University of Southern Queensland

Department of Mathematics and Computing
Faculty of Sciences

Performance issues for VOIP in Access Networks

A dissertation submitted by

Jishu Das Gupta

In partial fulfilment of the requirements for the degree of
Masters of Computing

Supervisor: Associate Professor R. G. Addie

2005

Abstract

The is a general consensus that the Quality of Service (QoS) of Voice over Internet Protocol (VOIP) is of growing importance for research and study.

In this dissertation we investigate the performance of VOIP and the impact of resource limitations in the performance of Access Networks. The impact of VOIP performance in access networks is particularly important in regions where Internet resources are limited and the cost of improving these resources is prohibitive.

It is clear that perceived VOIP performance, as measured by mean opinion score in experiments where subjects are asked to rate communication quality, is determined by end-to-end delay on the communication path, delay variation, packet loss, echo, the coding algorithm in use and noise. These performance indicators can be measured and the contribution in the access network can be estimated. The relation between MOS and technical measurement is less well understood. We investigate the contribution of the access network to the overall performance of VOIP services and the ways in which access networks can be designed to improve VOIP performance. Issues of interest include the choice of coding rate, dynamic variation of coding rate, packet length, methods of controlling echo, and the use of Active Queue Management (AQM) in Access Network routers.

Methods for analyzing the impact of the access network on VOIP performance will be surveyed and reviewed. Also, we consider some approaches for improving performance of VOIP by doing some experiment using NS2 simulation software with a view to gaining a better understanding of the design of access networks.

Certification

I certify that the ideas, experimental work, results, analysis, software and conclusions reported in this dissertation are entirely my own effort, except where otherwise acknowledged. I also certify that the work is original and has not been previously submitted for any other award, except where otherwise acknowledged.

-- --------------------------

Signature of Candidate Date

-- --------------------------

Signature of Supervisor Date

-- --------------------------

ENDORSEMENT Date

Acknowledgement

This dissertation would not have become what it is without the help of several people. First of all, I would like to thank my Principle Supervisor Assoc. Prof. Dr. Ron Addie for providing me with useful suggestions about the overall structure and contents of my thesis and for helping me with several simulation experiments.

Special thanks go to Dr. Karla Ziri-Castro for providing me feedback and suggestion with the project and dissertation writing and Zhi Li for providing support with the Network Simulator.

Lastly, this thesis is dedicated to my lovely wife for her endless love, support and continuous encouragement.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Motivation

In just over a decade, the Internet has grown to over 40,000 sites and 3,000,000 hosts [1]. The startling growth of Internet technology, coupled with the relatively low deployment cost of IP networks, has pushed for an integrated "IP-based core" - a single network for data, video and voice access. However, the diverse service-requirements and novel traffic characteristics of the emerging Internet applications have posed many technical challenges that the Internet community must address in the near future, as the emerging multimedia applications begin to constitute an ever-increasing fraction of Internet traffic. High quality interactive voice and video applications can tolerate little delay variation and packet loss. Based on the ITU-T publication in [1] Table 1.1 contrasts the traffic characteristics and service requirements of the traditional data applications and multimedia applications.

Applications	Traffic Behaviour	QoS Requirements
Electronic mail (SMTP), file transfer (FTP), remote terminal (Telnet)	Small, batch file transfers.	Very tolerant of delay & loss. Bandwidth requirement: Low. Best effort.
HTML web browsing	A series of small, bursty file transfers.	Tolerant of moderate delay & loss. Bandwidth requirement: Low to moderate. Best effort.
Client-server Business Applications	Many small two-way transactions, "chatty".	Sensitive to loss & delay. Bandwidth requirement: Low to moderate. Best effort, must be stable & reliable.
IP-based voice (VOIP)	Constant bit rate or bursty traffic (with silence suppression).	Very sensitive to delay & jitter. Sensitive to loss. Bandwidth requirement: Low & predictable.

		Requires predictable delay & loss.
IP-based Real-time Video	Constant or variable bit rate (Depending on the codec).	Extremely sensitive to delay & jitter, and loss. Bandwidth requirement: High, e.g. 384 Kbps. Requires predictable delay & loss.

Table 1.1: Heterogeneous Traffic Behaviour and QoS Requirements of Internet Applications [1]

The striking feature of the services listed in Table 1.1 is the diversity of traffic characteristics. Nevertheless, the internet appears to be capable of providing the necessary communication infrastructure for all these services. The economic advantages of provisioning this range of service by means of a single infrastructure are considerable. Effective networking for this diverse range of multimedia applications requires in-depth research in various fields of the internet based application. IP-enabling the existing legacy PBXs is one option, but a more aggressive upgrade that involves pushing IP phones to the desktop is becoming financially attractive for a growing number of firms. In the end, it will be the difficult-to-measure VoIP-related productivity gains that will determine many firms' migration decisions. In low-cost countries, IP-enabling PBXs looks like the most financially viable strategy, while in average-cost countries, there's a better business case for buying new IP PBXs or for renting a hosted VoIP service. Firms that have upgraded their hub-and-spoke architecture to a mesh network architecture using MPLS VPN’s may gain even bigger savings from moving aggressively to VoIP [1].

1.2. Research Issues

Quality **O**f **S**ervice (QoS) is a defined level of performance in a data communications system [2]. As an example, to ensure that real time voice and video are delivered without annoying blips, a guaranteed bandwidth is required. The plain old telephone system (POTS) has delivered the highest quality of service for years, because there is a dedicated channel between parties. However, when data is broken into packets that travel through the same routers in the LAN or WAN with all other data, QoS mechanisms are the only way to guarantee quality by giving real time data priority over non-real time data [3].

In a recent paper of Bur Goode [4] it was mentioned that, during the recent Internet stock bubble, articles in the trade press frequently said that, in the near future, telephone traffic would be just another application running over the Internet. Such statements gloss over many engineering details that prevent voice from being just another Internet application. A large number of factors are involved in making a high-quality VOIP call. These factors include the speech codec, packetization delay, packet loss, delay (coding, transmission, propagation and queuing), delay variation, and the network architecture to provide QoS. Other factors involved in making a successful VOIP call include the call setup signalling protocol, call admission control, security concerns, and the ability to traverse Network Access Translation (NAT*) [2]and firewalls.

**Short for Network Address Translation, an Internet standard that enables a local-area network (LAN) to use one set of IP addresses for internal traffic and a second set of addresses for external traffic. A NAT box located where the LAN meets the Internet makes all necessary IP address translations.*
NAT serves three main purposes:

- *Provides a type of firewall by hiding internal IP addresses*
- *Enables a company to use more internal IP addresses. Since they're used internally only, there's no possibility of conflict with IP addresses used by other companies and organizations.*
- *Allows a company to combine multiple ISDN connections into a single Internet connection.*

Although VOIP involves the transmission of digitized voice in packets, the telephone itself may be analog or digital. The voice may be digitized and encoded either before or concurrently with packetization. Figure 1.1 shows a business in which a PBX is connected to VOIP gateway as well as to the local telephone company central office. The VOIP gateway allows telephone calls to be completed through the IP network. Local calls can still be completed through the telephone company as in the past. The business may use the IP network to make all calls between its VOIP gateway connected sites or it may choose to split the traffic between the IP network and the PSTN based on a least-cost routing algorithms configured in the PBX. VOIP calls are not restricted to telephones served directly by the IP network. We refer to VOIP calls to telephones served by the PSTN as "off-net" calls. Off-net calls must be routed over the IP network to a VOIP/PSTN gateway near the destination telephone.

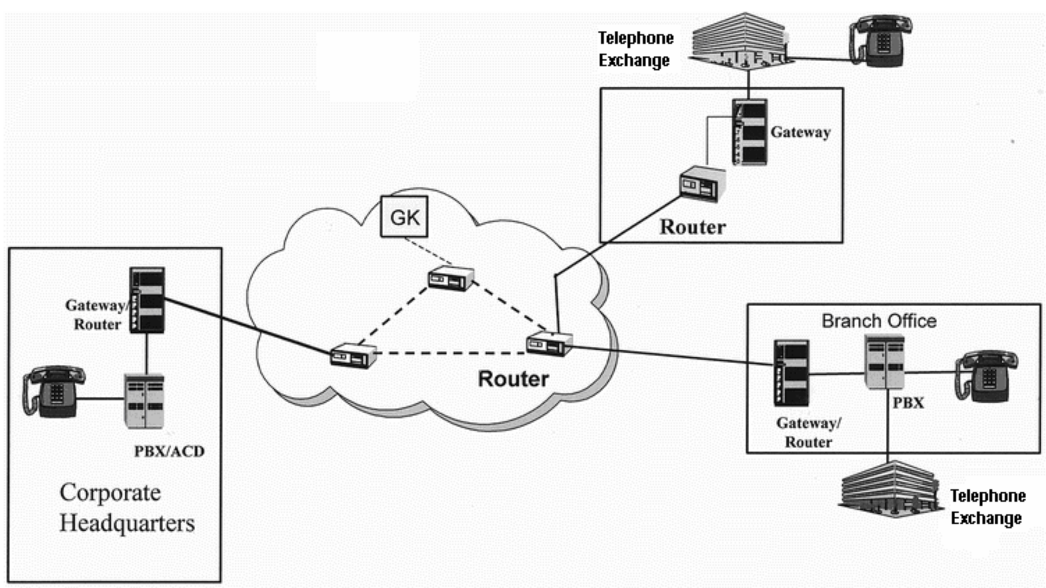


Figure 1.1 VOIP in Service [4]

An alternative VOIP implementation uses IP phones and does not rely on a standard PBX. Figure 1.2 is a simplified diagram of an IP telephone system connected to a wide area IP network. IP phones are connected to a LAN. Voice calls can be made locally over the LAN. The IP phones include codecs that digitize and encode (as well as decode) the speech. The IP phones also Packetsize and Depacketize the encoded speech. Calls between different sites can be made over the wide area IP network. Proxy servers perform IP phone registration and coordinate call signalling, especially between sites. Connections to the PSTN can be made through VOIP gateways.

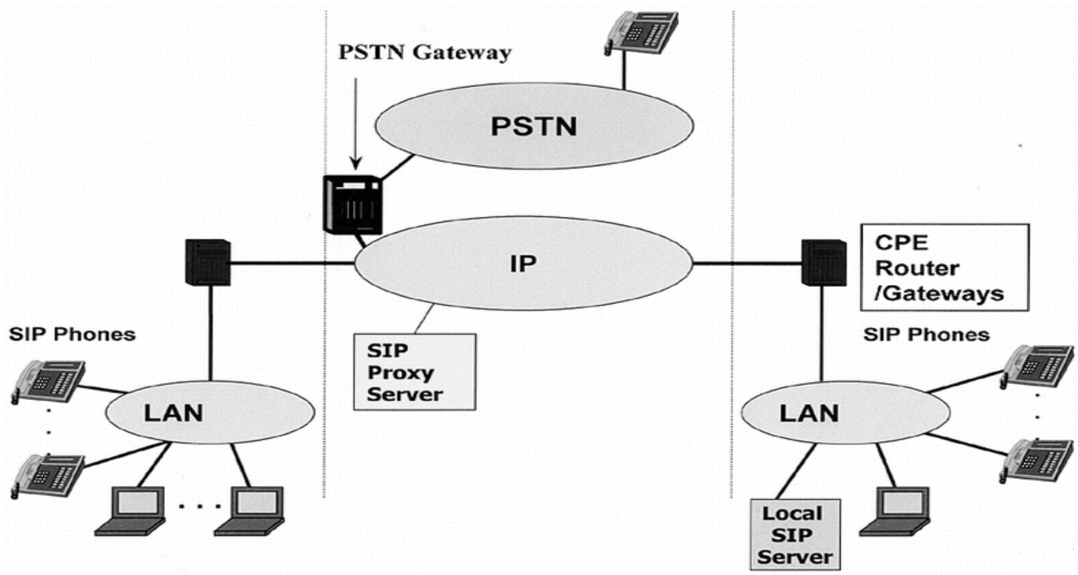


Figure 1.2 Simplified diagram of an IP telephone system [4]

** SIP : The Session Initiation Protocol (SIP) is an Internet Engineering Task Force (IETF) standard protocol for initiating an interactive user session that involves multimedia elements such as video, voice, chat, gaming, and virtual reality.*
SIP is a request-response protocol, dealing with requests from clients and responses from servers. Participants are identified by SIP URLs. Requests can be sent through any transport protocol, such as UDP, SCTP, or TCP. SIP determines the end system to be used for the session, the communication media and media parameters, and the called party's desire to engage in the communication. Once these are assured, SIP establishes call parameters at either end of the communication, and handles call transfer and termination [2]. See Chapter 7 for more details.

Figure 1.2 is a representation of a large network. For ascertaining the real problems we need to understand and investigate one simple link of this large network. For this issue we can focus on the point where the congestion (bottleneck) occurs. This will also facilitate us to develop an accurate understanding for the entire network. Since paths are made of links, understanding congestion in a path through the internet largely reduces to understanding congestion in one link. A simulation experiment investigating certain approaches to performance management in an access network will be presented in Chapter 6.

** Proxy Server: In an enterprise that uses the Internet, a proxy server is a server that acts as an intermediary between a workstation user and the Internet so that the enterprise can ensure security, administrative control, and caching service. A proxy server is associated with or part of a gateway server that separates the enterprise network from the outside network and a firewall server that protects the enterprise network from outside intrusion.*
A proxy server receives a request for an Internet service (such as a Web page request) from a user. If it passes filtering requirements, the proxy server, assuming it is also a cache server, looks in its local cache of previously downloaded Web pages. If it finds the page, it returns it to the user without needing to forward the request to the Internet. If the page is not in the cache, the proxy server, acting as a client on behalf of the user, uses one of its own IP addresses to request the page from the server out on the Internet. When the page is returned, the proxy server relates it to the original request and forwards it on to the user. [2]

1.3. Organization

In this Masters project we investigate and discuss the factors involved in making a high-quality VOIP call and the engineering tradeoffs that must be made between delay and the efficient use of bandwidth. After a discussion of codec selection and the delay budget, we shall discuss various techniques to achieve network quality of service. Since call setup is very important, the dissertation also presents an overview of several VOIP call signalling protocols, including H.323 and SIP.

Our first chapter describes the motivation and research issues and also elaborates the organization of the reminder of the dissertation. Chapter two mostly covers the general definition of the different components of VOIP. Chapter three focuses on the different advantages and disadvantages of VOIP. Chapter four investigates more closely the technical details of implementing the VOIP concept and details of its operation. In Chapter five we discuss the different factors, which create an effect on VOIP performance. Chapter six discusses a variety of methods for improving the quality of VOIP and presents some experiments concerning one such concept. In Chapter seven we describe the Session Initiation Protocol (SIP) in detail, with its different components and its structure. We also include the operational mechanisms of SIP. Finally in Chapter eight, we present conclusions and proposals for future work.

Chapter 2

VOIP Components

The rational of the Chapter 2 is to provide a clear picture of the components of VOIP. In this chapter we have presented the definition and the basic understanding of the different essential parts of the VOIP.

2.1. What is IP?

IP (Internet Protocol) is the method used to send data from one computer to another via the Internet; each data message is split into small parcels known as packets. IP Telephony covers the technologies that use this packet-switched approach to exchange voice, video, and other kinds of communication. VOIP is a part of all these technologies. Previously these have been carried using the circuit-switched connections on the voice orientated PSTN. The result of this is a converged IP-based network that provides a common transport platform for voice, video and data traffic [5].

2.2. What is VOIP?

Voice over Internet Protocol (VOIP) sends voice information in packets of digitized audio, in contrast to the constant flow of an analogue and digital signal in traditional circuit-switched telephone networks. In most cases, currently, conversion from analogue to digital occurs at the local exchange. VOIP provides telephone calls over an IP-based data network. Everyone is talking about VOIP, and everyone wants a piece of the action. Equipment developers and manufacturers see a window of opportunity to innovate and compete, and are consequently busy in developing new VOIP-enabled equipment attempting to break into a potentially lucrative market. Internet Service Providers (ISP's) see the possibility of competing with the PSTN for customers, particularly as today's PC users show a continued interest in the integration of voice and data applications, if not at work, then via Internet usage [6].

VOIP proponents expect an opportunity to provide a more flexible and feature-rich network at a significantly lower cost. The interest in VOIP arises out of the following factors:

Short-term:	Cheaper long distance calls, especially international calls.
Mid- to Long-term:	More efficient converged infrastructure for all types of traffic, simpler network and service management, and service access.
Long-term:	Open platform for telecom services and greater opportunities for innovation.

[7]

According to Bur Goode [4], although the Internet Protocol, is the most widespread of communication protocols, its main applications have been restricted to non-real time data, e.g. email, FTP, etc. Presently, in order to take the advantage of ubiquity of IP-based network and the bandwidth economies offered by packet transport, a whole new set of protocols and mechanisms is being introduced to carry real time traffic over such networks. Examples are – Multi Protocol Label Switching (MPLS), IP version 6 (IPv6), RSVP, Diffserv, IntServ. VOIP, as a technology, benefits from all such protocols and mechanisms. The VOIP technology can be used for PC to PC, PC to Phone and Phone to Phone telephony as well as for facsimile and video communication [4].

The building blocks of a VOIP network are – IP protocols, voice coding, delay control, public network interfaces and the SIP (IETF) and H.323 (ITU-T) standards. For signalling (setting up and clearing down calls) service users connect into the packet network via access networks at the originating and terminating ends [8]. Functionally, the customer premises equipment is at least a telephony device and the packet network comprises routers, media converters, protocol converters, multi point controllers, policy servers and billing and accounting servers.

2.3. IP Technology

IP determines the format of packets, also called datagrams, and in particular, provides the addressing for routing. Most services combine IP with one of two higher-level protocols, Transmission Control Protocol (TCP), which establishes a virtual connection between a destination and a source or User Datagram Protocol (UDP). UDP is more important than TCP for VOIP. In particular, for VOIP, retransmission of lost packets is inappropriate. TCP always retransmits lost packets, but UDP does not. IP by itself is something like the postal system. It allows you to address a package and drop it in the system, but there's no direct link between you and the recipient. TCP/IP, on the other hand, establishes a connection between two hosts so that they can send messages back and forth for a period of time [2].

IP-based infrastructure provides the flexibility to implement new products and services. The convergence of voice and data onto one delivery medium will enable the networks of the future to carry radio, television, data distribution, web, and telephony. Many enterprises are beginning to make use of IP for corporate networks, with its ability to provide for voice, data, and video through the same ‘pipe’. There is the potential to save not only on telephone costs, but also on infrastructure requirements when using IP. One of IP’s strengths has been the adaptation of packets to send data from one computer to another. The big plus point for the IP technology is it can handle low or high bit data over the internet. The requirement to leverage cost savings and flexibility from IP architecture has identified a number of shortcomings of the IP ‘best effort’ paradigm. This transport method is fine for emails and data files, where delivery is not time critical, but some of the new two-way applications, such as VOIP, now require data to be sent without any hold ups. For these new types of services, it is critical that routing delays, lost packets, and latency, do not impact on performance [5].

Here we should mention that the IP ‘pipe’ is the base carrier for both the TCP and UDP data. TCP retransmits the data if it is lost or destroyed. These characteristics give TCP an outstanding performance in the wide network. But retransmission is pointless for real time services because by the time a packet is retransmitted it will probably be out of date. For real time services, UDP is more appropriate.

2.4. The Access Network

The access network is the network to which customer premises equipment is directly connected, providing local services or wide area services via a Wide Area Network (WAN) [9].

Access networks connect to the "backbone," which is a network comprised of high-speed lines between major switching points. The switching points could be buildings on a campus or cities in a country.

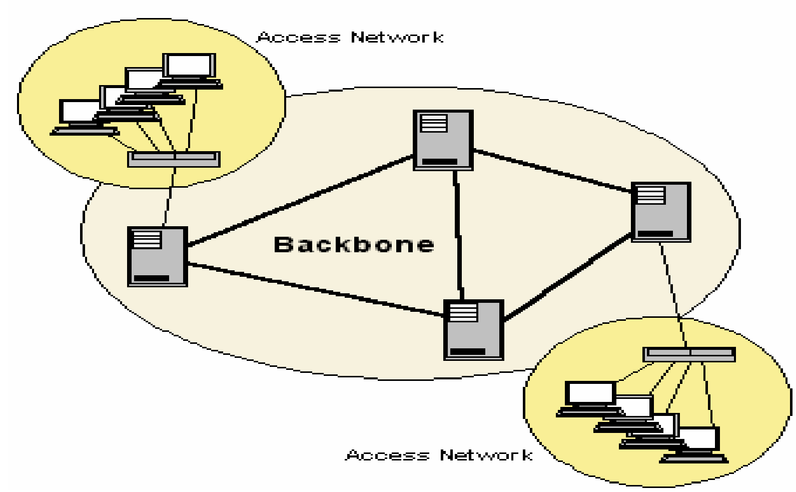


Figure 2.1 Typical Network

Achieving the global information infrastructure is one of the most challenging issues for future telecommunications networks in providing a common platform that uses open networking principles to offer multimedia services. Telecommunications networks can be divided into two parts, access networks and core networks.

For future multimedia services, one of the most crucial issues is building an access network that can accommodate multimedia services [1]. The access network is defined as a network entity providing access capabilities for various service applications (a wide variety of types of user terminals) to access various service providers (specific service nodes) located at the edges of the core networks. ITU-T is playing a leading role in studying GII for ITU-T standardization, and has the responsibility of studying architectural frameworks for telecommunications transport networks, including access networks and core networks. These studies should lead to versatile solutions for access networks which support a wide range of services, rather than a few restricted solutions. The studies will also lead to identifying new interfaces for various types of customer access networks and interfaces within the access networks to support GII and open networking concepts [10].

Although access networks provide only the first and last few kilometres of connectivity, because of the lower level of concentration of traffic, they make up a high proportion of the *cost* of communication. Also upgrading access networks e.g. from cable based communication to optical fibre, is very difficult & expensive. For this reason we will be forced to use all available technologies to get the best performance we can, from access networks for many years to come.

Difficulties of provisioning and design of access networks are more severe in regions of low population density such as regional and rural Australia and rural regions elsewhere. This is for the simple reason that the cables and ducts which physically comprise the network are longer & therefore more expensive and this cost must be borne by, or on behalf of, a smaller group of clients.

For this reason, getting the best performance from the limited resources is likely to be important, in relation to access networks, for a considerable time.

2.5. What is MOS?

In voice communications, particularly Internet telephony, the Mean Opinion Score (MOS) provides an aggregate numerical measure of the quality of human speech at the destination end of the circuit. The scheme uses subjective tests (opinion scores) that are mathematically averaged to obtain a quantitative indicator of overall quality.

Predictive codecs are commonly used in voice communications [11] because they conserve bandwidth. But they also degrade voice fidelity. The best codecs provide the most bandwidth conservation while producing the least degradation of the signal [12]. In addition, packetization delay is increased by the use of highly compressed coding, unless shorter packets are used. If shorter packets are used, the advantage of a highly compressed code is partially lost because of high level of overhead.

Technical quality like loss, delay, distortion, noise and echo can be measured by the technology. But subjective measures of quality are not possible to measure by the technology and for that we need MOS.

To determine MOS, a number of listeners rate the quality of test sentences read aloud over the communications circuit by male and female speakers. A listener gives each sentence a rating as follows:

(1) Bad;

(2) Poor;

(3) Fair;

(4) Good;

(5) Excellent;

The MOS is the arithmetic mean of all the individual scores, and can range from 1 (worst) to 5 (best) [2] [4].

2.6. Codecs

There are many codecs available for digitizing speech. Table 2.1 gives some of the characteristics of a few standard codecs. This data is based on the ITU-T standard [13].

Codec	Algorithm	*Frame Size/ Lookahead*	Usual Rate	Comments
G.711	PCM	0.125 ms/0	64 Kb/s	Universal use
G.722		0.125 ms/1.5 ms	48, 56 or 64 Kb/s	Wideband coder
G.726	ADPCM	0.125 ms/0	32 Kb/s	High quality, low complexity
G.728	LD-CELP	0.625 ms/0	16 Kb/s	High quality in tandem; Recommended for cable
G.729(A)	CS-ACELP	10 ms/5 ms	8 Kb/s	Widespread use
G.729e	Hybrid CELP	10 ms/5 ms	11.8 Kb/s	High quality/complexity; Recommended for cable
G.723.1(6.3)	MPC-MLQ	30 ms/7.5 ms	6.3 Kb/s	Video conferencing origin
G.723.1(5.3)	ACELP	30 ms/7.5 ms	5.3 Kb/s	Video conferencing origin
IS-127	RCELP	20 ms/5ms	Var. 4.2 Kb/s avg.	
AMR	ACELP	20 ms	Var. 4.75-12.2 Kb	Compatible w. No. Amer. & Japanese digital cellular, WCDMA (not CDMA2000); Nokia IPR

Table 2.1 Different Codecs and their characteristics

Bur Goode [4] has investigated the relation between MOS and codec choice in his paper on VOIP. The quality of a voice call through a codec is measured by subjective testing under controlled conditions using a large number of listeners to determine MOS. Several characteristics can be measured by varying the test conditions. Important characteristics include the effect of environmental noise, the effect of channel degradation (leading to bit errors), and the effect of tandem encoding/decoding when interworking with other wireless and terrestrial transport networks. The latter characteristic is important when VOIP networks interwork with switched circuit networks and wireless networks using different codecs. The general order of the fixed rate codecs listed in the table, from best to worst performance in tandem, is G.711, G.726, G.729e, G.728, G.729 and G.723.1.

