TRAFFIC BEHAVIOR AND PERFORMANCE IN VOIP

A Thesis
Submitted to the Department of Computer Science and Engineering
of
BRAC University
By

Ashis Kumar Saha
Student ID: 03101030

Under Supervision of

Sadia Hamid Kazi
Senior Lecturer
Department Of Computer Science & Engineering
BRAC University

In Partial Fulfillment of the
Requirements for the Degree
of
Bachelor of Science in Computer Science & Engineering
April 2008

DECLARATION

I hereby declare that this thesis is based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

---	---
Signature of Supervisor	Signature of Author

ACKNOWLEDGEMENT

I wish to express our heartiest gratitude ,sincere appreciation and indebtedness to my honorable Thesis Supervisor Sadia Hamid Kazi for her guidance,encouragement,support and the constructive suggestions throughout the completion of this report.

I would like to thank Niti Jabin,Hasan-Al-Banna & MD.Ruhanuzzaman for their valuable time during my pre-thesis semester. And I would like to thank all peoples who helped us to prepare this final report.

Finally I pay my deepest respect to His Almighty the beneficial and the merciful and thank him for the successful completion of the report.

ABSTRACT

Voice over Internet protocol - VoIP, or IP telephony is a technology by which the routing of voice communications are done through Internet or any other Internet Protocol (IP) based networks. Here the voice data is transmitted over a general purpose packet-switched network instead of dedicated traditional circuit-switched voice transmission lines. Voice Over Internet Protocol (VoIP) is a telephony technology used to transmit ordinary telephone calls over the Internet. VoIP takes analogue audio signals and turns them into digital signals (packets) that are transmitted using Internet Protocol (IP) networks. VoIP's advantages include low cost, flexibility, and mobility. Conversely, VoIP's disadvantages include sound quality such as latency (delay), jitter, and packet loss. To be more precise the main goal of my thesis was to study the traffic behavior of VoIP. To find out the factors that effect on the performance of VoIP and to mitigate that problems in the network was the main goal here.

TABLE OF CONTENTS

4.0 What is VoIP:

4.1 Overview:

Voice over Internet protocol – VoIP[1], or IP telephony is a technology by which the routing of voice communications are done through Internet or any other Internet Protocol (IP) based networks. Here the voice data is transmitted over a general purpose packet-switched network instead of dedicated traditional circuit-switched voice transmission lines.

VoIP is a part of the group of technologies called voice over packet networks. Other network protocols like asynchronous transfer mode (ATM)[2] can perform similar functions. Though the concept of VoIP is simple, the implementation and applications of it is a bit complicated. In order to send voice, the information has to be separated into packets just like data. Packets are chunks of information broken up into the most efficient size for routing. From there, the packets need to be sent and put back together in an efficient manner. For more efficient use, the voice data can be compressed so that it require less space and will certainly record only a limited frequency range. There are many ways to compress audio, the algorithm for which is referred to as a compressor/de-compressor (CODEC)[3]. Many a number of CODECs exist depending on the application (e.g., conversations, music, movies and sound recordings). The CODECs are optimized for compressing voice, which significantly reduce the bandwidth used compared to an uncompressed audio stream. Speech CODECs are optimized to improve spoken words at the expense of sounds outside the frequency range of human speech. Recorded music and other sounds do not generally sound very good when passed through a speech CODEC.

4.2 Technology Overview:

VoIP is a new form of communication that takes analogue audio signals and turns them into digital signals, or packets. This is an innovative alternate to the traditional circuit-switched[4] method of telecommunication, where a dedicated circuit between two parties is maintained. In order to set up a traditional phone call between two telephones, the switched and the intervening network establish a dedicated route from one end of the call to the other. Conversely, VoIP uses a packet-switched[4] method where audio signals are converted into digital data at the originating end, which is then transmitted over the Internet and converted back to analog signal at the receiving end. In other words, VoIP digitizes voice, inserts the digitized data into discrete packets, and sends them over the IP network. The packets have a destination address, but no fixed path through the network. The packets arrive at the address, where they are put back together and

converted back to analog audio signals. VoIP integrates voice and data communications and turns any Internet connection into a phone call. VoIP is a revolutionary technology that has the potential to drastically change the way people communicate and talk on the phone around the world.

5. How does VoIP works:

When you speak at the handset or a mike or a microphone, your voice generates electrical signals inside the gadget. These are analog signals i.e. the voltage level can take up any value within a range.

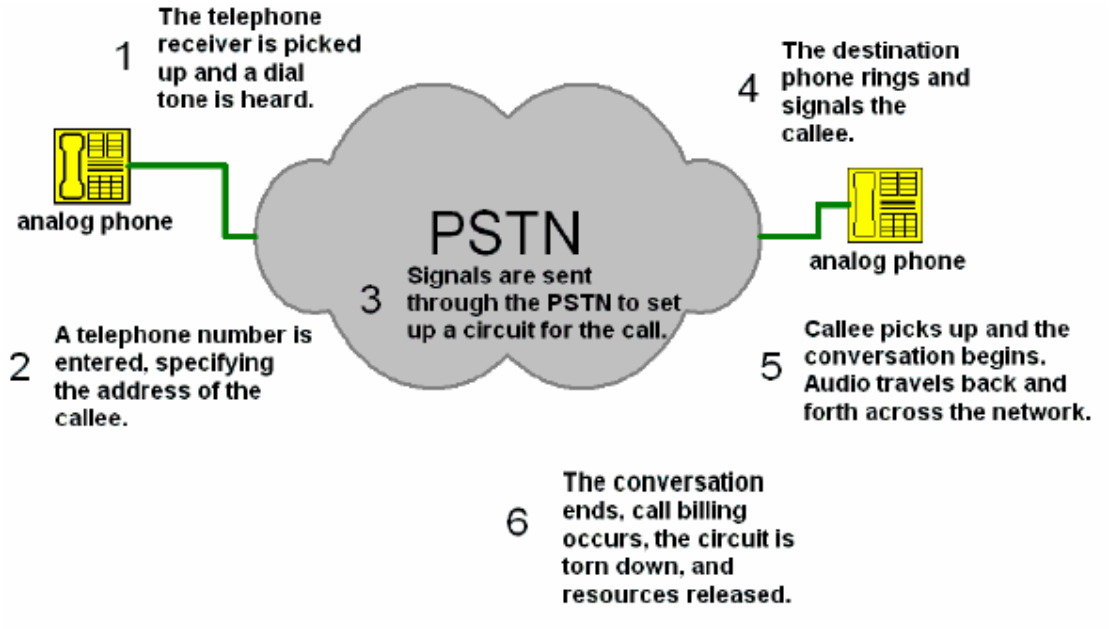


Fig: Typical PSTN form.

The analog signal is converted to a digital signal using an algorithm implemented by the device you are using. It can be a stand-alone VoIP phone or a softphone running on your PC. If you are using an analog phone, you will need a Telephony Adapter (TA)[5] for this purpose. The digitized voice is arranged in packets (i.e. collection of bytes) and sent over the IP network.

The data is channeled through gateways and servers to the destination. If the called number is on the PSTN[6], the server opens a connection to the PSTN and routes your call there.

While going to the PSTN or at the end device of a VoIP connection, the voice is gain brought back to its analog form so that it is perceptible to a human ear. During the entire process a protocol like SIP[7] or H.323[8] is used to control the call (e.g. setting up connection, dialing, disconnecting etc.)

and RTP[10] is used for reliable transmission of data packets and maintain Quality of Service.

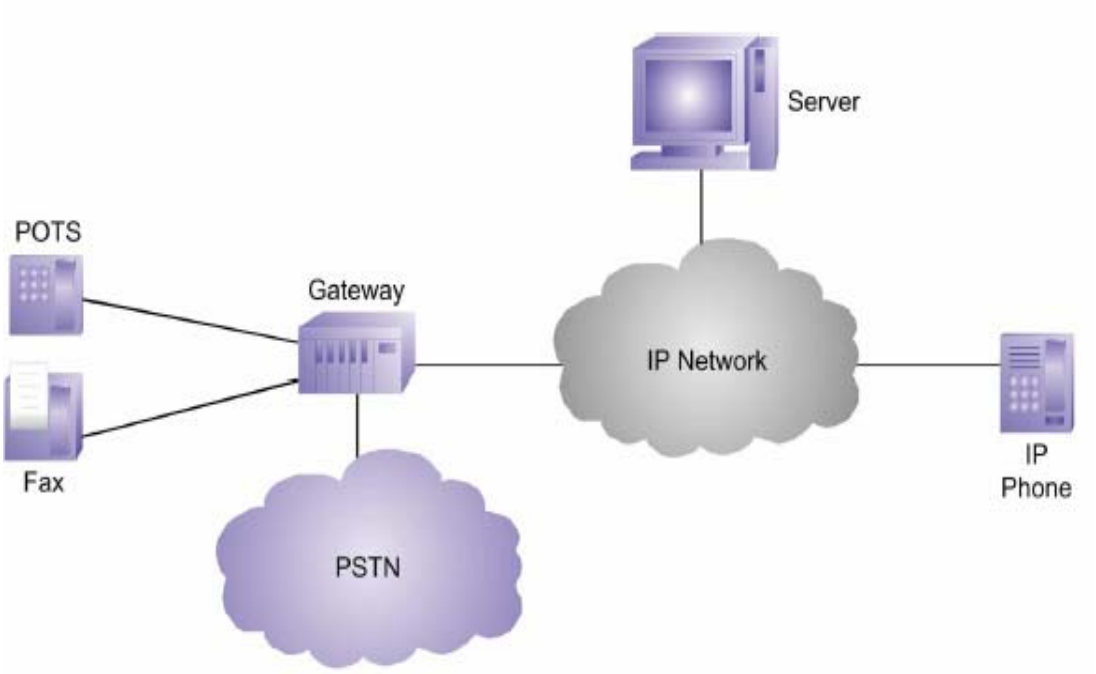


Figure 1. VoIP Components

5.1 The digitization of analog voice signals:

The digitization of analog voice signals[10] is a must to transmit voice over the digital IP network. It can be done in several ways:

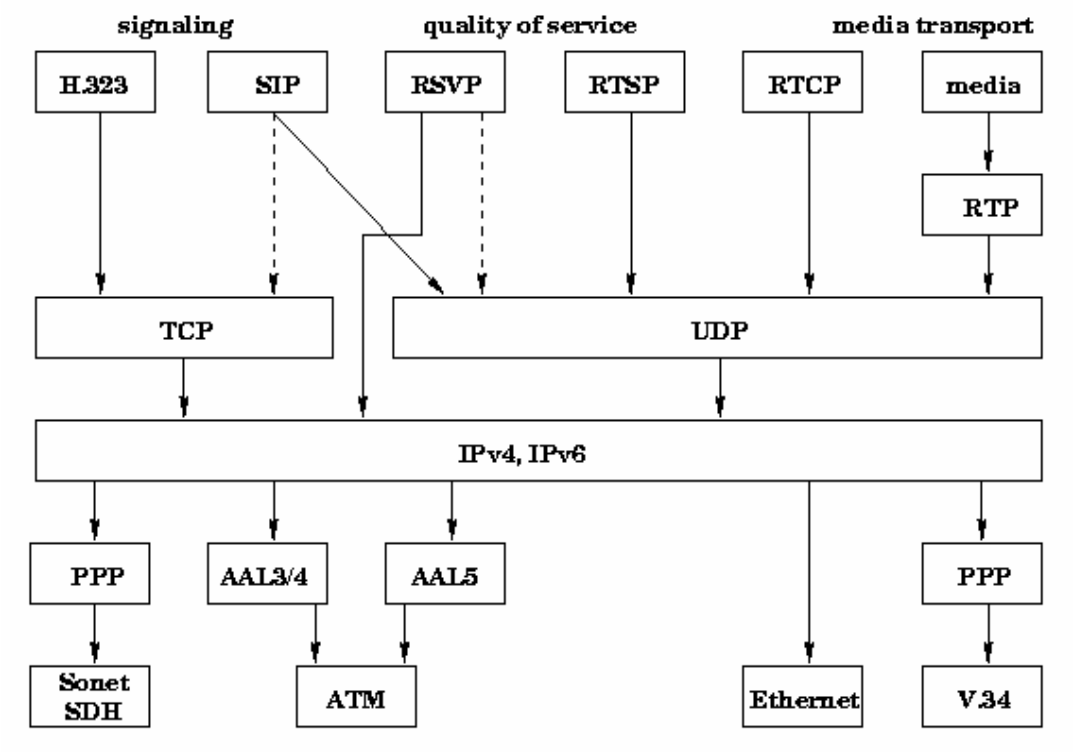
PCM (Pulse Code Modulation)[11] is a simple technique of sampling the sound signal at a fixed rate (8000 Times/second) and generate a number corresponding to each sample. It assumes no specific property of the signal. So it works reasonably well with all types of sounds.

