layer Security (TLS).

13.d. CONNECTION PROCEDURES:

This module graphically describes the steps involved in creating an H.323 call, establishing media communication, and releasing the call. The example network contains two H.323 terminals (end lines) connected to a gatekeeper (central line). Direct call signaling is assumed. It is also assumed that the media stream uses RTP encapsulation.

- **Call Establishment**

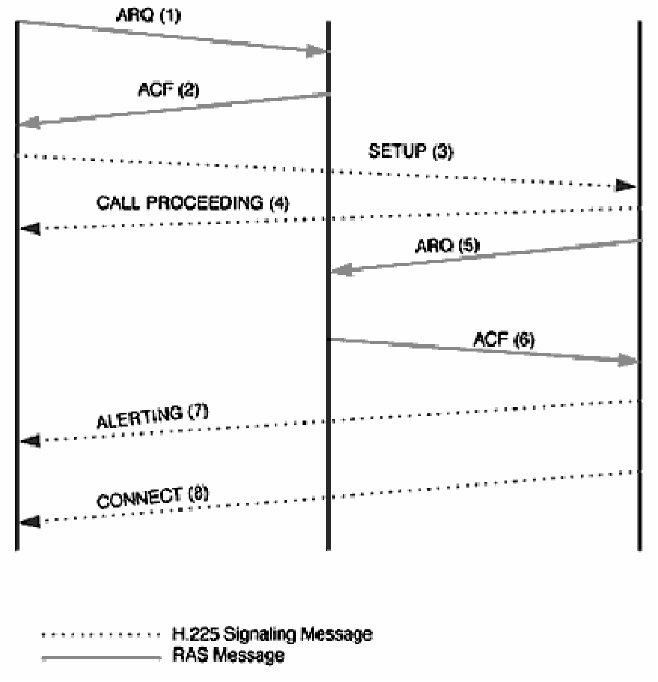


Fig14: Call establishment using H.323

- **Call Control Signaling:**

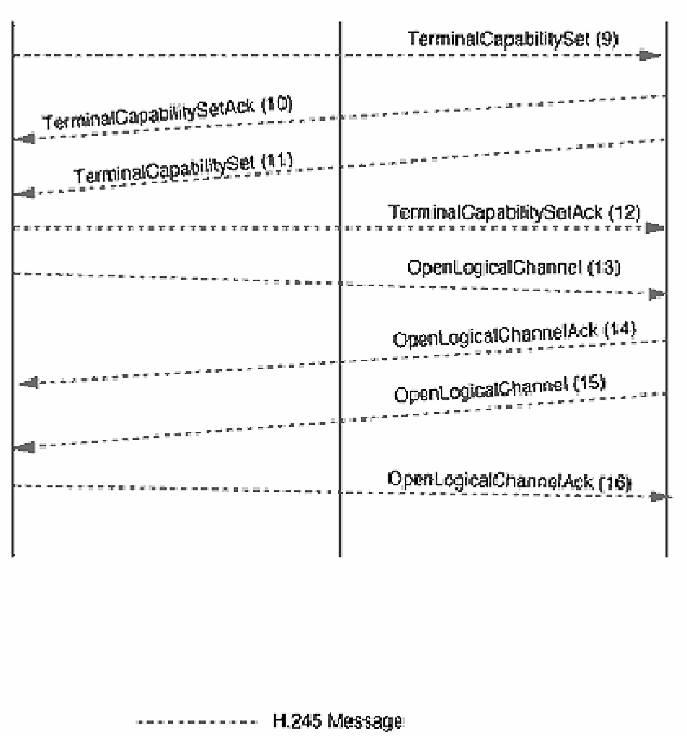

TerminalCapabilitySet (9)
TerminalCapabilitySetAck (10)
TerminalCapabilitySet (11)
TerminalCapabilitySetAck (12)
OpenLogicalChannel (13)
OpenLogicalChannelAck (14)
OpenLogicalChannel (15)
OpenLogicalChannelAck (16)
H.245 Message

Fig15: Call Control Signaling using H.323

- **Media Stream Flows:**

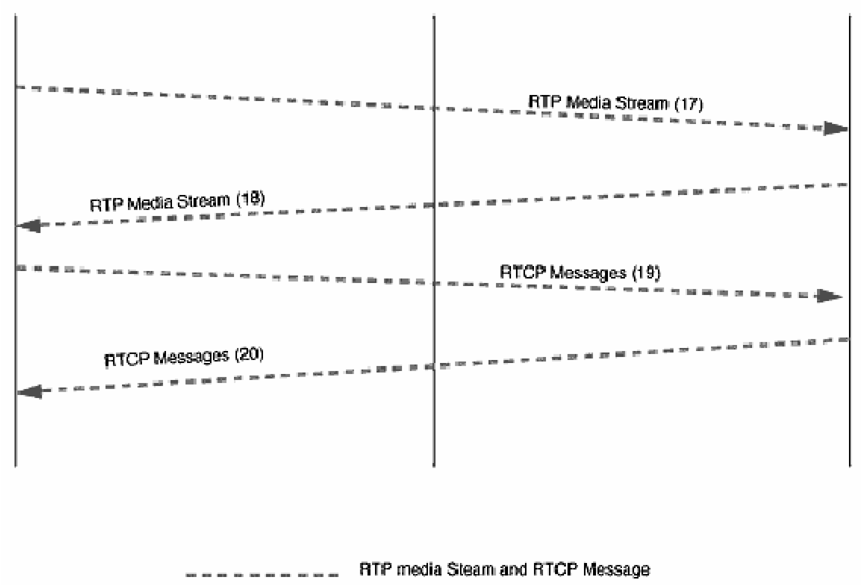


Fig16: Media Stream flows using H.323

- **Call Release:**

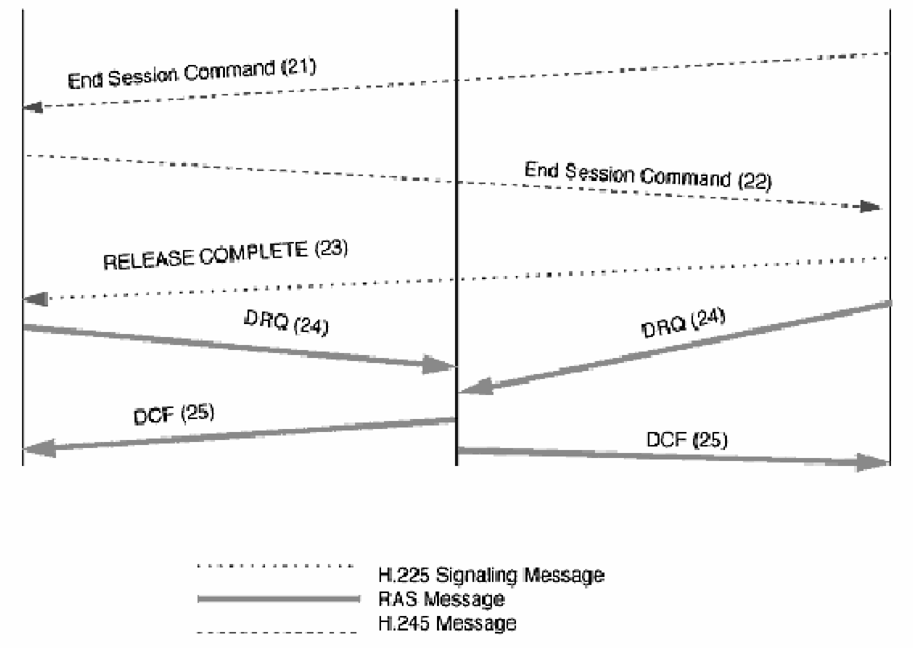


Fig17: Call Release using H.323

We have described H.323 as a control & signaling protocol. We will describe RTP and RTCP as real time VoIP support protocols in a later section.

14. SIP FAMILY:

This is the IETF's standard for establishing VOIP connections. The Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying and terminating sessions with one or more participants. These sessions include Internet multimedia conferences, Internet telephone calls and multimedia distribution. Members in a session can communicate via multicast or via a mesh of unicast relations, or a combination of these.

The architecture of SIP is similar to that of HTTP (client-server protocol). Requests are generated by the client and sent to the server. The server processes the requests and then sends a response to the client. A request and the responses for that request make a transaction. SIP has INVITE and ACK messages, which define the process of opening a reliable channel over which call control messages may be passed. SIP makes minimal assumptions about the underlying transport protocol. This protocol itself provides reliability and does not depend on TCP for reliability. SIP depends on the Session Description Protocol (SDP) for carrying out the negotiation for codec identification. SIP supports session descriptions that allow participants to agree on a set of compatible media types. It also supports user mobility by proxying and redirecting requests to the user's current location. The services that SIP provide include [RFC2543]:

- User Location: determination of the end system to be used for communication
- Call Setup: ringing and establishing call parameters at both called and calling party
- User Availability: determination of the willingness of the called party to engage in communications
- User Capabilities: determination of the media and media parameters to be used
- Call handling: the transfer and termination of calls

SIP requires support from a host of other protocols as shown below:

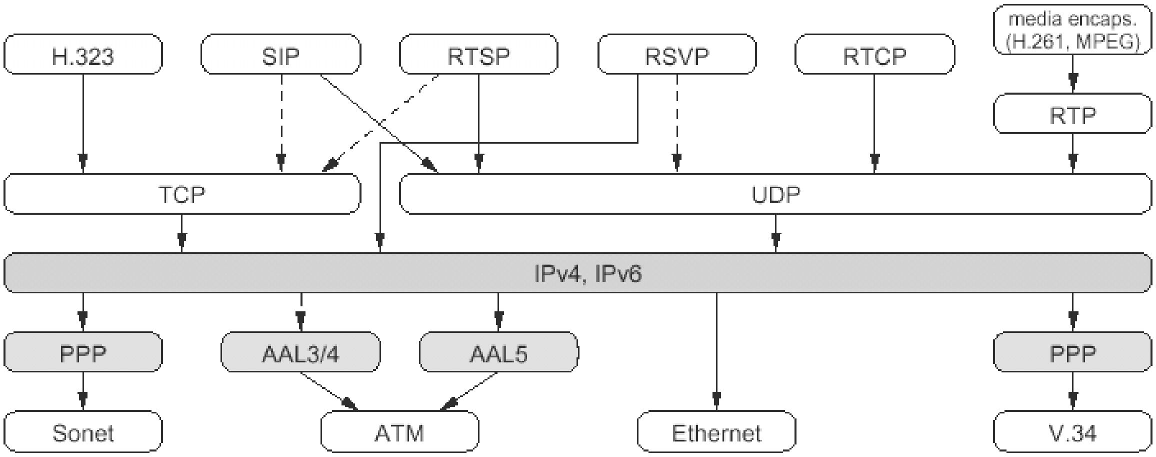


Fig18: SIP protocol architecture

14.a. SIP DEFINITIONS:

- **Client**: An application program that sends SIP requests. Clients may or may not interact directly with a human user. User agents and proxies contain clients (and servers).
- **Server**: A server is an application program that accepts requests in order to service requests and sends back responses to those requests. Servers are proxy, redirect or user agent servers or registrars.
- **Call**: A call consists of all participants in a conference invited by a common source. A SIP call is identified by a globally unique call-id. Thus, if a user is, for example, invited to the same multicast session by several people, each of these invitations will be a unique call. A point-to-point Internet telephony conversation maps into a single SIP call. In a multiparty conference unit (MCU) based call-in conference, each participant uses a separate call to invite himself to the MCU.
- **Call leg**: A call leg is identified by the combination of the Call-ID header field and the addr-spec and tag of the To and From header fields. Within the same Call-ID, requests with From A and To value B belong to the same call leg as the requests in the opposite direction, i.e., From B and To A.
- **Session**: From the SDP specification: ``A multimedia session is a set of multimedia senders and receivers and the data streams flowing from senders to receivers. A multimedia conference is an example of a multimedia session.". As defined, a callee can be invited several times, by different calls, to the same session. If SDP is used, a session is defined by the concatenation of the user name, session id, network type, address type and address elements in the origin field.
- **(SIP) transaction**: A SIP transaction occurs between a client and a server and comprises all messages from the first request sent from the client to the server up to a final (non-1xx) response sent from the server to the client. A transaction is identified by the CSeq sequence number within a single call leg.

14.b. SIP ARCHITECTURE: COMPONENTS:

- **User agent**: A user agent is an application, which contains both a user agent client and user agent server. The user agent client is the calling user agent (outgoing calls) and the user agent server is the called user agent (incoming calls).
- **Proxy server**: An intermediary program that acts both a server and a client for the purpose of making or forwarding requests on behalf of other clients.
- **Redirect server**: A redirect server is a server that accepts a SIP request, maps the address into zero or more new addresses to the client. Unlike the proxy server, it itself does not forward the message. Instead, it send back the forwarding address to the client asking him to resend the massage to the new address.

- **Location server**: A location server is used by a SIP redirect or proxy server to obtain information about a callee's possible location or locations.
- **Outbound proxy**: A proxy that is located near the originator of requests. It receives all outgoing requests from a particular UAC; including those requests whose Request-URLs identify a host other than the outbound proxy. The outbound proxy sends these requests, after any local processing, to the address indicated in the request-URI.
- **Registrar**: A registrar is a server that accepts REGISTER requests. A registrar is typically co-located with a proxy or redirect server and MAY make its information available.

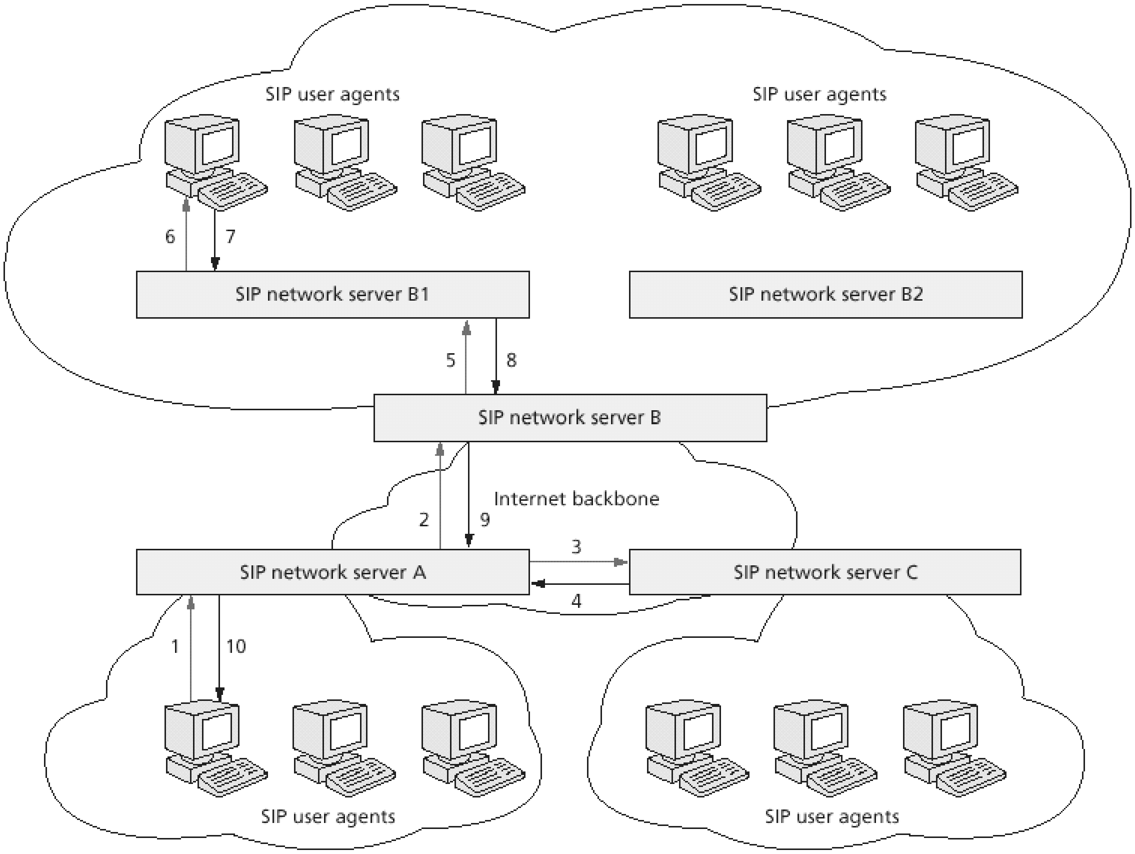


Fig19: Typical SIP architecture

These are all logical separation of functionality and are not necessarily distinct physical implementations.

14.c. SIP OPERATION:

Callers and callees are identified by SIP addresses. When making a SIP call, a caller first needs to locate the appropriate server and send it a request. The caller can either directly reach the callee or indirectly through the redirect servers. The Call ID field in the SIP message header uniquely identifies the calls.

SIP Addressing and locating a SIP server: To be invited and identified, the called party has to be named. SIP uses an e-mail type identifier of the form user@domain, user@host, user@IP_address or phonenumber@gateway because it is the most common form of addressing in the Internet. Using an e-mail address as an SIP address provides a scalable means by which a UAC can deliver a request to an SIP server, which likely knows how to forward the request to the final callee-the DNS (Domain name server). By performing a series of DNS lookups-such as searching for service (SRV), mail exchange (MX) and address (A) records-the caller can determine the address of a server that has naming authority for all users within a particular domain. The e-mail type address also allows SIP addresses to be easily transformed into URIs (Uniform resource identifiers) such as ***sip:a.kundaje@vjti.com***. The advantage of doing so is that they can be easily embedded in Web pages, so that clicking on the link initiates a call to that address, similar to the mailto: URL.

A SIP network server may either proxy or redirect the call to additional servers, eventually arriving at one that definitively knows the IP address where the user can be contacted. The process of determining the next-hop address is known as next-hop routing. As a result of its next-hop routing decision, a SIP network server may determine that several next-hop servers may be able to contact the user. In these cases, SIP allows a proxy server to fork an incoming request, sending it in parallel to multiple next-hop servers. Under normal conditions, each server will generate a response; SIP has rules for merging and returning the responses to the UAC.

SIP Transaction: Once the host part of the Request URI has been resolved to a SIP server, the client can send requests to that server. A request together with the responses triggered by that request make up a SIP transaction. The requests can be sent through reliable TCP or through unreliable UDP.

If a reliable stream protocol is used, request and responses within a single SIP transaction are carried over the same connection. Several SIP requests from the same client to the same server may use the same connection or may use a new connection for each request. If a client sends the request via a unicast datagram protocol such as UDP, the receiving user agent directs the response according to the information contained in the *Via header* fields. Each proxy server in the forward path of the request forwards the response using these Via header fields. For datagram protocols, reliability is achieved using retransmission

SIP Invitation: A successful SIP invitation consists of two requests: an INVITE followed by ACK. The INVITE request asks the callee to join a particular conference or establish a two party conversation. After the callee has agreed to participate in the call, the caller confirms that it has received that response by sending an ACK request. The INVITE request contains a session description that provides the called party with enough information to join the session. If the callee wishes to accept the call, it responds to the invitation by returning a response with a similar session description. After the callee has agreed to participate in the call, the caller confirms that it has received that response by sending an ACK request.

Locating a User: A callee may keep changing its position with time. These locations can be dynamically registered with the SIP server. When the SIP server is queried about the location of a callee, it returns a list of possible locations. A Location Server in the SIP system actually generates the list and passes it to the SIP server.

Changing an Existing Session: Sometimes we may need to change the parameters of an existing session. This is done by re-issuing the INVITE message using the same Call ID with a new body to convey the new information. This re INVITE must have a higher CSeq than any previous request from the client to the server.

14.d. SIP PROTOCOL- PROPERTIES:

- **Minimal State:** A single conference session or call involves one or more SIP request-response transactions. Each SIP transaction can take a different path through servers in the network. In a typical call, the request is an INVITE, which may traverse many network servers on its way to the callee. The response to the INVITE contains a reach address that can be used by the UAC to send subsequent transactions directly to the UAS. Because SIP network servers need not maintain the call state once a transaction is complete, a SIP server has no recollection of the caller or callee. This characteristic facilitates the scalability and reliability of a SIP server, because it can crash and recover without affecting any of the calls initiated through it. The duration of and amount of state maintained at a server are small compared to those in the global switched telephone network (GSTN), where a switch must maintain the call state for the entire duration of the call. However, a server that wishes the call state may do so. Through SIP's Route and Record-Route header fields, each proxy individually can insist on being the signaling path for subsequent transactions. Furthermore, the proxy can change its mind and remove itself from the signaling path later on. A SIP network server is not required to be stateful even for the duration of transaction. A proxy or redirect server can be completely stateless. After it receives a request, it either generates the response or proxies the response and forgets everything. The messages themselves contain all the information needed for a stateless proxy to correctly process and route them. This behavior aligns nicely with the Internet datagram architecture, whose packets contain enough information to be individually routed. Also, a stateful proxy can become stateless any time without affecting the operation. The administrator decides on a call-by-call basis, whether a proxy will be stateful or stateless. This flexibility allows large, central SIP servers to be stateless. It also allows smaller, localized servers to be stateful.
- **Lower-Layer-Protocol Neutral**: SIP makes minimal assumptions about the underlying transport and network-layer protocols. The lower layer can provide either a packet or a byte stream service, with reliable or unreliable service. In an Internet context, SIP is able to utilize both UDP and TCP as transport protocols, among others. UDP allows the application to more carefully control the timing of messages and their retransmission, to perform parallel searches without requiring TCP connection state for each outstanding request, and to use multicast. Routers can more readily snoop SIP UDP packets. TCP allows easier passage through

existing firewalls. When TCP is used, SIP can use one or more connections to attempt to contact a user or to modify parameters of an existing conference. Different SIP requests for the same SIP call may use different TCP connections or a single persistent connection, as appropriate. SIP may be used with Internet protocols as well as with protocols such as ATM AAL5, IPX, frame relay or X.25. User agents should implement both UDP and TCP transport. Proxy, registrar, and redirect servers must implement both UDP and TCP transport.