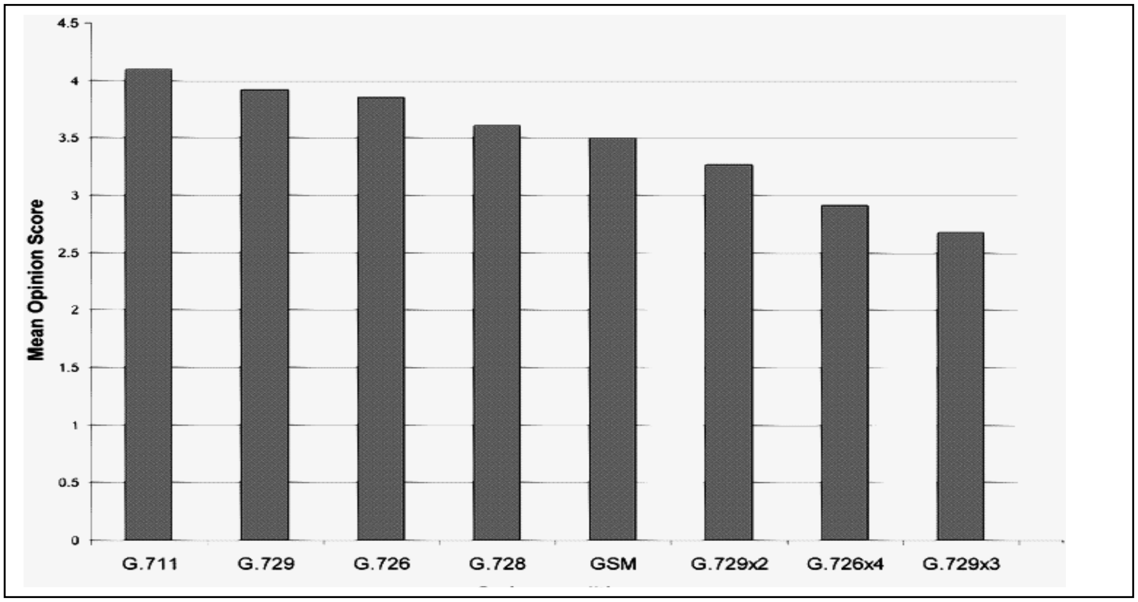


Figure 2.2 Effect of codec concatenation on an MOS [4].

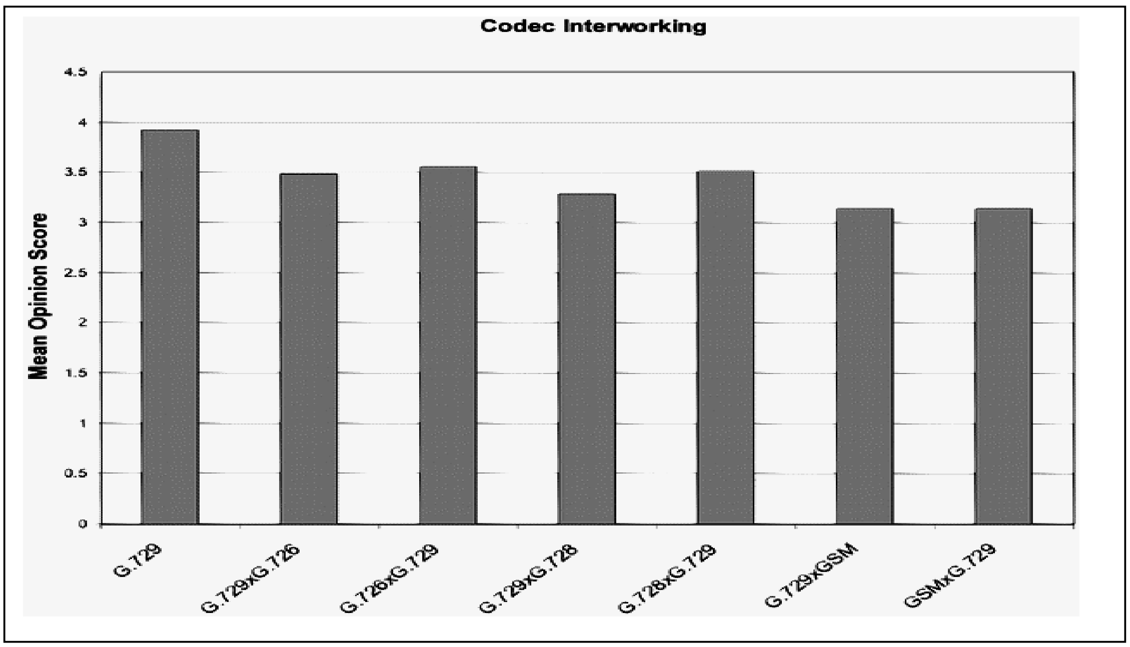


Figure 2.3 Effects of transcoding [4].

Provided quality depends on the measured quality characteristics of the network and devices used to conduct a call. The precise relationship between perceived and subjective quality and measured characteristics can only be found by carrying out experiments in which the technical qualities of the equipment are fixed and subjects are asked to rate the quality of the communication. Many studies of this sort have been carried out.

Figure 2.4. illustrates the underlying problem addressed by MOS experiments. Overall quality must be determined by technical, measurable, qualities such as loss (Loss of Volume, in dB), noise, delay, etc. However, there is no technical way of measuring quality. We can estimate quality by asking users to rate the quality of a series of calls. Because of random variation from one subject to another it is essential to repeat such experiments among times in order to obtain a sufficiently accurate measurement.

Measuring the quality of one particular voice call serves little purpose. However, measuring the quality of a voice call with known technical characteristics – for delay, dB loss, etc, is useful. In effect, in this way we can estimate the function depicted in Figure 2.4. This function, which gives the MOS of a call in terms of its technical parameters, can be reapplied many times over.

Knowledge of this function can be used to make better decisions regarding performance. There are a number of trade-offs between one performance parameter and another. For example packet loss can be reduced by increasing the jitter buffer, but this will increase delay. We should choose the size of jitter buffer which maximizes the MOS of calls. When we do this we are implicitly using our knowledge of the function shown in Figure 2.4.

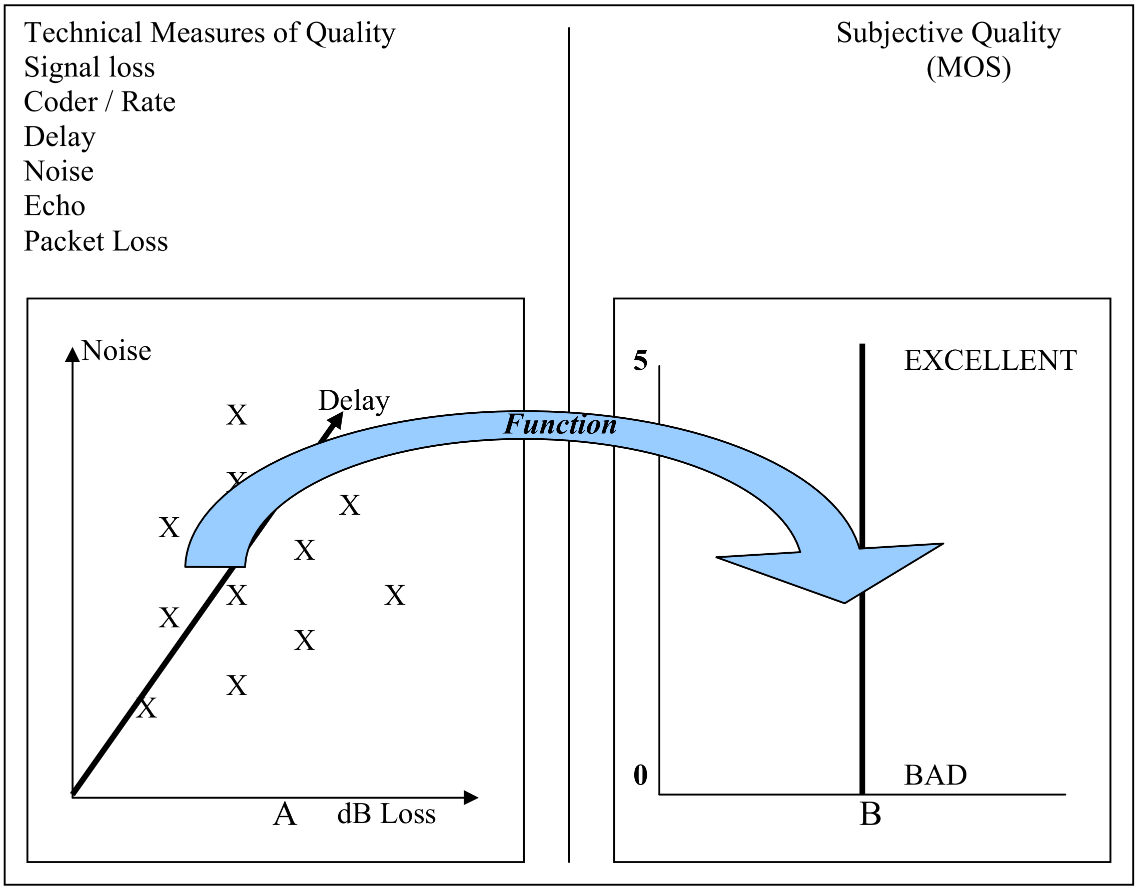


Figure 2.4 Relational Diagram.

2.7. Concatenation and Transcoding

The best packet network design codes the speech once near the speaker and decodes it once near the listener. Concatenation of low-bit-rate speech codecs, as well as the Transcoding of speech in the middle of the transmission path, degrades speech quality. Figure 2.2, from [4], shows the MOS of several codecs with and without concatenation. (A MOS of 5 is excellent, 4 is good, 3 is fair, 2 is poor, and 1 is very bad. Note that G.729 x 2 means that speech coded with G.729 was decoded and then recoded with G.729 before reaching the final decoder. G.729 x 3 means that three G.729 codecs were concatenated in the speech path between the speaker and listener [4].) Fig. 2.3 shows the MOS resulting from the interworking of different codecs, possibly in a Transcoding situation.

Since transcoding clearly reduces quality it would be preferable to avoid it whenever possible. Nowadays, end-to-end digital transmission, at least across the core network, is widely available. However, it will be desirable to negotiate the coder used at each end when setting us a connection and avoid any transcoding.

These may be cases, however, where the two terminals can not support the same coding algorithm. An important example is communication between a mobile phone (which uses an efficient coding) and a commercial phone (which communicates in analogue to a G.711 coder at the local exchange). Transcoding occurs regularly in this situation and will continue to be necessary for a period of time after VOIP is used in the same context.

2.8. DiffServ

DIFFerentiated SERVices is an architecture for guaranteed or at least managed quality of service (QoS) in IP networks under standardization by the IETF. Operating at layer 3 (IP) only, DiffServ uses the IP type of service (TOS) field as the DiffServ [14]. The DS code point uses 6 bits from the total 8 bits of the TOS field. DiffServ does not provide traffic engineering or hard quality of service similar to ATM. One possible approach is that service providers will use MPLS within the network and use DiffServ at the edges of the network for classification and assignment to the right connection [2].

Per-Hop Behavior (PHB) [15] is a term used in DiffServ (Differentiated Services). It defines the policy/priority applied to a packet when traversing a hop (such as a router) in a DiffServ network. DiffServ introduces the concept of Per Hop Behaviors (PHBs) that defines how traffic belonging to a particular behavior aggregate is treated at an individual network node. In IP packet headers, PHBs are not indicated as such; instead Differentiated Services Codepoint (DSCP) values are used. There are only 64 possible DSCP values, but there is no such limit on the number of PHBs. In a given network domain, there is a locally defined mapping between DSCP values and PHBs. Standardized PHBs recommend a DSCP mapping, but network operators may choose alternative mappings. In some cases it is necessary or desirable to identify a particular PHB in a protocol message, such as a message negotiating bandwidth management or path selection, especially when such messages pass between management domains. Examples where work is in progress include communication between bandwidth brokers, and MPLS support of diffserv. In certain cases, what needs to be identified is not an individual PHB, but a set of PHBs.

DiffServ [16], on the other hand, supports the needs of various applications by using a simple classification scheme [4]. This classification scheme helps the voice packet to be identified and thereby to obtain better performance when the data is transmitted over the internet. The DiffServ approach has several advantages over its predecessor, IntServ [17]:

• DiffServ is simpler than IntServ and does not require end-to-end signaling (no RVSP).

• DiffServ is efficient since classification and PHBs are based on per-class rather than per-flow information. Since the numbers of service classes are limited by the size of DS Codepoint field, and the amount of state information is proportional to the number of classes instead of number of flows, DiffServ approach is more scalable than IntServ.

• DiffServ requires minimal changes to the current network infrastructure.

A key challenge of the current telecommunication era is represented by the development of new architecture models for the Internet, aimed to satisfy the quality of service (QoS) requirements of innovative IP-based services (for example IP telephony). The relevance of this issue is related to the transformation of the Internet into a commercial infrastructure able to provide differentiated services to users with widely different service requirements. In this scenario, the differentiated services (DiffServ) approach is the most promising for implementing scalable service differentiation in Internet. The scalability is achieved by considering the aggregated traffic flows and conditioning the ingoing traffic at the edge of the network. Hence, the DiffServ approach permits to provide QoS employing a small, well defined, set of building blocks enabling a large variety of services. These building blocks include a small set of per-hop forwarding behaviours, packet classification and traffic conditioning functions such as marking, shaping and policing [18]. In particular, the marking function measures the temporal properties of the packet stream, in a particular service class, against a traffic profile specified in a traffic conditioning specification (TCS) for this class, which represents a traffic contract between the customer (or the administration requiring a service for a traffic aggregate) and the Service Provider. In this manner, the meter determines if a packet is in-profile or out-profile according to whether it is conforming or not to the TCS.

Most researchers agree in considering the token bucket algorithm as an efficient mechanism for marking, shaping & policing and in describing the traffic profile by means of the parameters obtained by the linear bounded arrival processes (LBAP) traffic characterization [18].

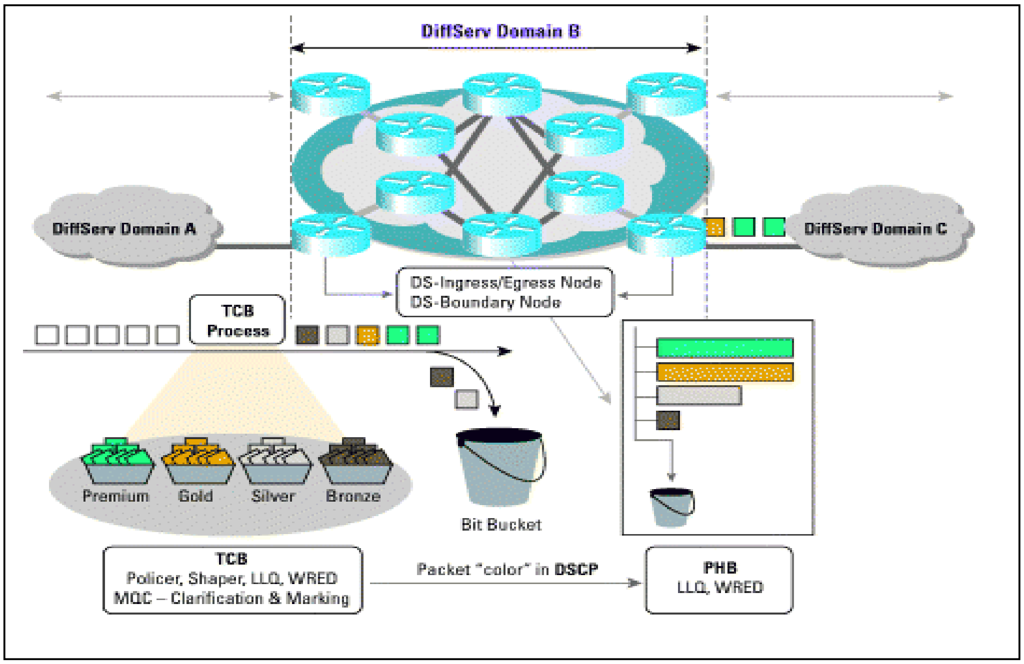


Figure 2.5 Diffserv Architecture [19].

2.9. SIP (Session Initiation Protocol)

Primarily used for voice over IP (VOIP) calls, SIP can also be used for video or any media type; for example, SIP has been used to set up multi-player Quake games [20]. With SIMPLE extensions for Instant Messenger and Network presence, SIP is also used for instant messaging [21].

SIP is a text-based protocol that is based on HTTP and MIME, which makes it suitable and very flexible for integrated voice-data applications. SIP is designed for real-time transmission, uses fewer resources and is considerably less complex than H.323, the signaling protocol defined by the ITU for establishing voice and video conference calls across the internet [22]. SIP's addressing scheme uses URLs and is human readable; An example of a SIP based address is:

sip:jd.sci@usq.edu.au

H.323, by contrast, most often uses numbers similar to telephone numbers [20]. SIP relies on the session description protocol (SDP) for session description and the real-time transport protocol (RTP) for actual transport. The RTP packets are then packaged in UDP datagram for delivery. Windows XP was the first version of Windows to natively support SIP for PC-based phone applications, and numerous vendors make SIP desktop phones [23] [20].

A SIP proxy is a server in a SIP-based IP telephony environment. It is required in large companies with numerous telephone numbers or when the Internet is the long distance transport. The SIP proxy takes over call control from the terminals and serves as a central repository for address translation (name to IP address).

The SIP proxy server provides similar functionality to a gatekeeper in an H.323 environment or a soft switch in an MGCP/MEGACO environment. SIP supports both stateless and stateful connections. A stateless connection establishes the call and then politely gets out of the way. A call stateful proxy stores all signaling events for the duration of the call. Some SIP proxy servers deposit cookies in the IP phone/terminal as a method of providing state information.

2.10. H.323

The H.323 standard [20] provides a foundation for audio, video, and data communications across IP-based networks, including the Internet. H.323 is an umbrella recommendation from the International Telecommunications Union (ITU) that sets standards for multimedia communications over Internet that do not provide a guaranteed Quality of Service (QoS). These networks dominate today's corporate desktops and include packet-switched TCP/IP and IPX over Ethernet, Fast Ethernet and Token Ring network technologies. Therefore, the H.323 standards are important building blocks for a broad new range of collaborative, LAN-based applications for multimedia communications. It includes parts of H.225.0 - RAS, Q.931, H.245 RTP/RTCP and audio/video codecs, such as the audio codecs (G.711, G.723.1, G.728, etc.) and video codecs (H.261, H.263) that compress and decompress media streams. Table 2.2 [24], tells us about the H.323 protocol suite group and its different components.

DVB	Digital Video Broadcasting
H.225	Covers narrow-band visual telephone services
H.225 Annex G	
H.225E	
H.235	Security and authentication
H.323SET	
H.245	Negotiates channel usage and capabilities
H.450.1	Series defines Supplementary Services for H.323
H.450.2	Call Transfer supplementary service for H.323
H.450.3	Call diversion supplementary service for H.323
H.450.4	Call Hold supplementary service
H.450.5	Call Park supplementary service
H.450.6	Call Waiting supplementary service
H.450.7	Message Waiting Indication supplementary service
H.450.8	Calling Party Name Presentation supplementary service
H.450.9	Completion of Calls to Busy Subscribers supplementary service
H.450.10	Call Offer supplementary service
H.450.11	Call Intrusion supplementary service
H.450.12	ANF-CMN supplementary service
H.261	Video stream for transport using the real-time transport
H.263	Bit stream in the Real-time Transport Protocol
Q.931	manages call setup and termination
RAS	Manages registration, admission, status
RTCP	RTP Control protocol
RTP	Real-Time Transport
T.38	IP-based fax service maps
T.125	Multipoint Communication Service Protocol (MCS).

Table 2.2. H.323 Protocols Suite

2.11. Echo cancellation

Echo is created in variety of ways. One of the important source of echo when the telephone company's telephone exchange (central office in US terminology) connects two-wire lines from the customer to four-wire lines for backbone trunks. The echo is exacerbated over longer distances and by certain kinds of network equipment. A delay of 30ms or more is generally noticeable, and 50ms is annoying. To eliminate it, the carriers put echo cancellers on their switch ports and in their long-distance trunks.

Echo cancellation is built into high-speed modems. Since telephone system echo cancellation is optimized for voice, the modem emits a 2100Hz signal to cancel it, allowing the modem's own cancellers to be used, which is more effective.

Echo cancellation may be built into a speakerphone to cancel the echo caused by the microphone picking up sound from the speaker. Since echo cancellation is not built into sound cards, making a PC-to-PC voice call using regular speakers and a microphone causes echo. Headsets help eliminate this problem.

Echo cancellation uses extremely sophisticated DSP circuits that are able to isolate the echo and transmit a reverse frequency to cancel it [25].

Video conferencing gives rise to echo when the audio signal from speakers is packed up by microphones and retransmitted. Because of the multiplicity of speakers and microphones this type of echo is difficult to control, and requires dedicated commercial hardware.

2.12. Packet Loss

Packet loss is caused by the discarding of data packets in a network when a device (switch, router, etc.) is overloaded and cannot accept any incoming data at a given moment. High-level transport protocols such as TCP/IP ensure that all the data sent in a transmission is received properly at the other end [25], however transmission will introduce significant delay, usually more than acceptable. For this reason VOIP will usually be sent via UDP packets.

Above some threshold rate, VOIP network packet loss introduces audio distortions that cause voice quality to decrease as the rate of packet loss increases. That said, on any particular connection this general effect can be modified by:

- The distribution of the lost packets
- The packet loss concealment (PLC) algorithm in use

Early work on VOIP generally repeated the assertion that packet loss did not become a significant problem until it reached a 5 percent rate. This assertion provided poor guidance.

According to [6], packet loss is a normal phenomenon in packet networks. Loss can be caused by many different reasons: overloaded links, excessive collisions on a LAN, physical media errors and others.

Audio Codec's also may take into account the possibility of packet loss; especially since RTP data is transferred over the unreliable UDP layer. The typical CODEC performs one of several functions that make an occasional packet loss unnoticeable to the user. For example, a CODEC may choose to use the packet received just before the lost packet instead of the lost one, or perform more sophisticated interpolation to eliminate any clicks or interruptions in the audio stream.

As we have mentioned earlier packet loss starts to be a real problem when the percentage of the lost packets exceeds a certain threshold (roughly 5% of the packets), or when packet losses are grouped together in large packet bursts. In those situations, even the best Codecs will be unable to hide the packet loss from the user, resulting in degraded voice quality. Thus, it is important to know both the percentage of lost packets, as well as whether these losses are grouped into packet bursts [26]. For further information see [26].

2.13. Delay

In modern digital telecommunication networks, delay is a key performance parameter which should be minimized. Although the delay of IP networks may exceed the typical delay of the PSTN, the degradation caused by additional delay might be compensated for by benefits provided by new network and service capabilities. These tradeoffs need to be quantified.

Delay can have two effects on voice performance. Firstly, it increases the subjective effect of any echo impairment. Secondly, as indicated in Recommendation G.114, even when echo is controlled, one-way delays above 150 ms can interfere with the dynamics of voice conversation, depending upon the type of conversation and degree of interaction. Recommendations G.114 and G.131 give additional information regarding effects of delay and echo.

In addition, delay can impair the performance of particular voice band data applications. Total delay of hybrid Internet/PSTN networks should be limited, even with the use of echo control. Recommendation G.114 should be consulted for additional information [26] [22].

2.14. Delay Variation (Jitter)

The variation in packet delay is sometimes called "jitter" [27]. This term, however, causes confusion because it is used in different ways by different groups of people. "Jitter" commonly has two meanings: The first meaning is the variation of a signal with respect to some clock signal, where the arrival time of the signal is expected to coincide with the arrival of the clock signal. This meaning is used with reference to synchronous signals and might be used to measure the quality of circuit emulation. The second meaning has to do with the variation of a metric (e.g., delay) with respect to some reference metric (e.g., average delay or minimum delay). This meaning is frequently used by computer scientists and frequently (but not always) refers to variation in delay.

Jitter buffer is one of the key issues which help us to control the balance between delay and loss. If we increase the jitter buffer it will reduce loss but will increase the delay. Again if we decrease the jitter buffer it will increase loss but will decrease delay. However, 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.

The approach selected will depend on the type of network the packets are traversing. The first approach is to measure the variation of packet level in the jitter buffer over a period of time and to incrementally adapt the buffer size to match the calculated jitter. This approach works best with networks that provide a consistent jitter performance over time (e.g., ATM networks). 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 (e.g., 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 QoS. Jitter has a diverse impact on perceived quality due to the necessity to increase the size of the playback buffer – leading to greater delay.

Chapter 3

Advantages & Disadvantages of VOIP

The focal point of Chapter 3 is the discussion of the Advantages and disadvantage of the VOIP. Here we have discussed the advantages as well as the limitation of the IP telephony.

3.1. Credentials of Voice over Internet Protocol

VOIP is under pressure as well as the PSTN [4], [3], [28]. In today's internet, handing voice traffic is not a big deal for the core network. It has enough capability to handle the voice data flow [4]. However, performance degradation in the access network requires further work.

The Voice over Internet Protocol, also called IP telephony, offers a new type of service that uses the Internet Protocol, LANs and WANs to deliver voice information. In contrast to traditional telephone services, which operates through a circuit-switched network, VOIP uses a packet switched network. This distinction results in differences in implementation, quality of service (QoS), and operating costs. During its long history, traditional telephone service has expanded across most of the world. In contrast, VOIP offers a relatively new type of telephone service that has been available for fewer than 10 years. At this point, the development of VOIP telephone services cannot be totally separated from traditional telephone networks. Although VOIP services have partially supplanted traditional toll telephone services, when users make a VOIP telephone call, they must, in most cases, still go through a local telephone network. Since the service was introduced to the public, VOIP toll telephone traffic has increased with astonishing speed. By the end of 2002, VOIP toll telephone traffic had surpassed traditional toll telephone traffic [1]. Four factors have contributed to this phenomenon [13].