LPC (Liner Predictive Coding)[10] assumes specific properties of human voice and uses a more complex algorithm to digitize and compress voice data. It works well for sending human utterances offering a low data rate but is not suitable for transmitting music or fax.

SBC (Sub Band Coder)[10] uses a different approach of representing sounds in terms of frequencies rather than sampling at regular intervals.

Hybrid coders like the CELP (Code Excited Linear Prediction)[10] use a mixture of the techniques to transmit sound of adequate quality.

5.2 Protocol Architechture:



5.3 RTP:

Real-time Transfer Protocol (RTP) is the Internet-standard protocol for the transport of real-time data, including audio and video. It can be used for media-on-demand as well as interactive services such as Internet telephony. RTP consists of a data and a control part. The latter is called RTCP.

The data part of RTP is a thin protocol providing support for applications with real-time properties such as continuous media (e.g., audio and video), including timing reconstruction, loss detection, security and content identification.

RTCP provides support for real-time conferencing of groups of any size within an internet. This support includes source identification and support for gateways like audio and video bridges as well as multicast-to-unicast translators. It offers quality-of-service feedback from receivers to the multicast group as well as support for the synchronization of different media streams.

While UDP/IP is its initial target networking environment, efforts have been made to make RTP transport-independent so that it could be used, say, over CLNP, IPX or other protocols. RTP is currently also in experimental use directly over AAL5/ATM. RTP does not address the issue of resource reservation or quality of service control; instead, it relies on resource reservation protocols such as RSVP .

Other applications, such as real-time control and distributed simulation, are also targets.

RTP was developed by the Audio-Video Transport Working Group of the IETF and first published in 1996 as RFC 1889 . It was originally designed as a multicast protocol, but has since been applied in many unicast applications. It is frequently used in streaming media systems (in conjunction with RTSP) as well as videoconferencing and push to talk systems (in conjunction with H.323 or SIP), making it the technical foundation of the Voice over IP industry. It goes along with the RTP Control Protocol (RTCP) and it's built on top of the User Datagram Protocol (UDP) (in OSI model).

According to RFC 1889 , the services provided by RTP include:

- Payload-type identification
- Sequence numbering
- Time stamping
- Delivery monitoring

5.4 SIP:

Session Initiation Protocol (SIP) is a protocol developed by IETF to assist in providing advanced telephony services across the Internet. Basically is a signaling protocol used for establishing sessions in an IP network. A session could be a simple two-way telephone call or it could be a collaborative multi-media conference session. The ability to establish these sessions means that a host of innovative services become possible, such as voice-enriched e-commerce, web page click-to-dial, Instant Messaging with buddy lists, and IP Centrex services.

SIP is modeled upon other Internet protocols such as SMTP[17] (Simple Mail Transfer Protocol) and HTTP[17] (Hypertext Transfer Protocol.) It is used to establish, change and tear down (end) calls between one or more users in an IP-based network. In order to provide telephony services there is a need for a number of different standards and protocols to come together - specifically to ensure transport (RTP), signaling inter-working with today’s telephony network, to be able to guarantee voice quality (RSVP, YESSIR), to be able to provide directories (LDAP), to authenticate users (RADIUS, DIAMETER), and to scale to meet the anticipated growth curves.

SIP can be regarded as the enabler protocol for telephony and voice over IP (VoIP) services. The following features of SIP play a major role in the ennoblement of IP telephony and VoIP:

* Name Translation and User Location - Ensuring that the call reaches the called party wherever they are located. Carrying out any mapping of descriptive information to location information. Ensuring that details of the nature of the call (Session) are supported.

* Feature Negotiation - This allows the group involved in a call (this may be a multi-party call) to agree on the features supported – recognizing that not all the parties can support the same level of features. For example video may or may not be supported; as any form of MIME type is supported by SIP, there is plenty of scope for negotiation. Call Participant Management - During a call a participant can bring other users onto the call or cancel connections to other users. In addition, users could be transferred or placed on hold.

* Call feature changes - A user should be able to change the call characteristics during the course of the call. For example, a call may have been set up as 'voice-only', but in the course of the call, the users may need to enable a video function. A third party joining a call may require different features to be enabled in order to participate in the call

* Media negotiation – The inherent SIP mechanisms that enable negotiation of the media used in a call, enable selection of the appropriate codex for establishing a call between the various devices. This way, less advanced devices can participate in the call, provided the appropriate codex is selected.

Below is are some of other SIP features that distinguish it among new signaling protocols

* SIP messages are text based and hence are easy to read and debug. Programming new services is easier and more intuitive for designers.

* SIP re-uses MIME type description in the same way that email clients do, so applications associated with sessions can be launched automatically.

* SIP re-uses several existing and mature internet services and protocols such as DNS, RTP, RSVP etc. No new services have to be introduced to support the SIP infrastructure, as much of it is already in place or available off the shelf.

* SIP extensions are easily defined, enabling service providers to add them for new applications without damaging their networks. Older SIP-based equipment in the network will not impede newer SIP-based services. For example, an older SIP implementation that does not support method/ header utilized by a newer SIP application would simply ignore it. SIP is transport layer independent. Therefore, the underlying transport could be IP over ATM. SIP uses the User Datagram Protocol, (UDP)[12] as well as the Transmission Control Protocol (TCP) protocol[13], flexibly connecting users independent of the underlying infrastructure.

* SIP supports multi-device feature leveling and negotiation. If a service or session initiates video and voice, voice can still be transmitted to non-video enabled devices, or other device features can be used such as one way video streaming.

SIP sessions utilize up to four major components: SIP User Agents, SIP Registrar Servers, SIP Proxy Servers and SIP Redirect Servers. Together,

these systems deliver messages embedded with the SDP protocol defining their content and characteristics to complete a SIP session.

5.5 H.323:

H.323 is a standard that specifies the components, protocols and procedures that provide multimedia communication services : real-time audio, video, and data communications over packet networks, including Internet protocol (IP) based networks. H.323 is part of a family of ITU-T recommendations called H.32x that provides multimedia communication services over a variety of networks.

H.323 was originally created to provide a mechanism for transporting multimedia applications over LANs but it has rapidly evolved to address the growing needs of VoIP networks. One strength of H.323 was the relatively early availability of a set of standards, not only defining the basic call model, but in addition the supplementary services, needed to address business communication expectations. H.323 was the first VoIP standard to adopt the IETF standard RTP to transport audio and video over IP networks.

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:

- **Terminals:** Used for real-time bi-directional multimedia communications, an H.323 terminal[8] 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. An H.323 terminal can either be a PC or a stand-alone device, running an H.323 stack and multimedia applications. The primary goal of H.323 is to inter-work 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.

- **Gateways:** A gateway[8] connects two dissimilar networks. An H.323 gateway provides connectivity between an H.323 network and a non–H.323 network. For example, a gateway can connect and provide communication between an H.323 terminal and SCN networks (SCN networks include all switched telephony networks, e.g., public switched telephone network [PSTN]). This connectivity of dissimilar networks is achieved by translating protocols for call setup and release, converting media formats between different networks, and transferring information between the networks connected by the gateway. A gateway is not

required, however, for communication between two terminals on an H.323 network.

- **Gatekeepers:** A gatekeeper[8] can be considered the brain of the H.323 network. It is the focal point for all calls within the H.323 network. Although they are not required, gatekeepers provide important services such as addressing, authorization and authentication of terminals and gateways; bandwidth management; accounting; billing; and charging. Gatekeepers may also provide call-routing services.
- **Multipoint Control Units:** MCUs[8] provide support for conferences of three or more H.323 terminals. All terminals participating in the conference establish a connection with the MCU. The MCU manages conference resources, negotiates between terminals for the purpose of determining the audio or video coder/decoder (CODEC) to use, and may handle the media stream. The gatekeepers, gateways, and MCUs are logically separate components of the H.323 standard but can be implemented as a single physical device.

Key Benefits of H.323 :

Codec Standards: H.323 establishes standards for compression and decompression of audio and video data streams, ensuring that equipment from different vendors will have some area of common support.

Interoperability: Users want to conference without worrying about compatibility at the receiving point. Besides ensuring that the receiver can decompress the information, H.323 establishes methods for receiving clients to communicate capabilities to the sender. The standard also establishes common call setup and control protocols.

Network Independence: H.323 is designed to run on top of common network architectures. As network technology evolves, and as bandwidth-management techniques improve, H.323-based solutions will be able to take advantage of those enhanced capabilities.

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.

Bandwidth Management:

Video and audio traffic is bandwidth-intensive and could clog the corporate network. H.323 addresses this issue by providing bandwidth management. Network managers can limit the number of simultaneous H.323 connections within their network or the amount of bandwidth available to H.323 applications. These limits ensure that critical traffic will not be disrupted.

Flexibility: An H.323 conference can include endpoints with different capabilities. For example, a terminal with audio-only capabilities can participate in a conference with terminals that have video and/or data capabilities. Furthermore, an H.323 multimedia terminal can share the data portion of a video conference with a T.120 data-only terminal, while sharing voice, video, and data with other H.323 terminals.

Inter-Network Conferencing: Many users want to conference from a LAN to a remote site. For example, H.323 establishes a means of linking LAN-based desktop systems with ISDN-based group systems. H.323 uses common codec technology from different videoconferencing standards to minimize transcoding delays and to provide optimum performance.

6.0 PSTN vs VoIP:

Many factors in the past have slowed the anticipated growth of Voice over IP (VoIP). Now, VoIP solutions that achieve quality and reliability, close to what we are used to from the Public Switched Telephony Network (PSTN), are emerging as the market is quickly growing. However, as will be shown in this article, there is no reason to limit the expectations to achieve only the same level of quality as in PSTN. It is quite well known that, by deploying wideband voice codecs, much better quality can be achieved. However, a little known fact is that there are ways to achieve better quality than a standard PSTN solution, even when using narrowband codecs. For example, the full available spectral bandwidth is not typically utilized in traditional PSTN solutions, something that can easily be done in a VoIP system. Implementing a wideband codec or expanding the bandwidth of narrowband codecs does not automatically guarantee great quality. There are many potential pitfalls when deploying VoIP. In this article we will also discuss implementation issues related to VoIP that will impact the final voice quality. We will discuss what level of quality can be achieved and describe how this can be implemented.

Speech Signals and Speech Coding Sampled digital signals can contain frequency content up to half the sampling frequency. Typically, a

young adult has a hearing span from about 20 to 20,000 Hz. Consequently, the sampling frequency of CD audio is chosen to be 44.1 kHz, which is more than double that of the highest frequency perceivable by most humans. Legacy telephony solutions are narrowband, which seriously limits the achievable quality. Wideband codecs could potentially be used in digital telephone systems, but this has never been practical enough to gain any real interest

In fact, in traditional telephony applications, the speech bandwidth is restricted much more than the inherent limitations of narrowband coding. Typical telephony speech is bandlimited to 300 – 3400 Hz (listen to Sound Pure digital connections are typically only found in enterprise environments. Due to poor connections or old wires, significant distortion is often generated in the analog part of the phone connection, a type of distortion that is entirely absent from VoIP implementations. The cordless phones so popular today also generate significant amounts of analog distortion due to radio interference and other implementation issues.

On the Public Switched Telephone Network (PSTN), calls between two parties are set up by a series of private and public switches. The resulting fixed communications link is dedicated for the duration of the call. When an individual makes a phone call over a circuit-switched network, a connection is made between a company's PBX and the local telephone company, also known as the PSTN. Depending on the destination, this connection might extend to the national or international exchange before reaching another local exchange, where it will be passed on to the PBX and the person who receives the call. This end-to-end link, established by a series of public and private switches, is 100% dedicated on a single, per-call basis and cannot be shared or used for another function as long as the call is in progress. For this reason, these dedicated circuits cannot be shared and the carrier bills the call on a time and distance rate. The Internet does not use switches to link calling parties. Instead, the analog voice signal is digitized by an Internet Protocol (IP) and broken up into thousands of small data packets by a router - the VoIP equivalent to a switch. These data packets are sent, or routed, over the public Internet to their destination, enabling calls to bypass the PSTN entirely.