- **Text-Based**: SIP is text based, using ISO 10646 in UTF-8 encoding throughout. This allows easy implementation in languages such as Java, Tcl and Perl, allows easy debugging, and most importantly, makes SIP flexible and extensible. As SIP is used for initiating multimedia conferences rather than delivering media data, it is believed that the additional overhead of using a text-based protocol is not significant.

14.e. SIP MESSAGES:

SIP is a text-based protocol. The message syntax is and header fields are identical to HTTP/1.1 specification. A SIP message is either a request or response. A request is generated by a client and a response is generated by a server. Both Request and Response messages use the generic-message format of RFC 822 for transferring the body of the message. Both types of messages consist of a start- line, one or more header fields, an empty line (i.e., a line with nothing preceding the carriage-return line-feed (CRLF)) indicating the end of the header fields, and an optional message body. The Request message format is shown below:

Request = *Request-Line*
**(general-header | request-header | entity-header)*
CRLF
[message-body]
Request-Line = *Method SP Request-URI SP SIP-Version CRLF*
Request-URI = *SIP-URL j absoluteURI*
SIP-Version = *"SIP/2.0"*
SP= *“ “*
Method = *"INVITE" | "ACK" | "OPTIONS" | "BYE" | "CANCEL" | "REGISTER" | extension-method*
extension-method = *token*

HEADER FIELDS:

SIP messages use header fields to specify specific things about the participants, the path and so on. SIP header fields are similar to HTTP header fields in both syntax and semantics. The order in which header fields appears is generally of no importance, except with the exception of the header fields that are hop-to-hop. These must appear before any header fields, which are end-to-end. Some of SIP's header fields are used in all messages and others only when necessary. An application containing SIP does not need to understand all header fields, though it is desirable. If a SIP participant does not

understand a header it simply ignores it. The total amount of SIP header fields is 37, which can be divided into four groups.
1. General header fields apply to both request and response messages.
2. Entity header fields define information about the message body or if no body is present, about the resource identified by the request.
3. Request header fields allow the client to pass additional information about the request, and about the client itself, to the server.
4. Response header fields give information about the server and about further access to the resource identified by the Request-URI.

Request-headers
Accept
Accept-Encoding
Accept-Language
Call-Id
Contact
Case
Date
Encryption
Expires
From
Record-Route
Timestamp
To
Via

General-headers
Authorization
Contact
Hide
Max-Forwards
Organization
Priority
Proxy-Authorization
Proxy-Require
Route
Require
Response-Key
Subject
User-Agent

Entity-headers
Content-Encoding
Content-Length
Content-Type
Server
Unsupported
Warning
WWW-Authenticate

Response-headers
Allow
Proxy-Authenticate
Retry-After

For further information about these header fields one must refer to ietf-sip-rfc 2543.

<u>*SIP METHODS:*</u>

The SIP methods are listed below. The method token is case-sensitive.

- **INVITE:** The INVITE request invites a user or service to participate in a session. The message body contains a description of the session.
- **ACK:** The ACK request confirms that the caller has received a final response to an INVITE request, acknowledge a successful response.
- **OPTIONS:** This method handles capability information and discovers the capabilities of the receiver.

- **BYE**: The BYE request terminates a call or call request, either a caller or callee can send a BYE request.
- **CANCEL**: The CANCEL request cancels a pending request, but does not affect a completed request. In other words the CANCEL method terminates incomplete call requests.
- **REGISTER**: The REGISTER method is used as a simple location service that registers the current location of a user. The REGISTER method is also needed when there are several SIP servers on a single host. In that case only one of the servers can use the default port number.
- **INFO**: represents mid-call information such as DTMF tones or ISUP.
- **PRACK**: is a provisional acknowledgement

Some other methods include COMET, SUBSCRIBE & NOTIFY. Methods that are not supported by a proxy or redirect server, are treated as an OPTIONS method and forwarded accordingly. Methods not supported by a user agent server or registrar cause a 501 Server Failure.
The header fields may be followed with a message body separated with an empty line. For ACK, INVITE and OPTIONS the message body is always a session description. The *Content-Type* must give the Internet media type. In the example below the message body type is the Session Description Protocol.

SIP RESPONSE CODES:

After receiving and interpreting a request message, the recipient responds with a SIP response message. The SIP response message contains Status-Line, header-field and a message body. The Status-Line consists of SIP-version, Status-Code and Reason- Phrase. The reason-phrase is a short textual description of the Status-Code.
The response format is as shown below:

Response = *Status-Line*
**(general-header | response-header | entity-header)*
CRLF
[message-body]
Status-Line = *SIP-version SP Status-Code SP Reason-Phrase CRLF*
Status-Code = *Informational | Success | Redirection | Client-Error | Server-Error | Global-Failure | extension-code*
extension-code = *3DIGIT*
Reason-Phrase = **<TEXT-UTF8, excluding CR, LF>*

The Status-Code is a 3-digit integer code; there are six different types of the Status-Code.

1xx: Informational: Request received, continuing to process the request.
180 RINGING.
2xx: Success: The action was successfully received, understood, and accepted.
200 OK.
3xx: Redirection: Further action needs to be taken in order to complete the request.

302 MOVED TEMPORARILY.
4xx: Client Error: The request contains bad syntax or cannot be fulfilled at this server.
404 NOT FOUND.
5xx: Server Error: The server failed to fulfill an apparently valid request.
501 NOT IMPLEMENTED.
6xx: Global Failure: The request cannot be fulfilled at any server.
600 BUSY EVERYWHERE.

<u>14.f. EXAMPLE: TWO-PARTY CALL:</u>

In the example below, Bell calls Watson. Bell indicates that he can receive RTP audio codings 0 (PCMU), 3 (GSM), 4 (G.723) and 5 (DVI4). The parts in italics indicate message body.

C->S: INVITE sip:watson@boston.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP kton.bell-tel.com
From: A. Bell <sip:a.g.bell@bell-tel.com>;tag=3
To: T. Watson <sip:watson@bell-tel.com>
Call-ID: 662606876@kton.bell-tel.com
CSeq: 1 INVITE
Contact: <sip:a.g.bell@kton.bell-tel.com>
Subject: Mr. Watson, come here.
Content-Type: application/sdp
Content-Length: ...
v=0
o=bell 53655765 2353687637 IN IP4 128.3.4.5
s=Mr. Watson, come here.
t=3149328600 0
c=IN IP4 kton.bell-tel.com
m=audio 3456 RTP/AVP 0 3 4 5
a=rtpmap:0 PCMU/8000
a=rtpmap:3 GSM/8000
a=rtpmap:4 G723/8000
a=rtpmap:5 DVI4/8000

S->C: SIP/2.0 100 Trying
Via: SIP/2.0/UDP kton.bell-tel.com
From: A. Bell <sip:a.g.bell@bell-tel.com>;tag=3
To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
Call-ID: 662606876@kton.bell-tel.com
CSeq: 1 INVITE
Content-Length: 0

```
S->C:  SIP/2.0 180 Ringing
       Via: SIP/2.0/UDP kton.bell-tel.com
       From: A. Bell <sip:a.g.bell@bell-tel.com>;tag=3
       To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
       Call-ID: 662606876@kton.bell-tel.com
       CSeq: 1 INVITE
       Content-Length: 0

S->C:  SIP/2.0 182 Queued, 2 callers ahead
       Via: SIP/2.0/UDP kton.bell-tel.com
       From: A. Bell <sip:a.g.bell@bell-tel.com>;tag=3
       To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
       Call-ID: 662606876@kton.bell-tel.com
       CSeq: 1 INVITE
       Content-Length: 0

S->C:  SIP/2.0 182 Queued, 1 caller ahead
       Via: SIP/2.0/UDP kton.bell-tel.com
       From: A. Bell <sip:a.g.bell@bell-tel.com>;tag=3
       To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
       Call-ID: 662606876@kton.bell-tel.com
       CSeq: 1 INVITE
       Content-Length: 0

S->C:  SIP/2.0 200 OK
       Via: SIP/2.0/UDP kton.bell-tel.com
       From: A. Bell <sip:a.g.bell@bell-tel.com>;tag=3
       To: <sip:watson@bell-tel.com> ;tag=37462311
       Call-ID: 662606876@kton.bell-tel.com
       CSeq: 1 INVITE
       Contact: sip:watson@boston.bell-tel.com
       Content-Type: application/sdp
       Content-Length: ...
       v=0
       o=watson 4858949 4858949 IN IP4 192.1.2.3
       s=I'm on my way
       t=3149329600 0
       c=IN IP4 boston.bell-tel.com
       m=audio 5004 RTP/AVP 0 3
       a=rtpmap:0 PCMU/8000
       a=rtpmap:3 GSM/8000

C->S:  ACK sip:watson@boston.bell-tel.com SIP/2.0
       Via: SIP/2.0/UDP kton.bell-tel.com
       From: A. Bell <sip:a.g.bell@bell-tel.com>;tag=3
       To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
```

```
        Call-ID: 3298420296@kton.bell-tel.com
        CSeq: 1 ACK

C->S:   BYE sip:watson@boston.bell-tel.com SIP/2.0
        Via: SIP/2.0/UDP kton.bell-tel.com
        From: A. Bell <sip:a.g.bell@bell-tel.com>;tag=3
        To: T. A. Watson <sip:watson@bell-tel.com> ;tag=37462311
        Call-ID: 3298420296@kton.bell-tel.com
        CSeq: 2 BYE
```

The example illustrates the use of informational status responses. Here, the reception of the call is confirmed immediately (100), then, possibly after some database mapping delay, the call rings (180) and is then queued, with periodic status updates. Watson can only receive PCMU and GSM. Note that Watson's list of codecs may or may not be a subset of the one offered by Bell, as each party indicates the data types it is willing to receive. Watson will send audio data to port 3456 at c.bell-tel.com, Bell will send to port 5004 at boston.bell-tel.com. By default, the media session is one RTP session. Watson will receive RTCP packets on port 5005, while Bell will receive them on port 3457. Since the two sides have agreed on the set of media, Bell confirms the call without enclosing another session description (ACK). To terminate the call, Bell sends a BYE request to Watson. If the callee wants to abort the call, it simply reverses the To and From fields. Note that it is unlikely that a BYE from the callee will traverse the same proxies as the original INVITE.

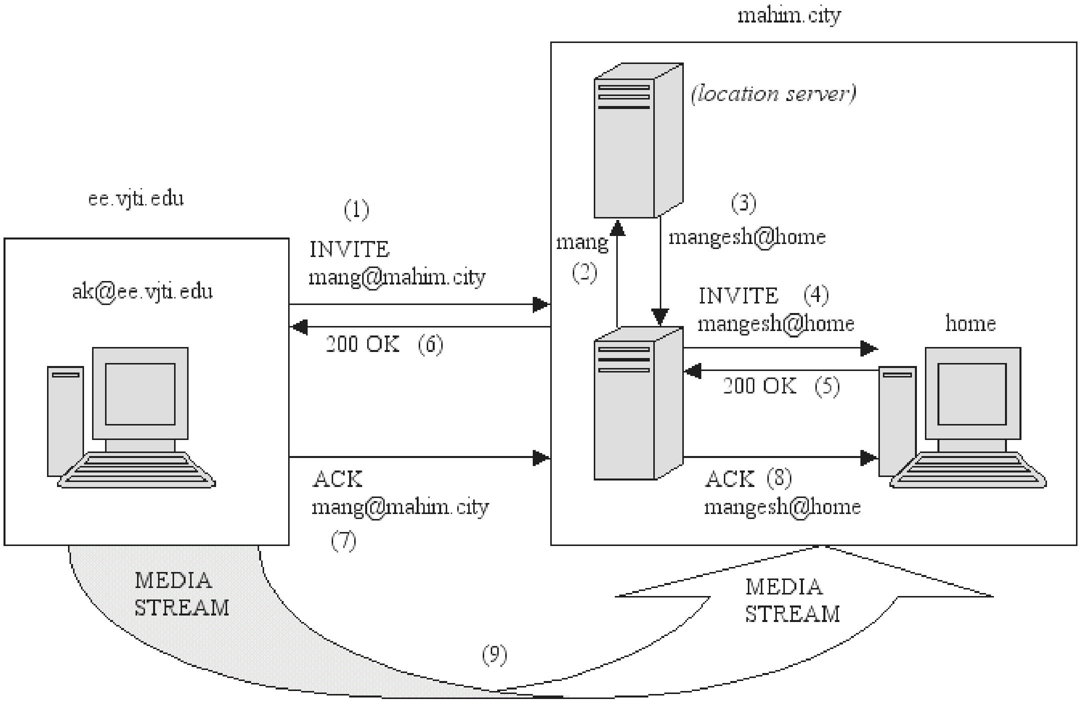
14.g. EXAMPLE: REGISTRATION

A user at host saturn.bell-tel.com registers on start-up, via multicast, with the local SIP server named bell-tel.com. In the example, the user agent on saturn expects to receive SIP requests on UDP port 3890. The registration expires after two hours. Any future invitations for watson@bell-tel.com arriving at sip.bell-tel.com will now be redirected to watson@saturn.bell-tel.com, UDP port 3890.

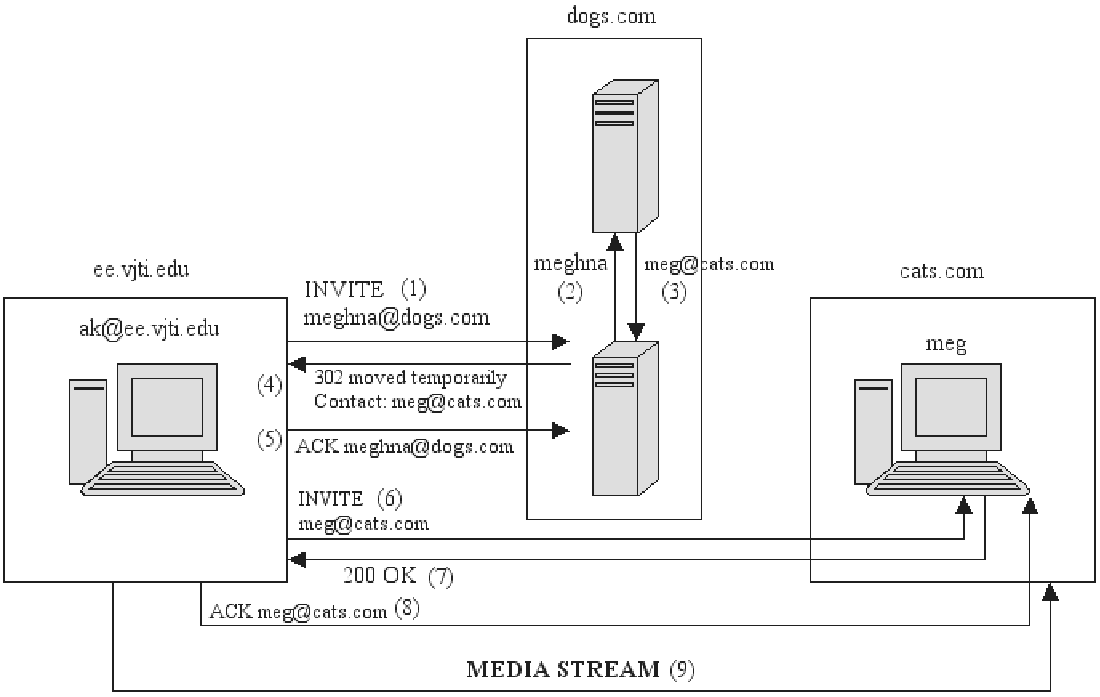
```
C->S:   REGISTER sip:bell-tel.com SIP/2.0
        Via: SIP/2.0/UDP saturn.bell-tel.com
        From: <sip:watson@bell-tel.com>;tag=19
        To: sip:watson@bell-tel.com
        Call-ID: 70710@saturn.bell-tel.com
        CSeq: 1 REGISTER
        Contact: <sip:watson@saturn.bell-tel.com:3890;transport=udp>
        Expires: 7200
```

14.h. CONNECTION PROCEDURES:

- **Operation in Proxy Mode:**

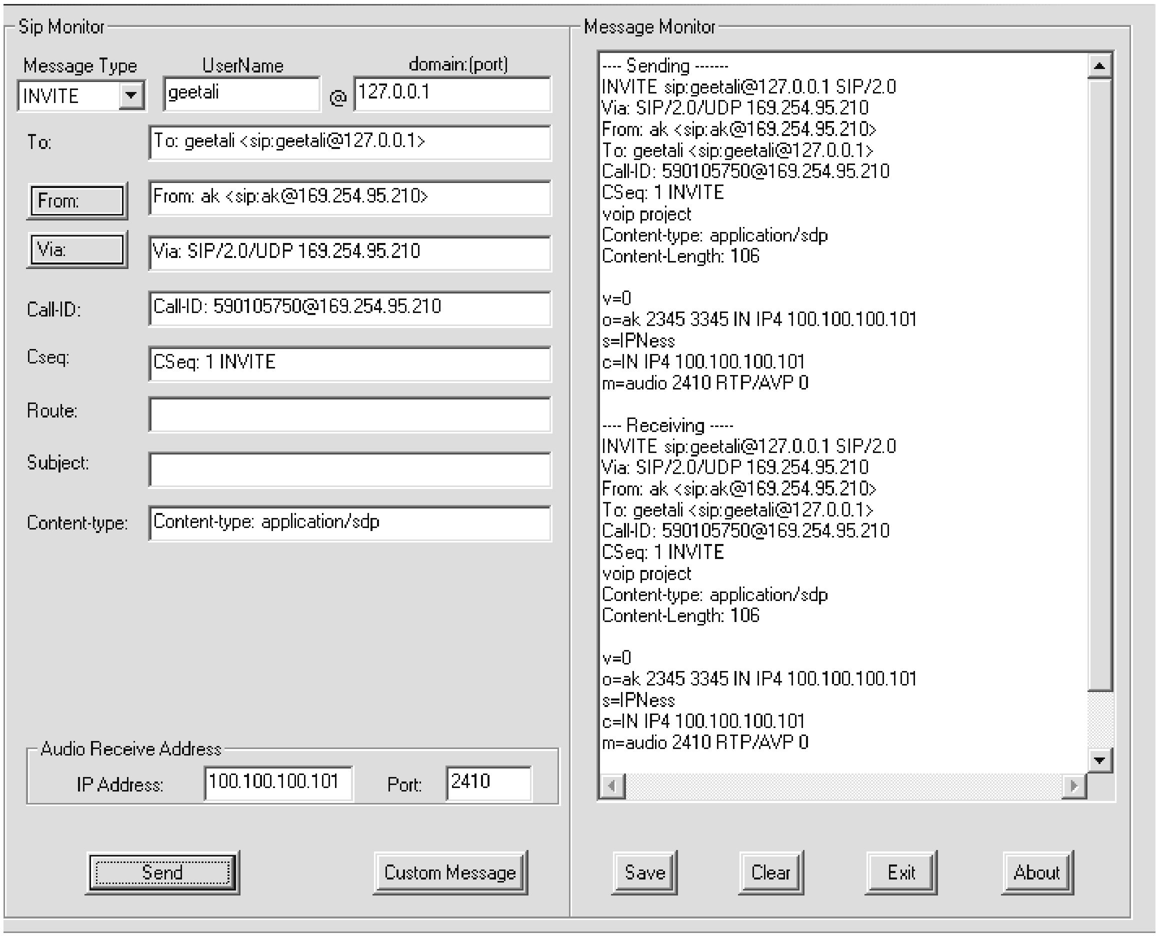


- **Operation in redirect mode:**



14.i. SIP MESSENGER:

We used the The IPNess™ SIP Messenger as a basic tool for learning how SIP sessions are performed. It provides an easy way to construct and send proper SIP messages to a remote SIP terminal, and at the same time to receive and monitor incoming SIP messages from remote SIP terminals. The complete SIP session, including all the outgoing and incoming messages can be viewed on the *Message Monitor* window. The displayed SIP messages are formatted and displayed including the SDP (Session Description Protocol) fields that are incorporated in them.