3.1.1. Price Advantage.

Given current price regulations on toll telephone services in the world, VOIP toll charges are only about one-third to one half the costs of traditional telephone charges [29]. Although VOIP telephone providers cannot guarantee that voice quality will match that of traditional telephone service, the price advantage alone remains attractive enough to draw customers to VOIP phone services.

3.1.2. New and profitable area for ISPs.

VOIP telephone services provide an opportunity for Internet service providers to earn higher profits, especially those ISPs new to the telecommunications market. This new opportunity has fostered a new class of additional parties in the world: Internet telephone service providers.

3.1.3. Benefits for traditional telephone service providers.

Although VOIP has made inroads into the toll telephone service market, total demand for traditional toll telephone services has actually increased. Both domestic and international calls more than doubled within last few years [29]. Because most VOIP toll telephone traffic must use the local telephone networks, the increase in VOIP services has brought an increase in local telephone network usage as well. Thus, the growth of VOIP services has actually increased profits for traditional telephone service providers.

3.1.4. Potential value in the transition to next-generation networks.

Development of VOIP telephone services can facilitate the transition from current telecommunications networks to IP based next generation networks. This technology, and the benefits it will provide, make its implementation crucial to telecommunications network and service providers. VOIP achieved rapid growth in both public and private networks.

Recently, some of world's large-scale intranets have begun integrating phone and data services into the same network through VOIP, reducing telecommunications costs by eliminating the use of a separate phone network.

Traditional telephone service	IP telephone service
Circuit-switching technology	Packet-switching technology
Uses synchronous time-division multiplexing in transmission resulting in lower channel utilization	Uses asynchronous time-division multiplexing in transmission, resulting in higher channel utilization
When network congestion occurs, calls will be blocked, but once call connection is established, the voice signal will not be lost	When network congestion occurs, calls can be blocked or IP packets can be lost, resulting in reduced voice quality
Uses the G.711 Pulse Code Modulation voice-encoding scheme without compression and achieves a transmission speed of 64 Kbps	Usually uses voice-compression encoding, with the bit rate of encoded voice data ranging from as high as 64 Kbps to as low as 5.3 Kbps
Short end-to-end transfer delay except in satellite communications and limited delay variation	Relatively long end-to-end transfer delay and significant delay variation
Guaranteed good voice quality	Voice quality affected significantly by the IP network's quality of service; absent specific measures, voice quality cannot be guaranteed
Given the separate network built to provide telephone services, reducing operational costs is difficult	Sharing network resources by combining with data and other multimedia services on the same IP network helps reduce operational costs

Table 3.1. Comparison of traditional and IP telephone services [7].

3.2. VOIP Limitations

In the past few years, tremendous use of IP technology has increased the trend of data-voice convergence, thus making voice over IP (VOIP) a preferred strategy that will replace the standard public switched telephone network (PSTN). However, the lack of end-to-end quality of service (QoS) guarantee has been the major problem in the current VOIP development. Apart from the bandwidth allocation guarantee, real-time applications like VOIP should not suffer much packet loss, excessive delay, and excessive jitter. Broadly speaking, the end-to-end delay of a link should be less than 120 ms, jitter should be less than 100 ms, and packet loss should be less than 1 percent [3]. Ideally, there should be no packet loss for VOIP; a packet loss is one of the key issues in the QoS of the IP telephony.

The main problem with VOIP at present is the lack of guaranteed QoS. There are different types of delay involved in the transmission of the IP packets on VOIP. From [4] we can focus on the different types of delay that are involved in the VOIP QoS. End-to-end delay includes delay due to codec processing as well as propagation delay. ITU-T Recommendation G.114 [26] recommends the following one-way transmission time limits for connections with adequately controlled echo (complying with G.131 [22]):

• 0 to 150 ms: acceptable for most user applications;

• 150 to 400 ms: acceptable for international connections;

• > 400 ms: unacceptable for general network planning purposes; however, it is recognized that in some exceptional cases this limit will be exceeded.

In the internet the Core Network is strong enough to handle the large amount of data and voice packets but when we consider the Access Network there remains a problem that occurs when big data chunk start moving its place. The codec plays also an important role in the quality of the service of VOIP. Coder performance generally follows bit rate (the higher the better). But in the VOIP environment more than one coder may be used on any end-to-end connection since the converging networks (digital cellular, broadband access, VOIP, PSTN, etc.) may use different coders. Therefore, in considering coder performance it is important to consider the effects of processing speech through them multiple times. Such processing is normally referred to as coder tandeming although when the multiple encode-decode episodes involve different coders the term transcoding is often used [30].

3.3. Summary

The future success of VOIP will be strongly influenced by customer opinions of call quality and how this quality compares with that of the public switched telephone network (PSTN). Here we can get a picture, which will consider how different components of a VOIP system can have an impact on a customer's perception of quality. The challenge is to design an efficient solution which delivers a 'just-good-enough' quality level for a particular application.

The unreliable nature of the Internet and initial product offerings, where cost savings were often gained at the expense of quality, has led to an image that VOIP is of a worse quality than circuit-switched networks. The 'poor quality' image has, to some extent, accompanied VOIP as it has moved from the domain of the Internet enthusiast to being used in today's carrier-scale networks. Carriers are eager to squeeze maximum efficiency from their networks but understand that there is a minimum quality level required to achieve customer acceptance. Enterprise customers, now adopting VOIP, are also sensitive to noise, distortions and general impairments in their voice communication systems. One of the most commonly discussed concerns is that VOIP will never deliver 'appropriate quality'. However, to achieve a high-quality VOIP solution is not impossible, rather an engineering challenge. Significant steps have been taken over the last few years to achieve higher quality systems, and this dissertation provides a view on some of the issues and choices that VOIP system designers face. It is important to realise that perceived quality is technology independent. The customer generally does not know, or want to know, about the underlying technology when using the telephone. Their demands are the same, regardless of whether VOIP or other voice network technology is used. Therefore most of the standardisation work for delivering a desired quality level is applicable to VOIP as well as the PSTN.

Although this appears to be a superfluous statement, all too often VOIP solutions fail to meet key requirements such as use of the recommended delay thresholds or inclusion of the correct telephony filtering or level control. This ultimately results in the customer receiving poor service.

For a VOIP system the three most significant influences on a user's perception of quality are:

· the performance of the VOIP device,

· the performance of the underlying network,

· the end-to-end delay.

A particular quality level can be achieved by trading one against another. However, for high-quality systems the scope for trade-offs is greatly reduced [31].

Chapter 4

Customer & Business Perception of VOIP Technology

Basic operation of any technology will give a better idea of the investigated result. The entire chapter is mostly focused on the step by step basic operation of the VOIP in contrast of the PSTN.

4.1. Basics:

The basic steps for transmitting voice over an IP network are as follows [32]:

- **i.** Audio from microphone or line input is A/D (Audio/Digital) converted at audio input device.
- **ii.** Samples are copied into a memory buffer in blocks of frame length.
- **iii.** The IP Telephony application estimates the energy levels of the block of samples if silence suppression is active.
- **iv.** Silence detector decides whether the block is to be treated as silence or part of a talk spurt.
- **v.** If the block is a talk spurt it is coded with the selected algorithm.
- **vi.** Header information is added to the block.
- **vii.** The block with headers is written into a socket interface.
- **viii.** The packet is transferred over a physical network and received by the peer.
- **ix.** The header information is removed, block of audio is decoded using the same algorithm with which it was encoded, and samples are written into a buffer.
- **x.** The block samples are copied from the buffer to the audio output device.
- **xi.** The audio output device D/A (Digital/Audio) converts the sample and outputs them. [32]

4.2. IP Telephony versus PSTN

The primary technical difference between the Internet and the PSTN is their switching architectures. The Internet uses dynamic routing (based on non-geographic addressing) versus the PSTN, which uses static switching (based on geographic telephone numbering). Furthermore, the Internet's 'intelligence' is very much decentralized, or distributed, as against the PSTN which bundles transport and applications resulting in intelligence residing at central points in the network.

The PSTN is a circuit switched network. It dedicates a fixed amount of bandwidth for each conversion and thus quality is guaranteed. When a caller places a typical call, they pick up the phone and hear the dial tone, dial a number, the telephone exchange establishes a connection, and the intended recipient of the call, and the caller, can talk to each other. When callers place an IP Telephony call, they pick up the phone and hear the dial tone from the PBX, dial the number, which is then forwarded to the nearest IP Telephony gateway located between the PBX and a TCP/IP network. The IP Telephony gateway modulates voice into IP packets and sends them on their way over the TCP/IP network as if they were typical data packets. Upon receiving the IP encoded voice packets, the remote IP Telephony gateway reassembles them into analogue signals to the recipient through the PBX. So, how does IP Telephony fit in, and meet, with business requirements? There are three basic business scenarios for IP Telephony [32].

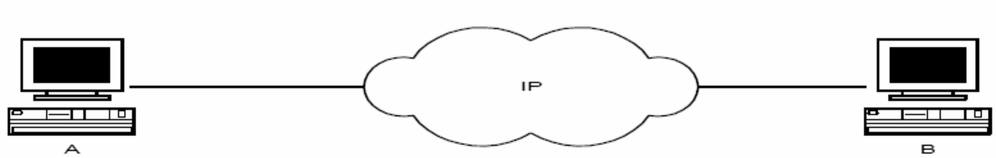


Figure 4.1 (IP terminal (computer) to IP terminal)[32]

In Figure 4.1 both the subscriber A and B are using a computer attached to an IP network as terminals. Voice is compressed and decompressed by the PC software. At present this case is not common but it is excepted to become more common & eventually should be the dominant case.

In Figure 4.2 one of the subscribers is using a computer for IP-voice and the other uses a phone on a PSTN /ISDN network. A gateway on the edge of the IP network translates IP-voice to voice and takes care of the signaling between the two networks.

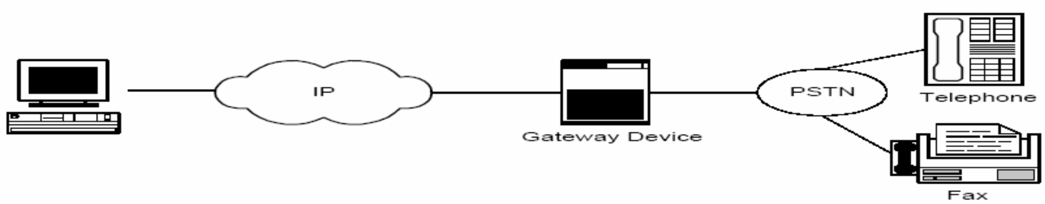


Figure 4.2 (IP terminal to phone or phone to IP terminal) [32]

In Figure 4.3 both subscribers are using conventional phones, and the IP network is used for the long distance connection. Gateways on both ends take care of traffic and signaling translation between the networks.

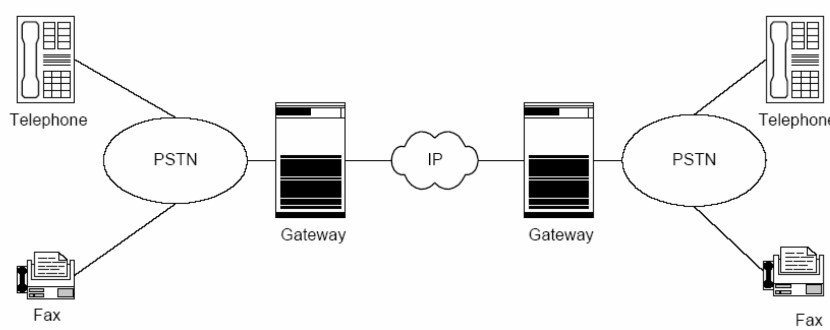


Figure 4.3 (Phone to Phone) [32]

4.3. Summary of Perception:

4.3.1. How traditional long distance works?

We pick up our phone and dial a long distance phone number; the call goes through our local telephone company to our long distance provider who charges us a connection fee and per minute charge, billed monthly in our long distance phone bill.

4.3.2. How long distance works with VOIP?

We pick up our phone and dial a long distance phone number, the call goes through our local telephone company to a VOIP provider, and this is a local call. The call then goes over the Internet to the receiver's local calling area, where a local call is placed (by the VOIP provider) to complete the connection. We have just circumvented our long distance company, and eliminated our long distance phone bill.

4.4. Overview

Voice over Ethernet was tried at Xerox Palo Alto Research Centre successfully [33] before the current Internet established. Internet Phone software was introduced to the market in the 90's [29]. Designed to run on a 486 / 33 MHz (or higher) personal computer, equipped with a sound card, speakers, microphone, and modem, the software compressed the voice signal and translated it into IP packets for transmission over the Internet. This PC-to-PC Internet telephony worked, however, only if both parties were using Internet Phone software [29]. In the relatively short period of time since then, Internet Telephony has advanced rapidly. Many software developers now offer PC telephony software but, more importantly, gateway servers are emerging to act as an interface between the Internet and the PSTN. Equipped with voice-processing cards, these gateway servers enable users to communicate via standard telephones over great distances without going over the 'Long Distance' telephone network. A call goes over the local PSTN to the nearest gateway server, which digitizes the analogue voice signal, compresses it into IP packets, and moves it onto the Internet for transport to a gateway server at the receiving end. This server converts the digital IP signal back to analogue and completes the call locally. With its support for computer-to-telephone calls, telephone-to-computer calls and telephone-to-telephone calls, VOIP represents a significant step towards the integration of voice and data networks.

Originally regarded as a novelty, Internet Telephony is attracting more and more users because it offers tremendous cost savings relative to the PSTN. Users can bypass long distance carriers and their per-minute usage rates and run their voice traffic over the Internet for a flat monthly Internet-access fee. VOIP provides a competitive threat to the providers of traditional telephone services that, at the very least, will stimulate improvements in cost and function throughout the industry.

Although the use of voice over packet networks is relatively limited at present, there is considerable user interest. Projection made in 2002 estimated that the compound annual growth rate for IP-enabled telephone equipment would be 132% over the period from 1997 to 2002 (from $47.3M in 1997 to $3.16B by 2002) [24]. This projection appears now to be over optimistic.

VOIP can be applied to almost any voice communications requirement, ranging from a simple inter-office intercom to complex multi-point teleconferencing/shared screen environments [24]. Widespread deployment of a new technology seldom occurs without a clear and sustainable justification, and this is also the case with VOIP. Demonstrable benefits to end-users are also needed if VOIP products (and services) are to be a long term success. Lots of long term effective estimation done for the growth on IP Telephony market and also on it prospects. But it is clear that a market has already been established and there exists a window of opportunity for developers to bring their products to market, and for consumers to realize significant savings.

4.4.1. What is IP Telephony good for?

The most significant benefit of IP Telephony, and driver of its evolution, is money saving. The second is easy implementation of innovative services. These reasons alone would be attractive to any business contemplating installation of the technology into their business. In future Internet Service Providers (ISP's) may use a single infrastructure for providing both Internet access and IP Telephony. Only data orientated switches could be deployed for switching data as well as packetized voice. Multiplexing data and voice could also result in better bandwidth utilization than in today's voice or nothing links. Not only the ISP's but also their customers will benefit from lower costs.

4.5. The Vision

According to the Duerinck [34], the futuristic vision of an office device that is both a voice and a data terminal capable of flexible, high-speed communications still has a long way to go. However, there have been significant advances in the technologies the experts believe will get us to this communications ideal. A few years ago, Computer Telephony Integration (CTI) was very much seen as the way forward. This was when phone systems were phone systems and computers were computers. What CTI solutions promised to do was pass information between the two to enrich the user experience and automate certain functions, such as dialling numbers and calling up database information on incoming callers as well as voice mails. In the early stages of the CTI revolution, a number of physical end user devices such as the ICL 'One Per Desk', a combined desktop voice and data terminal, were developed in readiness for an entirely CTI world. In recent years, the focus has switched to software products, with unified messaging software featuring 'all in one mailboxes' (where voice messages can be accessed from the same mailbox as email and fax messages); and fax server software (where software, running in servers connected to modems and computer-based fax cards, were used to replace traditional fax machines) being taken up in increasing numbers [34].

Despite these developments, technology suppliers are still a long way from being able to deliver a combination of media (voice, fax, email, Web information) to a single device down a single phone/data line or wireless connection, using CTI techniques supported throughout the industry.

Even the ‘multi-application communications servers’, that have emerged recently, and which combine applications, such as telephony switching, voice mail, fax, IVR, call transfer, call conferencing, email and Web access on a single platform, still typically rely on separate connections to the voice and data worlds; as well as two separate voice and data wires to the user’s desktop. What is set to change all this and bring about a true communications revolution in the workplace is Internet Protocol (IP). IP is already being used widely within the enterprise for data traffic, with enormous strides being made to improve the quality of voice over the Internet as well as huge investments being made in IP public network infrastructure. It is for this reason, that technology suppliers are now in almost universal agreement that IP, and not CTI, is the catalyst that will bring us closer to the ideal of a single connection to the outside world for voice and data traffic; and a single connection to the desktop or wireless device [35].

With some elements of the picture in place today, and some still very much in the laboratories (for the time being), companies need to take a pragmatic approach to IP Telephony enabling their businesses. In the short to medium term, the goal of technologists must be to move companies towards an IP Telephony future, and that means providing a transition path through supporting IP Telephony alongside traditional circuit switched telephony in a ‘hybrid’ technology environment. Companies must look to move forward while still protecting their existing infrastructure and systems investments. That is the concept and vision of the technology. However, this does bring issues and implications, involving Bandwidth, Quality of Service (QoS), Infrastructure requirements and Network Convergence, when considering implementation. The next chapter will address these issues.

Chapter 5

Factors that Impact the QoS of VOIP

This Chapter considers how different factors of a VOIP system can have an impact on a customer's perception of quality. The identification of the specific problems will help us on our investigation. The challenge is to design an efficient solution which delivers a 'just-good-enough' quality level for a particular application.

5.1. Speech quality

One reason for choosing VOIP is its potential to deliver voice and data services more efficiently than the PSTN. Good speech quality is critical in order to deliver a commercially viable VOIP system. However, there is a technical challenge in delivering high-quality speech while achieving high network efficiency. It is possible to create a high quality system, but low-bandwidth allocations typically result in some reduction in speech quality. The challenge is to deliver 'just-good-enough' speech quality at a specific bit rate. There are various ways to trade bandwidth and speech quality to meet a target quality level, but there is no definitive answer on how this is to be achieved. Based on the ITU-T Recommendations this section describes those aspects of a VOIP system design which have greatest bearing on the user perceived speech quality:

- speech coding,
- Packetization efficiency,
- silence suppression,
- error-concealment methods,
- jitter buffer implementation,
- Codec tandem performance.

5.1.1. Speech coding

Most domestic PSTN networks operate with speech sampled at 8 kHz and an 8-bit nonlinear quantization scheme according to ITU-T G.711 [36]. This encodes at 64 kbit/s and introduces little audible distortion for most types of signal. In a number of applications, however, a much lower bit rate is desirable either because capacity is limited (e.g. mobile) or to maximize the amount of traffic that can be carried over a trunk connection. Current ITU-T recommendations include codecs that compress to as low as 5.3 kbit/s, although quality at this bit rate is well below that of G.711. To maximize interoperability it is usual for VOIP gateways and clients to offer one or more standard codecs. Table 5.1 lists some common, standardized, voice codecs with their associated bit rates.

Codec	Bit Rate (kbit/s)	Coding Technique
G. 711	64	Pulse code modulation (PCM)
G. 726	40 to 16	Adaptive differential PCM (ADPCM)
G. 728	16	Low-delay codec excited linear prediction
G. 729	8	Algebraic code-excited linear prediction (ACELP)
G. 723.1	6.3/5.3	Multi-pulse-excited long term predictor (MP-MLQ)/ACELP)

Table 5.1. Voice codecs with coding Techniques [36].

The choice of codec is to some extent dictated by the bandwidth on offer which determines the maximum bit rate of the codec, and in turn the maximum speech quality that the system will achieve under ideal conditions. It is important to note the fact that because a codec is standardized does not in itself make it suitable for mass telecommunications applications. In general, the lower the bit rate, the lower the quality perceived by the listener. However, more modern codec designs are driving up the quality for a given bit rate. This shows the quality of speech passed through some of these standard codecs, measured using Perceptual Analysis/Measurement System (PAMS) [37]. The highest quality basic codec is still G.711 at 64 kbit/s. The low-complexity G.726 codec only offers good performance at 32 kbit/s or above, and is comparable with the more recent, though computationally intensive, GSMEFR at 12.2 kbit/s and G.728 (CELP) at16 kbit/s.

Given a demand for high quality, it is clear that choosing the lowest bit rate codec will not suit a large proportion of today's VOIP market. Therefore most systems tend to offer G.711 and at least one low bit rate codec. This allows the operator some flexibility to trade between quality and bandwidth — potentially on a per-call basis. Saving bandwidth is no longer the key reason why IP telephony is cheaper. IP bandwidth is cheap & getting cheaper. In some situation Bandwidth is still important.

The paper [28] of Carmen Peláez-Moreno, Ascensión Gallardo-Antolín, and Fernando Díaz-de-María, suggests that Voice codecs included in the H.323 protocol suite such as G.723.1 and G.729 are now the most commonly used ones even though for many years the PSTN operated strictly with the standard G. 711 [28]. Alternative codecs are emerging; the authors of [28] have investigated the G.723.1 standard codec because, together with the G.729, it is the most widely used in the VOIP environment [38]. Furthermore, G.723.1 seems to be more sensitive to packet loss, mainly due to its relatively slow frame rate (33.3 frames per second). A low frame rate implies that a considerable portion of voice is missing when a packet is lost. With the purpose of providing a better understanding of the coder technology, front-end, there follows an outline of the most relevant features of this codec. For a detailed description, we refer the reader to the standard recommendation [39].

The G.723.1 standard is an analysis-by-synthesis* linear predictive codec and provides a dual coding rate at 5.3 and 6.3 Kb/s. It is possible to switch between rates at the frame level and also; an option for variable rate operation is available using voice activity detection (VAD), which compresses the silent portions. The voice quality offered by G.723.1 can be rated as 3.8 on the M.O.S. scale in 5.3 kb/s mode and 3.9 in 6.3 kb/s modes. Therefore, even though toll quality is claimed, it is obvious that other algorithms provide a slightly better quality: G.729 and G.726 give 4.0 and 4.3, respectively. G.723.1 uses a frame length of 240 samples (30 ms) and an additional look ahead of 60 samples (7.5 ms), resulting in a total algorithmic delay of 37.5 ms. The frame is divided into four subframes of samples. A window of 180 samples is centred on every subframe and a tenth-order linear prediction (LP) analysis is performed. The prediction coefficients obtained in this way are used to implement a short-term filter.

**Analysis-by-synthesis: Speech coding technique which aims to minimize the mean-squared error between the input analogue speech and its synthesized version [40].*

5.1.2. Packetization efficiency

A further consideration when choosing a codec is how much speech can be placed in an IP packet. This is partly dictated by a codec's frame size. For example, GSM uses a fixed 20-ms frame and therefore packets must be a multiple of 20 ms, whereas G.711 can be any length. The amount of speech placed in a packet has a direct impact on the underlying network efficiency. VOIP is inefficient for small voice packets while large voice packets lead to long delays — which is inappropriate for real-time communications. Typically packets contain between 10 to 30 ms of speech which provides a practical trade-off between network efficiency and increased delay. Some increase in bandwidth overhead is justified by the economic benefits of running a single combined voice and data network [41]. Packetization efficiency reduces as the rate of the codecs reduces. As an example if a VOIP packet size is 128 Kbytes G.711 will process that data faster then G.729 as the rate of G.711 codec is higher then the G.729. A slow coder will take more time then a faster coder to generate a packet for delivery and that will create more packetization inefficiency.

5.1.3. Silence suppression

Silence suppression, or Discontinuous Transmission (DTX), is the process by which periods when the user is not speaking are not coded or transmitted. This allows the overall system bandwidth requirements to be reduced and can produce an average bandwidth saving of some 40%. DTX is implemented by the use of a Voice activity detection (VAD). One problem with silence suppression is its susceptibility to front and back-end clipping. This describes the situations where the VAD triggers too late or too early, cutting off speech at the beginning or end of a sentence. It has therefore proved difficult to design a VAD that works correctly in all circumstances, despite extensive research and development effort. In addition to the problem of deciding what is and is not speech there is the issue of how to generate a signal to fill in the silent periods. This generated signal is called comfort noise. Without comfort noise the telephone system feels dead to the end user, but a mismatch in noise between when the person is and is not talking is distracting and leads to an impression of poor quality. Also the background noise may be part of the message – e.g. the sound of a kitchen or a restaurant.

When choosing to develop a VAD an understanding of the application is critical. For example, a VAD that labels more of the signal as speech, suppressing transmission only when the talker is definitely silent, will deliver higher quality than a more aggressive VAD. This is of course balanced by a slight increase in bandwidth, but there can still be a significant saving over a system without DTX [1].

5.1.4. Error concealment

VOIP systems can suffer from a degree of packet loss during normal operation and there can also be packets dropped by the jitter buffer. The ability of a VOIP device to conceal packet loss makes a significant difference to its performance. Certain standardized codecs (e.g. G.729) include their own error-concealment methods. However, the use of proprietary error concealment can provide an improvement in quality over these standard methods and is applicable to other codecs. An alternative to error concealment is the insertion of silence in place of lost packets — this gives a clicking effect which users find annoying. For example, at moderate levels of packet loss a G.711 implementation with no error concealment can lead to a lower quality than a low bit rate codec with error concealment. At the expense of an increase in delay and bit rate, it is also possible to add redundancy to allow some errors to be corrected or lost packets reconstructed. Techniques available include forward error correction or duplication of frames across multiple packets. The distribution of quality scores measurements depends on the G.729 VOIP system, at different packet loss rates. The range of scores occurs because a lost packet may coincide with a critical or non-critical part of the voice stream [1].