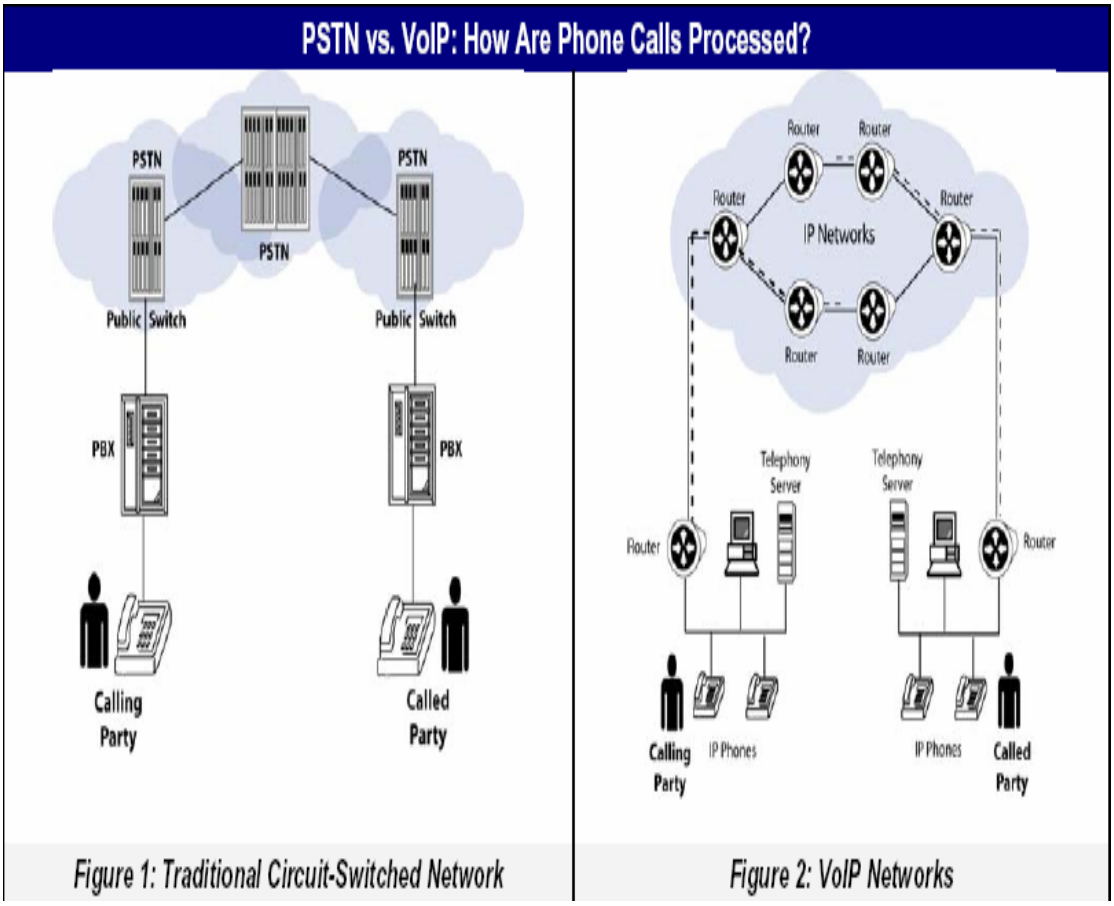


Figure 1: Traditional Circuit-Switched Network

Figure 2: VoIP Networks

7.0 Network Simulator: NS 2

A network simulator[14] is a piece of software or hardware that predicts the behavior of a network, without an actual network being present.

7.1 Uses of network simulators:

Network simulators serve a variety of needs. Compared to the cost and time involved in setting up an entire test bed containing multiple networked computers, routers and data links, network simulators are relatively fast and inexpensive. They allow engineers to test scenarios that might be particularly difficult or expensive to emulate using real hardware- for instance, simulating the effects of a sudden burst in traffic or a DoS attack on a network service. Networking simulators are particularly useful in allowing designers to test new networking protocols or changes to existing protocols in a controlled and reproducible environment.

Network simulators, as the name suggests are used by researchers, developers and QA to design various kinds of networks, simulate and then analyze the effect of various parameters on the network performance. A typical network simulator like NetSim encompasses a wide range of

networking technologies and help the users to build complex networks from basic building blocks like variety of nodes and links. With the help of simulators one can design hierarchical networks using various types of nodes like computers, hubs, bridges, routers, optical cross-connects, multicast routers, mobile units, MSAUs etc.

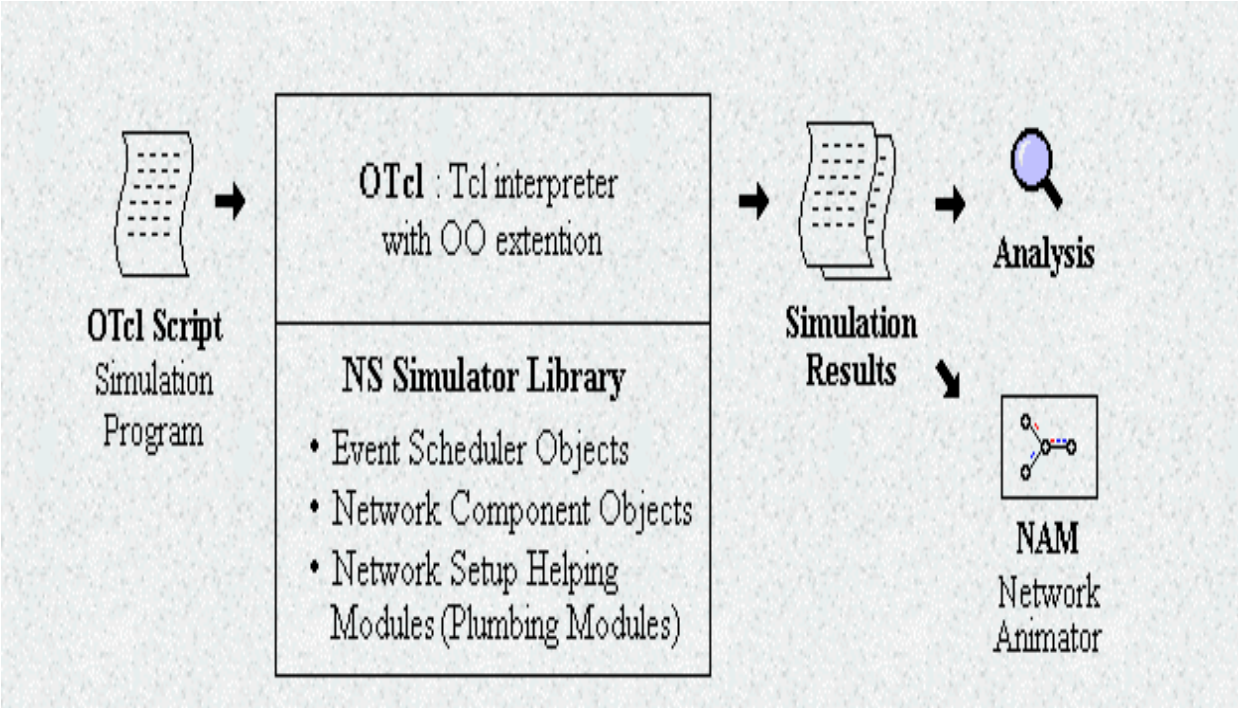
7.2 Types of network simulators:

Various types of Wide Area Network (WAN) technologies like TCP, ATM, IP etc and Local Area Network (LAN) technologies like Ethernet, token rings etc. can all be simulated with a typical simulator and the user can test, analyze various standard results apart from devising some novel protocol or strategy for routing etc.

There are a wide variety of network simulators, ranging from the very simple to the very complex. Minimally, a network simulator must enable a user to represent a network topology, specifying the nodes on the network, the links between those nodes and the traffic between the nodes. More complicated systems may allow the user to specify everything about the protocols used to handle network traffic. Graphical applications allow users to easily visualize the workings of their simulated environment. Text-based applications may provide a less intuitive interface, but may permit more advanced forms of customization. Others, such as GTNets, are programming-oriented, providing a programming framework that the user then customizes to create an application that simulates the networking environment to be tested.
Ns is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.

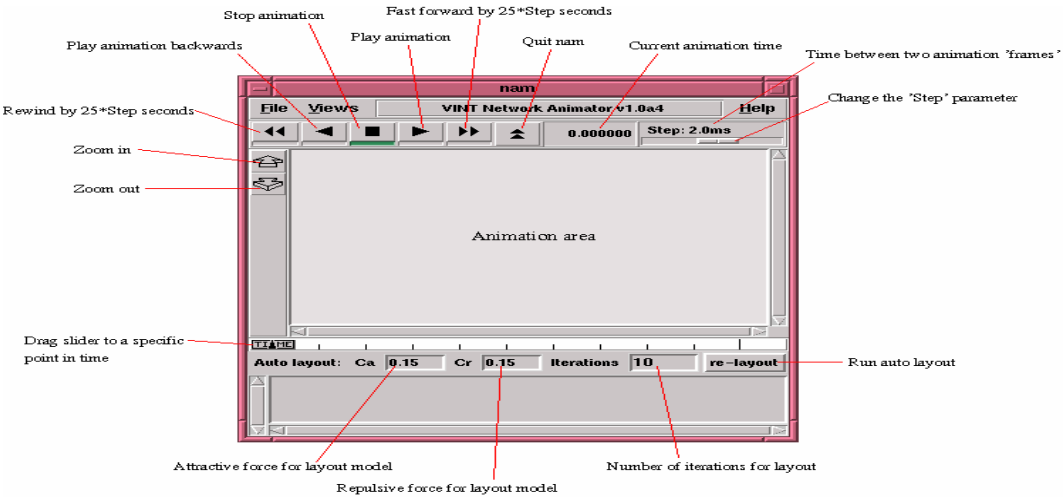
Ns began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. In 1995 ns development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is support through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI. Ns has always included substantial contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems.

7.3 How Does NS2 Works:



I have used NS2 for my thesis as a tool to simulated the scenarios. I have used network simulator version 2.Ns is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networksWe write tcl scripts and we run that script in NS2 and it uses necessary library to execute that code and then we get a network animator file we call it nam file and a trace file for doing the precise analysis of that scenario.

7.4 NS2 Screenshots:



8.0 VoIP uses UDP rather Than TCP:

8.1 Comparative analysis - TCP – UDP:

8.2 TCP:

Abbreviation of Transmission Control Protocol, and pronounced as separate letters. TCP is one of the main protocols in TCP/IP networks. Whereas the IP protocol deals only with packets, TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent.

TCP stands for Transmission Control Protocol. It is described in STD-7/RFC-793. TCP is a connection-oriented protocol that is responsible for reliable communication between two end processes. The unit of data transferred is called a stream, which is simply a sequence of bytes.

Being connection-oriented means that before actually transmitting data, you must open the connection between the two end points. The data can be transferred in full duplex (send and receive on a single connection). When the transfer is done, you have to close the connection to free system resources. Both ends know when the session is opened (begin) and is closed (end). The data transfer cannot take place before both ends have agreed upon the connection. The connection can be closed by either side; the other is notified. Provision is made to close gracefully or just abort the connection.

Being stream oriented means that the data is an anonymous sequence of bytes. There is nothing to make data boundaries apparent. The receiver has no means of knowing how the data was actually transmitted. The sender can send many small data chunks and the receiver receive only one big chunk, or the sender can send a big chunk, the receiver receiving it in a number of smaller chunks. The only thing that is guaranteed is that all data sent will be received without any error and in the correct order. Should any error occur, it will automatically be corrected (retransmitted as needed) or the error will be notified if it can't be corrected.

At the program level, the TCP stream look like a flat file. When you write data to a flat file, and read it back later, you are absolutely unable to know if the data has been written in only one chunk or in several chunks. Unless you write something special to identify record boundaries, there is nothing you can do to learn it afterward. You can, for example, use CR or CR LF to delimit your records just like a flat text file.

At the programming level, TWSocket is fairly simple to use. To send data, you just need to call the Send method (or any variation such as SendStr) to give

the data to be transmitted. TWSocket will put it in a buffer until it can be actually transmitted. Eventually the data will be sent in the background (the Send method returns immediately without waiting for the data to be transmitted) and the OnDataSent event will be generated once the buffer is emptied.