Message Type: INVITE
UserName: geetali @ domain:(port): 127.0.0.1
To: To: geetali <sip:geetali@127.0.0.1>
From: From: ak <sip:ak@169.254.95.210>
Via: Via: SIP/2.0/UDP 169.254.95.210
Call-ID: Call-ID: 590105750@169.254.95.210
Cseq: CSeq: 1 INVITE
Route:
Subject:
Content-type: Content-type: application/sdp

Audio Receive Address
IP Address: 100.100.100.101 Port: 2410

```
---- Sending -------
INVITE sip:geetali@127.0.0.1 SIP/2.0
Via: SIP/2.0/UDP 169.254.95.210
From: ak <sip:ak@169.254.95.210>
To: geetali <sip:geetali@127.0.0.1>
Call-ID: 590105750@169.254.95.210
CSeq: 1 INVITE
voip project
Content-type: application/sdp
Content-Length: 106

v=0
o=ak 2345 3345 IN IP4 100.100.100.101
s=IPNess
c=IN IP4 100.100.100.101
m=audio 2410 RTP/AVP 0

---- Receiving -----
INVITE sip:geetali@127.0.0.1 SIP/2.0
Via: SIP/2.0/UDP 169.254.95.210
From: ak <sip:ak@169.254.95.210>
To: geetali <sip:geetali@127.0.0.1>
Call-ID: 590105750@169.254.95.210
CSeq: 1 INVITE
voip project
Content-type: application/sdp
Content-Length: 106

v=0
o=ak 2345 3345 IN IP4 100.100.100.101
s=IPNess
c=IN IP4 100.100.100.101
m=audio 2410 RTP/AVP 0
```

Fig20: SIP Messenger

<u>14.j. SIP SECURITY:</u>

<u>CONFIDENTIALITY USING ENCRYPTION</u>:

SIP requests and responses can contain sensitive information about the communication patterns and communication content of individuals. The SIP message body may also contain encryption keys for the session itself. SIP supports three complementary forms of encryption to protect privacy:

- **End-to-end encryption:** relies on keys shared by the two user agents involved in the request. Typically, the message is sent encrypted with the public key of the recipient, so that only that recipient can read the message. All implementations should support PGP-based encryption and may implement other schemes. A SIP request (or response) is end-to-end encrypted by splitting the message to be sent into a part to be encrypted and a short header that will remain in the clear. Some parts of the SIP message, namely the request line, the response line and certain header fields need to be read and returned by proxies and thus must not be encrypted end-to-end. All fields that will remain in the clear must precede those that will be encrypted. The message is encrypted starting with the first character of the first header field that will be encrypted and continuing through to the end of the message body.
- **Hop-by-hop encryption:** Not all of the SIP request or response can be encrypted end-to-end because header fields such as *To* and *Via* need to be visible to proxies so that the SIP request can be routed correctly. Hop-by-hop encryption encrypts the entire SIP request or response on the wire so that packet sniffers or other eavesdroppers cannot see who is calling whom. Hop-by-hop encryption can also encrypt requests and responses that have been end-to-end encrypted. Note that proxies can still see who is calling whom, and this information is also deducible by performing a network traffic analysis, so this provides a very limited but still worthwhile degree of protection. Normally, proxies are not allowed to alter end-to-end header fields and message bodies. Proxies may, however, encrypt an unsigned request or response with the key of the call recipient. Proxies need to encrypt a SIP request if the end system cannot perform encryption or to enforce organizational security policies.

The PGP encryption scheme uses the following syntax:

Encryption = "Encryption" ":" "pgp" pgp-eparams
pgp-eparams = 1# (pgp-version | pgp-encoding)
pgp-encoding = "encoding" "=" "ascii" | token

encoding: Describes the encoding used by PGP. The value "ascii" refers to the standard PGP ASCII encoding. By default, the encrypted part is included as binary.

Example: Encryption: pgp version="2.6.2", encoding="ascii"

MESSAGE INTEGRITY & ACCESS CONTROL USING AUTHENTICATION:

Protective measures need to be taken to prevent an active attacker from modifying and replaying SIP requests and responses. The same cryptographic measures that are used to ensure the authenticity of the SIP message also serve to authenticate the originator of the message. However, the basic and digest authentication mechanism offer authentication only, without message integrity. Transport-layer or network-layer authentication may be used for hop-by-hop authentication. SIP also supports the *HTTP basic and digest schemes* and other HTTP authentication schemes to be defined that offer a rudimentary mechanism of ascertaining the identity of the caller. All SIP implementations should support *PGP-based authentication*.

The basic authentication is basically including a plain text password in a repeat request following an unauthorized or proxy authorization response to an original request.

The digest scheme consists of a challenge-response with a shared secret.

The Pretty Good Privacy (PGP) is based on the model that the client authenticates itself with a request signed with a private key. The server can then ascertain the origin of the request if it has access to the public key, preferably signed by a trusted third party. Even with encrypted requests there is a possibility, that an eavesdropper listens to messages and then injects unauthenticated responses that terminate, redirect or otherwise interfere with the call.

14.k. SIP BILLING:

The key components of an SIP billing scheme should include billing for:

- Transport ie. Resource reservation services
- SIP services such as call processing and security services
- PSTN gateway services for the integrated PSTN-VOIP scenario
- Media services such as media translation and storage

The means of providing these billing services:

- With the help of the resource reservation protocol used
- Using server log files.

14.l. SIP MINIMAL IMPLEMENTATION:

Client: All clients must be able to generate the INVITE and ACK requests. Clients must also generate and parse the Call-Id, Content-Length, Content-Type, CSeq, From and To headers. Clients must also parse the Require header. A minimal implementation must understand the protocol SDP. It must also be able to recognize the status code classes 1 through 6 and act accordingly.

Server: A minimally compliant server implementation must understand the INVITE, ACK, OPTIONS and BYE requests. A proxy server must also understand Cancel. It must parse and generate, as appropriate, the Call-Id, Content-Length, Content-Type, CSeq, Expires, From, Max-Forwards, Require, To and Via headers. It must echo the CSeq and Timestamp headers in the response. It should include the server header in its responses.

14.m. SIP INTEGRATION WITH EXISTING PROTOCOLS:

One of Sip's strengths is its remarkable ability to integrate with existing protocols used on the Internet. In particular, SIP integrates well with two dominant applications: Web and email.

Sip integrates with the Web on many levels. First, Sip carries around MIME content, as does HTTP. This characteristic enables SIP to return Web content as a result of a call invitation. Eg: a redirect response to an SIP INVITE request can contain an HTML document or text document. This document could relate detailed information on alternate places a user might be located (including pictures and sounds), or it could contain a form for submitting credit and authorization for the call. It could even return a JAVA applet, which accepts input from the caller to determine where a user might be reached. As a result SIP would integrate extremely well with Web browsers, uniting the Web and telephony to produce new, powerful services. Sip identifies a user by means of a URL, which can be embedded either in the web pages or in email, as can any other type of URL. Clicking on a URL can initiate calls just as clicking a web link accesses a new Web page.

One of the richest features of the Web is programmability. Web servers can use CGI (Common gateway Interface) to create dynamic content customized for each user. Because Sip looks like HTTP, CGI can be applied to SIP servers, and using backwards-compatible extensions, can provide a means for rich telephony programming. Services such as call-forwarding, mobility and VPN (virtual private network) are easily implemented with CGI. Furthermore, the wide array of software tools that exist to simplify CGI prototyping can be used for Internet telephony as well.

SIP also integrates well with email and SMTP. A SIP INVITE message can be sent by email when all else fails, because a SIP address is identical to an email address. A SIP proxy server can easily reformat an SIP message into an SMTP message, whose format is nearly identical and forward it to an SMTP MTA. This feature integrates voice-mail and email to enable call invitations to be delivered via email when a user is not available. In addition, SIP headers can contain mailto: URLs, which redirect callers to the user's email. SIP integrates very well with RTSP (Real-time Streaming Protocol), which provides VCR like controls ie: play, forward, rewind etc., over voice streams. This is especially useful with voice mail messages stored on a voice mail media server.

14.n. ONGOING SIP IMPLEMENTATIONS:

The current trend in SIP implementations concentrate on the following:

- Proxy and redirect servers for service creation
- PS-based user agents
- Ethernet phones
- Firewall and security enhancements
- SIP-H.323 translators
- Unified messaging

The key companies involved in these implementations include

3Com
AudioTalk networks
Broadsoft
Catapult
Cisco
Carnegie-Mellon University
Columbia University
Delta Information systems
Dynamicsoft
Ellemtel
Ericsson
Hewlett-Packard
Hughes software systems
Indigo Software
Iwatsu Electric
Komodo
Lucent
MCI Worldcom
Mediatrix
Microappliances
Netergy
Netspeak
Nokia
Object software
Nortel
Neura
Pingtel
RaveTel
Siemens
Telogy
Ubiquity
Vegastream
Vovida

14.o. SIP vs. H.323

	H.323 version 2	**SIP**
FUNCTIONALITY		
CALL CONTROL SERVICES:		
Call Holding	Yes	Yes
Call Transfer	Yes	Yes
Call Forwarding	Yes	Yes
Call Waiting	Yes	Yes
ADVANCED FEATURES:		
Third party control	No	Yes
Conference	Yes	Yes
Click-for-dial	Yes	Yes
Capability exchange	Yes & Better	Yes
QUALITY OF SERVICE		
Call setup delay	3~4 RT	2~3 RT
RELIABILITY:		
Packet loss delivery	Through TCP	Better
Fault detection	Yes	Yes
Fault tolerance	N/A	Good
MANAGEABILITY		
Admission Control	Yes	No
Policy Control	Yes	No
Resource reservation	No	No
SCALABILITY		
Complexity	More	Less
Server processing	Stateful	Stateful or Stateless
Inter-server communication	No	Yes
FLEXIBILITY		
Transport Protocol Neutrality	TCP	TCP/UDP
Extensibility of Functionality	Vendor Specified	Yes, IANA
Ease of Customization	Harder	Easier
INTEROPERABILITY		
Version Compatibility	Yes	Unknown
SCN Signaling Interoperability	Better	Worse
EASE OF IMPLEMENTATION		
Protocol Encoding	Binary	Text

15. SUPPORTING PROTOCOLS:

SIP works in conjunction with RSVP (Resource Reservation Protocol), RTP/RTCP (Real-time Transport Protocol), RTSP (Real-time Streaming Protocol), SAP (Session Announcement Protocol) and SDP (Session Description Protocol). RTP/RTCP is used for transporting real time data, RSVP for reserving resources, RTSP for controlled delivery of streams, SAP for advertising multimedia sessions and SDP for describing multimedia sessions. H.323 too works in conjunction with RTP and RTCP (Real-time Control Protocol). The present day voice gateways usually compose of two parts: the signaling gateway and the media gateway. The signaling gateway communicates with the media gateway using MGCP (Media Gateway Access Protocol). MGCP can interoperate with both SIP and H.323.

***15.a. MEDIA GATEWAY CONTROL PROTOCOL (MGCP)*:**

It is a protocol that defines communication between call control elements (Call Agents) and telephony gateways. It is a master/slave protocol, where the gateways are expected to execute commands sent by the Call Agents.

Call Agents are also known as *Media Gateway Controllers*. MGCP is a control protocol, allowing a central coordinator to monitor events in IP phones and gateways and instructs them to send media to specific addresses. It resulted from the merger of the Simple Gateway Control Protocol (SGCP) and Internet Protocol Device Control (IPDC). The call control intelligence is located outside the gateways and they are handled by external call control elements, the Call Agent. MGCP assumes that these call control elements will synchronize with each other to send coherent commands to the gateways under their control. It has introduced the concepts of connections and endpoints for establishing voice paths between two participants, and the concepts of events and signals for establishing and tearing down calls. The main emphasis of MGCP is simplicity and reliability and it allows programming difficulties to be concentrated in Call Agents, so it will enable service providers to develop reliable and cheap local access systems.

Endpoints and Connections:
Endpoints are the sources or sinks of data. An example could be an interface on a gateway that terminates a trunk connected to a PSTN switch.
Connections may be either point-to-point or multipoint. A connection is either an association between two endpoints (point-to-point) or it is an association between multiple endpoints (multipoint). Once the association is established, data transfer can take place. Connections can be established over a number of bearer networks i.e. TCP/IP, ATM etc.

Events and Signals:
A call agent may ask to be notified about certain events occurring in an endpoint, such as off-hook, on-hook, dialed digits, and may request that a certain signal be applied to an endpoint such as dial-tone, busy tone or ringing. Events and signals are grouped in

packages that are supported by a particular type of endpoint e.g., one package may support a certain group of events and signals for analog access lines.

Creating Connections:

Connections are created on the call agent at each endpoint that will be involved in the call. When the two endpoints are located on gateways that are managed by the same call agent, the creation is done via the following three steps:

- The Call Agent asks the first gateway to create a connection on the first endpoint. The response sent by the gateway includes a session description that contains pertinent information required by third parties to be able to send packets to the new connection that has been created.
- The Call Agent then sends the session description of the first gateway to the second gateway and asks it to create a connection on the second endpoint. The second gateway responds by sending its own session description.
- The Call Agent uses a modify connection command to provide this second session description to the first endpoint. Now communication can occur in both directions.
- When the two endpoints are located on gateways that are managed by the different call agents, these two call agents shall exchange information through a call agent to call agent signaling protocol, in order to synchronize the creation of the connection on the two endpoints.

MGCP Commands:

The MGCP implements the media gateway control interface as a set of transactions. The transactions are composed of a command and a mandatory response. There are 8 types of command:

- **CreateConnection**: The CreateConnection command is used to attach an endpoint to a specific IP address and port. To create a connection, a CreateConnection request is required for the remote endpoint also. If the request is successfully acknowledged by the gateway, then a ConnectionId is returned that uniquely identifies the connection.
- **ModifyConnection:** This command is used by the call agent to modify the parameters of an active connection. The ConnectionId is passed to identify the connection.
- **DeleteConnection:** This command is used by either the call agent or the gateway to delete an existing connection. The response includes a list of parameters about the status of the connection.
- **NotificationRequest**: If a call agent wants to be informed about the occurrence of specified events in an endpoint, then it can send this request to the gateway. Events could be: off hook transition, flash-hook, continuity tone detection, etc. A notification may be requested for a continuity tone detection event in a gateway.
- **Notify**: The response to the NotificationRequest is sent via the Notify command by the gateway. The notify command includes a list of events that the gateway observed.

- **AuditEndpoint**: This command is used by the call agent to get details about the status of an endpoint/several endpoints and the response from the gateway contains the requested information
- **AuditConnection**: To obtain information for a specific connection of an endpoint, the call agent uses this command. The connection is identified by the ConnectionId and the response from the gateway contains the requested information.
- **RestartInProgress**: This command is used by the gateway to indicate that an endpoint or a bunch of endpoints have been taken in/out of service. It also includes a parameter that indicates the type of restart (graceful restart/ forced restart/delayed restart)

15.b. REAL-TIME TRANSPORT PROTOCOL (RTP) & REAL-TIME TRANSPORT CONTROL PROTOCOL (RTCP):

So-called real-time delivery of traffic requires little in the way of transport protocol. In particular, real-time traffic that is sent over more than trivial distances is not retransmittable. In fact, a number of facets of an end-to-end protocol need to be re-designed or refined including:

- **Separate Flows for each Media Stream**: With packet multimedia data there is no need for the different media comprising a session to be carried in the same packets. In fact it simplifies receivers if different media streams are carried in separate flows (i.e., separate transport ports and/or separate multicast groups). This also allows the different media to be given different quality of service. For example, under congestion, a router might preferentially drop video packets over audio packets. In addition, some sites may not wish to receive all the media flows. For example, a site with a slow access link may be able to participate in a session using only audio and a white-board whereas other sites in the same session may also send and receiver video.
- **Receiver Adaptation**: Best-effort traffic is delayed by queues in routers between the sender and the receivers. Even reserved priority traffic may see small transient queues in routers, and so packets comprising a flow will be delayed for different times. Such delay variance is known as jitter. Real-time applications such as audio and video need to be able to buffer real-time data at the receiver for sufficient time to remove the jitter added by the network and recover the original timing relationships between the media data. In order to know how long to buffer, each packet must carry a timestamp which gives the time at the sender when the data was captured. Note that for audio and video data timing recovery, it is not necessary to know the absolute time that the data was captured at the sender, only the time relative to the other data packets.
- **Synchronisation:** As audio and video flows will receive differing jitter and possibly differing quality of service, audio and video that were grabbed at the same time at the sender may not arrive at the receiver at the same time. At the receiver, each flow will need a play-out buffer to remove network jitter. Inter-flow synchronisation can be performed by adapting these play-out buffers so that samples/frames that originated at the same time are play-out out at the same time.

This requires that the times that different flows from the same sender were captured are available at the receivers.

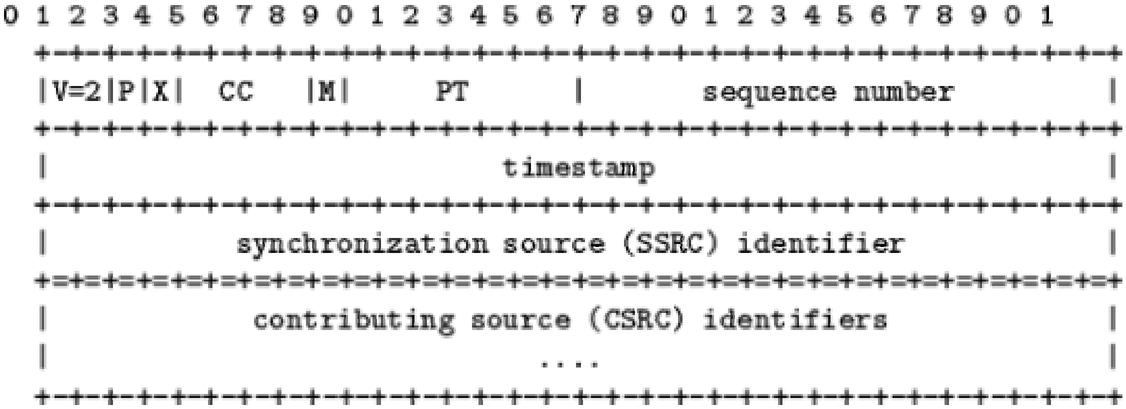
RTP supports the transfer of real-time media (audio and video) over packet switched networks. It is used by both SIP and H.323. The transport protocol must allow the receiver to detect any losses in packets and also provide timing information so that the receiver can correctly compensate for delay jitter. The RTP header contains information that assist the receiver to reconstruct the media and also contains information specifying how the codec bitstreams are broken up into packets. RTP does not reserve resources in the network but instead it provides information so that the receiver can recover in the presence of loss and jitter.The functions provided by RTP include:

- **Sequencing**: The sequence number in the RTP packet is used for detecting lost packets
- **Payload Identification**: In the Internet, it is often required to change the encoding of the media dynamically to adjust to changing bandwidth availability. To provide this functionality, a payload identifier is included in each RTP packet to describe the encoding of the media
- **Frame Indication:** Video and audio are sent in logical units called frames. To indicate the beginning and end of the frame, a frame marker bit has been provided
- **Source Identification:** In a multicast session, we have many participants. So an identifier is required to determine the originator of the frame. For this Synchronization Source (SSRC) identifier has been provided.
- **Intramedia Synchronization:** To compensate for the different delay jitter for packets within the same stream, RTP provides timestamps, which are needed by the play-out buffers.

RTCP is a control protocol and works in conjunction with RTP. In a RTP session, participants periodically send RTCP packets to obtain useful information about QoS etc. The additional services that RTCP provides to the participants are:

- **QoS feedback:** RTCP is used to report the quality of service. The information provided includes number of lost packets, Round Trip Time, jitter and this information is used by the sources to adjust their data rate.
- **Session Control:** By the use of the BYE packet, RTCP allows participants to indicate that they are leaving a session
- **Identification:** Information such as email address, name and phone number are included in the RTCP packets so that all the users can know the identities of the other users for that session.
- **Intermedia Synchronization:** Even though video and audio are normally sent over different streams, we need to synchronize them at the receiver so that they play together. RTCP provides the information that is required for synchronizing the streams.