5.1.5. Jitter buffer implementation

A jitter buffer is needed to smooth over the distribution of packet delay that is characteristic of IP. It is important to note that a gateway's or host's jitter buffer has an impact on the system's delay and speech quality. A careful balance is needed between adding too much delay, which impairs conversation, and dropping packets, which reduces the speech quality [2].

Jitter buffers are categorized into two types — static and dynamic. Static buffers use a fixed-length buffer and are the easiest to implement and manage. Any packet that arrives late is simply discarded. The jitter buffer size is normally configurable, which adds some flexibility for tuning a given VOIP system, but generally a static buffer requires a well managed underlying network to keep jitter within the size of the buffer. The more sophisticated dynamic jitter buffer has scope to adjust the play-out point in the buffer based on a history of the arriving packet jitter. This suits a network with a more erratic jitter profile, but can also be of benefit on a well-managed network. If the managed network is performing better than specified, e.g. below 10 ms of jitter rather than below the 30 ms planned, the buffer can adjust to reduce the overall delay of the connection. This delay reduction improves the perceived conversational quality experienced by the customers. An important design consideration for dynamic jitter buffers is when to adjust the play-out point. Although it is sometimes possible to do this during speech without the user noticing, it is far better to make the adjustment during silence where the user will not 'specifically notice' the change. The notion of not 'specifically noticing' a change in delay is because when asked to identify delay changes during silence the user is generally unable to do so. However, subjective tests show that delay changes are perceptible subconsciously during normal conversation. This suggests that delay changes should be made in small steps and kept to a minimum during a conversation or the perceived system quality will be reduced [3].

5.1.6. Codec tandem performance

When designing any voice transmission system it is important to know how well it will work with existing networks [7]. It is not enough to simply establish a connection; there is a need both to ensure adequate speech quality and to minimize delay. Typically where networks join, the speech traffic is passed as 64-kbit/s PCM. If the originating system has coded the speech in another format, decoding is required. This results in a transcoding or tandeming of speech codecs. Generally the quality of the combination cannot be better than the poorest link, and may be noticeably worse if two or more low bit rate codecs are included. The order is also important. Because these systems distort speech in a nonlinear way, G.729 followed by EFR* will not produce exactly the same quality as EFR followed by G.729. Delay also increases significantly with tandeming. For example, mobile networks introduce delays of around 100 ms each way. A mobile-to-mobile call with a VOIP trunk could easily exhibit 300-ms one-way delay. These problems can be avoided by ‘tandem-free operation’, where the system negotiates a common codec which is used end-to-end. It is expected that this will become more prevalent, but the potential quality improvement is balanced by a costly engineering challenge.

**Enhanced Full Rate or EFR or GSM-EFR is a speech coding standard that was developed in order to improve the quite poor quality of GSM-Full Rate (FR) codec. Working at 12.2kbit/s the EFR provides wire like quality in any noise free and background noise conditions. The EFR 12.2 kbit/s speech coding standard is compatible with the highest AMR mode.*

Current telephone systems typically allow the transmission of audio frequencies up to about 3.4 kHz. Speech and music contain many components at higher frequencies, which are filtered out and result in an unnatural sound. In contrast, CD audio produces frequencies up to about 20 kHz, the highest that can be detected by people with unimpaired hearing. Also codecs for wideband telephony and teleconferencing, transmitting audio frequencies up to about 7 kHz, have been available for some years. However, these have not become widespread due to problems with quality and interworking. New wideband codecs are under development and offer the potential for good quality at similar bit rates to the existing PSTN codecs, while allowing easy interoperation with the PSTN. Typically an existing narrowband codec, such as EFR (at 12.2 kbit/s), is extended by coding the higher frequencies separately — usually with relatively little overhead — and multiplexing the two data streams. The resulting bit stream at about 16 kbit/s can be decoded into wideband audio, or the higher frequency component can be dropped, without the need for re-coding, for transmission over a legacy connection.

5.2. Delay

Following [11] we would like to consider the view that in modern digital telecommunication networks, delay is a key performance parameter whose increase should be minimized. Although the delay of IP networks may exceed the typical delay of the PSTN, the degradation caused by additional delay might be compensated for by benefits provided by new network and service capabilities. These tradeoffs need to be quantified.

Delay can have two effects on voice performance. Firstly, it increases the subjective effect of any echo impairment. Secondly, as indicated in Recommendation G.114, even when echo is controlled, one-way delays above 150 ms can interfere with the dynamics of voice conversation, depending upon the type of conversation and degree of interaction.

Recommendations G.114 and G.131 and Annex A of G.173 [13] give additional information regarding effects of delay and echo.

In addition, delay can impair the performance of particular voice-band data applications, some applications being even more sensitive to the delay than voice applications. Total delay of hybrid Internet/PSTN networks should be limited, even with the use of echo control. Recommendation G.114 should be consulted for additional information [42].

5.2.1. Codec delay

Modern speech codecs operate on collections of speech samples known as frames. Each input frame is processed into a compressed frame. The coded speech frame is not generated until all speech samples in the input frame have been collected by the encoder. Thus, there is a delay of one frame before processing can begin. In addition many coders also look into the succeeding frame to improve compression efficiency. The length of this advance look is known as the look-ahead time of the coder. The time required to process an input frame is assumed to be the same as the frame length since efficient use of processor resources will be accomplished when an encoder/decoder pair (or multiple encoder/decoder pairs operating in parallel on multiple input streams) fully uses the available processing power (evenly distributed in the time domain). Thus, when a compression algorithm is in use (not in the case of G711) the delay through an encoder/decoder pair is normally assumed to be:

$$2 \times \text{frame size} + \text{look-ahead}$$

If the output facility is running at the same rate as the speech codec (e.g. an 8 kbit/s facility for G.729), then an additional frame of delay is incurred when clocking the compressed frame to the facility. Thus, the maximum delay attributable to codec-related processing in conventional systems (i.e. the PSTN) is:

$$3 \times \text{frame size} + \text{look-ahead}$$

If the output facility is an IP network, then the frame output by the encoder will instantaneously be dropped into an IP packet. The additional delay required for IP packet assembly and presentation to the underlying link layer will depend on the link layer. When the link layer is a LAN (e.g. Ethernet), this additional time may be quite small. If multiple speech frames are grouped together into a single IP packet, further delay is added to the speech signal. This delay will be the duration of one extra speech frame for each additional speech frame added to the IP packet:

$$(N + 1) \times \text{frame size} + \text{look-ahead}$$

where N is the number of frames in each packet [27] is analysis of delay doesn't apply to G.711.

5.2.2. IP terminal buffering delay

In the paper on IP terminal buffering [27] it was mentioned that audio cards and telephone cards in PCs usually include large internal buffers, in order to provide a fixed rate interface to A/D and D/A converter and an asynchronous interface to the application layer.

Additionally, modems and network adapters use internal buffers to increase network access efficiency. They have been optimized for data transmission where delay is not a problem, but this optimization may not be appropriate for voice transmission where delay is a critical issue.

There is also software buffering delays. Application or device drivers can store large amounts of data in order to process them easily and efficiently or to manage the delay jitter in received packets [27].

5.2.3. Packetization/buffering delays

Packetization delay is introduced while packets are being constructed. Buffering delay in the jitter buffer is introduced when they are being disassembled.

Buffering delay is due to queuing in the receiver and is usually used to compensate for network jitter. Voice playback requires equally spaced (in time) packets but network delays are variable, thus the receiver must delay packets that arrive early to synchronize them with those arriving later [27].

5.2.4. Network transmission delays

Transmission delay is the time spent by packets to reach their destination during transmission through the network. Components of network delay include [7] [43]:

- the transmission delay, introduced by sending a packet over a link (e.g. sending a 256 byte packet over a 64 kbit/s link takes 32 ms);
- the propagation delay, due to signal propagation over physical link. This delay is usually negligible if links are shorter than 1000 km;
- the node delay, due to router queuing and processing of packets;
- the protocol delay, due to packet retransmissions (if used, e.g. for TCP) or network access (e.g. CSMA-CD for Ethernet);
- gateway delay, introduced by interconnection between networks (e.g. packet disassembly/assembly and speech coding/decoding).

Network transmission delays may be negligible in fixed switched circuit network SCNs. However, significant transmission delays may be encountered in data networks (e.g. modem links or IP networks).

5.2.5. Delay variation

Packetized transmission systems exhibit variable delay in packet delivery times. Delay variation may have a negative impact on speech transmission quality. Depending on the nature of delay variations, the result may be experienced as time warping in speech or as impairments associated with lost speech packets [25].

Delay variations especially affect the performance of modems with auto-ranging echo cancellers. Accordingly, a guideline for limiting delay variation is desirable and is for further study for ITU-T.

5.2.6. End to End delay

From the paper of Anton, Borut and Saso [44] IP networks, to date, are not very suitable for transporting real-time traffic (like voice for example). This is basically because of the connectionless nature of networks and because of a low quality of service (QoS) offered by the IP networks. They were designed to carry data traffic and in order to facilitate the transmission of real-time traffic new mechanisms are needed. To support the transmission of real time voice traffic over IP networks, end to end delay performance parameters should be considered. One of the most important factors for the quality of interactive voice communication between two persons is the end to end delay.

Two problems that result from high end-to-end delay in a voice network, are echo and talker overlap. Echo is caused by the signal reflections of the speaker’s voice from the far end telephone equipment back into the speaker’s ear. Echo becomes a significant problem when the round-trip delay is more than 50 milliseconds [27]. Since echo is perceived as a significant quality problem, VoIP systems must address the need for echo control and implement some means of echo cancellation.

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 major constraint and requirement for reducing delay through a packet network is the end-to-end total delay time. The following are sources of delay in an end-to-end voice over packet call:

- Accumulation delay (or 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 waveform sample time (.125 microseconds) to many milliseconds. For a better understanding a representative list of standard voice coders and their frame times are given based on the ITU-T standard [13]:

    - G.726-ADPCM (16, 24, 32, 40 kb/s) 0.125 microseconds
    - G.728-LD-CELP(16 kb/s)-2.5 milliseconds
    - G.729-CS-ACELP (8 kb/s)-10 milliseconds
    - G.723.1-Multi Rate Coder (5.3, 6.3 kb/s)-30 milliseconds

- 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 [2]. 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.

- 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 that 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 because packet-delay variations can be as high as 70 msec to 100 msec in some frame relay networks and IP networks.

For the constant packet length and the unchanged path through the network, the only variable part of the total end to end delay is the time spent in queues of the network nodes on the transmission path. They are very hard to predict and depend heavily on the current network load. With the consideration of certain criteria the upper delay margin can be satisfactory defined.

Queuing delay can be reduced by the introduction of advanced queue scheduling mechanisms and implementation of QoS network architecture. If we assume that the network and the transmission path are fixed then we can reduce the fixed part of the network delay only by sending out shorter IP packets. Using the QoS mechanisms that make use of the different advanced queue-scheduling mechanisms can reduce the variable part of the network delay. There are several queue scheduling mechanisms that can be used. One of them is priority queuing that effectively reduce delay for the high priority traffic, for instance packet that's carry real-time voice. ITU-T Recommendation [27] also mentions that a similar principle is used in local area networks with the use of IEEE 802.1q and 802.1p standards for the IP telephony.

As an example Figure 6.2 [27] shows the simulation result for priority queuing in the IP network with four routers on the transmission path and 50% of the network load in the backbone. It is clearly visible that the delay for voice packets can be significantly reduced when the priority queuing is employed in the comparison to the FIFO queuing. The priority queuing successfully reduces the high-delay tail in the packet delay distribution [44].

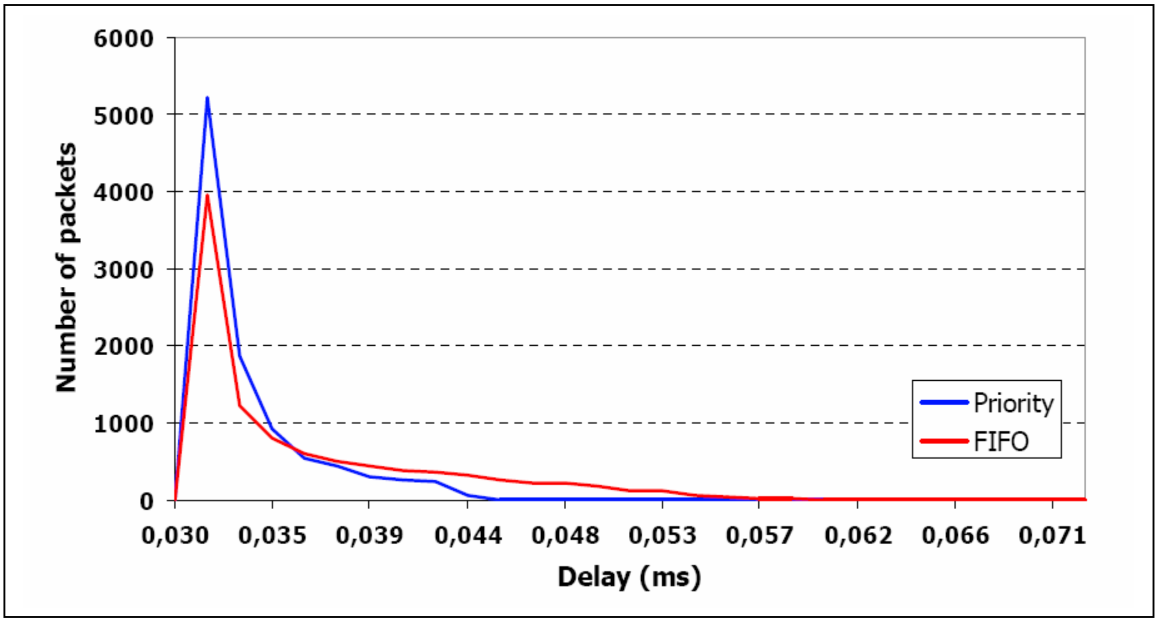


Figure 5.1 Packet delay Distribution [27]

5.3. Loss

A potentially more severe performance issue is that of lost or discarded IP packets. An IP packet may be lost due to congestion in the IP network. An IP packet may also be discarded at the destination. This would occur, for example, when a packet is sufficiently late that the destination declares the packet lost. Discarding packets that are very late is preferable to having the destination increase delay and delay variation by potentially waiting for long time periods [36].

Loss of a single IP packet will result in loss of one or more coded speech frames, depending on the speech coder in use and the number of frames per packet. As noted above, the speech coder should be robust with respect to loss of coded frames. In particular, if multiple frames are assembled into an IP packet, the performance of the speech coder must be assessed under frame loss conditions that reflect those of the network in use. The impact of large percentages of frame loss (as much as 20-30%) should be assessed [36]. This is an important consequence. It reduces the attractiveness of highly compressed coding.

VOIP network packet loss introduces audio distortions that cause voice quality to decrease as the rate of packet loss increases. That said, on any particular connection this general effect can be modulated by:

• The distribution of the lost packets

• The packet loss concealment (PLC) algorithm in use [30]

Early Papers on VOIP generally repeated the assertion that packet loss did not become a significant problem until it reached a 5 percent rate [41]. This assertion provided poor guidance. The data are collapsed using variety of algorithms and are based on the G.711 (64 kb/s) coder and a 20 ms packet size. The PLC algorithms used ranged from simply inserting silence for the missing audio to the use of the G.711 Appendix I algorithm that does a good job of masking the audio effects of up to 3 percent packet loss. Considering all the qualifying factors, it seems to me that VOIP networks must hold packet loss below 1 percent in order to deliver a level of voice quality that is public switched telephone network (PSTN) equivalent. Intuitively, it would seem that the negative effects of packet loss would be exacerbated where packets are lost in clusters or bursts rather than in isolation. But where the overall rate of packet loss is held constant, the data are more complicated [30].

5.4. Echo

Current assessment of delay in hybrid Internet/PSTN connections indicates that echo control is required for all types of calls. Another view is that echo cancellation should occur at the network edge or the terminal device. Control of echo from the PSTN should be provided in the gateway between the IP network and the PSTN [36].

5.4.1. Echo from H.323-based terminals

From ITU-T Recommendations it has been determined that when the terminal is using a microphone and loudspeaker as the transmitter and receiver, there will be an echo due to acoustic* coupling between transmitter and receiver [22]. This is a dominant feature but there could some other sorts of echo also be involved. The existing PSTN infrastructure probably will not provide adequate echo protection if the acoustic coupling loss in the terminal is too low and the delay is too high. Recommendation H.225.0 indicates that control of acoustic echo from the H.323 terminal is the responsibility of the terminal. In order to provide echo protection all H.323-based terminals should meet the Weighted Terminal Coupling Loss objective of 45 dB, as is specified for digital wire line terminals in Recommendation P.310. Such acoustic isolation may be achieved relatively easily in standard handset terminals by careful design. However, in hands-free operation (e.g. microphone and speaker), other more complex techniques may have to be used.

**Acoustic echo is defined as the coupling of received voice transmission between the earpiece and mouthpiece of a portable handset or the speaker and microphone of a hands-free mobile phone. When acoustic echo occurs, it is the PSTN user who is discomforted. Acoustic echo is a much more complex signal than hybrid echo. The received signal emits from the speaker and reflects from multiple surfaces inside an automobile. The reflections return the signal, at various time delays and amplitudes, into the microphone, and over the phone connection to the PSTN user's ear. In addition, the tail circuit is non-linear because of the speech compression.*

For example, introduction of advanced echo control technology capable of increasing acoustic isolation in hands-free terminals may be needed (standard echo cancellers may not be capable of providing sufficient isolation in a non-linear acoustic environment) [36].

5.4.2. Echo from the PSTN

The echo path, that arises at the PSTN end of the connection due to a poor impedance match at the 4-to-2 wire conversion point. This really doesn't normally occur in a VOIP system but for complete understanding we need to consider this case. Also there is the case where one end is PSTN to consider. In this configuration an echo canceller is applied in the Interworking Function to control echo. The echo cancelling function may, in practice, be implemented at any location within the system. However, practical considerations (i.e. capabilities of existing echo cancellers) indicate that the appropriate location is in the gateway.

Based on the existing PSTN infrastructure, the gateway should provide echo cancellation. It is likely that such echo cancellers will, in some configurations, be working in tandem with the PSTN echo control devices. This should not degrade the overall echo control function in the connection. In addition, effects of interaction of echo cancellers in the gateway with signal processing devices in the PSTN (e.g. PCME or conference bridges) is under study in the ITU-T.

Recommendation G.168 provides specifications for digital network echo cancellers. At a minimum, echo cancellers deployed in the gateway should meet these requirements [1].

5.5. Noise

5.5.1. Environmental (acoustic) noise

Acoustic noise at the transmitting end of a connection will have a negative impact on the performance of speech coders. The speech coders found in the G.72x-series have been tested for the effects of environmental noise at the sending end of the connection. However, some codecs have been tested more extensively than others. All have been shown to be fairly robust under conditions that included addition of circuit noise or speech babble to the input speech. If a particular type of background noise will be dominant in a given application, it is advisable to verify that performance of the speech codec is satisfactory under those conditions.

Pick-up of environmental noise with non-standard handsets may present special problems for the quality of speech when low bit rate codecs are used. In these situations, selection of a speech codec that is robust in the presence of acoustic background noise is especially important [3].

5.5.2. Idle channel noise

Idle channel noise in VOIP applications should be negligible. If present, however, background idle channel noise should be less than –68 dBm0p, according to Recommendation G.106 [1].

5.5.3. Noise contrast and comfort noise

Noise contrast occurs when background noise is interrupted due to digital speech processing, such as echo cancellation using centre clippers, and voice activity detection (silence removal). Comfort noise is noise that can be introduced to mask the negative effects of noise contrast. According to the ITU, recommendations on noise contrast limits, and comfort noise values, are for their future study [22].

For comfort noise insertion, some digital cellular systems (e.g. GSM) use an approach where noise parameters are extracted at the sending end and transmitted to the receiving end at a low bit rate. It is then possible to reconstruct (to good approximation) the background noise. This approach should provide superior subjective performance for voice users of circuits using voice activity detection and comfort noise insertion. The voice activity detectors and comfort noise generators described in Annex B of G.729 [13] and Annex A of G.723.1 [36] both operate in this fashion.

The best (subjective) performance will be realized when the noise inserted at the receiving end matches, as closely as possible, the background noise at the sending end. The following comments on CNGs can be made:

- the noise used should match the background noise, both in frequency content and level
- level of the inserted noise should match that of the background noise
- the time course of changes in the level of the inserted noise should match, as closely as possible, the level changes that occur in the background noise. If background "noise" is considered to be a valid component of the transmitted signal, silence suppression should not be used [3]

5.6. Combined Effect

Many factors determine voice quality, including the choice of codec, echo control, packet loss, delay, delay variation (jitter), and the design of the network. Packet loss causes voice clipping and skips. Some codec algorithms can correct for some lost voice packets. Typically, only a single packet can be lost during a short period for the codec correction algorithms to be effective. If the end-to-end delay becomes too long, the conversation begins to sound like two parties talking on a Citizens Band radio. A buffer in the receiving device always compensates for jitter (delay variation). If the delay variation exceeds the size of the jitter buffer, there will be buffer overruns at the receiving end, with the same effect as packet loss anywhere else in the transmission path.

For many years, the PSTN operated strictly with the ITU standard G.711. However, in a packet communications network, as well as in wireless mobile networks, other codecs will also be used. Telephones or gateways involved in setting up a call will be able to negotiate which codec to use from among a small working set of codecs that they support [4].

There are different types of combined effect which can play a distinguish role in the VOIP traffic in general. But in any project it’s not possible to investigate all the possible types of combination and their effects on the VOIP transmission. After investigation of different papers and comments we would like to present some of the research into combined effect of several factors.

5.6.1. Delay and Echo

In contrast to broadcast-type media transmission (e.g., RealAudio), a two-way phone conversation is quite sensitive to latency, Most callers notice round-trip delays when they exceed 250mSec, so the one-way latency budget would typically be 150 mSec. 150 mSec is also specified in ITU-T G.114 recommendation as the maximum desired one-way latency to achieve high-quality voice [26]. Beyond that round-trip latency, callers start feeling uneasy holding a two-way conversation and usually end up talking over each other. At 500 mSec round-trip delays and beyond, phone calls are impractical, where you can almost tell a joke and have the other guy laugh after you've left the room. For reference, the typical one way delay when speaking through a geo-stationary satellite is 150-500mSec [27].

The most important components of this latency are:

Backbone (network) latency, which is the delay incurred when traversing the VOIP backbone.

CODEC latency, which is each compression algorithm, has certain built-in delay.

And finally jitter buffer depth to compensate for the fluctuating network conditions; many vendors implement a jitter buffer in their voice gateways.

Echo is also playing a significant role in the VOIP transmission. Echo can be generated from different types of sources. From our investigations we have identified two different points of notation one Echo from H.323-based terminals and another is echo from the PSTN. The dominating echo in VOIP systems will usually be the echo due to acoustic coupling between transmitter and receiver. When the terminal is using a microphone and loudspeaker as the transmitter and receiver, it generates the acoustic coupling.

For a better understanding and clearer picture we would like to restructure a MOS [4].

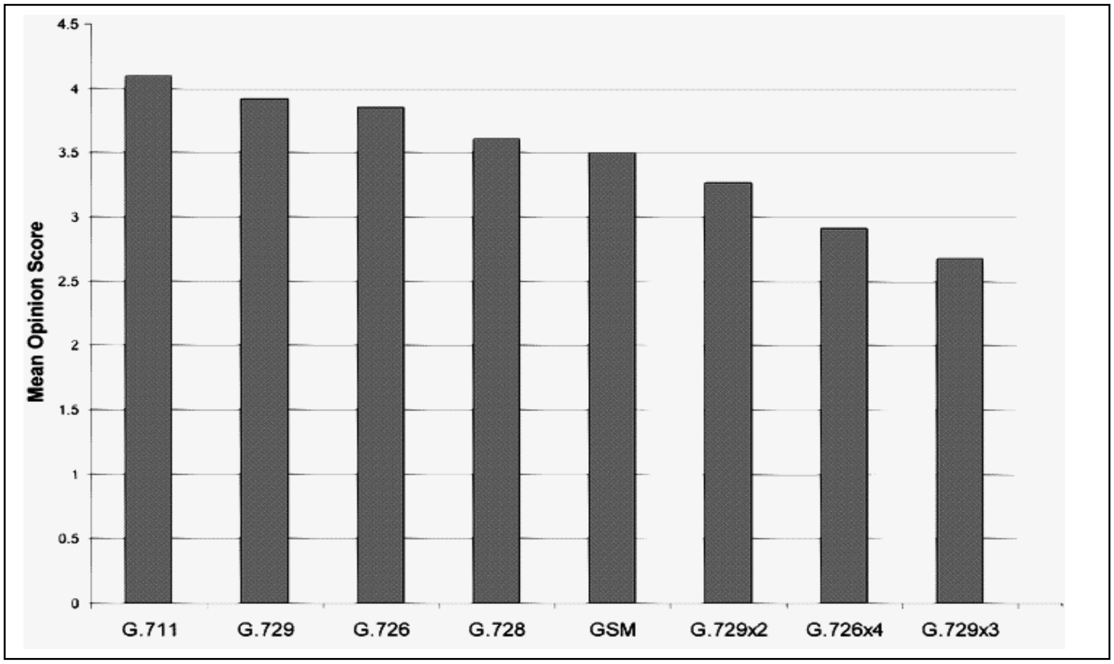


Figure 5.2 Mean Opinion Score Diagram

According to the MOS standard A MOS of 5 is excellent, 4 is good, 3 is fair, 2 is poor and 1 is very bad. As a very simple approximation we can assume that the effects on MOS are additive. So, if we consider that each of our delay and echo will create a negative unit 1 MOS the resultant output will be worse than the existing one by two. So the combined effect will be very poor. So combined with this level of noise, none of the codecs or combination of codecs considered in 5.1 will be satisfactory.