To receive data, a program must wait until it receives the OnDataAvailable event. This event is triggered each time a data packet comes from the lower level. The application must call the Receive method to actually get the data from the low-level buffers. You have to Receive all the data available or your program will go in an endless loop because TWSocket will trigger the OnDataAvailable again if you didn't Receive all the data.

As the data is a stream of bytes, your application must be prepared to receive data as sent from the sender, fragmented in several chunks or merged in bigger chunks. For example, if the sender sent "Hello " and then "World!", it is possible to get only one OnDataAvailable event and receive "Hello World!" in one chunk, or to get two events, one for "Hello " and the other for "World!". You can even receive more smaller chunks like "Hel", "lo wo" and "rld!". What happens depends on traffic load, router algorithms, random errors and many other parameters you can't control.

On the subject of client/server applications, most applications need to know command boundaries before being able to process data. As data boundaries are not always preserved, you cannot suppose your server will receive a single complete command in one OnDataAvailable event. You can receive only part of a request or maybe two or more request merged in one chunk. To overcome this difficulty, you must use delimiters.

Most TCP/IP protocols, like SMTP, POP3, FTP and others, use CR/LF pair as command delimiter. Each client request is sent as is with a CR/LF pair appended. The server receives the data as it arrives, assembles it in a receive buffer, scans for CR/LF pairs to extract commands from the received stream, and removes them from the receive buffer.

8.3 UDP:

Short for User Datagram Protocol, a connectionless protocol that, like TCP, runs on top of IP networks. Unlike TCP/IP, UDP/IP provides very few error recovery services, offering instead a direct way to send and receive datagrams over an IP network. It's used primarily for broadcasting messages over a network.

UDP stands for User Datagram Protocol. It is described in STD-6/RFC-768 and provides a connectionless host-to-host communication path. UDP has minimal overhead:; each packet on the network is composed of a small header and user data. It is called a UDP datagram.

UDP preserves datagram boundaries between the sender and the receiver. It means that the receiver socket will receive an OnDataAvailable event for each datagram sent and the Receive method will return a complete datagram for each call. If the buffer is too small, the datagram will be truncated. If the buffer is too large, only one datagram is returned, the remaining buffer space is not touched.

UDP is connectionless. It means that a datagram can be sent at any moment without prior advertising, negotiation or preparation. Just send the datagram and hope the receiver is able to handle it.

UDP is an unreliable protocol. There is absolutely no guarantee that the datagram will be delivered to the destination host. But to be honest, the failure rate is very low on the Internet and nearly null on a LAN unless the bandwidth is full.

Not only the datagram can be undelivered, but it can be delivered in an incorrect order. It means you can receive a packet before another one, even if the second has been sent before the first you just received. You can also receive the same packet twice.

Your application must be prepared to handle all those situations: missing datagram, duplicate datagram or datagram in the incorrect order. You must program error detection and correction. For example, if you need to transfer some file, you'd better set up a kind of zmodem protocol.

The main advantages for UDP are that datagram boundaries are respected, you can broadcast, and it is fast.

The main disadvantage is unreliability and therefore complicated to program at the application level.

8.4 Addressing:

TCP and UDP use the same addressing scheme. An IP address (32 bits number, always written as four 8-bit number expressed as unsigned 3-digit decimal numbers separated by dots such as 193.174.25.26) and a port number (a 16-bit number expressed as a unsigned decimal number).

The IP address is used by the low-level protocol (IP) to route the datagram to the correct host on the specified network. Then the port number is used to route the datagram to the correct host process (a program on the host).

For a given protocol (TCP or UDP), a single host process can exist at a time to receive data sent to the given port. Usually one port is dedicated to one process.

8.5 Advantages of tcp:

the operating system does all the work. you just sit back and watch the show. no need to have the same bugs in your code that everyone else did on their first try; it's all been figured out for you.

since it's in the os, handling incoming packets has fewer context switches from kernel to user space and back; all the reassembly, acking, flow control, etc is done by the kernel.

tcp guarantees three things: that your data gets there, that it gets there in order, and that it gets there without duplication. (the truth, the whole truth, and nothing but the truth...)

routers may notice tcp packets and treat them specially. they can buffer and retransmit them, and in limited cases preack them.

tcp has good relative throughput on a modem or a lan.

8.6 Disadvantages of tcp:

The operating system may be buggy, and you can't escape it. it may be inefficient, and you have to put up with it. it may be optimized for conditions other than the ones you are facing, and you may not be able to retune it.

Tcp makes it very difficult to try harder; you can set a few socket options, but beyond that you have to tolerate the built in flow control.

Tcp may have lots of features you don't need. it may waste bandwidth, time, or effort on ensuring things that are irrelevant to the task at hand.

Tcp has no block boundaries; you must create your own.

Routers on the internet today are out of memory. they can't pay much attention to tcp flying by, and try to help it. design assumptions of tcp break down in this environment.

Tcp has relatively poor throughput on a lossy, high bandwidth, high latency link, such as a satellite connection or an overfull t1.

Tcp cannot be used for broadcast or multicast transmission.

Tcp cannot conclude a transmission without all data in motion being explicitly acked.

8.7 Disadvantages of udp:

There are no guarantees with udp. a packet may not be delivered, or delivered twice, or delivered out of order; you get no indication of this unless the listening program at the other end decides to say something. tcp is really working in the same environment; you get roughly the same services from ip and udp. however, tcp makes up for it fairly well, and in a standardized manner.

Udp has no flow control. implementation is the duty of user programs.

Routers are quite careless with udp. they never retransmit it if it collides, and it seems to be the first thing dropped when a router is short on memory. udp suffers from worse packet loss than tcp.

8.8 Advantages of udp :

It doesn't restrict connection based communication model, so startup latency in distributed applications is much lower, as is operating system overhead.

All flow control, acking, transaction logging, etc is up to user programs; a broken os implementation is not going to get in your way. additionally, you only need to implement and use the features you need.

The recipient of udp packets gets them unmangled, including block boundaries.

Broadcast and multicast transmission are available with udp.

8.9 Disadvantages of tcp for file transfer :

Startup latency is significant. it takes at least twice rtt to start getting data back.

Tcp allows a window of at most 64k, and the acking mechanism means that packet loss is misdetected. tcp stalls easily under packet loss. tcp is more throttled by rtt than bandwidth.

Tcp transfer servers have to maintain a separate socket (and often separate thread) for each client.

Load balancing is crude and approximate. especially on local networks that allow collisions, two simultaneous tcp transfers have a tendency to fight with each other, even if the sender is the same.

8.10 Advantages of udp for file transfer :

Latency can be as low as rtt if the protocol is suitably designed.

Flow control is up to user space; windows can be infinite, artificial stalls nonexistant, latency well tolerated, and maximum speeds enforced only by real network bandwidth, yet actual speeds chosen by agreement of sender and receiver.

Receiving an image simultaneously from multiple hosts is much easier with udp, as is sending one to multiple hosts, especially if they happen to be part of the same broadcast or multicast group.

a single sending host with multiple transfers proceeding can balance them with excellent precision.The Internet runs on a hierarchical protocol stack. A simplified version of this is shown in figure 1 . The layer common to all Internet applications is the IP (Internet Protocol) layer. This layer provides a connectionless, unreliable packet based delivery service. It can be described as connectionless because packets are treated independently of all others. The service is unreliable because there is no guarantee of delivery. Packets may be silently dropped, duplicated or delayed and may arrive out of order. The service is also called a best effort service, all attempts to deliver a packet will be made, with unreliability only caused by hardware faults or exhausted resources.
As there is no sense of a connection at the IP level there are no simple methods to provide a quality of service (QoS). QoS is a request from an application to the network to provide a guarantee on the quality of a connection. This allows an application to request a fixed amount of bandwidth from the network, and assume it will be provided, once the QoS request has been accepted. Also a fixed delay, i.e. no jitter and in order delivery can be assumed. A network that supports QoS will be protected from congestion problems, as the network will refuse connections that request larger resources than can be supplied. An example of a network that supports QoS is the current telephone network, where every call is guaranteed the bandwidth for the call. Most users at some point have heard the overloaded signal where the network cannot provide the requested resource required to make a call.

The application decides which transport protocol is used. The two protocols shown here, TCP and UDP are the most commonly used ones. TCP provides a reliable connection and is used by the majority of current Internet applications. TCP, besides being responsible for error checking and correcting, is also responsible for controlling the speed at which this data is sent. TCP is capable of detecting congestion in the network and will back off transmission speed when congestion occurs. These features protect the network from congestion collapse.

As discussed in the introduction, VoIP is a real-time service. For real-time properties to be guaranteed to be met, a network with QoS must be used to provide fixed delay and bandwidth. It has already been said that IP cannot provide this. This then presents a choice. If IP is a requirement, which transport layer should be used to provide a system that is most likely to meet real-time constraints.

As TCP provides features such as congestion control, it would be the preferred protocol to use. Unfortunately due to the fact that TCP is a reliable service, delays will be introduced whenever a bit error or packet loss occurs. This delay is caused by retransmission of the broken packet, along with any successive packets that may have already been sent. This can be a large source of jitter.

TCP uses a combination of four algorithms to provide congestion control, slow start, congestion avoidance, fast retransmit and fast recovery. These algorithms all use packet loss as an indication of congestion, and all alter the number of packets TCP will send before waiting for acknowledgments of those packets. These alterations affect the bandwidth available and also change delays seen on a link, providing another source of jitter.

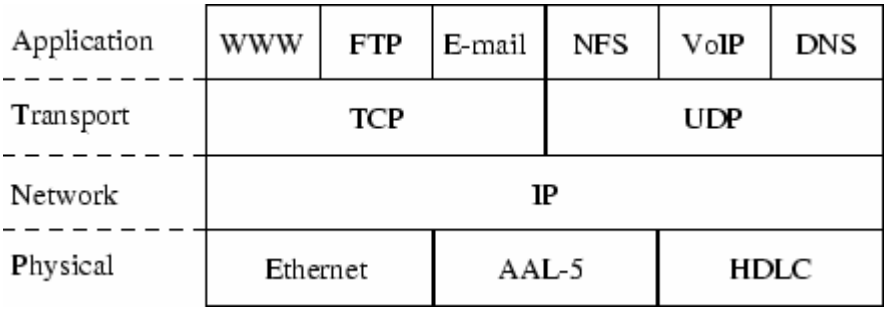


Figure 1: Simplified IP protocol stack

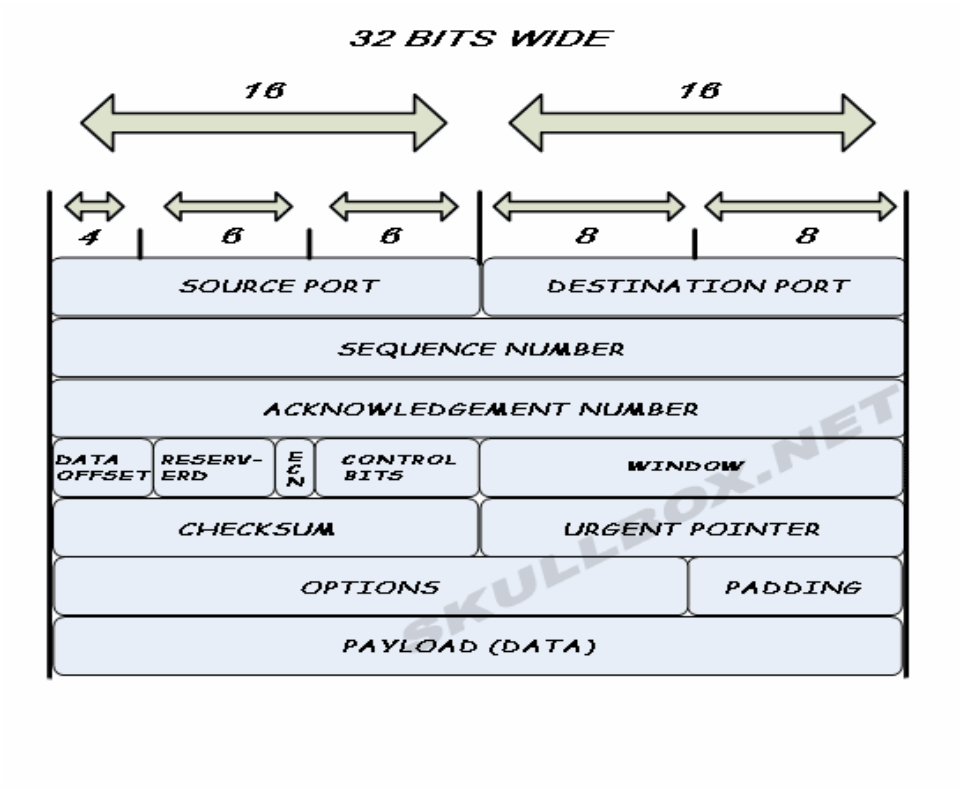
Combined, TCP raises jitter to an unacceptable level rendering TCP unusable for real-time services. Voice communication has the advantage of not requiring a completely reliable transport level. The loss of a packet or bit error will often only introduce a click or a minor break into the output.