RTP PACKET FORMAT:



```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |V=2|P|X|  CC   |M|     PT      |       sequence number         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           timestamp                           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           synchronization source (SSRC) identifier            |
 +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 |            contributing source (CSRC) identifiers             |
 |                             ....                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig21:RTP packet format

The first twelve octets are present in every RTP packet, while the list of CSRC identifiers is present only when inserted by a mixer.

- Version (V): 2 bits
  This field identifies the version of RTP. The version defined by this specification is two (2).
- Padding (P): 1 bit
  If the padding bit is set, the packet contains one or more additional padding octets at the end which are not part of the payload.
- Extension (X): 1 bit
  If the extension bit is set, the fixed header is followed by exactly one header extension, with a format defined in Section 5.2.1.
- CSRC count (CC): 4 bits
  The CSRC count contains the number of CSRC identifiers that follow the fixed header.
- Marker (M): 1 bit
  The interpretation of the marker is defined by a profile. It is intended to allow significant events such as frame boundaries to be marked in the packet stream.
- Payload type (PT): 7 bits
  This field identifies the format of the RTP payload and determines its interpretation by the application.
- Sequence number: 16 bits
  The sequence number increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence.
- Timestamp: 32 bits
  The timestamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations
- SSRC: 32 bits
  The SSRC field identifies the synchronization source.

- CSRC list: 0 to 15 items, 32 bits each
  The CSRC list identifies the contributing sources for the payload contained in this packet. The number of identifiers is given by the CC field. If there are more than 15 contributing sources, only 15 may be identified. CSRC identifiers are inserted by mixers, using the SSRC identifiers of contributing sources.

An extension mechanism is provided to allow individual implementations to experiment with new payload-format-independent functions that require additional information to be carried in the RTP data packet header. This mechanism is designed so that the header extension may be ignored by other interoperating implementations that have not been extended.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      defined by profile       |           length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        header extension                       |
|                             ....                              |
```

If the X bit in the RTP header is one, a variable-length header extension is appended to the RTP header, following the CSRC list if present. The header extension contains a 16-bit length field that counts the number of 32-bit words in the extension, excluding the four-octet extension header (therefore zero is a valid length). Only a single extension may be appended to the RTP data header. To allow multiple interoperating implementations to each experiment independently with different header extensions, or to allow a particular implementation to experiment with more than one type of header extension, the first 16 bits of the header extension are left open for distinguishing identifiers or parameters. The format of these 16 bits is to be defined by the profile specification under which the implementations are operating.

RTP HEADER COMPRESSION:
The combination of the IP, UDP and RTP control information adds up to a significant overhead for small media samples, particularly over low speed links, commonly in use by the domestic and small office user dialing up their Internet Service Provider at a few tens of kilobits per second. An IP Datagram has a 20 byte header, while the UDP header is 8 bytes (source and destination ports, plus length and checksum field). The RTP header adds 12 bytes to this, making a total of 40 bytes of control for a single sample, in some cases as little as 20ms worth of 8KHzspeech. By far the common case of such usage is as stated when dialing up an ISP, where the access router is connected to a high-speed backbone. There is already a technique for reducing the overhead of packets in such circumstances, designed for compressing TCP/IP. Casner and Jacobson adapted this technique to RTP headers.

The technique consists of two parts:

- noting fields in the packet headers that do not change over the life of a flow;
- noting that there are few flows at the edge of the network so that such information can be conveyed over the first hop by a single packet, and subsequently referred

to by a short connection identifier, which serves to index the full state so that the first hop router can reconstruct the full Internetwork packet.

In RTP, it turns out that there are also fields that change only by the same amount from each packet to the next, except in exceptional circumstances, so that this second order information can also be stored in the compression state vector at the first hop routers. This compression state is *soft-state* in that it can be recovered simply by loss since the packet conveys enough implicit information that end-to-end checksums are still computed, and hop-wise recomputed from the state vector and from the remaining data in the compressed header. In other words, if the router resets, or the route changes, or the end system radically alters state, the invalid checksum causes a reset of the compressed state, and an exchange of a full packet re-creates the necessary state from full anew. The compression shows a typical reduction to a header size of 3-4 bytes (better than ten-fold reduction in control overhead).

RTP MULTIPLEXING:

There are a number of circumstances in which one might wish to carry multiple media flows within a single RTP data payload between two points. The two most important cases are:

- IP paths between Internet Telephony Gateways
- Special hardware devices such as CODECs with non-negotiable multiplexed media.

There are at least two ways to multiplex data in RTP packets.

- One could (e,g. in the telephony case) assume that all the samples have the same payload types, and are just offset in different end-to-end flows. Here we need a mapping table in the gateways, which indicates the offset for each payload type and a list of the flows in each packet. In RTP, multiplexing is provided by the destination transport address (network address and port number), which defines an RTP session.
- The second approach is to adapt the ideas concerning RTP header compression and to allow for multiple compressed headers within a single RTP packet, one for each of the samples. This latter approach would not use precisely the same compression algorithm, since the fields differ for different reasons, but would permit multiple different media to be efficiently encapsulated in a single packet. This might address both types of application of RTP multiplexing more effectively.

RTP MIXERS & TRANSLATORS:

All sites might not want to or might not be able to receive media data in the same format. Consider the case where participants in one area are connected through a low-speed link to the majority of the conference participants who enjoy high-speed network access. Instead of forcing everyone to use a lower-bandwidth, reduced-quality audio encoding, an RTP-level relay called a *mixer* may be placed near the low-bandwidth area. This mixer resynchronizes incoming audio packets to reconstruct the constant 20 ms spacing generated by the sender, mixes these reconstructed audio streams into a single stream, translates the audio encoding to a lower-bandwidth one and forwards the lower-bandwidth packet stream across the low-speed link. These packets might be unicast to a single recipient or multicast on a different address to multiple recipients. The RTP header includes a means for mixers to identify the sources that contributed to a mixed packet so that correct talker indication can be provided at the receivers.

Some of the intended participants in the audio conference may be connected with high bandwidth links but might not be directly reachable via IP multicast. For example, they might be behind an application-level firewall that will not let any IP packets pass. For these sites, mixing may not be necessary; in which case another type of RTP-level relay called a *translator* may be used. Two translators are installed, one on either side of the firewall, with the outside one funneling all multicast packets received through a secure connection to the translator inside the firewall. The translator inside the firewall sends them again as multicast packets to a multicast group restricted to the site's internal network.

RTCP PACKET FORMAT:

RTCP is the Real-time Transport Control Protocol, which may be used as a lightweight companion to RTP to convey a number of statistics and other information about an RTP flow between recipients and senders. This specification defines several RTCP packet types to carry a variety of control information:

- SR: Sender report
  for transmission and reception statistics from participants that are active senders
- RR: Receiver report
  for reception statistics from participants that are not active senders
- SDES:
  Source description items, including CNAME (Canonical Identifier), NAME: User name, EMAIL: Electronic mail address, PHONE: Phone number, LOC: Geographic user location, TOOL: Application or tool name, NOTE: Notice/status, PRIV: Private extensions
- BYE:
  Indicates end of participation and a reason for leaving
- APP:
  Application specific functions

Each RTCP packet begins with a fixed part similar to that of RTP data packets, followed by structured elements that may be of variable length according to the packet type but always end on a 32-bit boundary. The alignment requirement and a length field in the fixed part are included to make RTCP packets "stackable". Multiple RTCP packets may be concatenated without any intervening separators to form a compound RTCP packet that is sent in a single packet of the lower layer protocol, for example UDP.

For the exact format of each of the packets one must refer to ietf-rfc1889.

RTCP SCALING & TIMING CONSIDERATIONS:

RTCP reports are designed to be sent periodically, with a frequency inversely proportional to the number of members. This can be set to constrain the bandwidth consumed by RTCP to be a known fixed fraction of the total capacity required for a many-to-many session. For multimedia conferencing this is an excellent solution for many applications requirements for such data. However there are some circumstances where this approach creates a problem: firstly, when a session starts, the members do not know the number of other members - initial membership for a scheduled session could rise very sharply, leading to a flood of initial packets and secondly, this effect can also happen for BYE messages.

Intra-stream Synch: inside a stream, need to know where in the "time structure" a bit goes. In the Internet, the RTP media specific timestamp provides a general-purpose way of carrying out this function.

Inter-Stream Synch: The easiest way of synchronizing between streams at different sites is based on providing a globally synchronized clock. There are two ways this might be done:

1. Have the network provide a clock.
2. Have a clock synchronization protocol, such as NTP (the network Time Protocol) or DTS (Digital Time Service). This operates between all the computers in a data network, and continually exchanges messages between the computers to monitor.

Inter-media Synch: Options include having a global clock provided from the network or using clock synch between computers. Or we could use NTP/DTS in each packet for clock synch calculation.

15.c. REAL-TIME STREAMING PROTOCOL (RTSP):

If our audio and video streams come to us over a network, and are available in the computer's memory, then at the very least we should expect to be able to control the streams in the same fashion as we control traditional media. We should be able to start the program playing, pause the stream if we wish, skip further along the stream to locate the interesting bits, and slow or speed up the program as required. Since the medium for the stream is a network connected to a computer, building the controls is simply a matter of designing a program and the associated techniques for talking to other programs. The stream is generated by a node on the network, which is called a media server, and is sent as a stream of packets over the network to the receiver(s), which appropriately process and pass the data up to the application. We have two possible ways of controlling the media streams - we could send messages back over the network to the source of the stream to ask the source to play, stop, pause etc., or we could allow the stream to come to us, and simply manipulate the media in memory, keeping control within the receiving computer. Both techniques are possible, but are suitable in different environments. When the stream of data coming over the network is owned by just one user, then the network and server resources are more efficiently used if the media stream is manipulated at source.

Stream control requires the receiver to send a message to the media server, asking it to perform some action. There may or may not be some response. This paradigm of request/response over a network has been well-known for many years, and has been formulated as a remote procedure call or RPC. Using RPC mechanisms makes defining the control interfaces easy. However there remain some subtleties in interpreting the precedence of control requests. There is the need for a small stateful protocol at the server to deal with the ordering of play and pause requests. Inherent in controlling streams of data remotely is the necessity to keep state at the server about the requests that have been made. Requests must have an associated sequence number, so that the server can service the incoming requests in the correct order, despite any re-ordering that may occur.

The Real-Time Streaming Protocol (RTSP) establishes and controls either a single or several time-synchronized streams of continuous media such as audio or video. RTSP acts a ``network remote control'' for multimedia servers. Rather than using an RPC mechanism directly, the designers of RTSP decided to use a variation on HTTP, since in its current incarnation of version 1.1; it approximates an application level RPC mechanism. The media streams are left unspecified by RTSP. RTSP only specifies the control and it is up to the client and server software to maintain the mapping between the control channel and the media streams.

RTSP URLs: Multimedia presentations are identified by URLs, using a protocol scheme of RTSP. The hostname is the server containing the presentation; whilst the port indicates which port the RTSP control requests should be sent to. Presentations may consist of one or more separate streams. The presentation URL provides a means of identifying and controlling the whole presentation rather than coordinating the control of each individual steam. So, *rtsp://media.example.com:554/twister/audiotrack* identifies the audio stream within the presentation twister, which can be controlled on its own.

RTSP MESSAGES: RTSP add a number of new requests to the existing HTTP requests. These are :

- DESCRIBE: Causes a server to return a description of the protocol using the Session Description Protocol.
- ANNOUNCE: Allows a client or server to register a description of a presentation.
- OPTIONS: Causes a server to return the list of supported methods.
- SETUP: Causes a server to allocate resources for a stream and starts an RTSP session. This manipulates state.
- PLAY: Starts data transmission on a stream allocated by SETUP. This manipulates state.
- RECORD: Starts a server recording an allocated stream. This manipulates state.
- PAUSE: Temporarily halts transmission of a stream without freeing server resources. This manipulates state.
- TEARDOWN: Frees resources associate with a stream, so that the RTSP session ceases to exist. This manipulates state.
- GET_PARAMETER and SET_PARAMETER: Placeholder methods to allow parameters of presentations and sessions to be manipulated.
- REDIRECT: Causes a client to go to another server for parts of a presentation.

The most obvious additions to the request header fields are a *Cseq field* to contain the sequence numbers of requests generated by the client, and a *Session field* to both request and response headers to identify the session. Session identifiers are generated in response to a SETUP request, and must be used in all stateful methods. The *Transport field* allows the client and server to negotiate and set parameters for the sending of the media stream.

In particular, it allows the client and server to set ports and multicast addresses for the RTP streams. There are a number of other header fields, such as the time range of the presentation upon which the method executes (Range), and various fields which interact with caches and other proxies.

TRANSPORT PROTOCOLS: The control requests and responses may be sent over either TCP or UDP. Since the order of the requests matters, the requests are sequenced, so if any requests are lost, they must be retransmitted. Using UDP thus requires the construction of retransmission mechanisms, so there are very few occasions when the application can get away with using UDP.

RTSP SESSIONS: A key concept in RTSP is the notion of a session. RTSP works by first requesting a presentation to be started by a server, receiving in return a session identifier which it then uses in all subsequent controls. Eventually, the client can request the teardown of session, which releases the associated resources. The session identifier represents the shared state between the client and server. If the state is lost, for example through one of the machines being rebooted, then the protocol relies on the transport of the media stopping automatically.

Descriptions of session use the Session Description Protocol, which provides a generic technique for describing the details of the presentation, such as transport and media types of the stream, and the presentation content. Importantly, it also provides the start and end times of the presentation, so that the client can PLAY from and to any point in the presentation they wish.

RTSP OPERATION: Media streams are referenced through specification of their times, either relative to the start time of the presentation, or in real time. RTSP allows the use of the standard time codes used in industry such as SMTPE or Normal Play Time, or by specifying an absolute time for presentations in real-time.

To display a presentation, the client software first requires the RTSP URL of the presentation. If it has this URL, it can then display the presentation by following these steps.

1. The client first requests the description of the presentation using the DESCRIBE method. This supplies details of the media streams, so that the client can start the appropriate media applications.
2. The client then requests that the session is SETUP, receiving a session identifier in return. The server would allocate state, such as the sockets through which the media will be sent, plus any reservations for bandwidth.
3. The client requests that the media streams of the session are PLAYed, by specifying the URL and the session identifier, and a time range to start and finish playing at.
4. At any point they may PAUSE the presentation, and continue form any other point in the presentation by using specifying a new range in the PLAY request.
5. When the client has completed, the client issues a TEARDOWN request to destroy the session, and deallocate any resources.

15.d. SESSION DESCRIPTION PROTOCOL (SDP):

A session description expressed in SDP is a short structured textual description of the name and purpose of the session, and the media, protocols, codec formats, timing and transport information that are required to decide whether a session is likely to be of interest and to know how to start media tools to participate in the session. SDP is purely a format for session description - it does not incorporate a transport protocol, and is intended to use different transport protocols as appropriate, including the Session Announcement Protocol, Session Initiation Protocol, Real-Time Streaming Protocol, electronic mail using the MIME extensions, and the Hyper-Text Transport Protocol (HTTP).

SDP includes the session name and a description of its purpose, the times the session is active, the media comprising the session, and information to receive those media (addresses, ports, formats and so on). As resources necessary to participate in a session may be limited, some additional information may also be desirable, including information about the bandwidth to be used by the conference and contact information for the person responsible for the session. In general, SDP must convey sufficient information to be able to join a session (with the possible exception of encryption keys) and to announce the resources to be used to non-participants that may need to know.

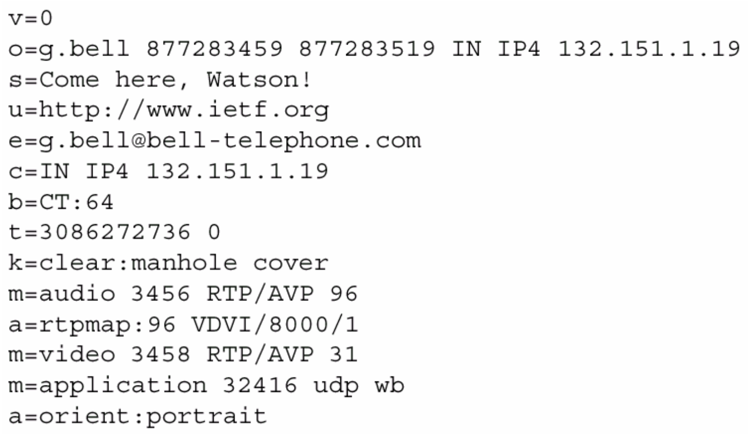
TIMING INFORMATION: Sessions may either be bounded or unbounded in time. Whether or not they are bounded, they may be only active at specific times. SDP can convey an arbitrary list of start and stop times bounding the session. For each bound, repeat times can be specified. This timing information is globally consistent. Modifiers can be specified to uniquely apply offsets.

MEDIA DESCRIPTIONS: A session description is composed of general information that applies to the whole session (such as the timing) followed by sections that are specific to each medium. For each medium SDP includes the type of media (video, audio, etc), the transport protocol (RTP/UDP/IP, H.320, etc), the format of the media (H.261 video, MPEG video, etc), and media or codec specific attributes.
For an IP multicast session, the multicast address for that medium and the transport port are also conveyed. This address and port are the destination address and destination port of the multicast stream, whether being sent, received, or both. For an IP unicast session, the remote address for the media stream and the transport port for the contact address are given. The response to the invitation would then give similar information about where the caller should send her audio and video streams.

SDP SYNTAX: An SDP session description consists of a number of lines of text of the form:
<type>=<value>

<type> is always exactly one character and is case-significant.
<value> is a structured text string whose format depends on <type>. In general <value> is either a number of fields delimited by a single space character or a free format string.