5.6.2. Loss and Noise

Since even well-engineered IP networks tend to have a small residual packet loss rate caused by low-probability statistical congestion events and amplification of bit errors in the hardware, it is attractive to use some kind of forward error correction (FEC) to ensure that the encoded voice stream can not be reconstructed if a few packets are lost. FEC is a good idea, but it may be too complicated in practice. This is typically applied at the packet level, since the encapsulated voice bit stream is typically only designed to tolerate low levels of bit or burst errors, rather than the loss of whole packets. In VOIP packet loss is a very typical issue that shouldn't be compromised in any way. If a packet gets lost in the transmission it can not be constructed from the information in the other packets.

Noise is another closely related issue for VOIP. From our previous investigation we found that there are several types of noise are playing role on the VOIP system. Packet loss will also appear to the customer like the addition of noise.

Combining loss and noise if we can focus on the MOS again we will find that the negative unit 1 will create an outrageous effect on the MOS as these two unit 1; will combine and create a effect of 2 units negative MOS. Now considering the Figure 5.1 we can see that the outcome for the resultant MOS will be most unsatisfactory than ever when the combination of codec occurs.

5.6.3. Encoding

In the existing internet there are different types of encoding which are happening in the two terminals of the network. When a wide range of cross transection VOIP is transferring over a large network one point may be using a higher quality encoding system then the other terminal. The QOS for the VOIP is facing a long term effect and losing its effectiveness due to this different coding system. These types of effect can create loss of data and noise in the other end. And as a combined effect of encoding, noise and loss the outcome will be significantly poor in quality and outcome will be devastating.

Although consumers are ready to experience a significant delay when a long distance call takes place in this scenario the addition of noise and loss will create unacceptable quality of service [30].

Chapter 6

Improving VOIP: An Engineering Challenge

QoS in a voice network includes, logical issues such noise & echo: However, in relation to packet networks we also use the term QoS to refer to just the key characteristics of delay, loss, bandwidth & throughput in the data & packet network. Specifically, QoS refers to the ability of a network to provide better, more predictable service to selected network traffic over various underlying technologies, including IP-routed networks. Traditionally packet switched networks did not require strict measures for QoS because the data wasn't multimedia and the end-user would not notice or be materially affected by latencies. But, as the use of internet has spread far beyond simple data transfer to intense multimedia applications, the need to address Quality of Service (QoS) issues has become more important. Both the enterprise and consumer markets are now beginning to demand data intensive, time-sensitive movement of things like audio and video around internet. Voice applications have different characteristics and requirements from those of traditional data applications. Because they are innately real-time, voice applications tolerate minimal delay in delivery of their packets. Additionally, they are intolerant of packet loss, out-of-order packets, and jitter. To effectively transport voice traffic over IP, mechanisms are required that ensure reliable conveyance of packets with low and controlled latency. Thus the goal in the context of VOIP, QoS, then would be to provide dedicated bandwidth, controlled jitter and latency (required by some real-time and interactive traffic), and improved loss characteristics [45].

The rational of this Chapter is to investigate solutions for improving VOIP QoS. As it is not possible to point out all the factors that we have identified in the earlier Chapter, we would like to focus on some of the issues.

6.1. AQM

According to the paper of Stephan, Khushboo, Gonzalo and Mike on AQM [46] a key feature of TCP is that it decreases the sending rate when it experiences packet losses. AQM tries to take advantage of this feature by dropping packets to control the packet arrival rates. Alternatively Early Congestion Notification (ECN) bits can be set in packets to control congestion [47]. Figure 6.1 situates AQM in the network. The question mark in Figure 6.1 denotes AQM principle task, deciding to drop a packet or not.

There are two ways that AQM decides to drop a packet. AQM may directly select which packet is to be dropped as in the case of adaptive virtual queue (AVQ) and stabilized RED (SRED). Or, as in the case of Floyd and Jacobson's RED, adaptive RED (ARED), random exponential marking (REM), proportional integrator (PI) controller, and BLUE, the AQM determines a drop probability and packets are dropped at random according to this probability. There has been extensive work toward modeling TCP's data rate. The typical assumption is that packets are dropped at random.

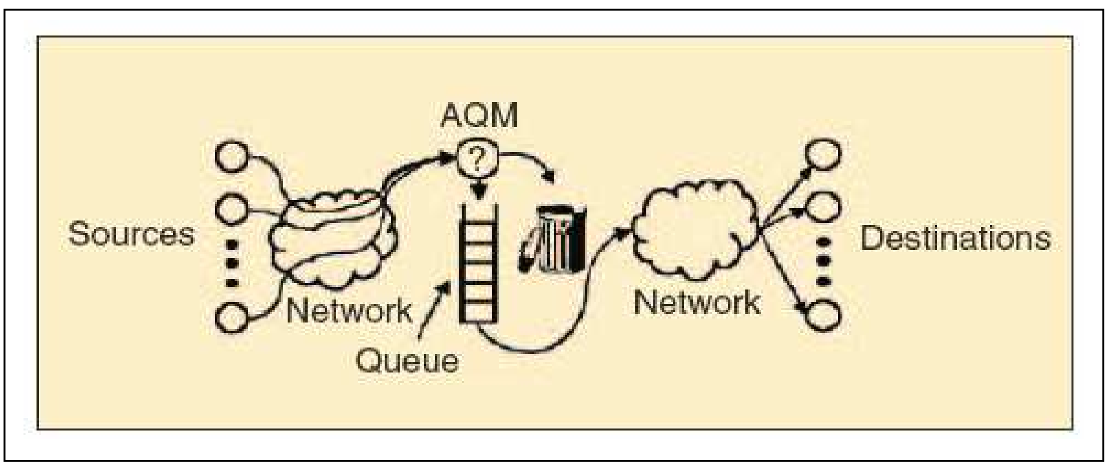


Figure 6.1 A pictorial representation of AQM [46]

The utility of an AQM scheme that performs well in a narrow sense is limited. Rather, an AQM scheme for "typical" networks must meet many objectives, some of which are difficult to quantify. Furthermore, these requirements change as new technology and new applications are introduced. For these reasons, listing the objectives of AQM is a difficult and perhaps controversial task. In the original paper by Floyd and Jacobson and in many of the AQM papers since Floyd and Jacobson's, the primary objective was to keep the average queuing delay low while maximizing the throughput. If the queue never empties, then the outgoing link is always busy, hence the highest possible throughput is achieved. While large average queue occupancy reduces the possibility of the queue emptying and the link going idle, it also increases the time it takes for packets to pass through the queue. Hence, Floyd and Jacobson desired a balance where the average queue occupancy is as small as possible without causing a significant decrease in throughput [46].

6.2. Speech Coding

The Internet is developing to become the ubiquitous communication network, which supports all kind of multimedia communications including telephony. Telephony still accounts for a major part of revenue for telephone companies, being the most important form of telecommunication between humans. Most telephone calls are conducted on PSTN systems. Internet Telephony on the other hand struggles to fulfill quality requirements and expectations. Therefore, enhancing the quality of telephone calls over packet networks, especially the Internet is a worthwhile goal. Qualities of Service (QoS) architectures have been introduced to guarantee quality levels. Diffserv [16] effectively assigns a higher priority to telephone calls and a lower priority to data transmission, so long as the total bandwidth of the voice traffic lies below a specified level. Emerging QoS architectures like DiffServ can treat, forward and drop packets according to their pre-defined DS Codepoint [48]. Novel approaches request different priorities for individual packets within a single flow. This leads to the problem to classify the importance of each packet correctly, so that the overall service quality can be optimized. The human perceived quality of a telephone call should be the main optimization criterion, because most calls are between humans. The quality of telephone calls compromises many aspects: One important factor is the quality of speech transmission. The perceived speech quality is often measured in the metric mean opinion source (MOS). Metrics like MOS, which are based on human based quality judging, are difficult to apply to plan and control a communication networks because they require that humans perform the quality evaluation. Networking based quality metrics like packet loss rate, throughput and delay are easier to measure, to control and to guarantee. But they do not reflect the experienced user quality precisely [41].

The standard coder that has been using the telecommunication industries is mostly G.711 [36]. Beside this coder there is several other ITU-T standard coders are in use with the local PSTN service. G.711 encodes at 64 kbit/s and introduces little audible distortion for most types of signal. In a number of applications, however, a much lower bit rate is desirable either because capacity is limited (e.g. mobile) or to maximize the amount of traffic that can be carried over a trunk connection. Current ITU-T recommendations include codecs that compress to as low as 5.3 kbit/s, although the quality at this bit rate is well below that of G.711. To maximize interoperability it is usual for VOIP gateways and clients to offer one or more standard codecs. Table 5.1 lists some common, standardized, voice codecs with their associated bit rates [31].

A mechanism we should consider is to introduce a dynamic coding system which will provide a strong and efficient support for improving the QoS of VOIP in general. A dynamic coder is the system, which will be able to identify the congestion in the network during an active call than switch between the higher rate codec to lower rate codec or the other way round. Customers will get a better quality service as a result. Beside this when we are considering a small network e.g. a house or a small office; the gateway will then be able to work more effectively with a variable rate codec. Let's consider a network, which is a combination of several computers as well as one or more VOIP connections. At any time of when all the components which to transfer data over the network at once, congestion will take place immediately. Under this circumstance the variable rate codec can handle the data more effectively by choosing a lesser bit rate coder instead of a higher bit rate coder for the VOIP. The reverse situation will also apply when the variable rate coder will be able to provide a higher bit rate coder to provide a better quality service. Such congestion will only persists for a short period of time. Consider sending a mail or just look into a certain web site in the GOOGLE search or even a bank transaction, which may take few seconds or fraction of seconds while a call can last for few hours. Sharing data with small data traffic flow could be more effective if the network can be handled more efficiently.

6.3. Packetization (Packet Loss)

UDP cannot provide a guarantee that packets will be delivered at all, much less in order [3]. Packets will be dropped under peak loads and during periods of congestion. Due to time sensitivity of voice transmissions, the normal TCP based retransmission scheme is not appropriate. Approaches used to compensate for packet loss include interpolation of speech by replaying the last packet and sending redundant information. Packet losses greater than 10 percent are generally intolerable, unless the encoding scheme provides extraordinary robustness. It is important to note that because of IP protocol inefficiencies, reducing codec bit rates below a certain level makes limited sense. This is primarily due to the IP, UDP and

RTP headers contributing towards a 40-byte overhead to each packet. At 20-ms packet spacing (a single GSM frame or two G.729 frames), this equates to a 16-kbit/s overhead, with efficiency defined as the reduction in bandwidth compared to a 64-kbit/s PSTN channel. By comparing only codec efficiency, G.729 shows an 87.5% saving over G.711, but, accounting for the IP headers this saving is reduced to 60%. These calculations do not take into account header compression, which would provide additional savings. Header Compression only applies in parts of an internet, typically the access network, however this might still be very useful.

When selecting a codec it is also important to know how well it will code non-speech signals such as background noise. When telephones are used in a noisy environment the ability of a particular codec to encode the background noise can have a significant effect on the perceived quality. Mobile telephone codecs are designed to cope with high noise environments and therefore it has been suggested that these are probably the best low bit rate codecs for noisy environments [31].

6.4. Silence Suppression

Silence suppression takes advantage of prolonged periods of silence in conversations to reduce the number of packets. In a normal interactive conversation, each speaker typically listens for about half the time, so it is not necessary to transmit packets carrying the speaker's silence. Many vendors take advantage of this to reduce the bandwidth and number of packets on a link [1].

According to the survey [1] voice in its inherent nature is random. It is found that on an average, human voice has a speech activity factor of about 42%. There are pauses between sentences and words with no speech in either direction. One can take advantage of these two characteristics to save bandwidth by halting the transmission of cells during these silent periods. This is known as silence suppression.

6.5. Echo

It is readily understood that the conversational impairment due to echo increases with its level and delay. A large amount of work has been conducted to determine the combined effect of talker echo with delay. Recommendations on its control are summarized in ITU-T G.131 [13]. There are essentially two sources of echo in today's PSTN; these are shown in Fig 6.3 [31].

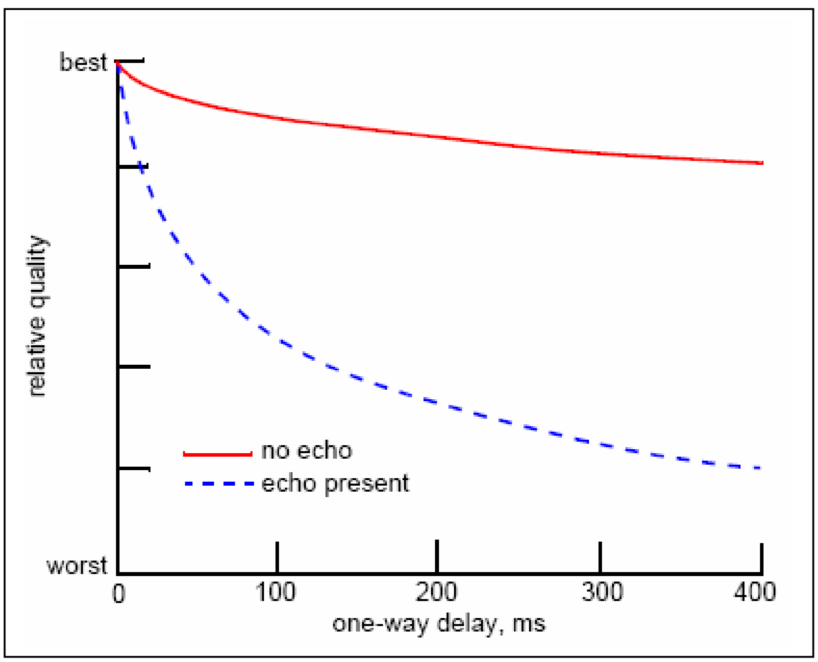


Figure 6.3 Impact of Delay on quality [31]

The delay introduced by packetizing the speech and removing network jitter, result in delays long enough to make the system susceptible to echo problems. Echo cancellation is therefore likely to be needed in most VoIP systems. This is in contrast to the PSTN where echo cancellation is only necessary on long-haul connections. Short-delay echoes are rarely distinguished from side tone unless either the round-trip delay exceeds 30 ms or the echo level is extremely high. For this reason echo cancellation is not required on short-haul

PSTN connections, where round-trip delays do not exceed 30 ms. However, round-trip delays of VoIP systems are unlikely to be less than 30 ms, ensuring that some form of echo cancellation is invariably required. If a VoIP system connects to a local PSTN, echo cancellation is probably needed to cancel the local hybrid reflections. If the system does not connect to a local PSTN, echo cancellation should still be included to remove any acoustic echo. As a final note, in order for a VoIP device to be considered as high quality, the performance of its echo canceller should adhere to ITU-T Recommendation G.168 as a minimum.

6.6. Jitter Buffer and Adaptive Alpha

A typical VOIP terminal buffers incoming packets and delays their playout in order to compensate for variable network delays, or jitter. This allows slower packets to arrive in time to be played out; packets which arrive too late for playout are regarded as lost. If the buffering delay is set too large, the overall latency increases to a level where interactivity of the conversation suffers; if it is set too small, the resulting increased packet loss rate decreases the perceived voice quality. The conflicting goals of minimizing buffering time and minimizing late packet loss have led to various playout algorithms. An adaptive playout mechanism makes it possible to trade off the buffering time a major component of end-to-end delay with the rate of packet loss. According to [49] a new VOIP adaptive playout algorithm that significantly improves this trade-off compared to the basic adaptive playout algorithm is needed.

The basic adaptive playout algorithm estimates two statistics, the delay itself and its variance for each incoming packet, and uses them to calculate the playout time

$$D_i = \alpha \, . \, D_{i-1} + (1-\alpha) \, . \, n_i$$

$$V_i = \alpha . V_{i-1} + (1-\alpha) . | Di - n_i |$$

Where D_i and V_i are the $_i$th estimates of delay and its variance, respectively, while n_i is the ith packet delay. The weighting factor α has a critical impact on the rate of convergence of these estimations, which in turn influences the calculated playout times and hence the delay/loss trade-off. According to the [49], α should be fixed at a high value, e.g., α = 0.998002. This was motivated by work on TCP round-trip time (RTT) estimation and assumed slow changes in RTT. However under typical present-day Internet conditions, the accuracy of the estimates and thus the resulting VOIP playout quality can be greatly improved by dynamically choosing the value of α [49].

6.7. Simulation and Scientific Experiment

In this project we have undertaken some simulation experiments to investigate possible approaches for improving QoS.

Figure 6.4 shows a complex network of the telecommunication 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. Figure 6.5. 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.

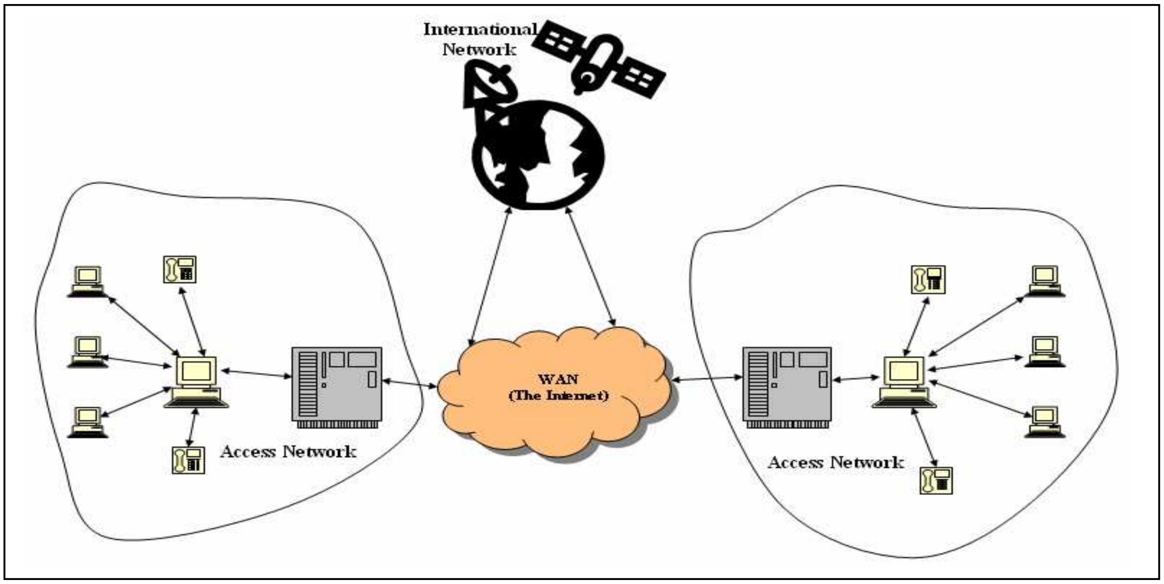


Figure 6.4 Real life picture of a typical network

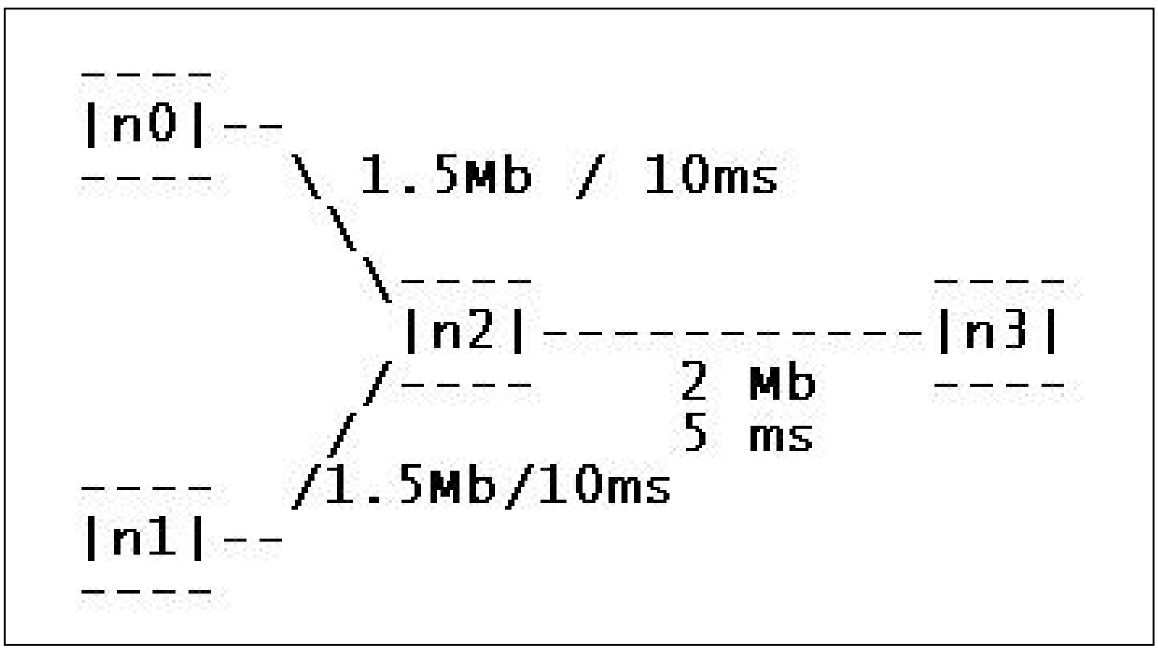


Figure 6.5 Basic Model for Simulation

In Figure 6.5 n0, n1, n2, n3 are four IP routing nodes. In these four nodes n0 is a TCP/IP node and FTP is the traffic generator for this node. The TCP/IP node has been attached to node n3 via n2 where a SINK is attached. These all are duplex links. Again on the other side node n1 is linked via n2 to n3 and a NULL agent, which just frees the packets received, is attached to it. With node n1 a CBR traffic generator is attached which generates the traffic for 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 network. In the Experiment 1, we have used the normal Droptail, AQM. In the Experiment 2 we have performed Experiment 3 using RED and finally we have performed an experiment using the Diffserv system. Two additional experiments, Experiment 2B and Experiment 3B have also been conducted in sequence of Experiment 2 and 3.

```
----          ----
|s1|---------|n0|--
----   10 Mb  ----  \ 1.5Mb / 10ms
        1 ms         \
                      \----          ----           ------
                       |n2|-----------|e2|-----------|dest|
                      /----    2 Mb   ----   10 Mb   ------
                     /         5 ms           1 ms
----          ----  /1.5Mb/10ms
|s2|---------|n1|--
----   10 Mb  ----
        1 ms
```

Figure 6.6 Model for Simulation with Diffserv extensions

In every experiment we have generated several different types of outputs: graphical output using NAM and graphical output using xgraph and in the graphs we have 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 Figure 6.5 n0 and n1 could be in an access network for example. This could be a small business or may be a home premise. In the third experiment we have introduced Diffserv and certain changes have been made in the network to cope with the Diffserv architecture.

6.8.1. Experiment 1:

Our first experiment was a simple network test without making use of any sorts of AQM's like RED or DiffServ. In this experiment the network consists of 4 nodes (n0, n1, n2, n3) as shown in Figure 6.5. The duplex links between n0 and n2, and n1 and n2 have 1.5 Mbps of bandwidth and 10 ms of delay. The duplex link between n2 and n3 has 2 Mbps of bandwidth and 20 ms of delay. Each node uses a Droptail queue, of which the maximum size is 25 packets. A "TCP" agent is attached to n0, and a connection is established to a TCP "sink" agent attached to n3. As default, the maximum size of a packet that a "TCP" agent can generate is 1KByte. A TCP "sink" agent generates and sends ACK packets to the sender (TCP agent) and digests the received packets. A "UDP" agent that is attached to n1 is connected to a "null" agent attached to n3. A "null" agent just frees the packets received. An "FTP" and a "CBR" traffic generator are attached to "TCP" and "UDP" agents respectively, and the "CBR" agent is configured to generate 128 Byte packets at the rate of 1 Mbps. The "CBR" is set to start at 1.0 sec and stop at 3.0 sec, again "CBR" is set to start at 3.5 sec and stop at 4.0 sec and the "FTP" agent is set to start at 0.1 sec and stop at 4.9 sec.