For these reasons most VoIP applications use UDP for the voice data transmission. UDP is a thin layer on top of IP that provides a way to distinguish among multiple programs running on a single machine. UDP also inherits all of the properties of IP that TCP attempts to hide. UDP is therefore also a packet based, connectionless, best-effort service. It is up to the application to split data into packets, and provide any necessary error checking that is required.

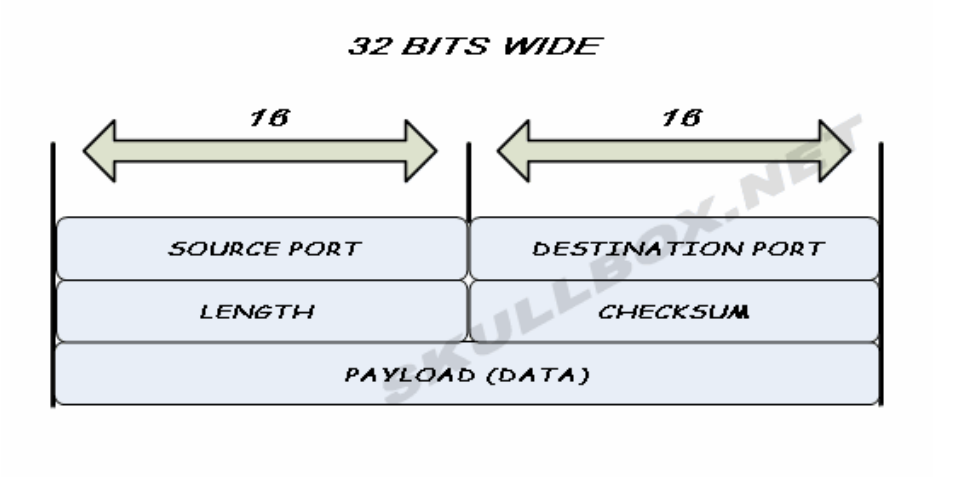
Because of this, UDP allows the fastest and most simple way of transmitting data to the receiver. There is no interference in the stream of data that can be possibly avoided. This provides the way for an application to get as close to meeting real-time constraints as possible.

UDP however provides no congestion control systems. A congested link that is only running TCP will be approximately fair to all users. When UDP data is introduced into this link, there is no requirement for the UDP data rates to back off, forcing the remaining TCP connections to back off even further. This can be though of as UDP data not being a ``good citizen''. The aim of this project is to characterise the quantity of this drop off in TCP performance.

8.11 TCP FRAME STRUCTURE:



8.12 UDP FRAME STRUCTURE



9.0 Factors that Impact the QoS of VOIP:

There are some factors that effect on the quality of the VoIP negatively. They are following

- Latency: Delay for packet delivery
- Jitter: Variations in delay of packet delivery
- Packet loss: Too much traffic in the network causes the network to drop packets

9.1 Latency:

Latency[14] is a time delay between the moment something is initiated, and the moment one of its effects begins or becomes detectable. The word derives from the fact that during the period of latency the effects of an action are latent, meaning "potential" or "not yet observed". Even within an engineering context, latency has several meanings depending on the engineering area concerned (i.e. communication, operational, simulation, mechanical or biomedical fiber stimulation latencies).

Latency in a packet-switched network is measured either one-way (the time from the source sending a packet to the destination receiving it), or round-trip (the one-way latency from source to destination plus the one-way latency from the destination back to the source). Round-trip latency is more often quoted, because it can be measured from a single point. Note that round trip latency excludes the amount of time that a destination system spends processing the packet. Many software platforms provide a service called ping that can be used to measure round-trip latency. Ping performs no packet processing; it merely sends a response back when it receives a packet (i.e. performs a no-op), thus it is a relatively accurate way of measuring latency.

Where precision is important, one-way latency for a link can be more strictly defined as the time from the start of packet transmission to the start of packet reception. The time from the start of packet reception to the end of packet reception is measured separately and called "Serialization Delay". This definition of latency is independent of the link's throughput and the size of the packet, and is the absolute minimum delay possible with that link.

However, in a non-trivial network, a typical packet will be forwarded over many links via many gateways, each of which will not begin to forward the packet until it has been completely received. In such a network, the minimal latency is the sum of the minimum latency of each link, plus the transmission delay of each link except the final one, plus the forwarding latency of each gateway. In practice, this minimal latency is further augmented by queuing and processing delays. Queuing delay occurs when a gateway receives multiple packets from different sources heading towards the same destination. Since typically only one packet can be transmitted at a time, some of the

packets must queue for transmission, incurring additional delay. Processing delays are incurred while a gateway determines what to do with a newly received packet. The combination of propagation, serialization, queuing, and processing delays often produces a complex and variable network latency profile.

Two key elements of network performance are bandwidth and latency. The average person is more familiar with the concept of bandwidth as that is the one advertised by manufacturers of network equipment. However, latency matters equally to the end user experience as the behavior of satellite Internet connections illustrates. Businesses use the term Quality of Service (QoS) to refer to measuring and maintaining consistent performance on a network by managing both bandwidth and latency in a coordinated fashion.

9.2 Jitter:

In voice over IP (VoIP), jitter[14] is the variation in the time between packets arriving, caused by network congestion, timing drift, or route changes. A jitter buffer can be used to handle jitter.
Jitter is the deviation in or displacement of some aspect of the pulses in a high-frequency digital signal. As the name suggests, jitter can be thought of as shaky pulses. The deviation can be in terms of amplitude, phase timing, or the width of the signal pulse. Another definition is that it is "the period frequency displacement of the signal from its ideal location." Among the causes of jitter are electromagnetic interference (EMI) and crosstalk with other signals. Jitter can cause a display monitor to flicker; affect the ability of the processor in a personal computer to perform as intended; introduce clicks or other undesired effects in audio signals, and loss of transmitted data between network devices. The amount of allowable jitter depends greatly on the application.

9.3 Packet Loss:

Packet loss[14] is the failure of one or more transmitted packets to arrive at their destination. This event can cause noticeable effects in all types of digital communications.

9.4 The effects of packet loss:

In text and data, packet loss produces errors.
In videoconference environments it can create jitter.
In pure audio communications, such as VoIP, it can cause jitter and frequent gaps in received speech.

In the worst cases, packet loss can cause severe mutilation of received data, broken-up images, unintelligible speech or even the complete absence of a received signal.

The causes of packet loss include inadequate signal strength at the destination, natural or human-made interference, excessive system noise, hardware failure, software corruption or overburdened network nodes. Often more than one of these factors is involved.

In a case where the cause can not be remedied, packet loss concealment may be used to minimize the effects of lost packets.

9.5 Packet recovery:

Some network transport protocols such as TCP provide for reliable delivery of packets. In the event of packet loss, the receiver asks for retransmission or the sender automatically resends any segments that have not been acknowledged. Although TCP can recover from packet loss, retransmitting missing packets causes the throughput of the connection to decrease. This drop in throughput is due to the sliding window protocols used for acknowledgment of received packets. In certain variants of TCP, if a transmitted packet is lost, it will be re-sent along with every packet that had been sent after it. This retransmission causes the overall throughput of the connection to drop.

Protocols such as UDP provide no recovery for lost packets. Applications that use UDP are designed to handle this type of pack.

9.6 Acceptable packet loss:

Perhaps the way to garner this number and still maintain a semblance of objectivity is to offer target numbers for different classes of service or types of architectures

For example, certain routers allow you to "prioritize" your traffic according to content [putting the different types into different queues]. For such architecture, you might specify that the highest priority for a critical service, packet types should pass with less than 1% packet loss. Lower priority packet types might pass with less than 5% and then 10% for the lowest of priority of services.

Packet loss is closely associated with quality of service considerations, and is related to the erlang unit of measure.

10.0 Differentiated services:

Differentiated Services[15] or DiffServ is a computer networking architecture that specifies a simple, scalable and coarse-grained mechanism for classifying, managing network traffic and providing Quality of Service (QoS)[14] guarantees on modern IP networks. DiffServ can, for example, be used to provide low-latency, guaranteed service (GS)[14] to critical network traffic such as voice or video while providing simple best-effort traffic guarantees to non-critical services such as web traffic or file transfers.

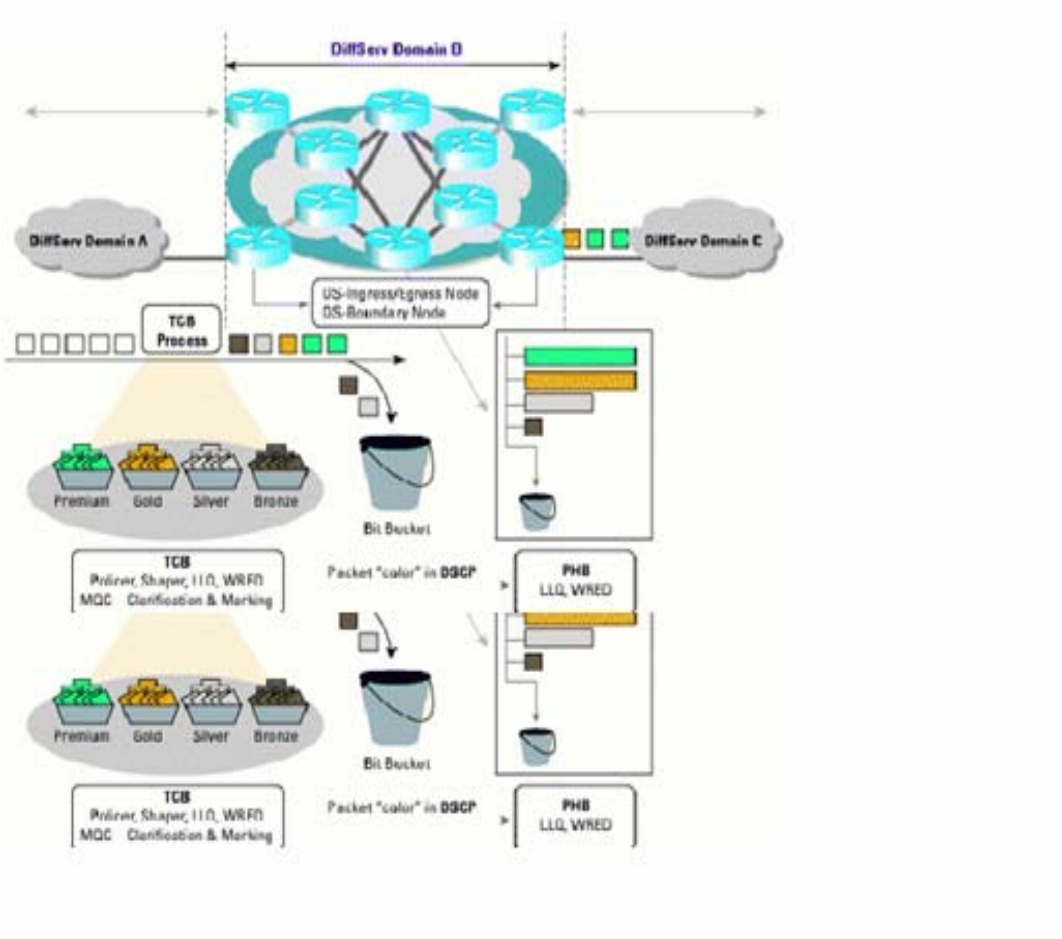


Fig: Diffserv Architecture

10.1 Traffic Management Mechanisms:

DiffServ is a coarse-grained, class-based mechanism for traffic management. In contrast, IntServ is a fine-grained, flow-based mechanism.

DiffServ operates on the principle of traffic classification, where each data packet is placed into a limited number of traffic classes, rather than differentiating network traffic based on the requirements of an individual flow. Each router on the network is configured to differentiate traffic based on its class. Each traffic class can be managed differently, ensuring preferential treatment for higher-priority traffic on the network.