A session description consists of a session-level description (details that apply to the whole session and all media streams) and optionally several media-level descriptions (details that apply only to a single media stream). *The session-level* part starts with a *`v='* line and continues to the first media-level section. The *media description* starts with an *`m='* line and continues to the next media description or to the end of the whole session description. In general, session-level values are the default for all media unless overridden by an equivalent media-level value.



```
v=0
o=g.bell 877283459 877283519 IN IP4 132.151.1.19
s=Come here, Watson!
u=http://www.ietf.org
e=g.bell@bell-telephone.com
c=IN IP4 132.151.1.19
b=CT:64
t=3086272736 0
k=clear:manhole cover
m=audio 3456 RTP/AVP 96
a=rtpmap:96 VDVI/8000/1
m=video 3458 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

Fig22:SDP example

The individual fields have the following meanings, and must be in this order, with ``*'' indicating an optional field:

v=	Protocol Version
o=	The Owner/Creator and Session Identifier
s=	Session Name
i=*	Session Information
u=*	URI of description
e=*	Email Address
p=*	Phone Number
c=*	Connection Information
b=*	Bandwidth Information
One or more time descriptions	
z=*	Time Zone Adjustments
k=*	Encryption Key
a=*	zero or more Session Attributes
Zero or more media descriptions	

Each time description consists of a ``t='' field, optionally followed by one or more ``r='' fields.

t=	Time the Session is Active
r=*	zero or more Repeat Times

Each media description consists of a ``m" field, with other optional fields providing additional information:

m=	Media Name and Transport Address
i=*	Media Title
c=*	Connection Information
b=*	Bandwidth Information
k=*	Encryption Key
a=*	zero or more Media Attributes

The connection (`c=') and attribute (`a=') information in the session-level section applies to all the media of that session unless overridden by connection information or an attribute of the same name in the media description.

15.e. SESSION ANNOUNCEMENT PROTOCOL (SAP):

This protocol is used for advertising the multicast conferences and other multicast sessions. A SAP announcer periodically multicasts an announcement packet to a well-known multicast address and port (port number 9875). A SAP listener learns of the multicast scopes using the Multicast Scope Zone Announcement Protocol and listens on the well-known SAP address and port for those scopes. There is no rendezvous mechanism -- the SAP announcer is not aware of the presence or absence of any SAP listeners. A SAP announcement is multicast with the same scope as the session it is announcing, ensuring that the recipients of the announcement can also be potential recipients of the session being advertised. If a session uses addresses in multiple administrative scope ranges, it is necessary for the announcer to send identical copies of the announcement to each administrative scope range. Multiple announcers may announce a single session, as an aid to robustness in the face of packet loss and failure of one or more announcers. The time period between repetitions of an announcement is chosen such that the total bandwidth used by all announcements on a single SAP group remains below a preconfigured limit. Each announcer is expected to listen to other announcements in order to determine the total number of sessions being announced on a particular group. SAP is intended to announce the existence of a long-lived wide area multicast sessions and involves a large startup delay before a complete set of announcements is heard by a listener. In order to reduce the delays inherent in SAP, it is recommended that proxy caches be deployed. A SAP proxy is expected to listen to all SAP groups in its scope and maintain an up-to-date list of all announced sessions along with the time each announcement was last received. SAP also contains mechanisms for ensuring integrity of session announcements, for authenticating the origin of an announcement and for encrypting such announcements.

The SAP packet format (for IPv4) is shown in the figure below. The Message Type (T) field indicates whether this packet announces a session, or deletes an announcement. One bit (E) indicates whether or not the payload is encrypted and one bit (C) indicates whether or not the payload is compressed. The combination of message ID hash and original source is supposed to provide a unique announcement ID that can be used to identify this particular version of this particular session. This is useful for caching or for ignoring packets that we previously failed to decrypt, but as this announcement ID is not guaranteed to be unique, care must be taken to also check the packet length and periodically check the packet contents themselves.

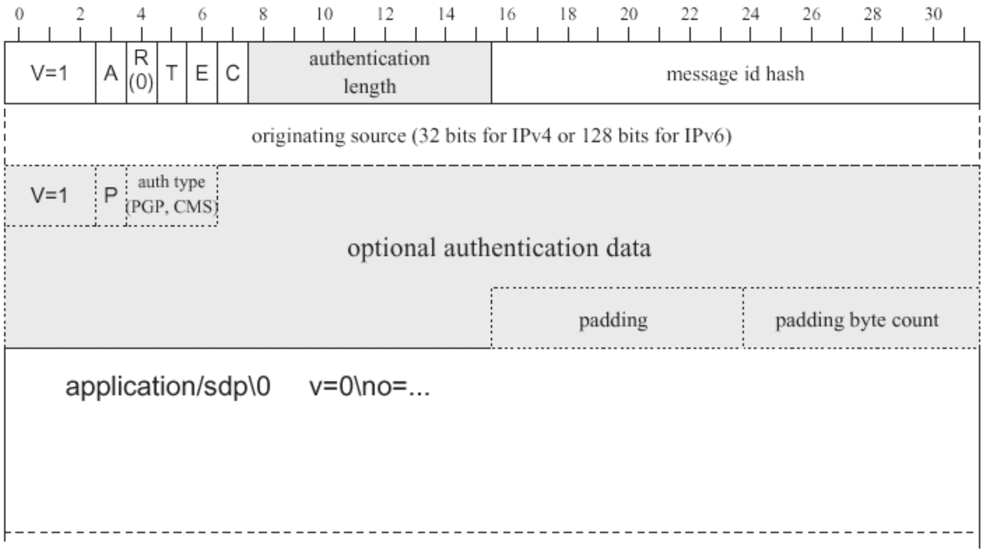


Fig23: SAP packet format

SAP announcements can be authenticated by including a digital signature of the payload in the optional authentication header. Both PGP and PKCS#7 based digital signatures are currently supported in the SAP protocol, although currently not many announcements are authenticated.

SAP announcements can also be encrypted. However, this does not mean that the standard way to have small private wide-area conferences will be to announce them with encrypted SAP - the SIP is a more appropriate mechanism for such conferences. The main uses for encrypted SAP announcements would appear to be in intranet environments where the SAP announcement bother few additional people, or for very large sessions where members are charged to participate. In the latter case, it would probably be a good idea to provide an additional level of access control beyond SAP encryption because it is easy for one misbehaving participant to leak a SAP key to other potential participant unless the keys are embedded in hardware.

SAP should be used for sessions of some public interest where the participants are not known in advance. If you know who you want in your session, a better mechanism is to explicitly invite them using SIP.

15.f. RESOURCE RESRVATION PROTOCOL (RSVP):

The Resource Reservation Protocol (RSVP) is a network-control protocol that enables Internet applications to obtain special qualities of service (QoSs) for their data flows. RSVP is not a routing protocol; instead, it works in conjunction with routing protocols and installs the equivalent of dynamic access lists along the routes that routing protocols calculate. RSVP occupies the place of a transport protocol in the OSI model seven-layer protocol stack.

RSVP DATA FLOWS: In RSVP, a data flow is a sequence of messages that have the same source, destination (one or more), and quality of service. QoS requirements are communicated through a network via a flow specification. A flow specification often guarantees how the internetwork will handle some of its host traffic.

RSVP supports three traffic types: best-effort, rate-sensitive, and the delay-sensitive. The type of data-flow service used to support these traffic types depends on QoS implemented.
1.Best-effort traffic: is traditional IP traffic and the service is called *Best-effort service*.
2.Rate-sensitive traffic: is willing to give up timeliness for guaranteed rate. Rate-sensitive traffic, for example, might request 100 kbps of bandwidth. If it actually sends 200 kbps for an extended period, a router can delay traffic. H.323 encoding is a constant rate or nearly constant rate, and it requires a constant transport rate. Such an application would use the rate-sensitive feature.The RSVP service supporting rate-sensitive traffic is called *guaranteed bit-rate service*.
3.Delay-sensitive traffic: is traffic that requires timeliness of delivery and varies its rate accordingly. MPEG-II video, for example, averages about 3 to 7 Mbps depending on the amount of change in the picture. RSVP services supporting delay-sensitive traffic are referred to as *controlled-delay service* (non-real time service) and *predictive service* (real-time service).

RSVP data flows are generally characterized by *sessions*, over which data packets flow. A session is a set of data flows with the same unicast or multicast destination, and RSVP treats each session independently. RSVP supports both unicast and multicast sessions, whereas a flow always originates with a single sender.

RSVP QoS: In the context of RSVP, quality of service (QoS) is an attribute specified in flow specifications that is used to determine the way in which data interchanges are handled by participating entities (routers, receivers, and senders). RSVP is used to specify the QoS by both hosts and routers. Hosts use RSVP to request a QoS level from the network on behalf of an application data stream. Routers use RSVP to deliver QoS requests to other routers along the path(s) of the data stream. In doing so, RSVP maintains the router and host state to provide the requested service.

RSVP SESSION STARTUP: A potential sender starts sending RSVP path messages to the IP destination address. The receiver application receives a path message and starts sending appropriate reservation-request messages specifying the desired flow descriptors using RSVP. After the sender application receives a reservation-request message, the sender starts sending data packets. If the receiver is part of a multicast group he must first join the group.

RSVP SOFT-STATE IMPLEMENTATION: In the context of an RSVP, a soft state refers to a state in routers and end nodes that can be updated by certain RSVP messages. The soft state characteristic permits an RSVP network to support dynamic group membership changes and adapt to changes in routing. In general, the soft state is maintained by an RSVP-based network to enable the network to change states without consultation with end points.

RSVP PROTCOL OPERATION: The RSVP resource-reservation process initiation begins when an RSVP daemon consults the local routing protocol(s) to obtain routes. A host sends IGMP messages to join a multicast group and RSVP messages to reserve resources along the delivery path(s) from that group. Each router that is capable of participating in resource reservation passes incoming data packets to a packet classifier and then queues them as necessary in a packet scheduler. The RSVP packet classifier determines the route and QoS class for each packet. The RSVP scheduler allocates resources for transmission on the particular data link layer medium used by each interface. If the data link layer medium has its own QoS management capability, the packet scheduler is responsible for negotiation with the data-link layer to obtain the QoS requested by RSVP.

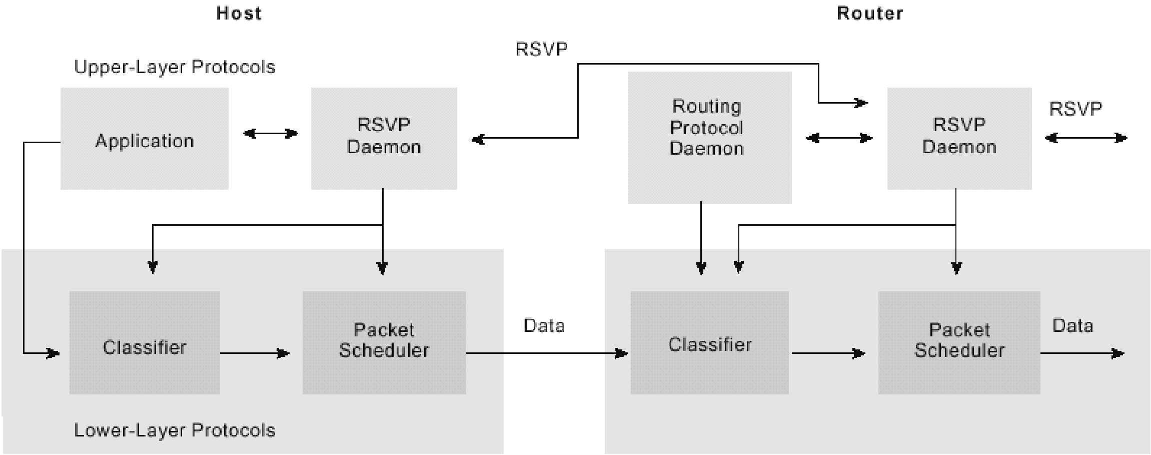


Fig24:RSVP operational environment

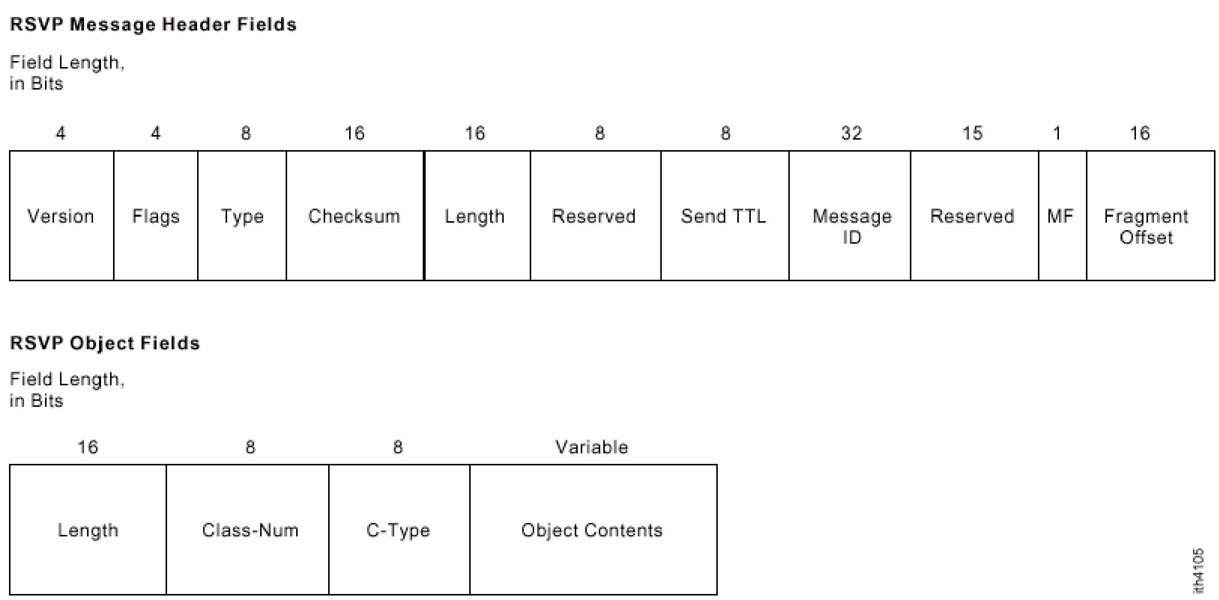
The scheduler itself allocates packet-transmission capacity on a QoS-passive medium, such as a leased line, and also can allocate other system resources, such as CPU time or buffers. A QoS request, typically originating in a receiver host application, is passed to the local RSVP implementation as an RSVP daemon. The RSVP protocol then is used to pass the request to all the nodes (routers and hosts) along the reverse data path(s) to the data source(s). At each node, the RSVP program applies a local decision procedure called admission control to determine whether it can supply the requested QoS. If admission control succeeds, the RSVP program sets the parameters of the packet classifier and scheduler to obtain the desired QoS. If admission control fails at any node, the RSVP program returns an error indication to the application that originated the request.

RSVP TUNNELLING: It is impossible to deploy RSVP or any new protocol at the same moment throughout the entire RSVP therefore must provide correct protocol operation even when two RSVP-capable routers are joined by an arbitrary cloud of non-RSVP routers. An intermediate cloud that does not support RSVP is unable to perform resource reservation, so service guarantees cannot be made. If, however, such a cloud has sufficient excess capacity, it can provide acceptable and useful real-time service. To support connection of RSVP networks through non-RSVP networks, RSVP supports tunneling. Tunneling requires RSVP and non-RSVP routers to forward path messages toward the destination address by using a local routing table. When a path message traverses a non-RSVP cloud, the path-message copies carry the IP address of the last RSVP-capable router. Reservation-request messages are forwarded to the next upstream RSVP-capable router.

RSVP MESSAGES: RSVP supports four basic message types:

- **Reservation-request messages:** A reservation-request message is sent by each receiver host toward the senders. A reservation-request message must be delivered to the sender hosts so that the hosts can set up appropriate traffic control parameters for the first hop.
- **Path messages**: An RSVP path message is sent by each sender forward along the unicast or multicast routes provided by the routing protocol(s). A path message is used to store the path state in each node.
- **Error and confirmation messages:** Three error and confirmation message forms exist namely path-error messages, reservation-request error messages, and reservation-request acknowledgment messages.
- **Teardown messages**: RSVP teardown messages remove the path and reservation state without waiting for the cleanup timeout period. Teardown messages can be initiated by an application in an end system or a router as the result of state timeout. RSVP supports two types of teardown messages-path-teardown messages and reservation-request teardown messages.

RSVP PACKET FORMAT:



RSVP Message Header Fields

Field Length, in Bits

4	4	8	16	16	8	8	32	15	1	16
Version	Flags	Type	Checksum	Length	Reserved	Send TTL	Message ID	Reserved	MF	Fragment Offset

RSVP Object Fields

Field Length, in Bits

16	8	8	Variable
Length	Class-Num	C-Type	Object Contents

Fig25: RSVP packet format

Some of the important RSVP message-header fields are:

- **Type**: 8-bit field with 6 possible (integer) values, as shown below.

Value	Message Type
1	Path
2	Reservation-request
3	Path-error
4	Reservation-request error
5	Path-teardown
6	Reservation-teardown
7	Reservation-request acknowledgment

- **Length**: 16-bit field representing the length of this RSVP packet in bytes, including the common header and the variable-length objects that follow. If the More Fragment (MF) flag is set or the fragment offset field is non-zero, this is the length of the current fragment of a larger message.
- **Send TTL**: 8-bit field indicating the IP time-to-live (TTL) value with which the message was sent.
- **Message ID**: 32-bit field providing a label shared by all fragments of one message from a given next/previous RSVP hop.

RSVP object fields are comprised of the following:

- **Length**: 16-bit field containing the total object length in bytes (must always be a multiple of 4 and be at least 4).
- **Class-Num**: Identifies the object class. Each object class has a name. The high-order bit of the Class-Num determines what action a node should take if it does not recognize the Class-Num of an object.
- **C-Type**: Object type, unique within Class-Num. The maximum object content length is 65528 bytes.
- **Object Contents**: The Length, Class-Num, and C-Type fields specify the form of the object content.

ROLE OF RSVP: We can exploit the principles of RSVP to create new ways of dynamic provisioning using in-band signaling. In particular, classes of aggregated flows may be defined between edge routers. Resource provisioning for a class of flows with similar QoS requirements can be performed by using RSVP signaling from the edge routers. As the need changes, the resources reserved over a path of aggregated flows may be increased or decreased using RSVP messages. The IP side of a gateway also acts as an edge router. It can use RSVP signaling to set up virtual trunk groups with other gateways. Voice connections arriving at a gateway are assigned to this trunk group if enough spare bandwidth, or free trunk, is available on the trunk group. Based on the time of day variation in call volume and on the measurements of call blocking at the SG (for a given PCG), the RSVP may request an increase or decrease in bandwidth (trunk group size). This procedure is akin to DPVC paths in ATM. Thus, using RSVP we can support the differentiated QoS model on an inherently best-effort IP network.

15.g. ROLE OF MULTIPROTOCOL LABEL SWITCHING (MPLS):

MPLS is not directly a part of the voice over IP family of protocols. However, its use, especially in conjunction with RSVP, greatly improves performance. IP-based routers typically forward a packet according to its destination IP address. There is a growing effort to reduce the routing burden and improve the traffic engineering capability of IP networks by using switched paths through these networks. Some approaches that may provide these switched paths are IP switching, multiprotocol over ATM (MPOA) and MPLS. Switching is faster than routing as it operates in hardware at layers 1&2 as opposed to the latter which is software implemented at layer 3.

MPLS with explicit routes is of particular importance to the internet voice service. MPLS can assign a label to each link of an explicit path between an ingress/egress pair of IP switches/routers in a subnetwork and can associate a collection of flows to this path. The ingress router/switch will prefix IP packets for any flow in this collection with the MPLS label before forwarding the packets to the next router/switch. Intermediate routers/switches use only the label (s) to forward the packets until they reach the egress router/switch, where normal forwarding methods can be used. Besides speeding up the forwarding function, MPLS adds tremendous flexibility to routing in the network. Elements such as explicit route selection and QoS-based routing are easier to implement with label switching than with conventional IP routing. Of course, labels have to be selected and distributed to all routers and edge devices before label switching can begin for an aggregated flow. Thus, like connection-oriented networking, label switching is only appropriate for long-lived flows, such as voice connections, video connections, and long file transfers. Because routers can provide both the usual datagram forwarding and label switching, they can take advantage of the ubiquitous IP infrastructure for routing short-lived flows while providing fast switching of long-lived flows, at least in some segment of the end-to-end path. Of course, when a collection of flows is grouped for label assignment, only the collection needs to be long lived.

In the general situation, the MPLS label needs additional overhead on top of the IP header overhead. However, for integrated voice services, MPLS can actually reduce the overhead significantly. Since the voice traffic does not originate from IP endpoints for these services, voice packets do not start out with an IP header. With a label-switching capability, however, we can set up an explicit label-switched path between a pair of gateways over an IP backbone. Provisioning servers interact with the MPLS path setup function to coordinate the path setup and bandwidth provisioning. Instead of encapsulating each voice packet first in RTP/UDP/IP headers and then in an MPLS header, the gateways encapsulate the voice packet in the MPLS label only, eliminating the need for expensive RTP/UDP/IP overhead. Since forwarding on this path is based only on the MPLS header, the missing headers are not relevant. At the egress gateway, the MPLS label is removed, and voice samples are extracted for transport over an STM network. The RTP/UDP/IP headers are not used in MPLS, so the overhead can be reduced from 49 bytes to about 13 bytes per voice packet (4 to 6 bytes of MPLS overhead, and 7 to 9 bytes of PPP overhead). Using MPLS would contribute tremendously to the efficiency of transporting voice over an IP network.

SECTION III

16. IP TELEPHONE DESIGN:

16.a REFERENCE DESIGN:

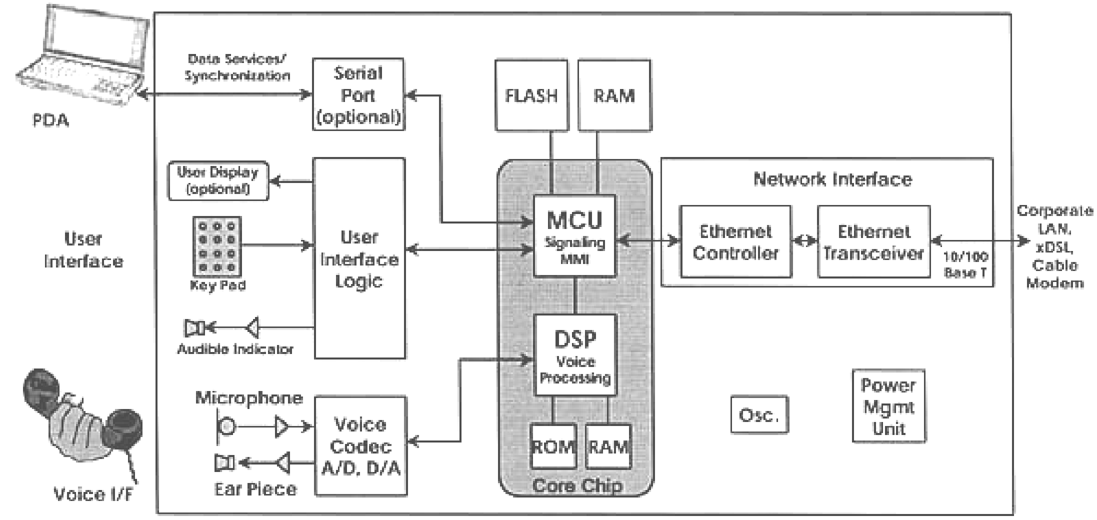


Fig26:IP Phone reference design

An IP Telephone consists of the following components: User Interface, Voice Interface, Network Interface, and Processor Core and associated logic.

The User Interface: provides the traditional user interface functions of a telephone. At a minimum, this consists of a keypad for dialing numbers (0-9, *, #) and an audible indicator for announcing incoming calls to the user. On more sophisticated telephone sets, additional keys are provided for features such as mute, redial, hold, transfer, conferencing, etc. A display is also typically provided for displaying user prompts, number dialed, CallerID information for incoming calls, etc. In certain models, the telephone will be equipped with a serial interface to allow communications to a device such as a PDA to allow synchronization of phone information, facilitate automatic dialing, etc.

The Voice Interface: provides the conversion of analog voice into digital samples. Speech signals from the microphone are sampled at a rate of 8 KHz to create a digitized 64kbps data stream to the processor via a pulse code modulation (PCM) codec. Similarly, the processor passes a 64kbps data stream in the return path to the speaker through the PCM codec to convert digital samples back into speech.

The Network Interface: allows transmission and reception of voice packets from/to the telephone. For corporate LANs this is most often either 10BaseT or 100BaseT Ethernet running TCP/IP protocols. The IP Telephone may offer a second RJ-45 Ethernet connector to allow a PC to plug in and share one connection to the wall jack.

The Processor Core: performs the voice processing, call processing, protocol processing, and network management software functions of the telephone. This consists of a Digital Signal Processor (DSP) for the voice-related functions and a Micro Controller Unit (MCU) for the remaining functions. To ensure software upgradeability the telephone will make use of Flash memory.

16.b. SOFTWARE ARCHITECTURE:

The software consists of the following major subsystems: User Interface, Voice Processing, Telephony Signaling Gateway, Network Interface Protocols, Network Management Agent, and System Services.

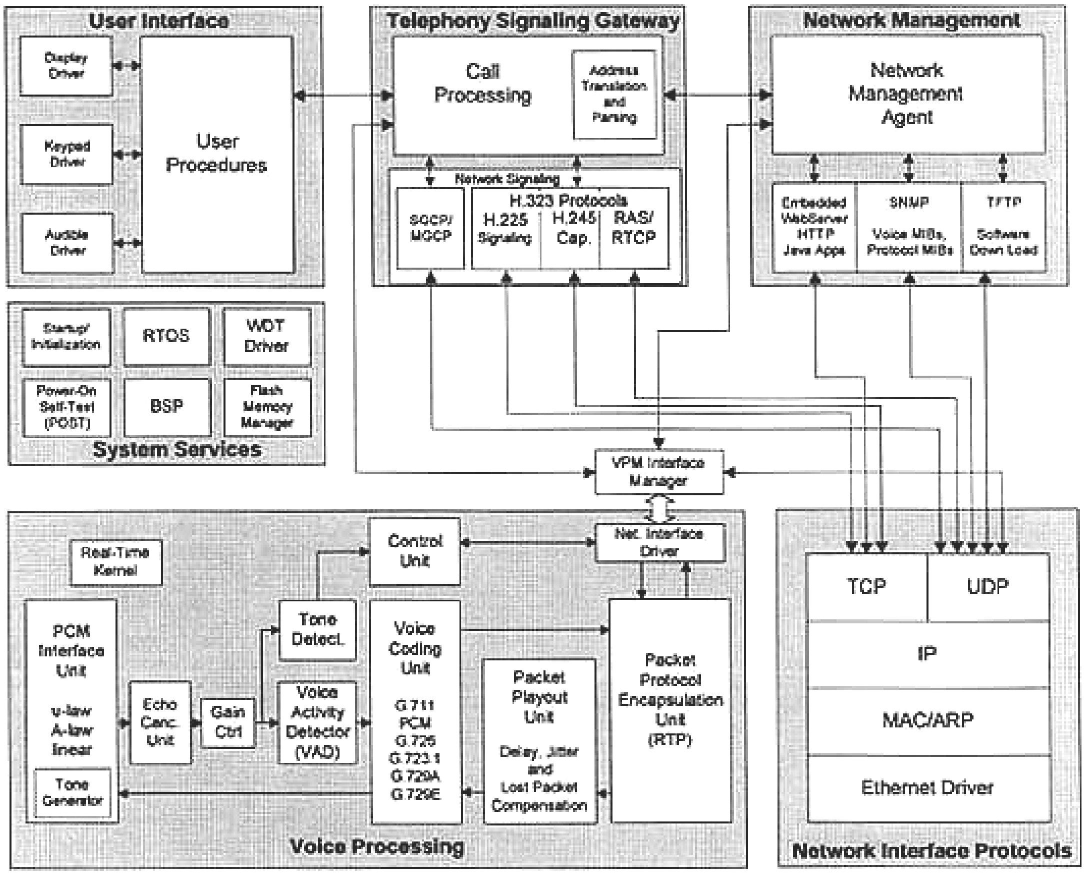


Fig27:IP Phone Software architecture

User Interface: provides the software components that handle the interface to the user of the IP Telephone and consists of the following software modules:

- Display Driver: Controls the hardware that generates characters to the display
- Keypad Driver: Performs keypad scanning and debounces key presses entered by the user.
- AudibleDriver: Performs control of the hardware that generates ringing to the user.
- UserProcedures: Controls the information displayed by the Display Driver and processes user key inputs and converts them into primitives for Call Processing.

Voice processing: The Voice Processing software is composed of the following software modules:

- PCM Interface Unit: Receives PCM samples from the analog interface and forwards then to the appropriate DSP software module for processing
- Tone Generator
- Echo Canceller Unit
- Voice Activity Detector
- Voice Codec Unit: Performs packetization of the 64 kbps data stream received from the user
- Packet Playout Unit
- RTP encapsulation Unit
- Voice Encryption Unit
- Control Unit: Coordinates the exchange of monitor and control information between the Voice Processing Module and other modules

Telephony signaling gateway : The Telephony Signaling Gateway (TSG) subsystem performs the functions for establishing, maintaining and terminating a call. It could use SIP-SDP-SAP, MGCP, SGCP, H.323 or other signaling and control protocols.

Network Management: The Network Management subsystem supports remote administration of the IP Telephone by a Network Management System. The Network Management Agent consists of the following software modules:

- Network Management Agent: Performs the network management functions of the IP Telephone
- Embedded Web Server: Provides administration support via a standard web browser
- It must support management protocols such as SNMP
- TFTP (to download software updates into flash memory)
- Transport protocols namely TCP & UDP
- IP
- MAC & ARP protocols
- Ethernet or other datalink-physical drivers

System Services: System Services consists of the following software modules:

- Startup/Initialization: Provides startup and initialization of the hardware and software components of the IP Telephone.
- POST: Provides Power-On Self-Test (POST) functions of the IP Telephone.
- RTOS: The Real-Time Operating System (RTOS) provides functions such as task management, memory management and task synchronization.
- BSP: Board Support Package (BSP) software provides hardware interface drivers, interrupt vectors, etc. that allow the real-time operating system to operate on the target hardware platform.
- Antilocking mechanism: to prevent the IP Telephone from locking up due to a software or intermittent hardware failure.

- Flash Memory Manager: Provides functions for reading/write data from/to the Flash memory.
- DSP Interface Manager: Provides the driver for exchanging information between the MCU and DSP, including software download, voice packets and network management functions.

17. HANDS-ON EXPERIENCE:

Using the simulated environment provided by the CISCO VoIP network simulator we studied the configuration of CISCO 3640 routers for:

- Analog voice over IP
- Voice over IP with E&M signaling
- Digital Voice over IP
- Interactive voice response (IVR)

To do so we first need to study the CISCO Internetworking Operating System (IOS) command-line-interface (CLI).

17.a. CISCO IOS COMMAND LINE INTERFACE:

CLI ARCHITECTURE:

A Cisco IOS router command line interface can be accessed through either a console connection, modem connection, or a telnet session. Regardless of which connection method is used, access to the Cisco IOS command line interface is generally referred to as an EXEC session.

As a security feature, Cisco IOS separates EXEC sessions into two different access levels - user EXEC level and privileged EXEC level. For example, when an EXEC session is started, the router will display a "Router>" prompt. The right arrow (>) in the prompt indicates that the router is at the user EXEC level. The user EXEC level does not contain any commands that might control the operation of the router. To list the commands available at the user EXEC level, a question mark (?) must be typed at the Router> prompt.

To change to the privileged EXEC level, we must type "enable" at the Router> prompt. If an enable password is configured, the router will then prompt for that password. When the correct enable password is entered, the router prompt will change to "Router#" indicating that the user is now at the privileged EXEC level. To switch back to user EXEC level, we must type "disable" at the Router# prompt. Typing a question mark (?) at the privileged EXEC level will now reveal many more command options than those available at the user EXEC level. The text below illustrates the process of changing EXEC levels.