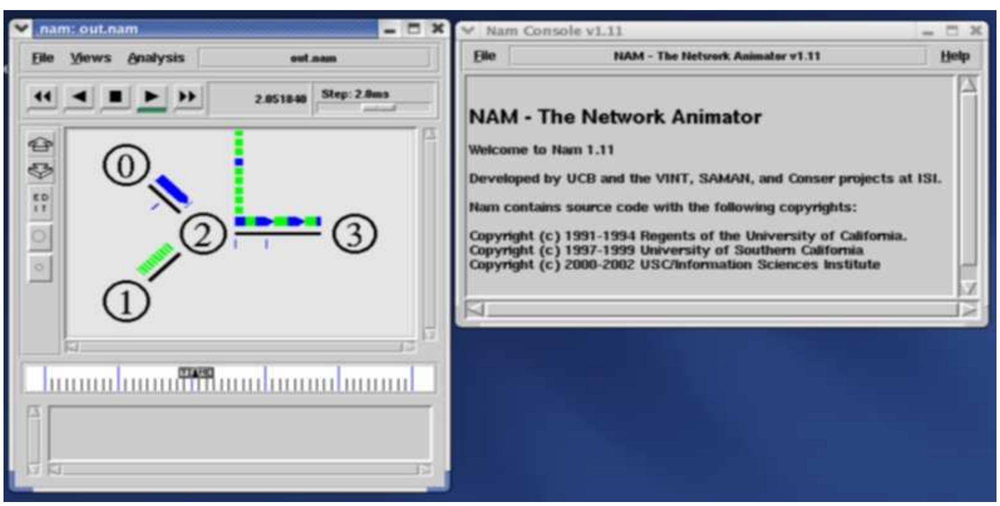


Figure 6.7: The NAM outlook

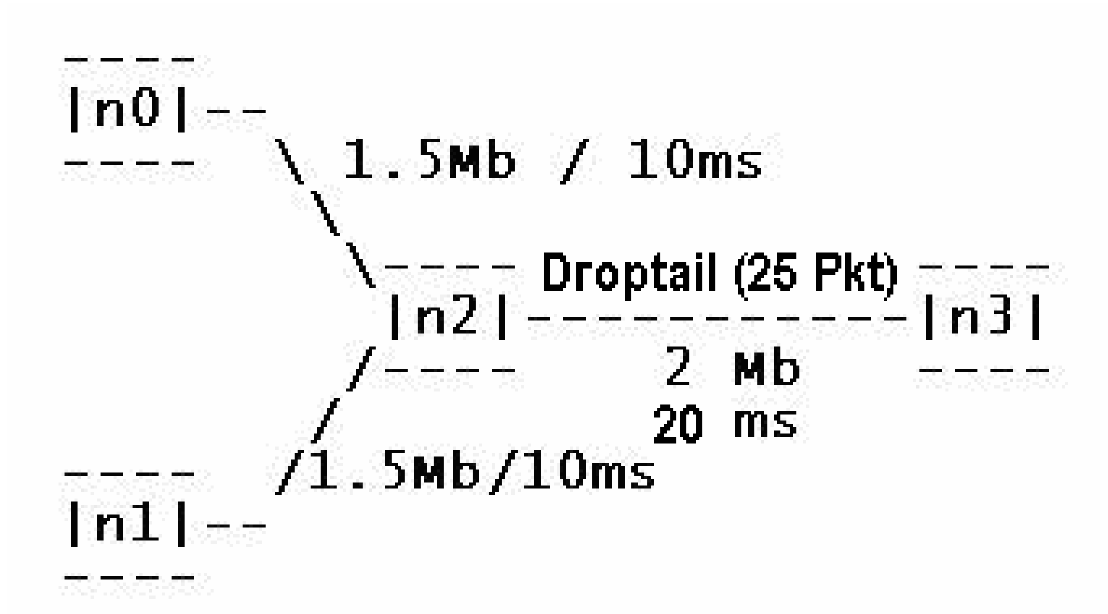


Figure 6.8 The network for Experiment 1

NS2 provides an animation tool, called NAM, which shows all significant events during a simulation by means of a graphical display. Figure 6.7 is an example from such simulation

In the simulation, using NAM output, it will be noticed that when the queue limit is reached the TCP traffic backs off and this back off situation happens when the TCP packets are dropped and node n0 doesn't receive the acknowledgements from the SINK. On the other hand, the UDP traffic just keeps on going as it doesn't wait for acknowledgements.

Comparing Experiment 1 with the Experiment 2, it is apparent that using a standard queuing technique, the data traffic handling condition can be improved. Retransmission of lost voice packets would be pointless- that is why UDP has been used..

This experiment can be modified in any way and we could then see the real time picture of the data flow.

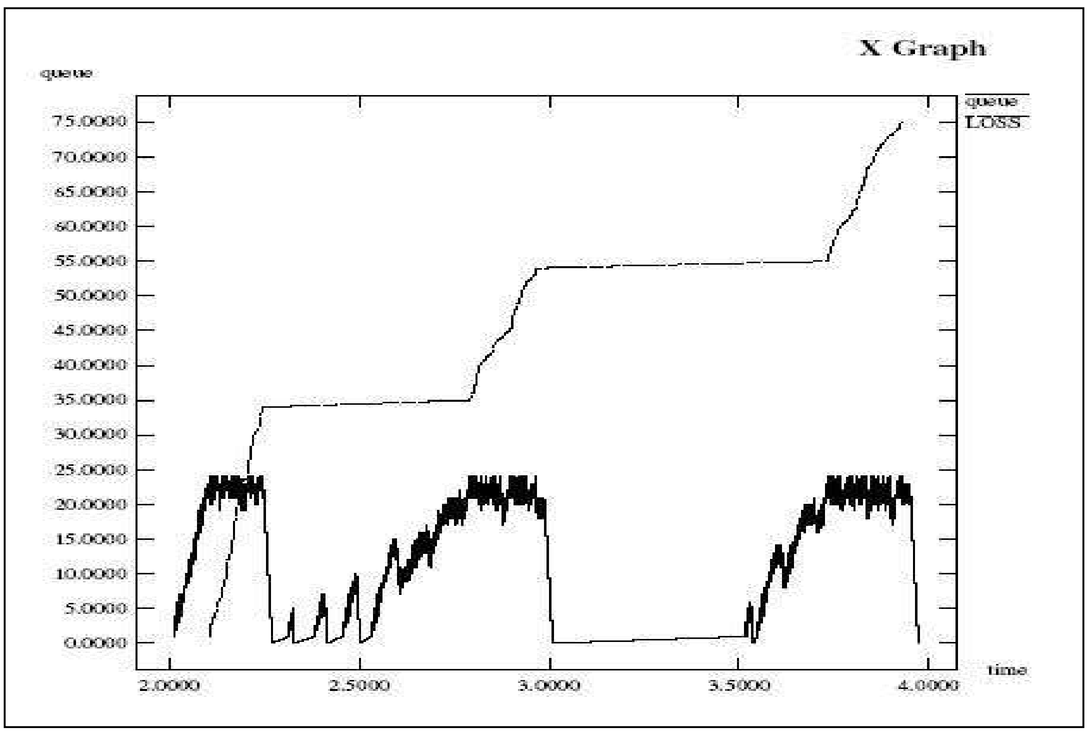


Figure 6.9 Experiment 1 xgraph output

In Experiment 1 we set the CBR packet size to 128 bytes which is often used with the VOIP calls. The Droptail queue system is used and we trace the queue and draw the xgraph from the acquired data. From the NAM output we have seen that lots of CBR packets are dropped. The size of the TCP packets in this experiment is 1000 bytes. TCP is expected to back off when there is too much data flow and dropped packets are occurring. As a consequence TCP starts sending repeat acknowledgements (acknowledgement for the same packet more than once). The FIFO Droptail queue starts dropping more CBR packets than TCP as CBR packets don't back off. Our main concern is to transmit the CBR packets without excessive loss. The TCP lost packets will be retransmit with a delay which is acceptable for that service. However massive loss of UDP traffic will create terrible performance in the voice transmission.

Figure 6.9 is an xgraph plot of queue and loss versus time. Here the queue limit is reached regularly & packets will be lost when this occurs. The loss displayed on the graph is accumulated loss. It is massive in this experiment. Modification of the Droptail AQM over the existing link is not feasible because increasing the buffer size will cause massive delay. We will discuss this experiment further as we compare it to Experiments 2 and 3.

In the presence of the Droptail, AQM Droptail substantial amount of packets are dropping. These are combination of UDP and TCP packets. Using NAM we have observed that the drop rates of UDP packets are higher than TCP packets. But for VOIP we would prefer to drop only the TCP packets.

6.8.2. Experiment 2:

Experiment 2A

Our Second experiment uses the same network with the RED AQM at the buffer at the head of the link from n2 to n3. The network topology and traffic conditions are exactly the same as Experiment 1. Each buffer except for the one at the head of the link n2 to n3 uses a Droptail queue, of which the maximum size is 25 packets.

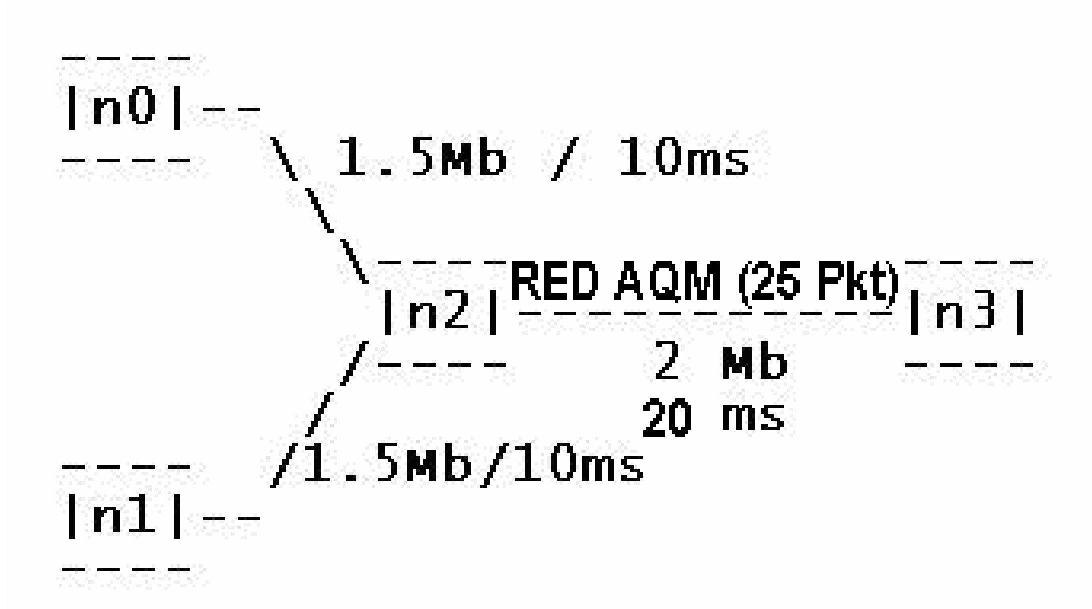


Figure 6.10 Experiment 2 diagram

In the Experiment 2 simulation, using NAM output, we also have noticed that when the queue limit is reached the TCP traffic backs off and this back off situation happens when the TCP packets are dropped and node n0 receive repeated acknowledgement or doesn't receive the acknowledgements from SINK. This is very similar to the scenario of the Experiment 1. We tried to force two points to behave identically. The UDP traffic, as before, starts and keeps on going as it doesn't wait for acknowledgements. And in the Experiment 2, using the RED AQM the data traffic handling condition has been improved. Overall loss is similar but buffer level is reduced, especially in the second half of the experiment.

This experiment can be modified in any and we can see the real time picture of the data flow, using NAM.

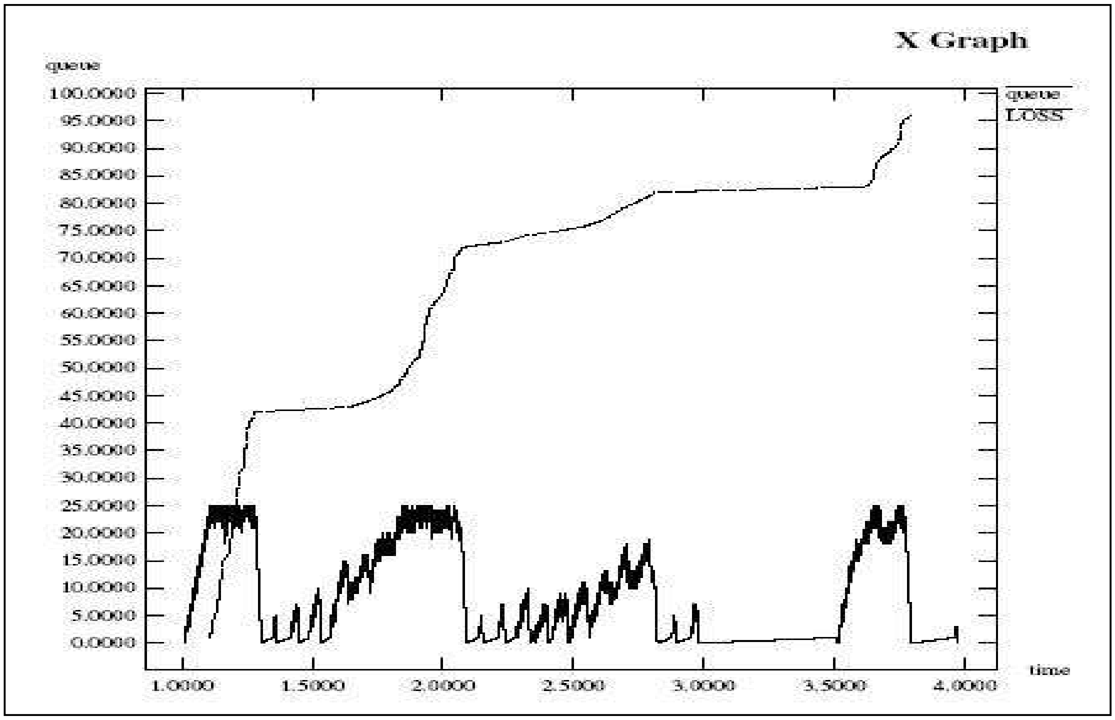


Figure 6.11 Experiment 2 xgraph output

In presences of AQM, RED and after implementing much the same experiment as Experiment 1 we found that we can control the buffer level better. But still from the NAM output we can see that noticeable UDP packets are dropping. Although in regard to managing the queue we have improved the situation in Experiment 2, we are not still delivering all the UDP packets, QoS for VOIP has not been greatly improved on this point.

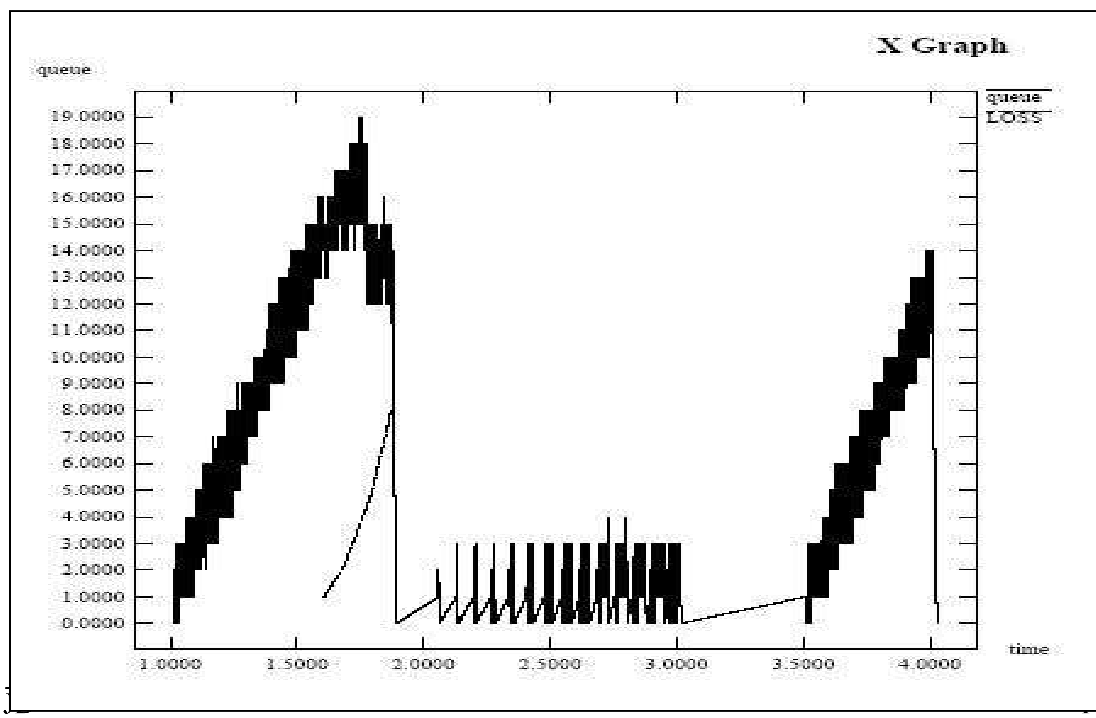


Figure 6.12 Experiment 2 xgraph output with tuning

Experiment 2B

Let's make an adjustment to Experiment 2 and see the consequences. In Experiment 2B we have changed the bottle neck bandwidth from 2 MB to 2.45 Mb. The delay is the same as before, 20ms. Introducing such a change we can see an enormous change in the queue & loss versus time graph shown in Figure 6.12.

The improvement we observe is a consequence of this change. Many other changes can be made. The factors that can be adjusted include delay, queue limit etc.

After doing more adjustment with bottleneck we can see in Figure 6.12, the queue length and the loss has been substantially decreased. Although the NAM output is showing us we have few UDP packets drop which will create an effect on our VOIP QoS.

The main dilemma with RED is not being able to differentiate between CBR & TCP packets. We would prefer no drops of CBR packets. And for that we introduce our third experiment, with Diffserv, which will solve this problem.

6.8.3. Experiment 3:

Experiment 3A

Our third experiment is a network test with the DiffServ system, as implemented in ns2. In this experiment the network consists of 6 nodes (s1, s2, n0, n1, e2, dest) as shown in Figure 6.13.

The core topology of Experiment 3 is the same as Experiments 1 & 2. The details of the additional links are as follows: All the Droptail, AQMs' have 1.5 Mbps of bandwidth and 10 ms of delay. Another two simplex links n2 and e2 use dsRED/core, and e2 and n2 dsRED/edge have 2 Mbps of bandwidth and 20 ms of delay.

The duplex links between s1 and n0, and s2 and n1, and e2 and dest have 10 Mbps of bandwidth and 1 ms of delay and use Droptail. The AQM's are in use as follows: The simplex link from n0 to n2 uses the dsRED/edge AQM, which marks the packets; the link from n2 and n0 uses dsRED/core which drops packets, in a manner similar to RED, except that voice & data packets are given different treatment, the link from n1 to n2 uses dsRED/edge; and the link from n2 to n1 uses dsRED/core. The link from n2 to e2 used dsRED/core.

The traffic conditions are basically the same as in experiments 1 & 2, however, the agents are attached at different locations because the network topology is different. The "TCP" agent is attached to s1, and a connection is established to a TCP "sink" agent attached to dest. As default, the maximum size of a packet that a "TCP" agent can generate is 1KByte. A TCP "sink" agent generates and sends ACK packets to the sender (TCP agent) and frees the received packets. A "UDP" agent that is attached to s2 is connected to a "null" agent attached to dest. A "null" agent just frees the packets received. A "FTP" and a "CBR" traffic generator are attached to "TCP" and "UDP" agents respectively, and the "CBR" is configured to generate 1 KByte packets at the rate of 1 Mbps. The "CBR" is set to start at 1.0 sec and stop at 3.0 sec, again "CBR" is set to start at 3.5 sec and stop at 4.0 sec and "FTP" is set to start at 0.1 sec and stop at 4.9 sec.

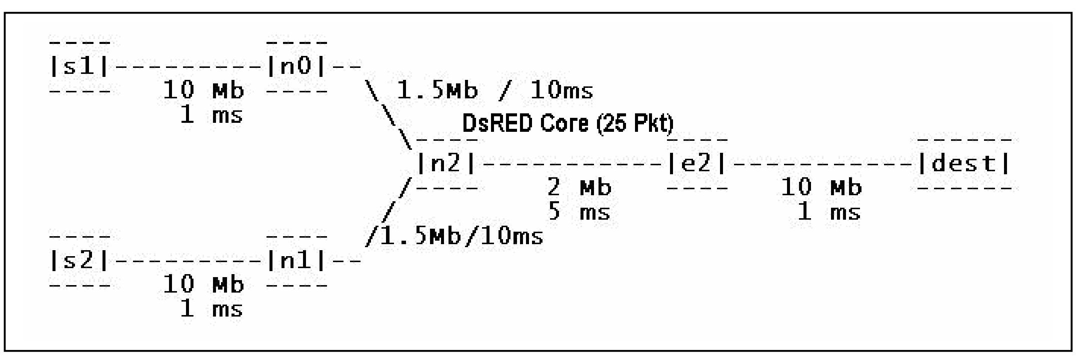


Figure 6.13 Experiment 3 diagram

In Experiment 3 simulation, using NAM output, we have noticed that when congestion occurs the TCP traffic backs off, as expected. The important factor that we need to focus on is that in this experiment the performance experienced by the UDP packets is very different. The following parameters are used to configure the Diffserv devices.

```
-----------------------------------------
set cir0 1000000
set cbs0 3000
set rate0 2000000
set cir1 1000000
set cbs1 10000
set rate1 3000000
-----------------------------------------
```

Here is the ns2 code which defines the topology in the network simulated in Example 3.

```
----------------------------------------------------------------------------
if ($diff) {
        $ns duplex-link $s1 $n0 10Mb 1ms DropTail
        $ns duplex-link $s2 $n1 10Mb 1ms DropTail
        $ns simplex-link $n0 $n2 1.5Mb 10ms dsRED/edge
        $ns simplex-link $n2 $n0 1.5Mb 10ms dsRED/core
        $ns simplex-link $n1 $n2 1.5Mb 10ms dsRED/edge
        $ns simplex-link $n2 $n1 1.5Mb 10ms dsRED/core

        $ns simplex-link $n2 $e2 2Mb 20ms dsRED/core
        $ns simplex-link $e2 $n2 2Mb 20ms dsRED/edge

        $ns simplex-link $n2 $e2 3Mb 20ms dsRED/core
        $ns simplex-link $e2 $n2 3Mb 20ms dsRED/edge
        $ns duplex-link $e2 $dest 10Mb 1ms DropTail
----------------------------------------------------------------------------

        -------------------------------------------------------
        $qn2e2 meanPktSize $packetSize
        $qn2e2 set numQueues_ 1
        $qn2e2 setNumPrec 2
        $qn2e2 addPHBEntry 10 0 0
        $qn2e2 addPHBEntry 11 0 1
        $qn2e2 configQ 0 0 5 15 0.1
```

```
#$qn2e2 configQ 0 0 15 25 0.05
$qn2e2 configQ 0 1 5 15 0.1
$qn2e2 set bytes_ false
$qn2e2 set queue_in_bytes_ false
--------------------------------------------------------
```

This block of the script configures all of the parameters for the edge queue between nodes n2 and e2. The meanPktSize command is required for the RED state variables to be calculated accurately. Because neither the scheduling or RED mode type is set, they default to Round Robin scheduling and RIO-C Active Queue Management. The CIR and CBS values used in the policies are the ones set at the beginning of the script. The addPHBEntry commands map each code point to a queue/precedence pair. Although each code point in this example maps to a unique queue/precedence pair, that need not be the case; multiple code points could receive identical treatment. Finally, the configQ commands set the RED parameters for each virtual queue. This command configures the RED parameters for one virtual queue.

```
---------------------------------------------
$dsredq configQ 0 1 10 20 0.10
---------------------------------------------
```

The above example specifies that physical queue 0 and virtual queue 1 has a minimum value of 10 packets, a maximum value of 20 packets, and a maximum drop probability value of 10% [50].

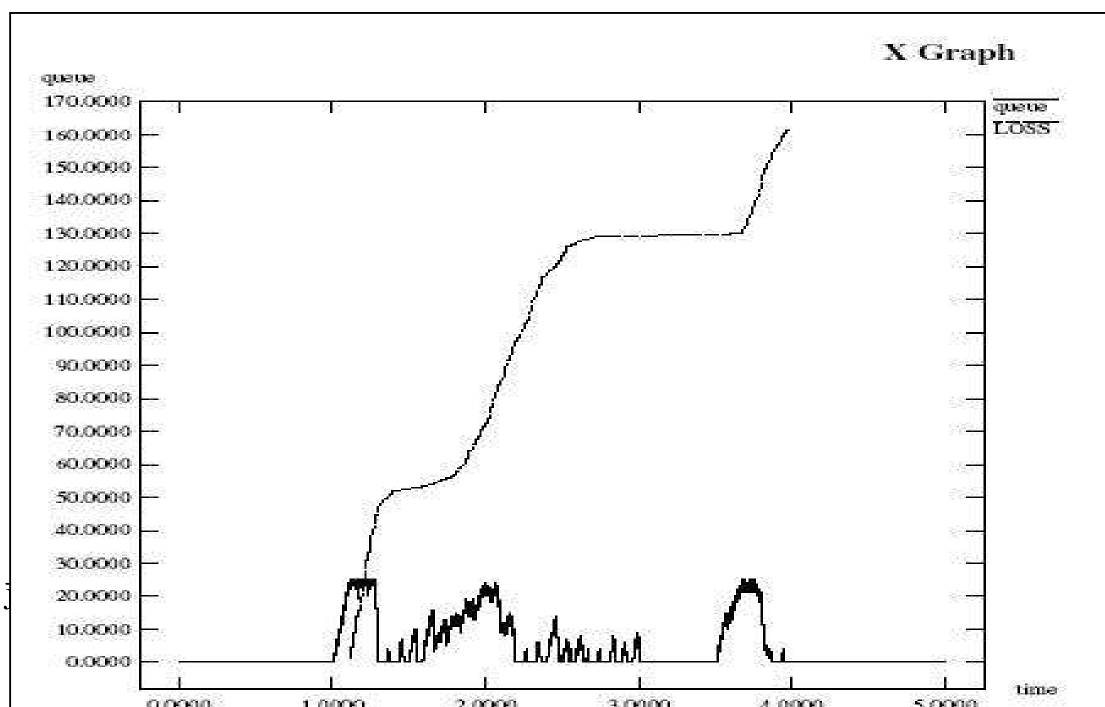


Figure 6.14 Experiment 3 xgraph output

In this experiment we have also experienced high level of packet loss. But using diffserv we have successfully eliminated the CBR loss; it is to zero. All lost packets are TCP. TCP backs off and retransmits the lost data so it will not be big problem. Careful choice of parameters will produce better performance than has been shown in Figure 6.14. Here we should mention that congestion in the Access network will normally occur for a modest proportion of the time. Once the CBR stops sending the packets, the TCP flow will transmit more rapidly. The performance of the Diffserv devices can be modified by adjusting the parameters. Beside this there are several parameters, like bottleneck bandwidth, delay etc., that can be modified to get abetter performance.