The DiffServ model does not make judgement on what types of traffic should be given priority treatment since that is left up to the network operator. DiffServ simply provides a framework to allow classification and differentiated treatment. DiffServ does recommend a standardized set of traffic classes (discussed below) to make interoperability between different networks and different vendors' equipment simpler.

DiffServ relies on a mechanism to classify and mark packets as belonging to a specific class. DiffServ-aware routers implement Per-Hop Behaviors (PHBs), which define the packet forwarding properties associated with a class of traffic. Different PHBs may be defined to offer, for example, low-loss, low-latency forwarding properties or best-effort forwarding properties. All the traffic flowing through a router that belongs to the same class is referred to as a Behavior Aggregate (BA).

10.2 DiffServ Domain:

A group of routers that implement common, administratively defined DiffServ policies are referred to as a DiffServ Domain.

10.3 Classification and Marking:

Network traffic entering a DiffServ domain is subjected to classification and conditioning. Traffic may be classified by many different parameters, such as source address, destination address or traffic type and assigned to a specific traffic class. Traffic classifiers may honor any DiffServ markings in received packets or may elect to ignore or override those markings. Because network operators want tight control over volumes and type of traffic in a given class, it is very rare that the network honors markings at the ingress to the DiffServ domain. Traffic in each class may be further conditioned by subjecting the traffic to rate limiters, traffic policers or shapers.

10.4 Per-Hop Behavior:

The Per-Hop Behavior (PHB) is indicated by encoding a 6-bit value—called the Differentiated Services Code Point (DSCP)—into the 8-bit Differentiated Services (DS) field of the IP packet header. The DS field is the same as the TOS field, and ECN occupies the upper 2 bits.

In theory, a network could have up to 64 (i.e. 26) different traffic classes using different markings in the DSCP. The DiffServ RFCs recommend, but do not require, certain encodings. This gives a network operator great flexibility in defining traffic classes. In practice, however, most networks use the following commonly-defined Per-Hop Behaviors:

Default PHB—which is typically best-effort traffic
Expedited Forwarding (EF) PHB—for low-loss, low-latency traffic
Assured Forwarding (AF)—behavior group
Class Selector PHBs—which are defined to maintain backward compatibility with the IP Precedence field.

10.5 Default PHB:

A default PHB is the only required behavior. Essentially, any traffic that does not meet the requirements of any of the other defined classes is placed in the default PHB. Typically, the default PHB has best-effort forwarding characteristics. The recommended DSCP for the default PHB is '000000'.

10.6 Expedited Forwarding (EF) PHB:

The IETF defines Expedited Forwarding in RFC 3246. The EF PHB has the characteristics of low delay, low loss and low jitter. These characteristics are suitable for voice, video and other realtime services. EF traffic is often given strict priority queuing above all other traffic classes. Because an overload of EF traffic will cause queuing delays and affect the jitter and delay tolerances within the class, EF traffic is often strictly controlled through admission control, policing and other mechanisms. Typical networks will limit EF traffic to no more than 30%—and often much less—of the capacity of a link.

10.7 Assured Forwarding (AF) PHB Behavior Group:

The IETF defines the Assured Forwarding behavior group in RFC 2597. Assured forwarding allows the operator to provide assurance of delivery as long as the traffic does not exceed some subscribed rate. Traffic that exceeds the subscription rate faces a higher probability of being dropped if congestion occurs.

The AF behavior group defines four separate AF classes. Within each class, packets are given a drop precedence (high, medium or low). The combination of classes and drop precedence yields twelve separate DSCP encodings from AF11 through AF43

Some measure of priority and proportional fairness is defined between traffic in different classes. Should congestion occur between classes, the traffic in the higher class is given priority. Rather than using strict priority queueing, more balanced queue servicing algorithms such as fair queueing or weighted fair queuing are likely to be used. If congestion occurs within a class, the packets with the higher drop precedence are discarded first. To prevent issues associated with tail drop, the random early detection (RED) or weighted random early detection (WRED) algorithms are often used to drop packets.

Usually, traffic policing is required to encode drop precedence. Typically, all traffic assigned to a class is initially given a low drop precedence. As the traffic rate exceeds subscription thresholds, the policer will increase the drop precedence of packets that exceed the threshold.

10.8 Class Selector PHB:

Prior to DiffServ, IP networks could use the Precedence field in the Type of Service (TOS) byte of the IP header to mark priority traffic. The TOS byte and IP precedence was not widely used. The IETF agreed to reuse the TOS byte as the DS field for DiffServ networks. In order to maintain backward compatibility with network devices that still use the Precedence field, DiffServ defines the Class Selector PHB.

The Class Selector codepoints are of the form 'xxx000'. The first three bits are the IP precedence bits. Each IP precedence value can be mapped into a DiffServ class. If a packet is received from a non-DiffServ aware router that used IP precedence markings, the DiffServ router can still understand the encoding as a Class Selector codepoint.

10.9 Advantages of DiffServ:

One advantage of DiffServ is that all the policing and classifying is done at the boundaries between DiffServ clouds. This means that in the core of the Internet, routers can get on with doing the job of routing, and not care about the complexities of collecting payment or enforcing agreements.That is, DiffServ requires no advance setup, no reservation, and no time-consuming end-to-end negotiation for each flow, as with integrated services. This leads DS relatively easy to implement.

10.10 Disadvantages of DiffServ:

End-to-end and peering problems
One disadvantage is that the details of how individual routers deal with the type of service field is somewhat arbitrary, and it is difficult to predict end-to-

end behaviour. This is complicated further if a packet crosses two or more DiffServ clouds before reaching its destination.

From a commercial viewpoint, this is a major flaw, as it means that it is impossible to sell different classes of end-to-end connectivity to end users, as one provider's Gold packet may be another's Bronze. Internet operators could fix this, by enforcing standardised policies across networks, but are not keen on adding new levels of complexity to their already complex peering agreements. One of the reasons for this is set out below.

Diffserv operation only works if the boundary hosts honour the policy agreed upon. However, this assumption is naive as human beings rarely agree. A host can always tag its own traffic with a higher precedence, even though the traffic doesn't qualify to be handled with that importance. This in fact has already been exploited: Microsoft Windows 2000 always tags its traffic with IP precedence 5, making the traffic classing useless. On the other hand, the network is usually quite within its rights to traffic shape and otherwise ration the amount of network traffic ingress with any particular precedence, and so where this is enforced, overall network traffic flow provided to a host would be reduced by such a tactic.

10.11 DiffServ vs. more capacity:

The greatest disadvantage of DiffServ is that at the very highest level, some regard it as a technical solution for a technical problem which does not exist if the capacity of Internet links is properly engineered.

Since DiffServ is simply a mechanism for deciding which packets to delay or drop at the expense of others in a situation where there is not enough network capacity, consider that when DiffServ is working by dropping packets selectively, traffic on the link in question must already be very close to saturation. Any further increase in traffic will result in Bronze services being taken out altogether. Since Internet traffic is highly bursty, this is almost certain to happen on a regular basis if traffic on a link is near the limit at which DiffServ becomes needed. (However, the network can be provisioned to provide a minimum Bronze bandwidth, by limiting the maximum amount of higher priority traffic.)

For this reason, many people think that DiffServ will always be inferior to adding sufficient network capacity to avoid packet loss on all classes of traffic.

As of 2003, there is a glut of fibre capacity in most parts of the telecoms market, with it being far easier and cheaper to add more capacity than to employ elaborate DiffServ policies as a way of increasing customer satisfaction. In fact, this is what is generally done in the core of the Internet, which is generally fast and dumb with "fat pipes" connecting its routers.

However with wireless links, such as EV-DO, where the air-interface bandwidth is several orders of magnitude less than the backhaul, QoS is being used to efficiently deliver VoIP packets where not otherwise achievable.

10.12 Effects of dropped packets:

Dropping packets wastes the resources that have already been expended in carrying these packets so far through the network. In many cases, this traffic will be re-transmitted, causing further bandwidth consumption at the congestion point and elsewhere in the network.[citation needed] To minimize this waste, packets must be discarded as close to the edge of the network as possible, while Diffserv is often implemented throughout a network (edge and core)

Thus, dropping packets amounts to betting that congestion will have resolved by the time the packets are re-sent,[citation needed] or that (if the dropped packets are TCP datagrams) TCP will throttle back transmission rates at the sources to reduce congestion in the network. The TCP congestion avoidance algorithms are subject to a phenomenon called TCP global synchronization unless special approaches (such as Random early detection) are taken when dropping TCP packets. In Global Synchronization, all TCP streams tend to build up their transmission rates together, reach the peak throughput of the network, and all crash together to a lower rate as packets are dropped, only to repeat the process.

Delays caused by re-scheduling packets due to Diffserv can cause packets to drop by the IPsec anti-replay mechanism.

DiffServ as rationing:

Hence, DiffServ is for most ISPs mainly a way of rationing customer network utilisation to allow greater overbooking of their capacity. A good example of this is the use of DiffServ tools to suppress or control peer-to-peer traffic, because of its ability to saturate customer links indefinitely, disrupting the ISP's business model which relies on 1%-10% link utilization for most online customers.

10.13 Bandwidth Broker:

RFC 2638 from IETF defines the entity of the Bandwidth Broker in the framework of DiffServ. According to RFC 2638, a Bandwidth Broker is an agent that has some knowledge of an organization's priorities and policies and allocates bandwidth with respect to those policies. In order to achieve an end-to-end allocation of resources across separate domains, the Bandwidth Broker managing a domain will have to communicate with its adjacent peers, which allows end-to-end services to be constructed out of purely bilateral

agreements. Bandwidth Brokers can be configured with organizational policies, keep track of the current allocation of marked traffic, and interpret new requests to mark traffic in light of the policies and current allocation. Bandwidth Brokers only need to establish relationships of limited trust with their peers in adjacent domains, unlike schemes that require the setting of flow specifications in routers throughout an end-to-end path. In practical technical terms, the Bandwidth Broker architecture makes it possible to keep state on an administrative domain basis, rather than at every router and the DiffServ architecture makes it possible to confine per flow state to just the leaf routers.

11.0 Simulation and Scientific Experiment:

In this project we have undertaken some simulation experiments to investigate possible approaches for improving QoS. Below we have a complex network of the telecommunica tion system (in outline). 10000's of such access networks of the sort shown here, together with the core network create an entire real world network. Studying an individual link in this network, will lead us towards a solution for the entire network. It shows the model we have used in the simulations. In this network both VOIP and TCP/IP data traffic will be used to test the link and bottleneck condition.

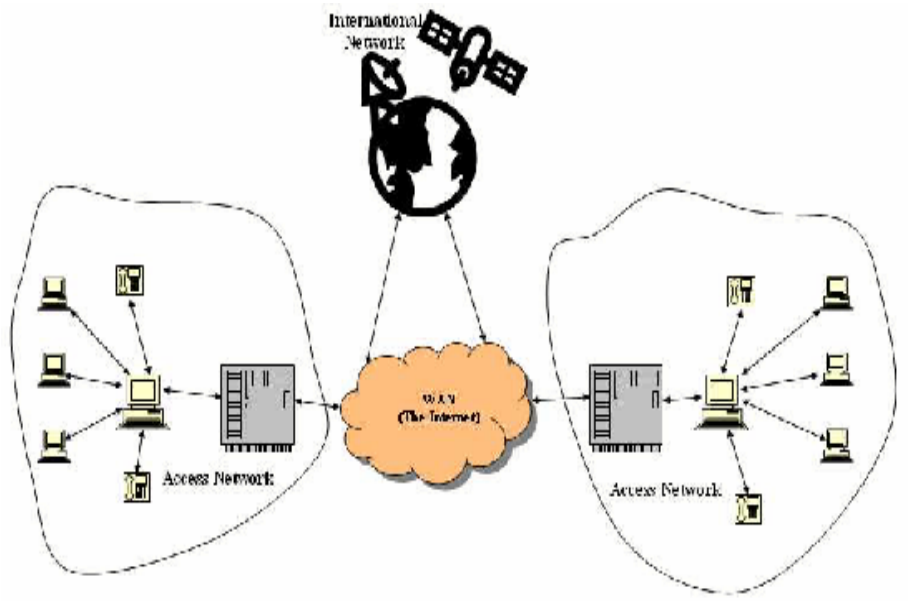
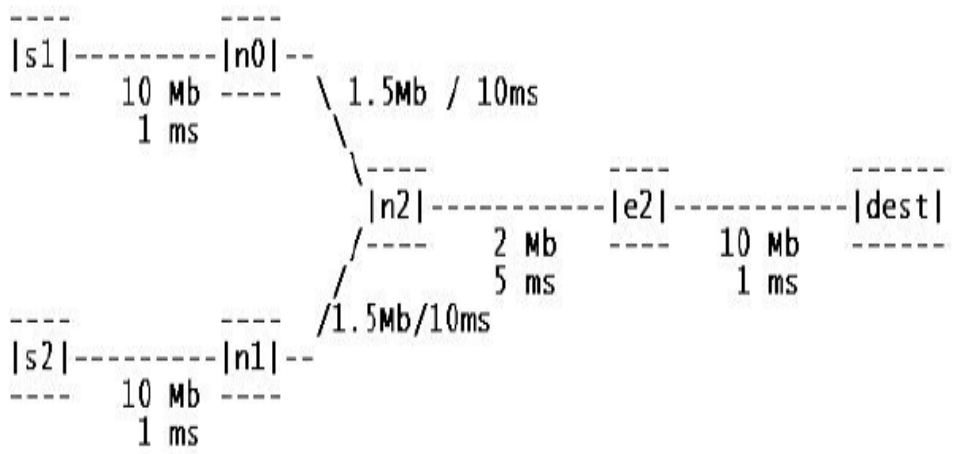