```
Router> enable
Password: [enable password]
Router# disable
Router>
```

Once an EXEC session is established, commands within Cisco IOS are hierarchically structured.

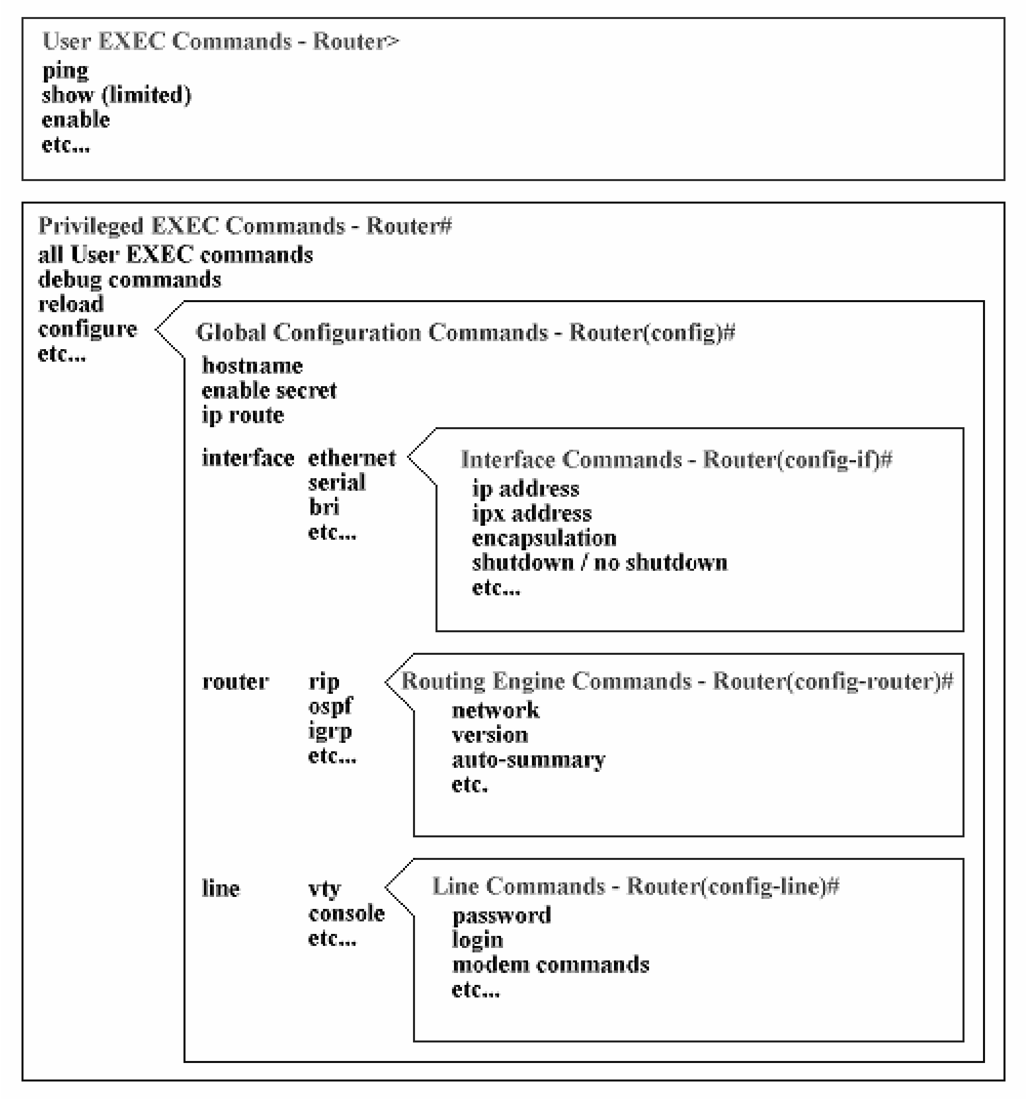


Fig28:Cisco IOS CLI hierarchy

<u>*CLI EDITOR FEATURES:*</u>

Context Sensitive Help:

Cisco IOS CLI offers context sensitive help. This is a useful tool for a new user because at any time during an EXEC session, a user can type a question mark (?) to get help. Two types of context sensitive help are available - word help and command syntax help.
Word help can be used to obtain a list of commands that begin with a particular character sequence. The following is an example of word help:

```
Router# co?
configure connect copy
```

Command syntax help can be used to obtain a list of command, keyword, or argument options that are available based on the syntax the user has already entered. The router will then display a list of available command options with <cr> standing for carriage return. The following is an example of command syntax help:

```
Router# configure ?
  memory              Configure from NV memory
  network             Configure from a TFTP network host
  overwrite-network    Overwrite NV memory from TFTP network
host=20
  terminal            Configure from the terminal
  <cr>
```

Command Syntax Check:

If a command is entered improperly, the router will inform the user and indicate where the error has occurred. A caret symbol (^) will appear underneath the incorrect command, keyword, or argument. The following example displays what happens if the keyword "ethernet" is spelled incorrectly.

```
Router(config)#interface ethernat
                           ^
% Invalid input detected at '^' marker.
```

Command Abbreviation:

Commands and keywords can be abbreviated to the minimum number of characters that identifies a unique selection. Eg: you can abbreviate the "configure" command to "conf" because "configure" is the only command that begins with "conf".

Hot Keys:

For many editing functions, the Cisco IOS CLI editor provides hot keys. The following table lists some editing shortcuts that are available.

Delete	- Removes one character to the right of the cursor.
Backspace	- Removes one character to the left of the cursor.
TAB	- Finishes a partial command.
Ctrl-A	- Moves the cursor to the beginning of the current line.
Ctrl-R	- Redisplays a line.
Ctrl-U	- Erases a line.
Ctrl-W	- Erases a word.
Ctrl-Z	- Ends configuration mode and returns to the EXEC.
Up Arrow	- Allows user to scroll forward through former commands.
Down Arrow	- Allows user to scroll backward through former commands.

ROUTER CONFIGURATION:

Entering Configurations:

Perhaps the best way to illustrate Cisco IOS CLI navigation is by walking through a simple router configuration. The comments in the example do not attempt to explain the meaning of each individual command, but rather intend to display where configuration commands are entered within the Cisco IOS command structure. Global parameters must be configured at the global configuration level (indicated by the "Router(config)#" prompt) whereas interface specific commands are entered after switching to the particular interface (indicated by the "Router(config-if)#" prompt).

Router> enable	- switches to privileged EXEC level
Router# configure terminal	- switches to global configuration level
Router(config)# enable secret cisco	- configures router with an enable secret (global)
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3	- configures a static IP route (global)
Router(config)# interface ethernet0	- switches to configure the ethernet0 interface
Router(config-if)# ip address 10.1.1.1 255.0.0.0	- configures an IP address on ethernet0 (interface)
Router(config-if)# no shutdown	- activates ethernet0 (interface)
Router(config-if)# exit	- exits back to global configuration level
Router(config)# interface serial0	- switches to configure the serial0 interface
Router(config-if)# ip address 20.2.2.2 255.0.0.0	- configures an IP address on serial0 (interface)
Router(config-if)# no shutdown	- activates serial0 (interface)
Router(config-if)# exit	- exits back to global configuration level
Router(config)# router rip	- switches to configure RIP routing engine
Router(config-router)# network 10.0.0.0	- adds network 10.0.0.0 to RIP engine (routing engine)
Router(config-router)# network 20.0.0.0	- adds network 20.0.0.0 to RIP engine (routing engine)
>Router(config-router)# exit	- exits back to global configuration level
Router(config)# exit	- exits out of configuration level
Router# copy running-config startup-config	- saves configuration into NVRAM
Router# disable	- disables privileged EXEC level
Router>	- indicates user is back to user EXEC level

As seen the exit command is used to back up a level within the Cisco IOS hierarchy

Taking Interfaces Out Of Shutdown:

Routers ship from the factory with all interfaces deactivated. Deactivated interfaces are referred to as being in a shutdown state. Before an interface can be used, it must be taken out of the shutdown state. To take an interface out of shutdown, one must type "no shutdown" at the appropriate interface configuration level. The example above includes these commands for both the ethernet and serial interfaces.

Removing Commands / Resetting Default Values:
Cisco IOS provides an easy way to remove commands from a configuration. To remove a command from the configuration, simply navigate to the proper location and type "no" followed by the command to be removed. Some configuration commands in Cisco IOS are enabled by default and assigned a certain default value. When left at the default value, these commands will not be displayed when the configuration is listed. If the value is altered from the default setting, issuing a "no" form of the command will restore the value to the default setting.

Saving Configurations:
A Cisco IOS router stores configurations in two locations - RAM and NVRAM. The running configuration is stored in RAM and is used by the router during operation. Any configuration changes to the router are made to the running-configuration and take effect immediately after the command is entered. The startup-configuration is saved in NVRAM and is loaded into the router's running-configuration when the router boots up. If a router loses power or is reloaded, changes to the running configuration will be lost unless they are saved to the startup-configuration. To save the running-configuration to the startup configuration, we must type the following from privileged EXEC mode.

```
Router# copy running-config startup-config
Router#write memory
```

<u>ROUTER MANAGEMENT</u>:

Cisco IOS supports many different types of show commands. This section covers a few of the common show commands used to both manage and troubleshoot a router.

Displaying Configurations:
To display the running-configuration, we type the following command in privileged EXEC mode:

```
Router#show running-config
```

To display the startup-configuration that is stored in NVRAM, type the following command in privileged EXEC mode:

```
Router#show startup-config
```

The following is the show running-config output from the example used in the Router Configuration section.

```
Current configuration:
version 11.2
hostname cisco
enable password cisco
interface Ethernet0
 ip address 10.1.1.1 255.0.0.0
interface Serial0
 ip address 20.2.2.2 255.0.0.0
```

```
router rip
 network 10.0.0.0
 network 20.0.0.0
ip route 0.0.0.0 0.0.0.0 20.2.2.3
line vty 0 4
 password telnet
 login
end
```

When displaying a configuration, exclamation marks (!) (not shown above) function as line separators to make reading easier. We can also see how commands entered at the interface configuration level appear indented underneath the respective interface. This type of display allows a user to easily identify which configuration parameters are set at the global configuration level and which are set at the various configuration sub-levels.

Displaying Software Version And More:

The *show version command* provides a lot of information in addition to the version of software that is running on the router. The following information can be collected with the show version command:

Software Version	- Cisco IOS software version (stored in flash)
Bootstrap Version	- Bootstrap version (stored in Boot ROM)
System up-time	- Time since last reboot
System restart info	- Method of restart (e.g. power cycle, crash)
Software image name	- Cisco IOS filename stored in flash
Router Type and Processor type	- Model number and processor type
Memory type and allocation (Shared/Main)	- Main Processor RAM - Shared Packet I/O buffering
Software Features	- Supported protocols / feature sets
Hardware Interfaces	- Interfaces available on router
Configuration Register	- Bootup specifications, console speed setting, etc.

The following is a sample output of a show version command.

```
Router# show version
Cisco Internetwork Operating System Software
IOS (tm) 3600 Software (C3640-J-M), Version 11.2(6)P, SHARED PLATFORM,
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1997 by cisco Systems, Inc.
Compiled Mon 12-May-97 15:07 by tej
Image text-base: 0x600088A0, data-base: 0x6075C000

ROM: System Bootstrap, Version 11.1(7)AX [kuong (7)AX], EARLY
DEPLOYMENT
RELEASE SOFTWARE (fc2)

Router uptime is 1 week, 1 day, 38 minutes
System restarted by power-on
System image file is "flash:c3640-j-mz_112-6_P.bin", booted
via flash
Host configuration file is "3600_4-confg", booted via tftp
from 171.69.83.194

cisco 3640 (R4700) processor (revision 0x00) with 107520K/23552K bytes
of memory.
Processor board ID 03084730
R4700 processor, Implementation 33, Revision 1.0
Bridging software.
SuperLAT software copyright 1990 by Meridian Technology Corp).
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
TN3270 Emulation software.
Primary Rate ISDN software, Version 1.0.
2 Ethernet/IEEE 802.3 interface(s)
97 Serial network interface(s)
4 Channelized T1/PRI port(s)
DRAM configuration is 64 bits wide with parity disabled.
125K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102
```

Displaying Interface States:

To view information about a particular interface, we use the *show interface* command. The show interface command provides the following list of important information:

Interface State (e.g. UP, DOWN, LOOPED)

Protocol addresses

Bandwidth

Reliability and Load

Encapsulation type

Packet Rates

Error Rates

Signaling Status (i.e. DCD,DSR,DTR,RTS,CTS)

The following is an example of a "show interface serial0" output:

```
Router#show interface serial 0
Serial0 is up, line protocol is down
Hardware is QUICC Serial
Internet address is 10.1.1.2/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
LMI enq sent 207603, LMI stat recvd 113715, LMI upd recvd 0, DTE LMI
down
LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
LMI DLCI 1023 LMI type is CISCO frame relay DTE
Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts
62856
Last input 1w, output 00:00:08, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/64/0 (size/threshold/drops)
```

```
Conversations 0/1 (active/max active)
Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 1000 bits/sec, 1 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
1012272 packets input, 91255488 bytes, 0 no buffer
Received 916 broadcasts, 0 runts, 0 giants
18519 input errors, 0 CRC, 17796 frame, 0 overrun, 0 ignored, 723 abort
283132 packets output, 13712011 bytes, 0 underruns
0 output errors, 0 collisions, 31317 interface resets
0 output buffer failures, 0 output buffers swapped out
3 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

17.b. CONFIGURING ANALOG VOICE OVER IP:

CISCO 3640 ROUTERS:

VoIP is primarily a software feature; however, to use this feature on a Cisco 3600 series router, you must install a voice network module (VNM). The VNM holds either one or two voice interface cards (VIC). Each two-port VIC is specific to a particular signaling type associated with a voice port type, such as FXS, FXO, or E&M analog voice ports.