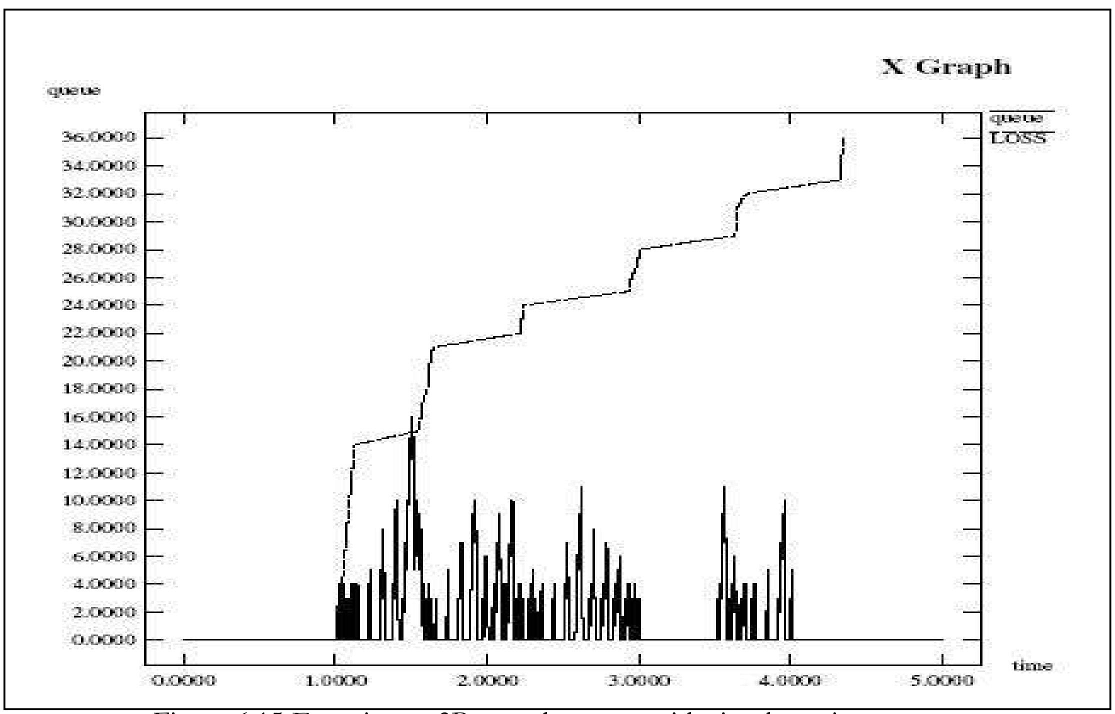


Figure 6.15 Experiment 3B xgraph output with simple tuning

Experiment 3B

We have done an adjustment on the existing diffserv code which has been projected on Figure 6.15. From the NAM output we have observed that not a single UDP packet has been dropped.

The main reason we have conducted this experiment with Diffserv is to help explain the somewhat high loss rate in Experiment 3. In Experiment 3B this has dropped considerably even though the only change has been increased buffer size (50 packets instead of 25). Because we have specified buffer lengths (queue limit) in packets rather than lengths, and UDP packets are shorter than TCP packets, the effective buffer size in Experiment 3 was less than in Experiment 2. This is probably the explanation of the high packet loss rate in Experiment 3. Experiment 3B helps to confirm this explanation.

Some more effective advance studies can open the door of more opportunities for controlling and handling the data traffic over the network. There are several parameter setting options available in the diffserv. Using those, the performance of the network can also be more modified. The parameters and tuning with variable codec can be effective future study and research topic.

Using Different system in our experiment 3 and implementing the same network we can see that the queue length and loss has been fairly controlled. Using the parameter setup option in Diffserv system we have gained the success of controlling UDP packet drops. Diffserv system has successfully stoped the UDP packet drops, which is important for VOIP QoS. The loss we are seeing in the graph, using NAM we have observed that all of the packets are from TCP.

It appears from Experiment 3 that using adjustment of parameters in the Diffserv system we can substantially increase the quality of service in the Access Network.

Chapter 7

Signalling

Chapter 7 has been dedicated towards the better understanding of signalling, Session Initiation Protocol (SIP) and also H.323. We have explained basic operation of SIP as well as made a comparison with the H.323 protocol in this Chapter.

7.1. Background

The meteoric ascent of the Internet as a rival to the circuit-switched telephone network has given rise to strong economic and technological reasons for convergence of services and architectures. In order to assimilate telephony services with the ubiquitous technology of IP, a signalling protocol is required to set up and tear down connections. A number of different solutions have been put forward, each coloured by their own priorities and interests. The Internet community wanted to introduce innovative services based on enhanced web-authoring tools like XML and more open, peer-to-peer protocols and call models. The IETF offered SIP. SIP was originally intended to create a mechanism for inviting people to large-scale multipoint conferences on the Internet Multicast Backbone (MBone). When the Multicast Backbone was introduced, IP telephony didn't really exist. It was soon realised that SIP could be used to set up point-to-point conferences - phone calls. The SIP approach exemplifies classic Internet-style innovation: build only what you need, to address only what is lacking in existing mechanisms. Because the SIP approach is modular and free from underlying protocol or architectural constraints, and because the protocols themselves are simple, SIP has caught on as an alternative to H.323 and to vendor-proprietary mechanisms for transporting Signalling System number 7 (SS7) protocols over IP [23].

7.2. The Basics

The Session Initiation Protocol (SIP) is a signalling 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 becomes possible, such as voice-enriched e-commerce, web page click-to-dial, Instant Messaging with buddy lists, and IP Centrex services.

Over the last couple of years, the Voice over IP community has adopted SIP as its protocol of choice for signalling. SIP is an RFC standard (RFC 3261) from the Internet Engineering Task Force (IETF) [36], the body responsible for administering and developing the mechanisms that comprise the Internet. SIP is still evolving and being extended as technology matures and SIP products are combined in the marketplace.

The IETF's philosophy is one of simplicity: specify only what you need to specify. SIP is very much of this mould; having been developed purely as a mechanism to establish sessions, it does not know about the details of a session, it just initiates, terminates and modifies sessions. This simplicity means that SIP scales, it is extensible, and it sits comfortably in different architectures and deployment scenarios.

SIP is a request-response protocol that closely resembles two other Internet protocols, HTTP and SMTP (the protocols that power the World Wide Web and email); consequently, SIP sits comfortably alongside Internet applications. Using SIP, telephony becomes another web application and integrates easily into other Internet services. SIP is a simple toolkit that service providers can use to build converged voice and multimedia services.

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), to authenticate users (RADIUS, DIAMETER), to provide directories (LDAP), to be able to guarantee voice quality (RSVP, YESSIR) and to inter-work with today's telephone network. Here we will only cover SIP [51].

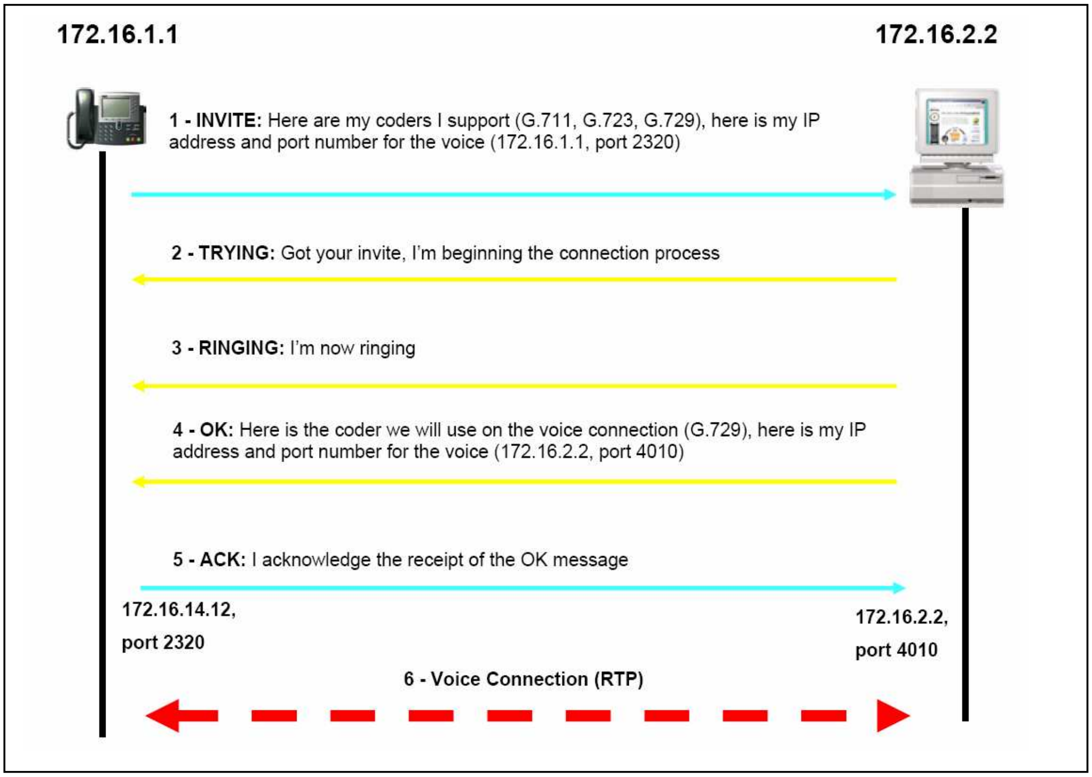


Figure 7.1 SIP basic session initiation flow between two SIP clients [52].

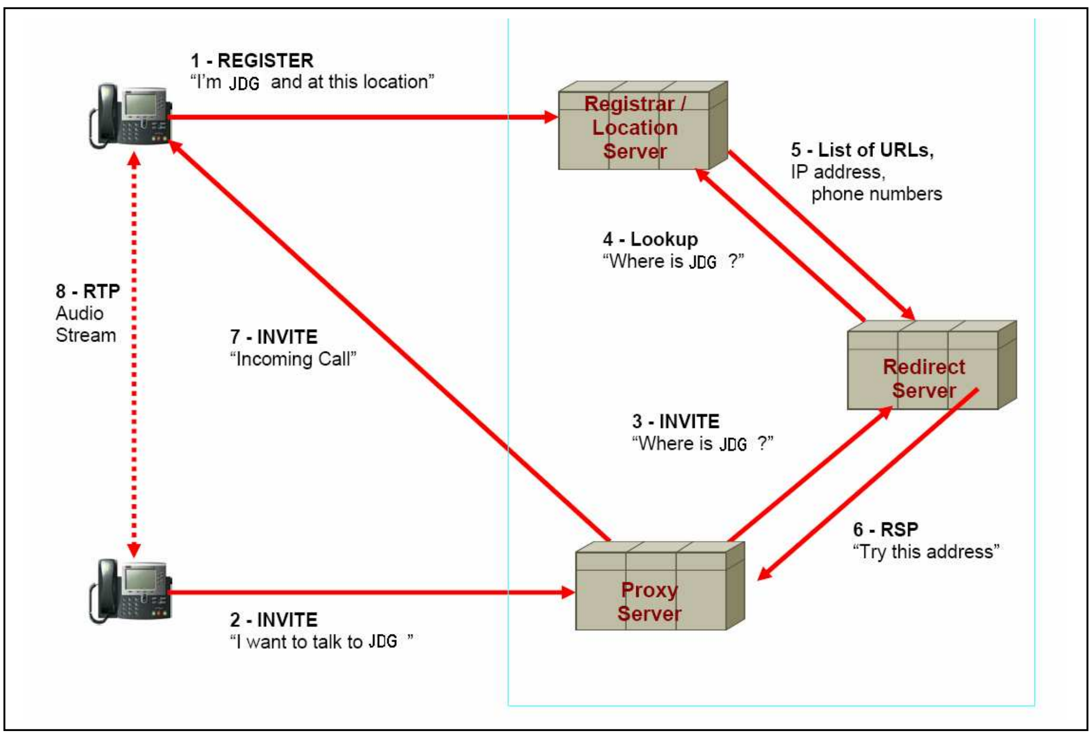


Figure 7.2 Session initiation utilizing SIP registration, proxy, and redirect services [52].

7.3. Architecture

There are two basic components within SIP:

The SIP user agent and The SIP network server.

The user agent is the end system component for the call and the SIP server is the network device that handles the signaling associated with multiple calls. The user agent itself has a client element, the User Agent Client (UAC) and a server element, the User Agent Server (UAS). The client element initiates the calls and the server element answers the calls. This allows peer-to-peer calls to be made using a client-server protocol. SIP user agents can be lightweight clients suitable for embedding in end-user devices such as mobile handsets or PDAs. Alternatively, they can be desktop applications that bind with other software applications such as contact managers. The main function of the SIP servers is to provide name resolution and user location, since the caller is unlikely to know the IP address or host name of the called party, and to pass on messages to other servers using next hop routing protocols [23].

SIP servers can operate in two different modes: Stateful and Stateless.

The difference between these modes is that a server in a stateful mode remembers the incoming requests it receives, along with the responses it sends back and the outgoing requests it sends on. A server acting in a stateless mode forgets all information once it has sent a request. These stateless servers are likely to be the backbone of the SIP infrastructure while stateful-mode servers are likely to be the local devices close to the user agents, controlling domains of users.

Other functions fulfilled by the SIP servers are re-direct and forking. A re-direct server receives requests but, rather than passing these onto the next server, it sends a response to the caller indicating the address for the called user. Forking is the ability to split or "fork" an incoming call so that several locations can ring at once. The first location to answer takes the call. These functions are further illustrated in SIP signalling. Together these components make up a basic SIP infrastructure. Application servers can sit above these components delivering SIP services to end-users. Application servers host service modules such as IM and presence, third party call control and user profiling. They also interact with other media servers and can be responsible for load balancing across a distributed architecture. These servers will also typically contain the management interface [53].

Custom services can be created by accessing subroutines in the application servers using APIs (Application Program Interfaces). When service modules are used in combination the service possibilities are vast.

SIP follows the client/server model that has proved so successful with the Internet. Backbone service providers can offer SIP infrastructure as part of their IP service offering to other service providers. These can, in turn, offer their own SIP services over this infrastructure in a ISP/ASP model. It is even possible for applications to be written by end-users in the same way that web applications are today.

7.4. SIP Signalling

From [54] we can get a better view of the total signalling process. SIP is based on the request-response paradigm. The following sequence is a simple example of a call set-up procedure:

1. To initiate a session, the caller (or User Agent Client) sends a request with the SIP URL of the called party.

2. If the client knows the location of the other party it can send the request directly to their IP address; if not, the client can send it to a locally configured SIP network server.

3. The server will attempt to resolve the called user's location and send the request to them. There are many ways it can do this, such as searching the DNS or accessing databases. Alternatively, the server may be a redirect server that may return the called user location to the calling client for it to try directly. During the course of locating a user, one SIP network server can proxy or redirect the call to additional servers until it arrives at one that definitely knows the IP address where the called user can be found.

4. Once found, the request is sent to the user and then several options arise. In the simplest case, the user's telephony client receives the request, that is, the user's phone rings. If the user takes the call, the client responds to the invitation with the designated capabilities of the client software and a connection is established. If the user declines the call, the session can be redirected to a voice mail server or to another user [54].

SIP has two additional significant features. The first is a stateful SIP server's ability to split or "fork" an incoming call so that several extensions can be rung at once. The first extension to answer takes the call. This feature is handy if a user is working between two locations (a lab and an office, for example), or where someone is ringing both a boss and their secretary.

The second significant feature is SIP's unique ability to return different media types within a single session. For example, a customer could call a travel agent, view video clips of possible holiday destinations, complete an on-line booking form and order currency - all within the same communication session.

7.4.1. SIP Methods

The commands that SIP uses are called methods. SIP defines the following methods:

SIP Method	Description
INVITE	Invites a user to a call
ACK	Used to facilitate reliable message exchange for INVITEs
BYE	Terminates a connection between users or declines a call
CANCEL	Terminates a request, or search, for a user
OPTIONS	Solicits information about a server's capabilities
REGISTER	Registers a user's current location
INFO	Used for mid-session signalling

Table7.1: Based on the TETF RFC 2543 [53]

7.4.2. Addressing and naming

To be able to locate and invite participants there has to be a way the called party could be addressed. The entities addressed by SIP are users at hosts, identified by a SIP URL. The SIP URL has an email-like identifier of the form user@host. Where the user part can be a user name, a telephone number or a conventional name. The host part is either a domain name or a numeric network address. In many cases a user's SIP URL could be guessed from the users email address. Examples of SIP URLs could be:

SIP:patrik@example.com

SIP: beagleboy@176.7.6.1

This URL may well be placed in a web page, so that clicking on the link, as in the case mail URLs, initiates a call to that address.

When using the email address SIP has to resolve the name@domain to user@host. This could lead to different addresses depending on time of the day, media to be used and so on [53].

7.4.3. Locating a Server

When a client wishes to send a request it first obtains the address of the participant to be contacted. If the address consists of a numeric IP address the client contact the SIP server there. Otherwise the address is of the form name@domain the client has to translate the domain part to an IP address where a server may be found. This is done with a DNS lookup. Once the IP address is found the request is sent using either UDP or TCP [53].

7.4.4. Locating a User

When the SIP server receives a request it has to locate the user in its domain. The user's location could be of different kind. He could for example be logged in at zero to many hosts or at a different domain. To find the user's actual location there is an outside SIP entity, a location server. Being asked for a user's location the location server returns a list of zero to many locations where the user could be found.

The location can dynamically be registered with the SIP server. The user may also install call handling features at the server. This is done sending a REGISTER request. If the location of the user was not found the server sends a response to the client indicating this. Otherwise the action taken by the server varies with the type of the SIP server:

Proxy Server: A SIP proxy server can send the request in sequence or in parallel to the locations listed.

Redirect Server: A SIP redirect server can return a response with the list placed in Contact headers. Then the client can send directly to the users' location(s).[53]

7.5. SIP and H.323

SIP is, more or less, equivalent to the Q.931 and H.225 components of H.323. These protocols are responsible for call setup and call signalling. Consequently, both SIP and H.323 can be used as signalling protocols in IP networks.

A comparison:

SIP	H. 323
PHILOSOPHY	
"New World" - a relative of Internet protocols - simple, open and horizontal	"Old World" - complex, deterministic and vertical
IETF	ITU
Carrier-class solution addressing the wide area	Borne of the LAN - focusing on enterprise conferencing priorities
CHARACTERISTICS	
A simple toolkit upon which smart clients and applications can be built. It re-uses Net elements (URLs, MIME and DNS)	H.323 specifies everything including the codec for the media and how you carry the packets in RTP
Leaves issues of reliability to underlying network	Assumes fallibility of network - an unnecessary overhead
SIP messages are formatted as text. (Text processing lies behind the web and email)	Binary format doesn't sit well with the internet - this adds complexity
SIP allows for standards-based extensions to perform specific functions.	Extensions are added by using vendor-specific non-standard elements
Hierarchical URL style addressing scheme that scales	Addressing scheme doesn't scale well
Minimal delay - simplified signalling scheme makes it faster	Possibilities of delay (up to 7 or 8 seconds!)
Slim and Pragmatic	The suite is too cumbersome to deploy easily
SERVICES	
Standard IP Centrex services	Standard IP Centrex services
Ability to 'fork' calls	Not possible in the existing standard
User profiling	-
'Unified messaging'	-
Presence management	-
Unique ability to mix media (e.g. IVR)	Cannot mix media within a session
URLs can be embedded in web browsers and email tools	H.323 has no URL format

Works smoothly with media gateway controllers controlling multiple gateways - crucial in a multi-operator environment	"Shoehorn" interworking with SS7 is problematic - H.323 has trouble connecting calls to and from PSTN endpoints
Seamless interaction with other media - services are only limited by the developers imagination	Services are nailed-down and constricted - voice only ceiling
STATUS	
Industry endorsed	Popularity due to the fact that it was the first set of agreed-upon standards
Many vendors developing products	The majority of existing IP telephony products rely on the H.323 suite

Table 7.2 : Based on The Session Initiation Protocol (SIP) preamble.[52]

Chapter 8

Conclusion and Further Development

8.1. Conclusion

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 signalling protocol. There are also engineering tradeoffs between delay and efficiency of bandwidth utilization.

Voice over Internet Protocol (VoIP) is a rapidly developing technology. Most cable, broadband and phone service providers are planning to start adding Internet telephony service to their standard packages. Despite hardware constraints, the operators have to tackle the problem of providing voice quality over the current Internet. VoIP is a time-sensitive application and requires real-time support for its quality of service (QoS) requirements. The traditional Internet, which uses a best-effort mechanism, fails to support the QoS requirement of most multimedia application like VoIP. Differentiated Service (Diffserv) is a scheme designed to support multimedia QoS requirements in a scalable manner. Two per-hop-behaviours (PHB): Expedited forwarding (EF) and Assured Forwarding (AF) have been defined for Diffserv. They are designed to provide low loss, low latency end-to-end service and assured bandwidth service respectively. In addition, AF is capable of being configured as a low latency service. In this thesis, simulations of VoIP using RED and Diffserv were carried out which gave some indication of the potential importance of DiffServ.

8.2. Future Work

There is a wide range of possibilities for future research in this area. It would be interesting to look at the performance of data traffic in our VOIP architecture. This would involve a study of TCP's ability to utilize the residual bandwidth (bandwidth unused by voice traffic) in the link.

Research could be undertaken into an Adaptive variable codec which could respond to the variation of the traffic flow. This will improve the QoS as well as allowing user to experience better quality when network conditions do allow.

Future research could be done on the automated DiffServ design and parameter setup so that DiffServ can be implemented from anywhere in the internet. DiffServ needs adaptive parameter configuration in order to achieve its potential.

Nomenclature (Classification)

ACD	Automatic call distributor.
API	Application Program Interfaces
ALG	Application level gateway.
ATM	Asynchronous transfer mode, a cell- switched communications technology.
BGP-4	Border gateway protocol 4, an interdomain routing protocol.
BRI	Basic rate interface (ATM interface, usually 144 kb/s).
Codec	Coder/decoder.
CR-LDP	Constrained route label distribution protocol.
CTI	Computer Telephony Integration
DiffServ	Differentiated services
DHCP	Dynamic host configuration protocol.
DSL	Digital subscriber line.
DTMF	Dual tone multiple frequency.
DTX	Discontinuous Transmission
EF	Expedited forwarding.
EFR	Enhanced Full Rate
FTP	File transfer protocol.
FXO	Foreign Exchange Office.
GII	Global Information Infrastructure
H.323	An ITU-T standard protocol suite for real-time communications over a packet network.
H.225	An ITU-T call signaling protocol (part of the H.323 suite).
H.235	An ITU-T security protocol (part of the H.323 suite).
H.245	An ITU-T capability exchange protocol (part of the H.323 suite).
HTTP	Hypertext transfer protocol.
IANA	Internet assigned numbers authority.

IETF	Internet engineering task force.
IntServ	Integrated services Internet.
ITAD	Internet telephony administrative domain.
ITSP	Internet telephony service provider.
ITU	International Telecommunications Union.
IP	Internet protocol.
IS-IS	Intermediate system-to-intermediate system, an interior routing protocol.
LAN	Local area network.
LDP	Label distribution protocol.
LS	Location server.
LSP	Label switched path.
LSR	Label switching router.
Megaco/H.248	An advanced media gateway control protocol standardized jointly by the IETF and the ITU-T.
MG	Media gateway.
MGCP	Media gateway control protocol.
MOS	Mean opinion score.
MPLS	Multiprotocol label switching.
MPLS-TE	MPLS with traffic engineering.
NAT	Network addresses translation.
OSPF	Open shortest path first, an interior routing protocol.
PBX	Private branch exchange, usually used on business premises to switch telephone calls.
PHB	Per hop behaviour.
PRI	Primary rate interface (ATM interface, usually 1.544 kb/s or 2.048 Mb/s).
PSTN	Public switched telephone network.
POTS	Plain Old Telephone System
RAS	Registration, admission and status. RAS channels are used in H.323 gatekeeper communications.
RFC	Request for comment, an approved IETF document.

RSVP	ReSerVation setup protocol.
RSVP-TE	RSVP with traffic engineering extensions.
RTP	Real-time transport protocol.
RTCP	Real-time control protocol.
RTSP	Real-time streaming protocol.
QoS	Quality of service.
SCN	Switched Circuit Network
SDP	Session description protocol.
SG	Signaling gateway.
SIP	Session initiation protocol.
SS7	Signaling system 7.
SCTP	Stream control transmission protocol.
SOHO	Small office/ home office.
TCP	Transmission control protocol.
TLS	Transport layer security.
TDM	Time-division multiplexing.
TRIP	Telephony routing over IP.
URI	Uniform resource identifier.
URL	Uniform resource locator.
UDP	User datagram protocol.
UAC	User Agent Client
UAS	User Agent Server
VAD	Voice activity detection
VOIP	Voice over Internet protocol.

Bibliography

1. *General Information on the Internet System Survey,* ITU-T Publication , ITU-T Publication , Jan 2000.

2. SearchNetworking, A whatis.com,

   http://searchnetworking.techtarget.com/gDefinition/0,294236,sid7_gci786677,00.html

3. *Quality of Service for Voice Over IP,* Cisco System, 2001, PP. 3-18,

   http://www.cisco.com/univercd/cc/td/doc/cisintwk/qosvoip.pdf.

4. Bur Goode *Voice Over Internet Protocol,* Proceeding of The IEEE, IEEE, Sep 2002, **90**(9).

5. ONI, 2001, The guide to Internet Protocol (IP) Telephony Optically Networked Issue

6. Ray Woabank, December 2000, Voice Over Internet Protocol - Fine Tuning Voice Over Packet Service, Butler group (Reaearch and advisory service).