Fig: Real life picture of a typical network

VoIP Model with Diffserv Extension:

In the following Figure n0, n1, n2, n3 are four IP routing nodes. In these four nodes n0 is a TCP/IP node and FTP is the traffic gene rator for this node. The TCP/IP node has been attached to node n3 via n2 where a SINK is attached. These all a re duplex links. Again on the other side node n1 is linked via n2 to n3 and a NULL agent, whic h just frees the packe ts received, is attached to it. With node n1 a CBR traffic generator is attache d whic h generates the traffic or the UDP connection. There are buffers at the head of every link in the network. We focus on one of these buffers, namely the one at the head of the link from n2 to n3. We have performed three different simulation experiments using the NS2

simulation software. These three simulations are based on the same basic model and we have performed these tests by changing the traffic control methods in the networkusing RED and finally we have performed an experiment using the Diffserv system.



---

we have generated several different types of outputs: graphical output using NAM and graphical output using xgraph and in the graphs we ha ve shown queue and loss versus time. All these tests were performed under specific controlled circumstances which will generate data and the output will lead us towards a clearer understanding and that can be further explored in later work. In the Figure n0 and n1 could be in an access network for example. This could be a small business or may be a home premise.

12.0 Experiment Outcomes:

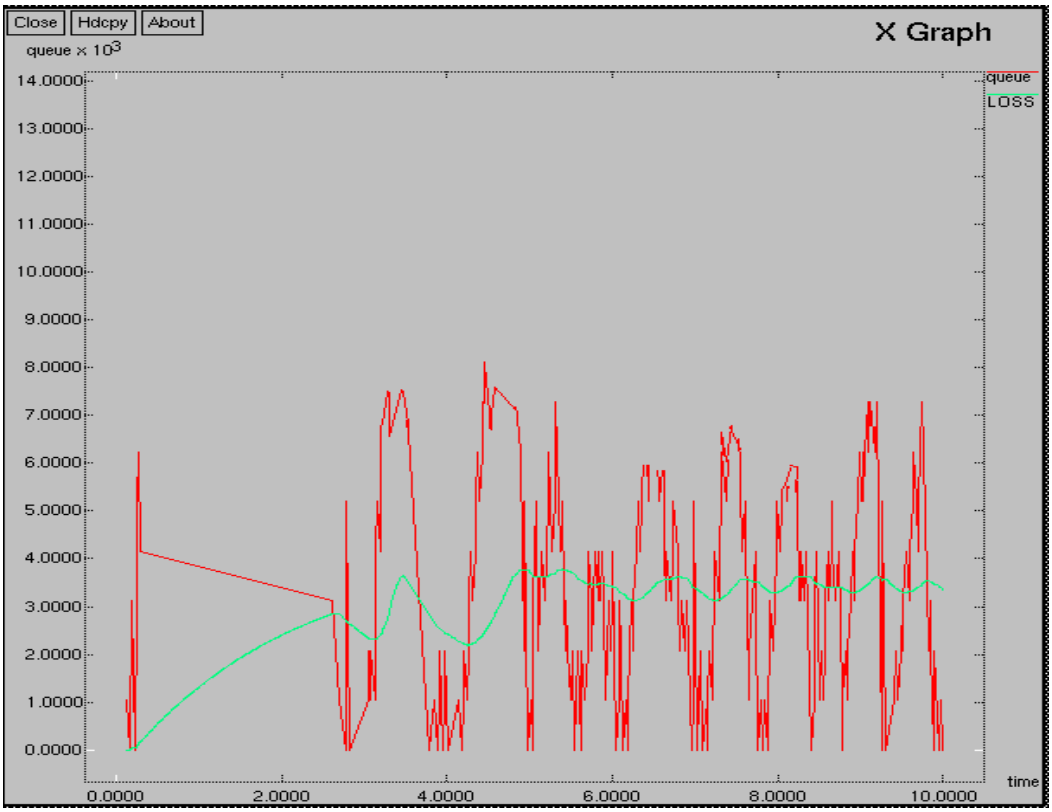


Figure 01

Buffer size 35 Bandwidth 2 Mbps

In this graph we ge can see drop of packects which is not little in number. Here we have buffer size 35 and bandwidth 2 Mbps. Such kind of packet drop is not desired. This rate of loss lead us simulate the next experiments.

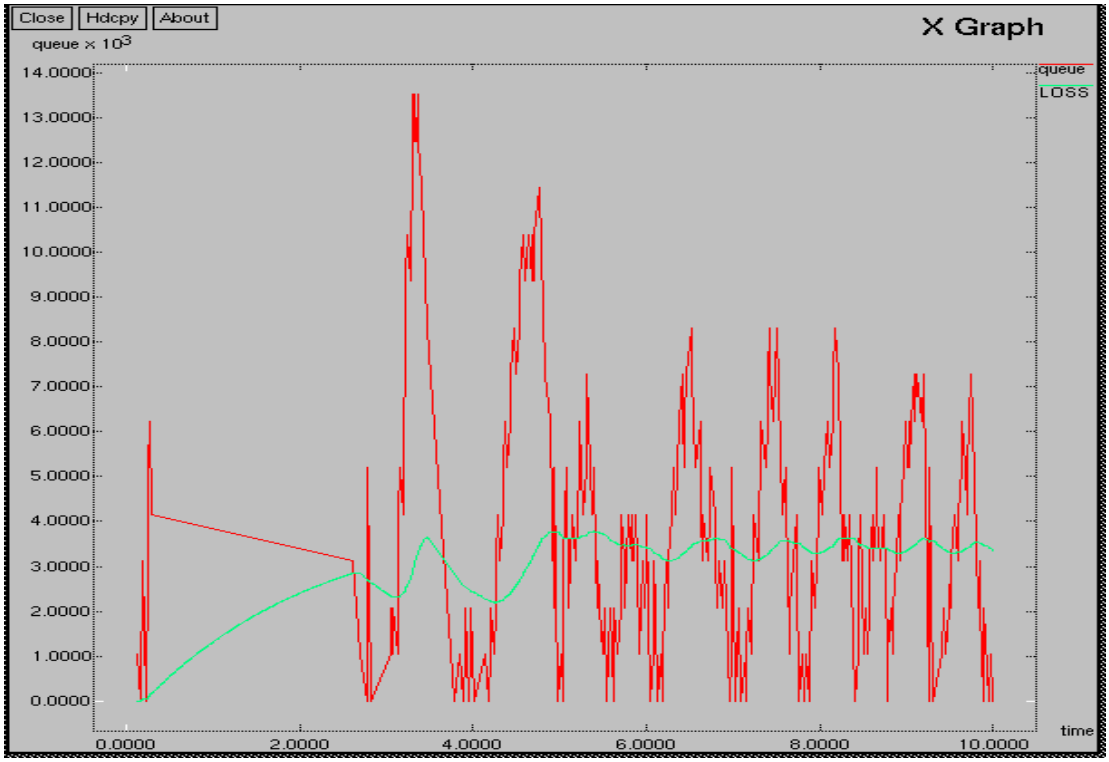


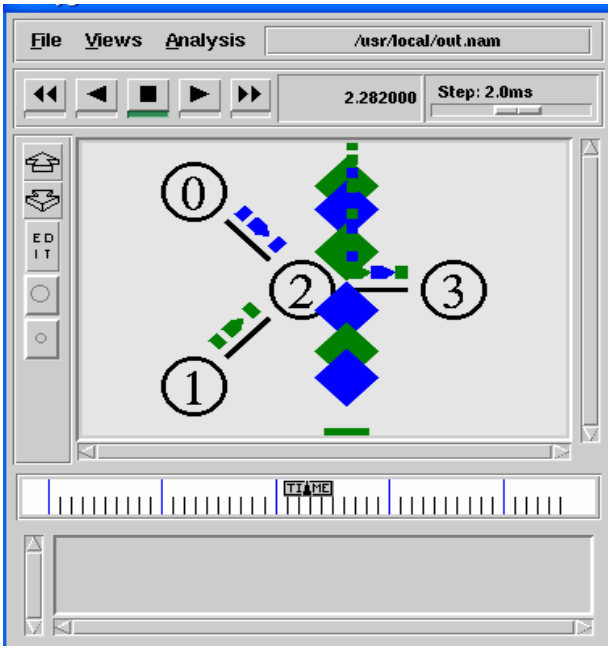
Figure 02

Buffer size 50 Bandwidth 2 Mbps

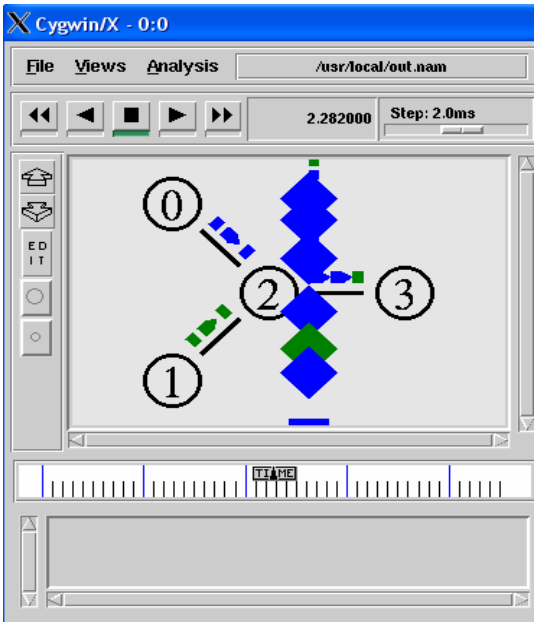
In this graph we see the similar scenario.we have increased the buffer size to 50 and bandwidth to 2 Mbps. In this experiment we see a large number of packet drop.

Network animator output:

In these following network animator file we see drop of packets. The blue slices are TCP packets and the green slices are the UDP packets.



In the first one we see similar rate of drop of both UDP and TCP packets.



But after increasing the buffer rate we see that a huge change in drop of packets. We see a less drop of UDP packets.