A Cisco 3640 can house up to three modules with up to a total of six VICs that slide into the voice/fax network modules and provide the interface to the telephony equipment.

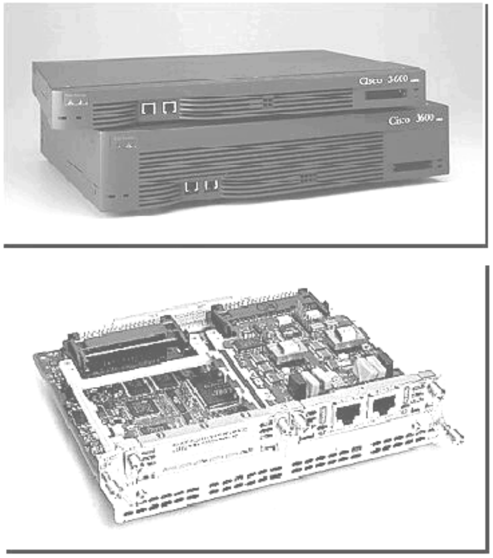


Fig29: CISCO 3640 router

OBJECTIVES:

To set up a basic FXS analog phone connection we needed to complete these objectives:

1. Set up IP connectivity between the serial interfaces of routers 3640-A and 3640-B.
2. Locate and configure the FXS voice port to use on 3640-A.
3. Set up the POTS and VoIP dial peers on 3640-A.
4. Locate and configure the FXS voice port to use on 3640-B.
5. Set up the POTS and VoIP dial peers on 3640-B.
6. Test the connection.

NETWORK TOPOLOGY & PARAMETERS:

Item	**Router 3640-A**	**Router 3640-B**

Serial 2/0 IP address	1.1.1.1	1.1.1.2
Subnet mask	255.255.255.0	255.255.255.0
Dial-peer phone number	555-1234	655-5678
FXS voice port number	1/0/0	1/0/0
Encapsulation	HDLC	HDLC

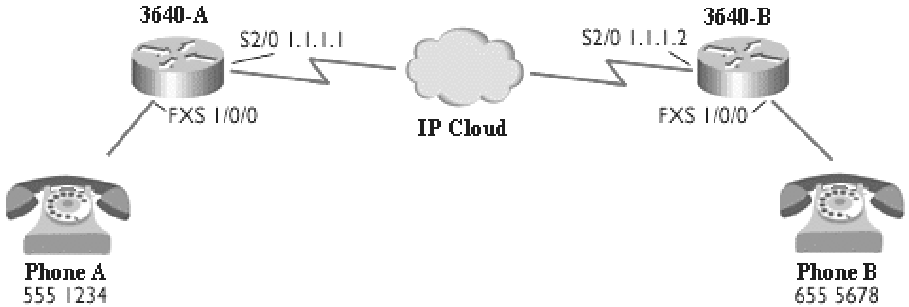


Fig30:Topology for analog VoIP

17.c. CONFIGURING VOICE OVER IP USING E&M SIGNALING:

OBJECTIVES:

To set up a basic analog E&M connection through a PBX and use the **prefix** command, we completed the following objectives:

1. Review and test current configurations on routers 3640-A and 3640-B.
2. Set up the dial peers on routers 3640-A and 3640-B.
3. Test the voice configuration.
4. Practice debugging a successful call.

In this exercise, basic IP connectivity was already established, and the main goal was to configure the necessary voice features so that we could successfully complete a phone call over VoIP.

NETWORK TOPOLOGY & PARAMETERS:

Router	**3640-A**	**3640-B**
Serial 2/0 IP Address	1.1.1.1	1.1.1.2
Subnet Mask	255.255.255.0	255.255.255.0
Phone Numbers for Connected Phones	Phone A 555-1234	Phone B 655-5678
E&M Voice Port	3/0/0	3/0/0

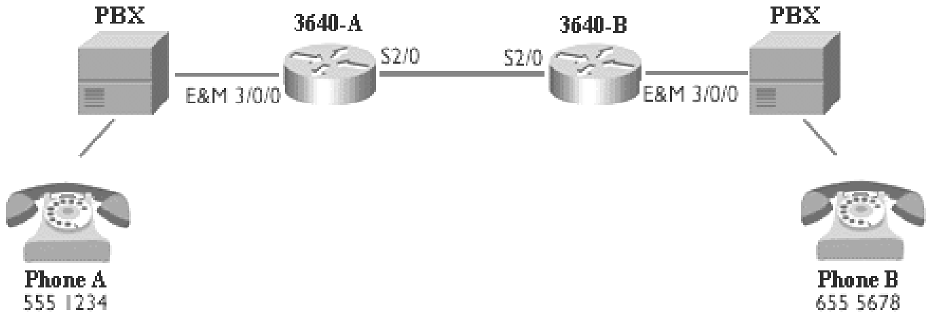


Fig31: Topology for VoIP using E&M

As the diagram shows, the two Cisco 3640 routers connected to PBXs via their E&M ports, and each PBX has a connected analog telephone.

17.d. CONFIGURING DIGITAL VOICE OVER IP:

HARDWARE REQUIRED:

A Cisco 2600 or 3600 series router can be configured for T1 connectivity with the use of a digital T1 packet voice trunk network module. A digital T1 packet voice trunk network module is made up of a network module with installed packet voice data modules (PVDMs), and one T1 multiplex trunk voice/WAN interface card with either one or two T1 ports, as illustrated below.

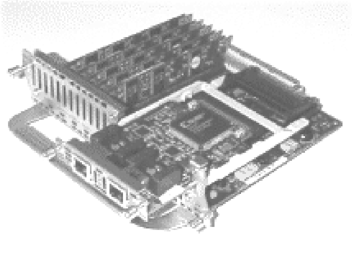
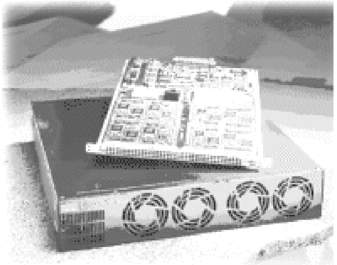


Fig32: Digital T1/E1 Packet Voice Trunk Network Module

The digital T1/E1 packet voice module uses a real-time CPU and powerful DSPs that provide high levels of voice quality, eliminating the processing burden from the main CPU of the router. This setup allows voice and fax traffic to travel efficiently across a user's WAN or directly over the PSTN.

AS5300 ACCESS SERVER:



The following voice features are available on the Cisco AS5300:

- Interfaces to PBXs and PSTN/digital switches via T1/channel-associated signaling (CAS), Q.931, and E1/R2
- VoIP, fax over IP, and dialup remote access server (modem/ISDN)
- H.323-compliant, CODEC support includes G.711, G.729a, G.723.1
- Echo cancellation and voice activity detection (VAD)
- Integrated IVR
- Direct inward dialing (DID) and two-stage dialing (account and PIN number entry)
- Cisco IOS® software, multiprotocol routing, quality of service (QoS), and Remote Access Dial-In Service (RADIUS)

OBJECTIVES:

To complete the exercise and set up a basic common channel signaling (CCS) digital Voice over IP (VoIP) connection, we needed to complete the following seven objectives:

1. Examine and understand the initial configurations on both devices.
2. Confirm the existence of IP connectivity on both devices.
3. Configure ISDN connectivity between the AS5300 and the private branch exchange (PBX) to which it is connected.
4. Configure the T1 controller on the AS5300.
5. Configure a set of dial peers so you can complete a call and a fax from the Cisco 3640 to the A5300.
6. Configure a set of dial peers so you can complete a call and a fax from the AS5300 to router 3640-A.
7. Test the VoIP connectivity, then run some debug and show commands. Place voice and fax calls from Phone A to Phone B and Phone B to Phone A.

NETWORK TOPOLOGY & PARAMETERS:

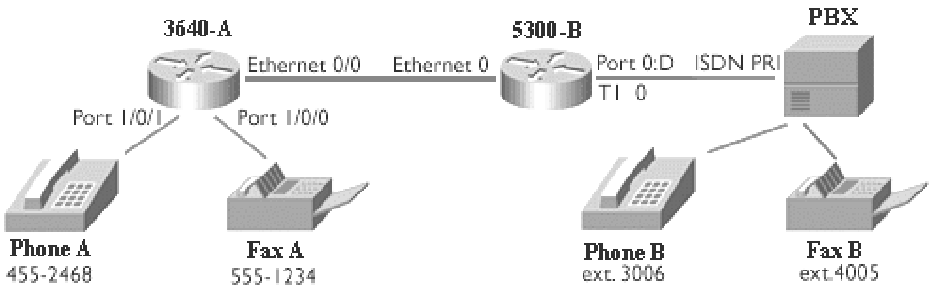


Fig33: Topology for Digital VoIP

Item	3640-A	5300-B

Ethernet Port and IP Address	Ethernet 0/0 10.2.1.1	Ethernet 0 10.2.1.2
Subnet Mask	255.255.255.0	255.255.255.0
Phone Dial-Peer Phone Number	Phone A 455-2468	Phone B 3006
Fax Dial-Peer Phone Number	Fax A 555-1234	Fax B 4005

17.e. CONFIGURING INTERACTIVE VOICE RESPONSE (IVR):

WHAT IS IVR?

IVR can be used to automate information access and retrieval using a touch-tone-telephone keypad as an input device. In its most basic application, prerecorded voice prompts are played in response to received DTMF tones. IVR provides the ability to:

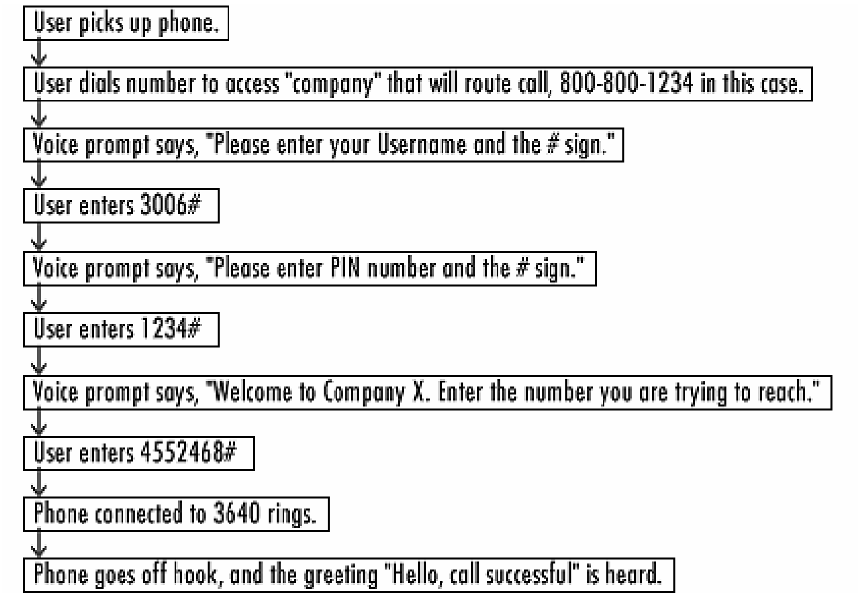
- Collect account numbers and/or personal identification numbers (PINs)
- Collect destination phone numbers
- Perform authorization, authentication, and accounting (AAA) tasks interacting with a variety of servers

OBJECTIVES:

To finish the IVR lab we had to complete the following objectives:

1. Examine and Understand Configurations of Both Routers.
2. Check IP connectivity between the Routers.
3. Review Supported Application Scripts.
4. Download the voice prompt files from the TFTP server to 5300-B.
5. Enable IVR feature; Define a Username and Password.
6. Examine **show** and **debug** command output.

In the IVR Configuration Lab, we configured IVR on router 5300-B to call the application script called: **clid_authen_collect**. The **clid_authen_collect** application causes a Cisco AS5300 to offer voice prompting for digit collection from the callers for authentication purposes, such as an account number and PIN number, and to identify the destination of a call. In this lab you will provide the correct account number and PIN number while going through the authentication process.



User picks up phone.

User dials number to access "company" that will route call, 800-800-1234 in this case.

Voice prompt says, "Please enter your Username and the # sign."

User enters 3006#

Voice prompt says, "Please enter PIN number and the # sign."

User enters 1234#

Voice prompt says, "Welcome to Company X. Enter the number you are trying to reach."

User enters 4552468#

Phone connected to 3640 rings.

Phone goes off hook, and the greeting "Hello, call successful" is heard.

Above shown is the call progress chart. We placed calls from Phone B to Phone A and heard the appropriate voice response prompts.

NETWORK TOPOLOGY & PARAMETERS:

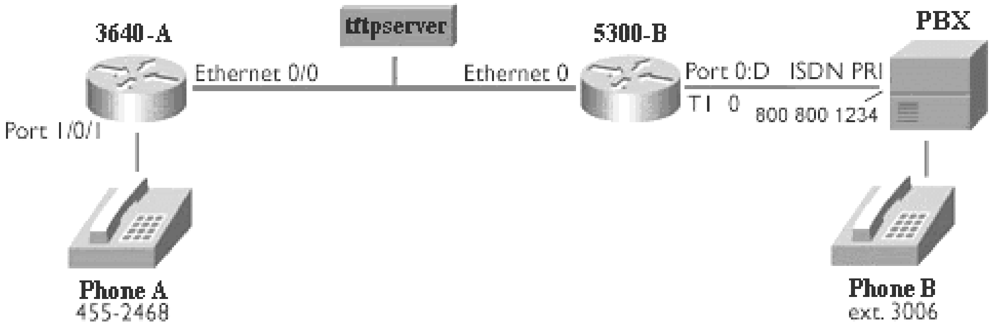


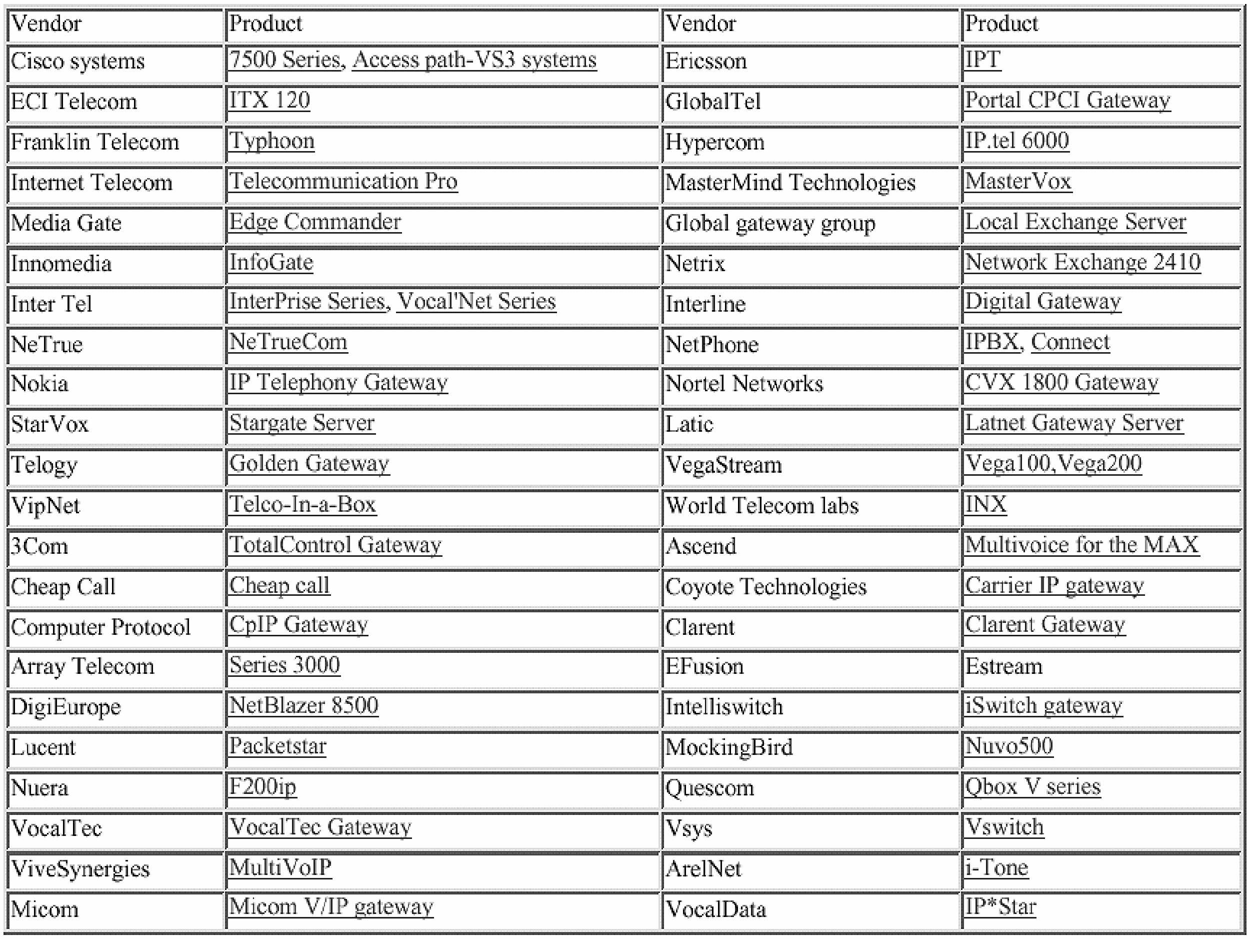
Fig34:Topology for IVR simulation

Item	Router 3640-A	Router 5300-B	TFTP Server
Ethernet Port and IP address	Ethernet 0/0 10.2.1.1	Ethernet 0 10.2.1.2	10.2.1.3
Subnet Mask	255.255.255.0	255.255.255.0	255.255.255.0
Dial-Peer Phone Number	Phone A 455-2468	Phone B 3006	N/A

SUPPLEMENT

18. PRODUCTS, VENDORS, SERVICES:

18.a. VoIP GATEWAY VENDORS:



Vendor	Product	Vendor	Product
Cisco systems	7500 Series, Access path-VS3 systems	Ericsson	IPT
ECI Telecom	ITX 120	GlobalTel	Portal CPCI Gateway
Franklin Telecom	Typhoon	Hypercom	IP.tel 6000
Internet Telecom	Telecommunication Pro	MasterMind Technologies	MasterVox
Media Gate	Edge Commander	Global gateway group	Local Exchange Server
Innomedia	InfoGate	Netrix	Network Exchange 2410
Inter Tel	InterPrise Series, Vocal'Net Series	Interline	Digital Gateway
NeTrue	NeTrueCom	NetPhone	IPBX, Connect
Nokia	IP Telephony Gateway	Nortel Networks	CVX 1800 Gateway
StarVox	Stargate Server	Latic	Latnet Gateway Server
Telogy	Golden Gateway	VegaStream	Vega100,Vega200
VipNet	Telco-In-a-Box	World Telecom labs	INX
3Com	TotalControl Gateway	Ascend	Multivoice for the MAX
Cheap Call	Cheap call	Coyote Technologies	Carrier IP gateway
Computer Protocol	CpIP Gateway	Clarent	Clarent Gateway
Array Telecom	Series 3000	EFusion	Estream
DigiEurope	NetBlazer 8500	Intelliswitch	iSwitch gateway
Lucent	Packetstar	MockingBird	Nuvo500
Nuera	F200ip	Quescom	Qbox V series
VocalTec	VocalTec Gateway	Vsys	Vswitch
ViveSynergies	MultiVoIP	ArelNet	i-Tone
Micom	Micom V/IP gateway	VocalData	IP*Star