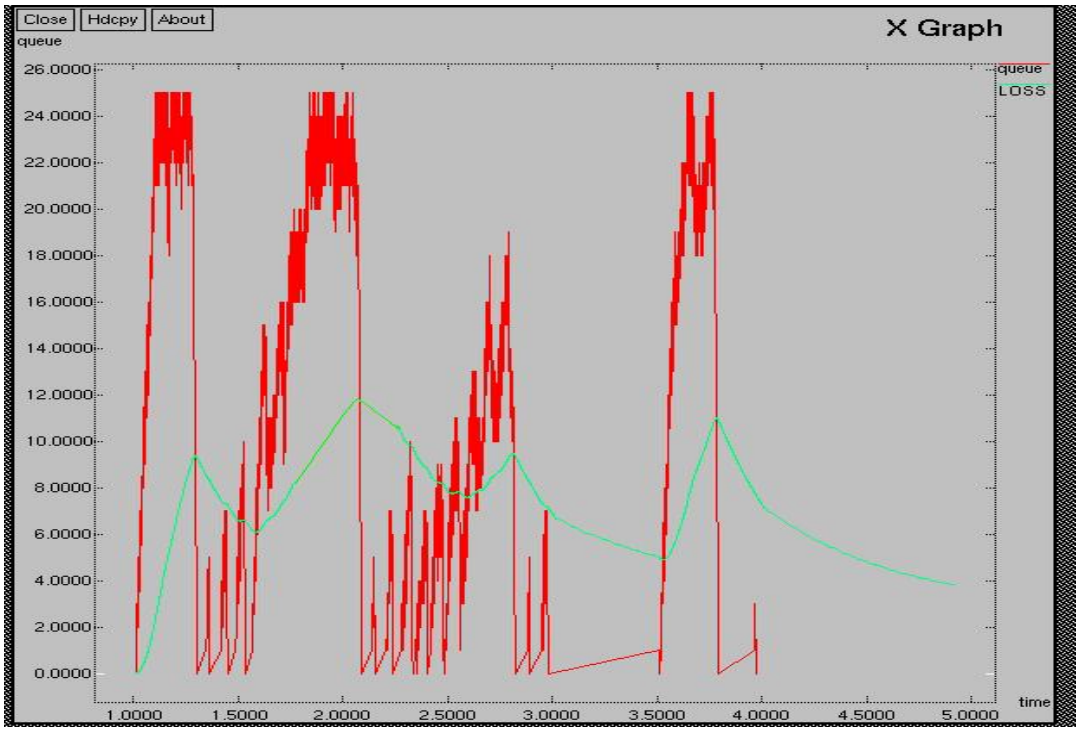


Figure 03

Buffer size 50 Bandwidth 5 Mbps

we have increased the buffer size to 50 and bandwidth to 5 Mbps. In this experiment we see improvement in packet drop.This experiment lead us to simulate with a greater bandwidth and in the later simulation we have even better result then this one.

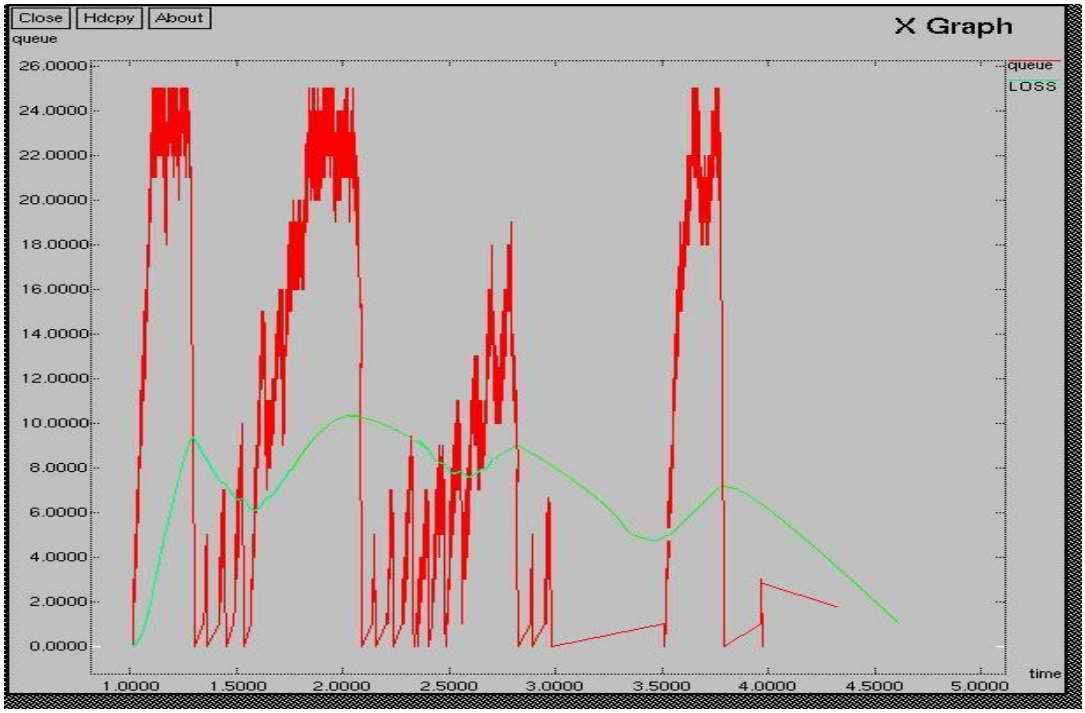


Figure 04

Buffer size 50 Bandwidth 10Mbps

In this figure we see improvement in terms of packet loss. By considering the buffer size constant if we increase the bandwidth to 10 Mbps then we have that result. We came to a conclusion that in this rate of badwidth and buffer size we can improve the VoIP performance by decreasing the packet drop.

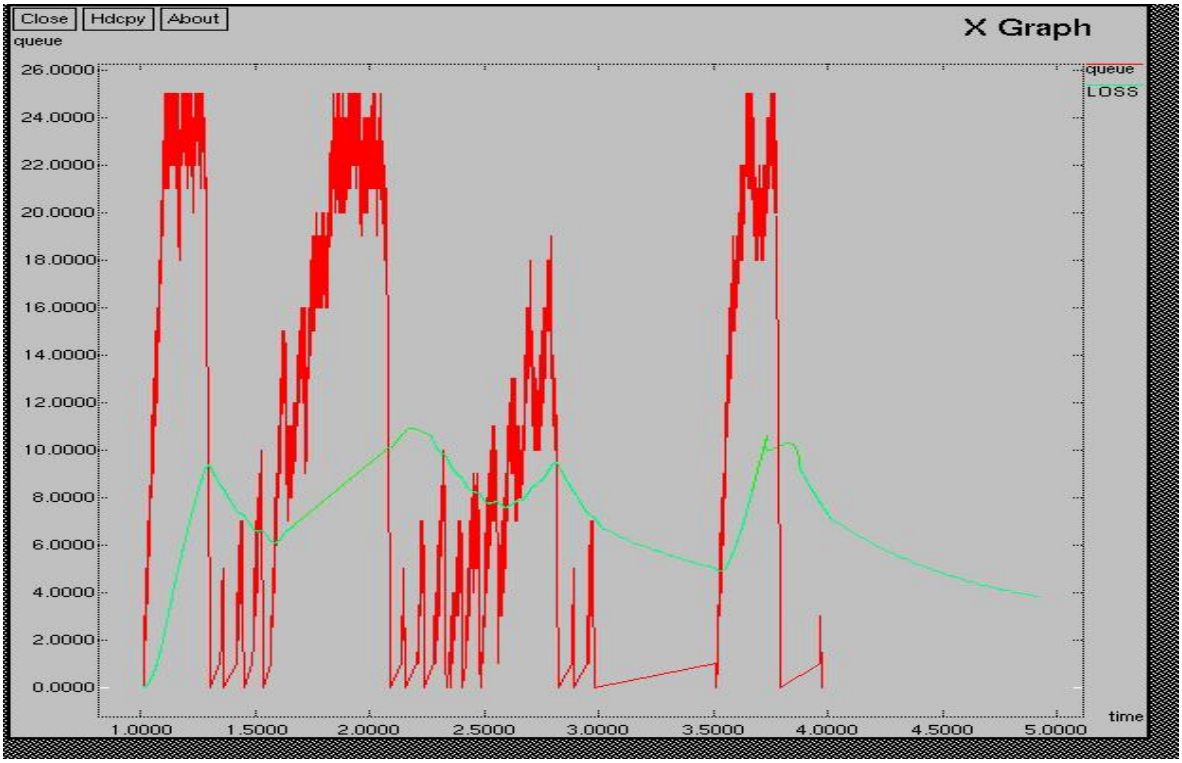


Figure 05

Buffer size 50 Bandwidth 20 Mbps

But again we have increased the bandwidth considering the buffer size constant to 50.There was no significant change in the rate of packet drop. It gives us a interesting result that after a certain limit if we increase the bandwidth then there will be no effect in terms of packet losss. It is a optimum situation.

By performing these simulation we can come to a position where we can say that bandwidth is not the only matter to increase or decrease the VoIP performance.

13.0 Future Work:

Providing reliable, high-quality voice communications over a network designed for data communications is a complex engineering challenge. Factors involved in designing a high-quality VoIP system include the choice of codec and call signaling protocol. There are also engineering tradeoffs between delay and efficiency of bandwidth utilization. In this thesis, simulations of VoIP using RED and Diffserv were carried out which gave some indication of the potential importance of DiffServ.

If we able to find the possible combination and permutation of the parameters and rate to increase to decrease that than we will be able to create a ideal scenario or a protocol to work for voip.

Study and simulate using different parameters rather then my one could be a research topic.

If we ever be able to find the perfect combination and permutation of different parameter rate and ratio then we might be able to create an ideal scenario.

14.0 Conclusion:

In the Conclusion we can say that simulating with different parameters requires a lot of time. We could study more and more parameters to understand the Traffic Behavior but one thing that we could not manage that was time.In this thesis we have studied one parameters and as I said earlier that was packet loss and more precisely the study of UDP packet drop. We have came to a solution where we can decrease the number of UDP packet loss in a simulated scenario and in a small VoIP Topology. Thousands of this small network build a real life network. If we are successfull in this scale then we will be able to successful in a braoder aspect.

15.0 LIST OF REFERENCES:

[1] Todd Rapposelli *"VoIP Overview" Release date* September 21, 2004
E-mail: Todd.Rapposelli@acterna.com

[2] *"Asynchronous transfer mode Fundamentals"*
From http://www-net.cs.umass.edu/cs653-2002/documents/atm_iec.pdf

[3] *"VoIP Codec Overview"* By Fujutsu Corporation
From http://www.fujitsu.com/downloads/MICRO/fma/pdf/voip.pdf

[4] Martin B.H. Weiss{mbw+@pitt.edu}Telecommunications Program,University of Pittsburgh,Pittsburgh PA15260JunseokHwang{hwang@tele.pitt.edu} Telecommunications Program,University of Pittsburgh,Pittsburgh PA 15260 *"Internet Telephony or Circuit Switched Telephony"* September 4, 1998

[5] "Belkin Telephone Adapter" From Belkin Corporation,501 West Walnut Street,Compton • CA • 90220 • USA

[6] "Overview of the PSTN" Public Switched Telephone Network,Neil Abramson ARRIS Digital Applications Engineering,Phone (720) 895-7158

[7] *"Understanding SIP",* Today's Hottest Communications ProtocolComes of Age From http://www.sipcenter.com/sip.nsf/html/WEBB5YNVK8/$FILE/ Ubiquity_SIP_Overview.pdf

[8] *"H.323 Protocol Overview"* By Paul E. Jones(October 2007) From http://www.packetizer.com/ipmc/h323/papers/h323_protocol_overview.pdf

[9] *"Real-timeTransport Protocol"* By Krzysztof Hebel,Multimedia Communications Laboratory, Electrical &Computer,Engineering Department,University of Waterloo,February 21st 2006

[10] *"Waveform Coding Techniques",*http://www.cisco.com/warp/public/788/ signalling/waveform_coding.pdf

[11] *"PULSE CODE MODULATION STANDARDS",*http://www.irig106.org/docs/106-05/chapter4.pdf

[12] *"User Datagram Protocol",*http://www.javvin.com/protocol/rfc768.pdf

[13] *"TRANSMI S S ION CONTROL PROTOCOL (TCP) FUNDAMENTALS AND GENERAL OPERATION"* http://nostarch.com/download/ tcpip_ch46.pdf

[14] *"A Network Simulator Differentiated Services Implementation"* ,Open IP, Nortel Networks,By Peter Pieda <ppieda@nortelnetworks.com>Jeremy Ethridge<jethridg@nortelnetworks.com> Mandeep BainesFarhan ShallwaniJuly 26, 2000

[15] *"Differentiated Services*"A Tutorial Overview with a Voice over IP Slant,Kathleen Nichols kmn@cisco.com ETSI Workhop on Voice over IP June 9, 1999

[16] "Impact of VoIP and QoS on Open and Distance Learning"P. C. SAXENA SCSS, Jawaharlal Nehru University,New Delhi,INDIA, Sanjay JASOLA,Computer Division Indira Gandhi National Open University,New Delhi, INDIA,Ramesh C. SHARMA Indira Gandhi National Open University ,New Delhi, INDIA

[17] *"Python Standard Library Network Protocols"* Niklaus Wirth URL: http://effbot.org/media/downloads/librarybook-network-protocols.pdf

[18] Jishu Das Gupta, University of Southern Queensland,Department of Mathematics and Computing ,Faculty of Sciences.

[19] Tariq Latif & Karanti Kumar Malkajgiri,Lulea University of Technology,Masters Thesis"Adonptation Of VoIP Technology"

[20] *" Traffic Behavious of VoIP in a simulated Access Netwrok"* By: Jishu DasGupta, Srecko Howard and Angela Howard