- 3Com Corp., http://www.3com.com/
- ACT Networks, http://www.acti.com/
- Ascend Communications, Inc. http://www.ascend.com/
- Cisco Systems Inc., http://www.cisco.com/
- ITXC, http://www.itxc.com/intro.html?
- Franklin Telecommunications Corp., http://www.ftel.com/
- Inter-tel Inc., http://www.inter-tel.com/
- IPAXS Corp, http://www.ipaxs.com
- Lucent Technologies Inc., http://www.lucent.com/
- Mitel Corporation, http://www.mitel.com/
- Motorola Inc., http://www.mot.com/
- Netrix Corp., http://www.netrix.com/

- Netspeak Corp., http://www.netspeak.com/
- Nortel Networks, http://www.nortelnetworks.com/
- RADvision Ltd., http://www.radvision.com/
- Sonus Networks, Inc., http://www.sonusnetworks.com/
- Unisphere Networks, http://www.unispherenetworks.com
- VIVE Synergies, Inc., http://www.vive.com
- VocalTec Inc., http://www.vocaltec.com/

18.b. GATEKEEPER PRODUCTS:

- Ericsson H.323 Gatekeeper
- VocalTec Gatekeeper
- Nortel Networks IPConnect
- Elemedia H.323 gatekeeper GK2000S

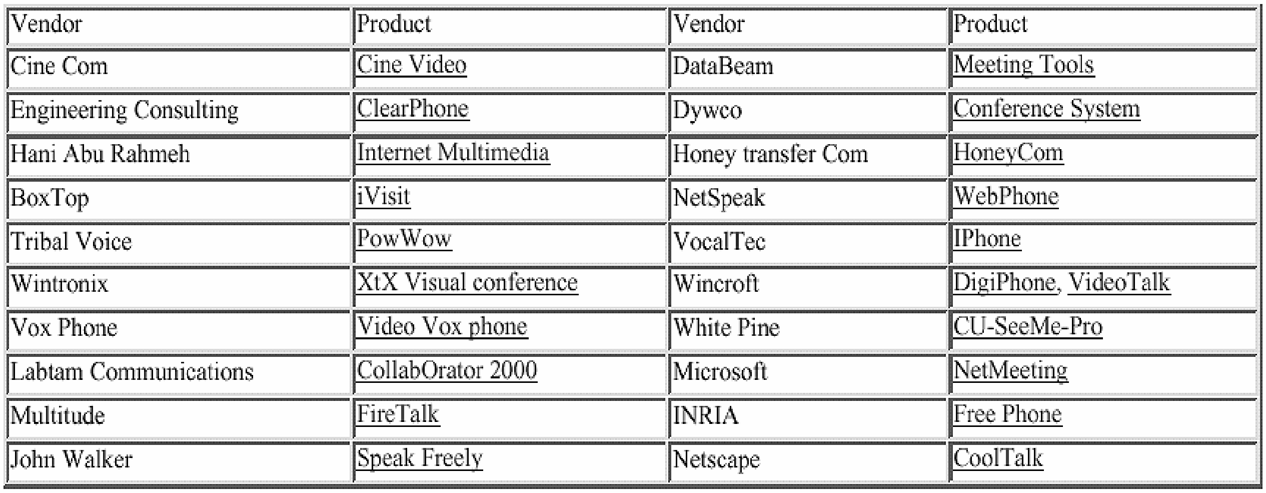
18.c. IP TELEPHONES:

- CISCO IP phones
- Selsius IP Phones
- Nokia Systems IPCourier

18.d. PC-BASED SOFTWARE PHONES:

- VocalTec Iphone
- Netscape’s Cooltalk
- White Pine’s CU-Seeme Pro
- Microsoft Netmeeting
- Motorola Vanguard series
- Lucent softswitch

18.e. GROUP CONFERENCE SOFTWARE PRODUCTS:



Vendor	Product	Vendor	Product
Cine Com	Cine Video	DataBeam	Meeting Tools
Engineering Consulting	ClearPhone	Dywco	Conference System
Hani Abu Rahmeh	Internet Multimedia	Honey transfer Com	HoneyCom
BoxTop	iVisit	NetSpeak	WebPhone
Tribal Voice	PowWow	VocalTec	IPhone
Wintronix	XtX Visual conference	Wincroft	DigiPhone, VideoTalk
Vox Phone	Video Vox phone	White Pine	CU-SeeMe-Pro
Labtam Communications	CollabOrator 2000	Microsoft	NetMeeting
Multitude	FireTalk	INRIA	Free Phone
John Walker	Speak Freely	Netscape	CoolTalk

18.f. PC TO PHONE SERVICES:

- VocalTec Surf&Call
- Dialpad.com

18.g. PC TO PC SERVICES:

- Microsoft Netmeeting
- VocalTec Iphone
- TaoTalk.com

18.h. PHONE TO PHONE SEVICES:

- AT&T
- America Online
- IDT Corporation
- AcculinQ
- USATEL
- Qwest
- Deutche Telecom in Germany
- France Telecom
- Sprint

18.i. SERVICES FOR SERVICE PROVIDERS:

- ITXC
- IP Telephony for carriers by DELTA Three & Ericsson
- CISCO AVVID (Architecture for Video, Voice & Integrated Data)

19. VoIP RELATED IETF WORKING GROUPS:

19.a. MULTIMEDIA WORKING GROUPS:

- IP Telephony (iptel), http://www.ietf.org/html.charters/iptel-charter.html
- Internet Fax (fax), http://www.ietf.org/html.charters/fax-charter.html
- PSTN and Internet Interworking (pint), http://www.ietf.org/html.charters/pint-charter.html
- Audio/Video Transport (avt), http://www.ietf.org/html.charters/avt-charter.html
- Multiparty Multimedia Session Control (mmusic), http://www.ietf.org/html.charters/mmusic-charter.html

19.b. MULTICAST WORKING GROUPS:

- MBONE Deployment (mboned), http://www.ietf.org/html.charters/mboned-charter.html
- Inter-Domain Multicast Routing (idmr), http://www.ietf.org/html.charters/idmr-charter.html
- Multicast Extensions to OSPF (mospf), http://www.ietf.org/html.charters/mospf-charter.html

- Protocol Independent Multicast (pim), http://www.ietf.org/html.charters/pim-charter.html
- Multicast Address Allocation (malloc), http://www.ietf.org/html.charters/malloc-charter.html

19.c. QUALITY OF SERVICE WORKING GROUPS:

- Differenticated Services (diffserv), http://www.ietf.org/html.charters/diffserv-charter.html
- Integrated Services (intserv), http://www.ietf.org/html.charters/intserv-charter.html
- Integrated Services over Specific Link Layers (issll), http://www.ietf.org/html.charters/issll-charter.html
- Resource Reservation (rsvp), http://www.ietf.org/html.charters/rsvp-charter.html
- RSVP Admission Policy (rap), http://www.ietf.org/html.charters/rap-charter.html
- IP Next Generation (ipngwg), http://www.ietf.org/html.charters/ipngwg-charter.html
- Multiprotocol Label Switching (mpls), http://www.ietf.org/html.charters/mpls-charter.html

20. ORGANISATIONS:

- International Multimedia Teleconferencing Consortium, http://www.imtc.org/
- IMTC FTP Site, http://www.imtc.org/u/u_ftp.htm
- Conferencing over IP (COIP) Forum - a working group of IMTC, http://www.imtc.org/act_coip.htm
- Voice On The Net (VON) Coalition, http://www.von.org/
- MIT Internet Telephony Consortium, http://itel.mit.edu/
- Enterprise Computer Telephony Forum (ECTF), http://www.ectf.org/
- Internet Fax Routing Forum
- Voice Profile for Internet Mail (VPIM) Work Group of EMA, http://www.ema.org/vpimdir/index.htm
- ITU http://www.itu.org
- IETF http://www.ietf.org

21. ITU STANDARDS:

- ITU-T H.320 Standards for Video Conferencing, http://www.imtc.org/h320.htm
- H.323 ITU Standards, http://www.imtc.org/h323.htm
- H.324 ITU Standards, http://www.imtc.org/h324.htm
- VPIM Technical Specification, http://www.ema.org/vpimdir/specs.html
- Simple Computer Telephony Protocol Home Page, http://www.phonezone.com/sctp/index.htm

REFERENCES

22. TECHNICAL PAPERS:

- Voice over IP: Protocols & Standards- Rakesh Arora
- Migrating Corporate voice traffic to the Data network- Quintum Technologies Inc.
- Bridging the gap to IP Telephony- Paul G., A.Sijben et al
- Voice and multiservice Network Design over ATM & IP networks- Zheng Chen et al.
- Protocols, Performance, and Controls for Voice Over Wide Area Packet networks- Bharat T. Doshi et al.
- H.323 Tutorial- Trillium Digital Systems
- Comparison of H.323 & SIP for IP Telephony-Ismail Dalgic et al
- Programming Internet Telephony Services- Henning Schulzrinne, Jonathan Rosenberg et al.
- SIP Telephony gateway on DTM- Mattias Eriksson, Lars Lundstedt
- The session initiation protocol: Advanced telephony services over the Internet- Henning Schulzrinne
- Resource Reservation protocol-CISCO
- Configuring CISCO Voice over IP for the CISCO 3600 Series-CISCO
- QoS Networking –CISCO
- The Pathstar Access Server-John M.Fossaceca et al.
- CISCO IP telephony network design guide-CISCO
- Voice over ATM-Mahdi S.Chambers et al.
- Interaction of call setup and resource reservation protocols in Internet telephony- Henning Schulzrinne et al.
- Signalling for Internet Telephony-Henning Schulzrinne et al.
- IP telephony gateways- Gonzalo Camarillo
- An application service architecture for communication services-Dynamicsoft
- H.323 tutorial- Dynamicsoft
- IP telephone design & implementation Issues- Telogy networks
- IP telephony and the fusion gateway platform- Natural Microsystems
- Voice over IP cookbook- CISCO
- H.323 and associated protocols- Asim Karim
- Pulse code modulation- Data Compression Reference Center
- News articles from Pulver.com
- Imerge Centrex feature gateway- AG Communications systems
- Future of Internet Telephony- Bill Griffin
- Adaptive Jitter Buffer Management for VoIP - Mapletree Networks.
- CG6000C: The platform for a new era of communications – Natural Microsystems.

- Digital Loop Carrier Solutions for Voice and Data Networks – Hoo Yin Khoe, A.Craig Bolling, Arupjyoti Bhuyan.
- Echo Cancellation for Voice over IP – Mapletree Networks.
- Echo Cancellation: Providing performance for PSTN and IP telephony applications – Natural Microsystems.
- Emerging trends in Signaling System #7 (Whitepaper) – RadiSys Inc.
- Additional extensions to G.729 in perspective – Sipro Lab Telecom Inc, Canada.
- Carrier class, High density VoP (Whitepaper) – W.E.Witowsky, Dennis.R.Gatens.
- An introduction to Cisco Open Packet Telephony for service providers – Cisco Systems.
- Packet voice Networking – Cisco Systems.
- Packet voice Primer – Cisco Systems.
- µ-Law Compression – Texas Instruments Inc.
- Simultaneous Voice and Data technology – AT&T Paradyne.
- Switch and Router design – PMC Sierra Inc.
- Voice over IP on Linux – www.techguide.com
- Voice over Cable (Whitepaper) – Edward Morgan, Debbie Greenstreet (Telogy n/w)
- Integrated voice and data service – SITA (www.sita.com)
- Voice LAN – www.techguide.com
- Voice over IP – Telogy Networks.

23. INTERNET RFCs:

- RFC 3054, Megaco IP Phone Media Gateway Application Profile. P. Blatherwick, R. Bell, P. Holland. January 2001. ftp://ftp.isi.edu/in-notes/rfc3054.txt
- RFC 3050, Common Gateway Interface for SIP. J. Lennox, H. Schulzrinne, J. Rosenberg. January 2001. ftp://ftp.isi.edu/in-notes/rfc3050.txt
- RFC 3015, Megaco Protocol 1.0. F. Cuervo, N. Greene, A. Rayhan, C. Huitema, B. Rosen, J. Segers. November 2000. ftp://ftp.isi.edu/in-notes/rfc3015.txt
- RFC 2995, Pre-Spirits Implementations of PSTN-initiated Services. H. Lu, Editor, I. Faynberg, J. Voelker, M. Weissman, W. Zhang, S. Rhim, J. Hwang, S. Ago, . Hwang, S. Ago, S. Moeenuddin, S. Hadvani, S. Nyckelgard, J. Yoakum, L. Robart. November 2000. ftp://ftp.isi.edu/in-notes/rfc2995.txt
- RFC 2976, The SIP INFO Method. S. Donovan. October 2000. ftp://ftp.isi.edu/in-notes/rfc2976.txt
- RFC 2974, Session Announcement Protocol. M. Handley, C. Perkins, E. Whelan. October 2000. ftp://ftp.isi.edu/in-notes/rfc2974.txt
- RFC 2897, Proposal for an MGCP Advanced Audio Package. D. Cromwell. August 2000. ftp://ftp.isi.edu/in-notes/rfc2897.txt
- RFC 2880, Internet Fax T.30 Feature Mapping. L. McIntyre, G. Klyne. August 2000. ftp://ftp.isi.edu/in-notes/rfc2880.txt
- RFC 2879, Content Feature Schema for Internet Fax (V2). G. Klyne, L. McIntyre. August 2000. ftp://ftp.isi.edu/in-notes/rfc2879.txt

- RFC 2871, A Framework for Telephony Routing over IP. J. Rosenberg, H. Schulzrinne. June 2000. ftp://ftp.isi.edu/in-notes/rfc2871.txt
- RFC 2833, RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals. H. Schulzrinne, S. Petrack. May 2000. ftp://ftp.isi.edu/in-notes/rfc2833.txt
- RFC 2824, Call Processing Language Framework and Requirements. J. Lennox, H. Schulzrinne. May 2000. ftp://ftp.isi.edu/in-notes/rfc2824.txt
- RFC 2806, URLs for Telephone Calls. A. Vaha-Sipila. April 2000. ftp://ftp.isi.edu/in-notes/rfc2806.txt
- RFC 2805, Media Gateway Control Protocol Architecture and Requirements. N. Greene, M. Ramalho, B. Rosen. April 2000. ftp://ftp.isi.edu/in-notes/rfc2805.txt
- RFC 2658, RTP Payload Format for PureVoice(tm) Audio. K. McKay. August 1999. ftp://ftp.isi.edu/in-notes/rfc2658.txt
- RFC 2423, VPIM Voice Message MIME Sub-type Registration. G. Vaudreuil, G. Parsons. September 1998. ftp://ftp.isi.edu/in-notes/rfc2423.txt
- RFC 2422, Toll Quality Voice - 32 kbit/s ADPCM MIME Sub-type Registration. G. Vaudreuil, G. Parsons. September 1998. ftp://ftp.isi.edu/in-notes/rfc2422.txt
- RFC 2421, Voice Profile for Internet Mail - version 2. G. Vaudreuil, G. Parsons. September 1998. ftp://ftp.isi.edu/in-notes/rfc2421.txt
- RFC 2871, A Framework for Telephony Routing over IP. J. Rosenberg, H. Schulzrinne. June 2000. ftp://ftp.isi.edu/in-notes/rfc2871.txt
- RFC 2848, The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services. S. Petrack, L. Conroy. June 2000. ftp://ftp.isi.edu/in-notes/rfc2848.txt
- RFC 2833, RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals. H. Schulzrinne, S. Petrack. May 2000. ftp://ftp.isi.edu/in-notes/rfc2833.txt
- RFC 2806, URLs for Telephone Calls. A. Vaha-Sipila. April 2000. ftp://ftp.isi.edu/in-notes/rfc2806.txt
- RFC 2719, Framework Architecture for Signaling Transport. L. Ong, I. Rytina, M. Garcia, H. Schwarzbauer, L. Coene, H. Lin, I. Juhasz, M. Holdrege, C. Sharp. October 1999. ftp://ftp.isi.edu/in-notes/rfc2719.txt
- RFC 2705, Media Gateway Control Protocol (MGCP) Version 1.0. M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. October 1999. ftp://ftp.isi.edu/in-notes/rfc2705.txt
- RFC 2543, SIP: Session Initiation Protocol. M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg. March 1999. ftp://ftp.isi.edu/in-notes/rfc2543.txt
- RFC 2542, Terminology and Goals for Internet Fax. L. Masinter. March 1999. ftp://ftp.isi.edu/in-notes/rfc2542.txt
- RFC 2532, Extended Facsimile Using Internet Mail. L. Masinter, D. Wing. March 1999. ftp://ftp.isi.edu/in-notes/rfc2532.txt
- RFC 2531, Content Feature Schema for Internet Fax. G. Klyne, L. McIntyre. March 1999. ftp://ftp.isi.edu/in-notes/rfc2531.txt
- RFC 2458, Toward the PSTN/Internet Inter-Networking--Pre-PINT Implementations. H. Lu, M. Krishnaswamy, L. Conroy, S. Bellovin, F. Burg, A.,

DeSimone, K. Tewani, P. Davidson, H. Schulzrinne, K. Vishwanathan. November 1998. ftp://ftp.isi.edu/in-notes/rfc2458.txt

- RFC 2423, VPIM Voice Message MIME Sub-type Registration. G. Vaudreuil, G. Parsons. September 1998. ftp://ftp.isi.edu/in-notes/rfc2423.txt
- RFC 2422, Toll Quality Voice - 32 kbit/s ADPCM MIME Sub-type Registration. G. Vaudreuil, G. Parsons. September 1998. ftp://ftp.isi.edu/in-notes/rfc2422.txt
- RFC 2421, Voice Profile for Internet Mail - version 2. G. Vaudreuil, G. Parsons. September 1998. ftp://ftp.isi.edu/in-notes/rfc2421.txt
- RFC 2423, VPIM Voice Message MIME Sub-type Registration. G. Vaudreuil, G. Parsons. September 1998. ftp://ftp.isi.edu/in-notes/rfc2423.txt
- RFC 2422, Toll Quality Voice - 32 kbit/s ADPCM MIME Sub-type Registration. G. Vaudreuil, G. Parsons. September 1998. ftp://ftp.isi.edu/in-notes/rfc2422.txt
- RFC 2421, Voice Profile for Internet Mail - version 2. G. Vaudreuil, G. Parsons. September 1998. ftp://ftp.isi.edu/in-notes/rfc2421.txt
- RFC 2327, SDP: Session Description Protocol. M. Handley, V. Jacobson. April 1998. ftp://ftp.isi.edu/in-notes/rfc2327.txt
- RFC 2306, Tag Image File Format (TIFF) - F Profile for Facsimile. G. Parsons, J. Rafferty. March 1998. ftp://ftp.isi.edu/in-notes/rfc2306.txt
- RFC 2305, A Simple Mode of Facsimile Using Internet Mail. K. Toyoda, H. Ohno, J. Murai, D. Wing. March 1998. ftp://ftp.isi.edu/in-notes/rfc2305.txt
- RFC 2304, Minimal FAX address format in Internet Mail. C. Allocchio. March 1998. ftp://ftp.isi.edu/in-notes/rfc2304.txt
- RFC 2303, Minimal PSTN address format in Internet Mail. C. Allocchio. March 1998. ftp://ftp.isi.edu/in-notes/rfc2303.txt
- RFC 2302, Tag Image File Format (TIFF) - image/tiff MIME Sub-type Registration. G. Parsons, J. Rafferty, S. Zilles. March 1998. ftp://ftp.isi.edu/in-notes/rfc2302.txt
- RFC 2301, File Format for Internet Fax. L. McIntyre, S. Zilles, R. Buckley, D. Venable, G. Parsons, J. Rafferty. March 1998. ftp://ftp.isi.edu/in-notes/rfc2301.txt
- RFC 2159, A MIME Body Part for FAX, ftp://ftp.isi.edu/in-notes/rfc2159.txt
- RFC 1911, Voice Profile for Internet Mail, ftp://ftp.isi.edu/in-notes/rfc1911.txt
- RFC 1789, INETPhone: Telephone Services and Servers on Internet, ftp://ftp.isi.edu/in-notes/rfc1789.txt
- RFC 0978, Voice File Interchange Protocol (VFIP), ftp://ftp.isi.edu/in-notes/rfc0978.txt
- RFC 0741, Specifications for the Network Voice Protocol (NVP), ftp://ftp.isi.edu/in-notes/rfc0741.txt
- RFC 0511, Enterprise phone service to NIC from ARPANET sites, ftp://ftp.isi.edu/in-notes/rfc0511.txt

24. BOOKS:

- Computer Networks-Andrew.S.Tanenbaum
- Data & Computer Communications-William Stallings
- TCP/IP Illustrated Volume 1-W.R.Stevens
- TCP/IP Illustrated Volume 2-W.R.Stevens
- TCP/IP Illustrated Volume 3-W.R.Stevens
- Unix Network Programming-W.R.Stevens

25. HANDS ON EXPERIENCE:

- CISCO Interactive Mentor: Voice Networking-Basic Voice over IP