7. Mark Stuward, 2004, *VOIP DEVELOPMENT,* Computer Journals.

8. Hong Goo Kang and Hong Kook Kim and Richard V Cox, *Improving The Transcoading Capability of Speech Coaders,* IEEE Transactions On Multimedia, Mar 2003, **5**(1) 1520-9210.

9. Autorité de régulation des telecommunications, May 2001, The French Telecommunication Regulator, French Telecommunication.

10. Koichi Asatani and Kogakuin and Yoichi Maeda, *Access Network Architectural Issues for Future Telecommunication Networks,* IEEE Communication Magazine, August 1998, PP. 110-114 http://www.comsoc.org/ci/private/1998/aug/pdf/Asatani.pdf.

11. *Transmission planning for voiceband services over hybrid Internet/PSTN connections: ITU-T Recommendation G.177,* SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS, ITU-T.

12. Amarnath Subramaniyam, *QoS of Telecommunication,* IEEE Communication Journal, June 2003, **4**(2) PP. 63-69.

13. *ITU-T PUBLICATION ,* ITU-T Proceeding , 2003 , http://www.itu.int/ITU-T/.

14. K Nichols and S Blake and F Baker and D Baker, *Definition of the Differentiated Services Field (DS Field),* Request for Comments, Dec 1998, **2474,** http://www.javvin.com/protocol/rfc2474.pdf.

15. S Brim and B Carpenter and F Le Faucheur, May 2000, Per Hop Behaviour Identification Codes, Network Working Group, Standards Track, http://www.faqs.org/rfcs/rfc2836.html.

16. S Blake and D Blake and M Carlson and E Davies and Z Wang and W Weiss Dec, 1998, An architecture for Differentiated Services, RFC Informational.

http://www.sci.usq.edu.au/courses/CSC8407/resources/rfc2475.txt

17. R Braden and D Clark and S Shenker, June 1994, Integrated Services in the Internet Architecture: an Overview,

http://www.sci.usq.edu.au/courses/CSC8407/resources/braden94integrated.pdf

18. Rosario G. Garroppon and Stefano Giordano and Michele Pagano, *Estimation of token bucket parameters for aggregated VoIP sources,* INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS, September 2002,

http://www3.interscience.wiley.com/cgi-bin/fulltext/102522521/PDFSTART

19. DiffServ The Scalable End-to-End QoS Model , March 2001 , Cisco Systems , http://www.cisco.com/warp/public/cc/pd/iosw/ioft/iofwft/prodlit/difse_wp.htm

20. A portal for the commercial development of SIP Session Initiation Protocol http://www.sipcenter.com/

21. I E T F 2003, Session Initiation Protocol and IP Telephony Signaling Protocol, Lucent Technologies.

22. ITU-T *Stability and Echo CCITT Recommendation G.131* 1988

http://www.itu.int/ITU-T/

23. Fredrik Fingal and Patrik Gustavsson, Feb 1999, A SIP of IP Telephony, Department of Communication Systems, Lund University.

24. RADCOM, H.323 Protocol Suite

www.protocols.com

25. Technology and Communication Answer dot Com,

http://www.answers.com/topic/echo-cancellation

26. One way Transmission time, ITU-T Recommendation G114, 1996, http://www.itu.int/ITU-T/

27. Goodman B *Internet telephony and modem delay.* Network, IEEE, June 1999, **13,**(3,) PP. 8-16, 0890-8044.

28. Carmen Moreno and Ascension Antolin and Fernando Diaz, *Recognizing Voice Over IP: A Robust Front-End for Speech Recognition on the World Wide Web,* IEEE Transactions On Multimedia, Jun 2001, **3,**(2 ,).

29. 2001 , VOIP Revolution , Vocaltec Newsletter , http://www.vocaltec.com/.

30. James J H and Bing Chen and Garrison L, *Implementing VOIP: a voice transmission performance progress report,* Communication Magazine, IEEE Jul 2004 **42**(7) PP. 36-41 0163-6804.

31. R J B Reynolds and A W Rix, *Quality VOIP - An Engineering Challenge,* BT Technol Journal, Apr 2001 **19**(2) .

32. Koistinen T, *IP Telephony,* Nokia Communication Journal, Oct 1998, PP. 11-18

33. Xerox Palo Alto Research Centre, History of Palo Alto Research Center, http://inventors.about.com/gi/dynamic/offsite.htm?site=http://www.parc.xerox.com/history.html

34. Duerinck, *The guide to Internet Protocol (IP) Telephony,* Optically Networked Issue , 2001.

35. Brian and Shankar, *Strategies for moving to IP enabled voice and data networks,* IEEE Communication Letters, 1999 **6**(6).

36. *Pulse code modulation (PCM) of voice frequencies,* ITU-T Recommendation G. 711 , NOV 1988 , http://www.itu.int/ITU-T/.

37. Antony Rix and Richard Reynolds and Mike Hollier, 2000, Perceptual measurement of end-to-end speech quality over audio and packetbased networks, BT Laboratories, Martlesham Heath, Ipswich IP5 3RE, United Kingdom.

http://www.psytechnics.com/downloads/1999-P02.pdf

38. V V Digalakis and L G Neumeyer and M Perakakis, *Quantization of cepstral parameters for speech recognition over the world wide web,* IEEE Selected areas of Communication, Jan 1999, **17,** PP. 82-90.

39. S Euler and J Zinke, 1994, *The Influence of speech coding algorithms on automatic speech recognition,* ICASSP, 1994, Australia

40. Wikimedia Foundation, Wikipedia , www.wikipedia.com

41. Christian Hoene and Berthold Rathke and Adam Wolisz, *On the Importance of a VOIP Packet,* Proceeding of ISCA Tutorial and Research Workshop on the Auditory Quality of Systems, Technical University of Berlin, Apr 2003.

42. ITU-T, *International telephone connections and circuits - Transmission plan aspects of special circuits*

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS, ITU-T Recommendation G.177 .

43. Runsheng Wang and Xiaouri Hu, *VOIP Development in Asia,* IEEE Computer Society, 2004, **04,** 0018-9162.

44. Anton Kos and Borut Klepec and Saso, 2001, Techniques for Performance Improvement of VOIP Applications, University of Ljubliana and Triglav Insurance Company.

45. Mona Habib and Nirmala Bulusu, Project Research paper, Computer Communication Magazine.

46. Khushboo Stephan, Mikedavis, *Signal Processing Challanges in Active Queue Management ,* IEEE Signal Processing Magazine , Sep 2004 , **04 ,** 1053-5888.

47. Michhele C Weigle and Kevin Jeffay and F Donelson Smith, November 2004, Delay-Based Early Congestion Detection and Adaptation in TCP: Impact on web performance http://www.cs.clemson.edu/~mweigle/papers/comcom04.pdf .

48. Octavio and Medina and Laurent Toutain, 2001, State of Art in DiffServ Technical Report, RTIPA.

49. Miroslaw Narbutt and Liam Murphy, *Improving Voice Over IP Subjective Call Quality,* IEEE Communication Letters, May 2004, **8**(5) ,1089-7798

50. Peter Pieda and Jeremy Ethridge and Mandeep Baines and Farhan Shallwani, A Network Simulator Differentiated Services Implementation, July 2000,

http://www-sop.inria.fr/mistral/personnel/Eitan.Altman/COURS-NS/DOC/DSnortel.pdf

51. Arnold J Rozer, SIP Information, June 2000, SIP CT Magazine

http://www.cs.columbia.edu/sip/articles/SIP_tutorial_CT_Magazine_June2000.pdf

52. The Session Initiation Protocol (SIP) preamble

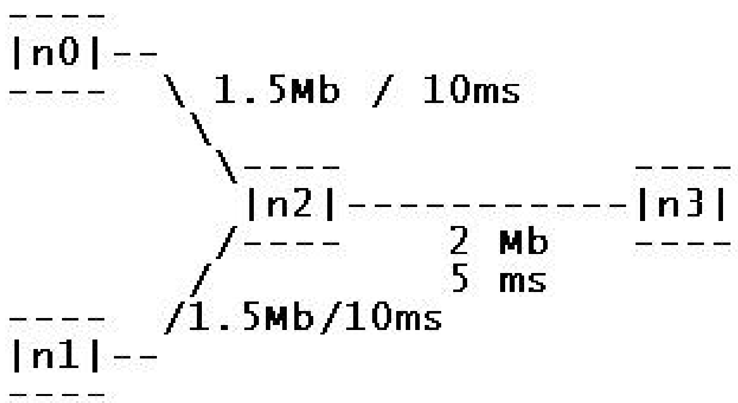
http://www.cs.columbia.edu/hgs/teaching/ais/slides/sip_long.pdf

53. M Handley and H Schulzrinne and E Schooler and J Rosenberg, March 1999, SIP: Session Initiation Protocol, Internet Engineering Task Force, RFC 2543, IETF.

54. Luca Veltri and Donald Papalilo and Cori Tel, *SIP Security Issues: The SIP Authentication Procedure and its Processing Load* IEEE network, Nov/Dec 2002, **02** 0890-8044.

Appendix

A. Experiment 1



Basic model with Droptail implementation in between n2 to n3

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Open the Trace file
set tf [open out.tr w]
$ns trace-all $tf

#Define a 'finish' procedure
proc finish {} {
        global ns nf
        $ns flush-trace
        #Close the NAM trace file
```

```
    close $nf

    #Execute NAM on the trace file
    exec /home/postgrad/dasgupta/ns/ns2.28/ns-allinone-2.28/nam-1.11/nam out.nam &
#change----------------------
global tchan_
  set awkCode {
      {
        if ($1 == "Q" && NF>2) {
            print $2, $3 >> "temp.q";
            set end $2
        }
        else if ($1 == "a" && NF>2)
        print $2, $3 >$ns> "temp.a";
      }
  }
  set f [open temp.queue w]

  if { [info exists tchan_] } {
      close $tchan_
  }
  exec rm -f temp.q temp.a
  exec touch temp.a temp.q

  exec awk $awkCode all.q

  puts $f \"queue
  exec cat temp.q >@ $f
  puts $f \n\"ave_queue
  exec cat temp.a >@ $f

  set awkCode1 {
      BEGIN{dropNo = 0;}
      {
```

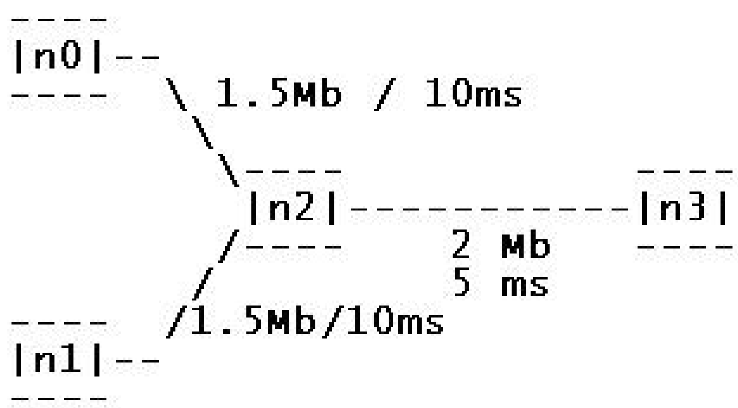
```
                if($1 == "d" && $3 == 2 && $4 == 3) {
                        dropNo = dropNo + 1;
                        print $2, dropNo >> "temp.l";
                }
        }
                }
        exec rm -f temp.l
        exec touch temp.l

        exec awk $awkCode1 out.tr
        puts $f \"LOSS
        exec cat temp.l >@ $f
   close $f
   exec /home/postgrad/dasgupta/ns/ns2.28/ns-allinone-2.28/xgraph-12.1/xgraph -bb -tk -x
time -y queue temp.queue &
#-----------------------------------------------------



exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 1.5Mb 10ms DropTail
$ns duplex-link $n1 $n2 1.5Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 20ms DropTail
```

```
#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 25

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

# (change) Tracing a queue
set q [[$ns link $n2 $n3] queue]
set tchan_ [open all.q w]
$q trace curq_
#$q trace ave_
$q attach $tchan_
#-----------------------------------
#Setup a TCP connection
set TCP [new Agent/TCP]
$TCP set class_ 2
$ns attach-agent $n0 $TCP
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $TCP $sink
$TCP set fid_ 1

#Setup a FTP over TCP connection
set FTP [new Application/FTP]
$FTP attach-agent $TCP
$FTP set type_ FTP
```

```
#Setup a UDP connection
set UDP [new Agent/UDP]
$ns attach-agent $n1 $UDP
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $UDP $null
$UDP set fid_ 2

#Setup a CBR over UDP connection
set CBR [new Application/Traffic/CBR]
$CBR attach-agent $UDP
$CBR set type_ CBR
$CBR set packet_size_ 128
$CBR set rate_ 1mb
$CBR set random_ false



#Schedule events for the CBR and FTP agents
$ns at 0.1 "$FTP start"
$ns at 2.0 "$CBR start"
$ns at 3.0 "$CBR stop"
$ns at 3.5 "$CBR start"
$ns at 4.0 "$CBR stop"
$ns at 4.9 "$FTP stop"



#Detach TCP and sink agents (not really necessary)
#$ns at 4.5 "$ns detach-agent $n0 $TCP ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$CBR set packet_size_]"
```

```
puts "CBR interval = [$CBR set interval_]"

#Run the simulation
$ns run
```

B. Experiment 2



Basic model with RED implementation in between n2 to n3

```
# Notes: A DS-RED script that uses CBR traffic agents and the Token Bucket policer.
# Through the arg we are controlling the individual diff and red test


#Find out if it is a DiffServ test
set diff $argc==1

if ($diff) {
        puts "Doing a DiffServ run"
}

#Create a simulator object
set ns [new Simulator]

#Giving priority to all packets
 set cir0  50000000
 set cbs0  2000000
 set cir1  100000000
 set cbs1  10000000
```

```
#Setting for a generic prority
# set cir0  500000
# set cbs0  20000
# set cir1  1000000
# set cbs1  100000

#Giving no priority at all
# set cir0  5
# set cbs0  2
# set cir1  1
# set cbs1  1
set rate0 1000000
set rate1 3000000

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Open the Trace file
set tf [open out.tr w]
$ns trace-all $tf

set packetSize 1000
# used in the diffServ queue setup



#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace
```

```
    #Close the NAM trace file
    close $nf

#Execute NAM on the trace file (Setup for common Uni computers)
# exec /usr/local/src/ns-allinone-2.28/nam-1.11/nam out.nam &
#Execute NAM on the trace file (Setup for my computers)
exec /home/postgrad/dasgupta/ns/ns2.28/ns-allinone-2.28/nam-1.11/nam out.nam &



#change (Sorting out data for drafting graph)
global tchan_
  set awkCode {
      {
        if ($1 == "Q" && NF>2) {
            print $2, $3 >> "temp.q";
            set end $2
        }
        else if ($1 == "a" && NF>2) {
          print $2, $3 >$ns> "temp.a";
        }
      }
  }

  set f [open temp.queue w]

  if { [info exists tchan_] } {
      close $tchan_
  }
  exec rm -f temp.q temp.a
  exec touch temp.a temp.q

  exec awk $awkCode all.q
```

```
    puts $f \"queue
    exec cat temp.q >@ $f
    puts $f \n\"ave_queue
    exec cat temp.a >@ $f


set awkCode1 {
        BEGIN{dropNo = 0;}
        {
            if($1 == "d" && $3 == 2 && $4 == 3) {
                dropNo = dropNo + 1;
                print $2, dropNo >> "temp.l";
            }
        }
       }
       exec rm -f temp.l
       exec touch temp.l

       exec awk $awkCode1 out.tr
       puts $f \"LOSS
       exec cat temp.l >@ $f
       close $f

#Execute XGRAPH on the trace file (Setup for common Uni computers)
#exec /usr/local/src/ns-allinone-2.28/bin/xgraph -bb -tk -x time -y queue temp.queue &

#Execute XGRAPH on the trace file (Setup for my computers)
exec /home/postgrad/dasgupta/ns/ns2.28/ns-allinone-2.28/xgraph-12.1/xgraph -bb -tk -x
time -y queue temp.queue &

#-------------------------------------------------------

exit 0
```

```
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

# create some extraa nodes for diffserv
if ($diff) {
  set s1 [$ns node]
  set s2 [$ns node]
  set e2 $n3
  set dest [$ns node]
}

#Create links between the nodes
if ($diff) {
        $ns duplex-link $s1 $n0 10Mb 1ms DropTail
        $ns duplex-link $s2 $n1 10Mb 1ms DropTail
        $ns simplex-link $n0 $n2 1.5Mb 10ms dsRED/edge
        $ns simplex-link $n2 $n0 1.5Mb 10ms dsRED/core
        $ns simplex-link $n1 $n2 1.5Mb 10ms dsRED/edge
        $ns simplex-link $n2 $n1 1.5Mb 10ms dsRED/core

   $ns simplex-link $n2 $e2 2Mb 20ms dsRED/core
   $ns simplex-link $e2 $n2 2Mb 20ms dsRED/edge

#Setup for higher bandwidth in bottleneck
#   $ns simplex-link $n2 $e2 3Mb 20ms dsRED/core
#   $ns simplex-link $e2 $n2 3Mb 20ms dsRED/edge

   $ns duplex-link $e2 $dest 10Mb 1ms DropTail
```

```
    #Set Queue Size of link (s1-n0) etc to 25
    $ns queue-limit $s1 $n0 25
    $ns queue-limit $s2 $n1 25
    $ns queue-limit $dest $e2 25

    set qn2e2 [[$ns link $n2 $e2] queue]
    set qe2n2 [[$ns link $e2 $n2] queue]

    set qn0n2 [[$ns link $n0 $n2] queue]
    set qn1n2 [[$ns link $n1 $n2] queue]
    set qn2n0 [[$ns link $n2 $n0] queue]
    set qn2n1 [[$ns link $n2 $n1] queue]

puts "Packet size is: "
puts $packetSize

# Set DS RED parameters from n0 to Core n2:
$qn0n2 meanPktSize $packetSize
$qn0n2 set numQueues_ 1
$qn0n2 setNumPrec 2
$qn0n2 addPolicyEntry [$s1 id] [$dest id] TokenBucket 10 $cir0 $cbs0
$qn0n2 addPolicyEntry [$s2 id] [$dest id] TokenBucket 10 $cir1 $cbs1
$qn0n2 addPolicerEntry TokenBucket 10 11
$qn0n2 addPHBEntry 10 0 0
$qn0n2 addPHBEntry 11 0 1
$qn0n2 configQ 0 0 25 50 0.1
$qn0n2 configQ 0 1 25 50 0.1
$qn0n2 set bytes_ false
$qn0n2 set queue_in_bytes_ false

# Set DS RED parameters from n1 to Core n2:
$qn1n2 meanPktSize $packetSize
$qn1n2 set numQueues_ 1
```

```
$qn1n2 setNumPrec 2
$qn1n2 addPolicyEntry [$s1 id] [$dest id] TokenBucket 10 $cir0 $cbs0
$qn1n2 addPolicyEntry [$s2 id] [$dest id] TokenBucket 10 $cir1 $cbs1
$qn1n2 addPolicerEntry TokenBucket 10 11
$qn1n2 addPHBEntry 10 0 0
$qn1n2 addPHBEntry 11 0 1
$qn1n2 configQ 0 0 25 50 0.1
$qn1n2 configQ 0 1 25 50 0.1
$qn1n2 set bytes_ false
$qn1n2 set queue_in_bytes_ false

# Set DS RED parameters from e2 to Core n2:
$qe2n2 meanPktSize $packetSize
$qe2n2 set numQueues_ 1
$qe2n2 setNumPrec 2
$qe2n2 addPolicyEntry [$s1 id] [$dest id] TokenBucket 10 $cir0 $cbs0
$qe2n2 addPolicyEntry [$s2 id] [$dest id] TokenBucket 10 $cir1 $cbs1
$qe2n2 addPolicyEntry [$dest id] [$s1 id] TokenBucket 10 $cir0 $cbs0
$qe2n2 addPolicyEntry [$dest id] [$s2 id] TokenBucket 10 $cir1 $cbs1
$qe2n2 addPolicerEntry TokenBucket 10 11
$qe2n2 addPHBEntry 10 0 0
$qe2n2 addPHBEntry 11 0 1
$qe2n2 configQ 0 0 25 50 0.1
$qe2n2 configQ 0 1 25 50 0.1
$qe2n2 set bytes_ false
$qe2n2 set queue_in_bytes_ false

# Set DS RED parameters from n2 to e2:
$qn2e2 meanPktSize $packetSize
$qn2e2 set numQueues_ 1
$qn2e2 setNumPrec 2
$qn2e2 addPHBEntry 10 0 0
$qn2e2 addPHBEntry 11 0 1
$qn2e2 configQ 0 0 5 15 0.1
```

```
#$qn2e2 configQ 0 0 15 25 0.05
$qn2e2 configQ 0 1 5 15 0.1
$qn2e2 set bytes_ false
$qn2e2 set queue_in_bytes_ false

# Set DS RED parameters from n2 to n0:
$qn2n0 meanPktSize $packetSize
$qn2n0 set numQueues_ 1
$qn2n0 setNumPrec 2
$qn2n0 addPHBEntry 10 0 0
$qn2n0 addPHBEntry 11 0 1
$qn2n0 configQ 0 0 5 15 0.1
$qn2n0 configQ 0 1 5 15 0.1
$qn2n0 set bytes_ false
$qn2n0 set queue_in_bytes_ false

# Set DS RED parameters from n2 to e2:
$qn2n1 meanPktSize $packetSize
$qn2n1 set numQueues_ 1
$qn2n1 setNumPrec 2
$qn2n1 addPHBEntry 10 0 0
$qn2n1 addPHBEntry 11 0 1
$qn2n1 configQ 0 0 5 15 0.1
$qn2n1 configQ 0 1 5 15 0.1
$qn2n1 set bytes_ false
$qn2n1 set queue_in_bytes_ false

   $ns queue-limit $n0 $n2 50
   $ns queue-limit $n1 $n2 50
   $ns queue-limit $n2 $e2 25
   $ns queue-limit $e2 $n2 50

#Give node position (for NAM)
$ns duplex-link-op $s1 $n0 orient right
```

```
$ns duplex-link-op $s2 $n1 orient right
$ns duplex-link-op $e2 $dest orient right
} else {
        $ns duplex-link $n0 $n2 1.5Mb 10ms DropTail
        $ns duplex-link $n1 $n2 1.5Mb 10ms DropTail

        #change
        $ns duplex-link $n2 $n3 2Mb 20ms RED
        #$ns duplex-link $n2 $n3 3Mb 20ms RED

  #Set Queue Size of link (n2-n3) to 10
  $ns queue-limit $n2 $n3 25
  $ns queue-limit $n3 $n2 25
  $ns queue-limit $n0 $n2 25
  $ns queue-limit $n1 $n2 25
  set redq [[$ns link $n2 $n3] queue]
  $redq set gentle_ false
  $redq set bytes_ false
  $redq set queue_in_bytes_ false
}

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

if ($diff) {} else {
  #Monitor the queue for link (n2-n3). (for NAM)
  $ns duplex-link-op $n2 $n3 queuePos 0.5
  #change
  $ns duplex-link-op $n3 $n2 queuePos 0.5
}

puts "Positions for NAM are set"
```

```
set lastDrops 0

proc queueTrace interval {
        global tchan_ ns n0 n1 n2 e2 lastDrops
        set redq [[$ns link $n2 $e2] queue]
        puts $tchan_ "Q [$ns now] [$redq getCurrent 0 ] "
        puts $tchan_ "a  [$ns now] [$redq getAverage 0 ] "
        puts $tchan_ "d  [$ns now] [expr [$redq getStat drops ] - $lastDrops] "
      set lastDrops [$redq getStat drops ]
        # puts $tchan_ "Q [$ns_ now] [$redq getCurrentV 0 0 ] "
        # puts $tchan_ "a  [$ns_ now] [$redq getAverageV 0 2 ] "
        $ns at [expr [$ns now] + $interval] "queueTrace $interval"
}

# Tracing a queue
   set redq [[$ns link $n2 $n3] queue]
   set tchan_ [open all.q w]
if ($diff) {
   queueTrace 0.002
} else {
   $redq trace curq_
   $redq trace ave_
   $redq attach $tchan_
}

#Setup a TCP connection
set TCP [new Agent/TCP]
$TCP set class_ 2
set sink [new Agent/TCPSink]
if ($diff) {
   $ns attach-agent $s1 $TCP
   $ns attach-agent $dest $sink
} else {
```

```
    $ns attach-agent $n0 $TCP
    $ns attach-agent $n3 $sink
}
$ns connect $TCP $sink
$TCP set fid_ 1

#Setup a FTP over TCP connection
set FTP [new Application/FTP]
$FTP attach-agent $TCP
$FTP set type_ FTP

#Setup a UDP connection
set UDP [new Agent/UDP]
if ($diff) {
    $ns attach-agent $s2 $UDP
} else {
    $ns attach-agent $n1 $UDP
}
set null [new Agent/Null]
if ($diff) {
    $ns attach-agent $dest $null
} else {
    $ns attach-agent $n3 $null
}
$ns connect $UDP $null
$UDP set fid_ 2

#Setup a CBR over UDP connection
set CBR [new Application/Traffic/CBR]
$CBR attach-agent $UDP
$CBR set type_ CBR
$CBR set packet_size_ 128
$CBR set rate_ 1mb
$CBR set random_ false
```

```
puts "About to schedule the traffic start and stop events"

#Schedule events for the CBR and FTP agents
$ns at 0.1 "$FTP start"
$ns at 1.0 "$CBR start"
$ns at 3.0 "$CBR stop"
$ns at 3.5 "$CBR start"
$ns at 4.0 "$CBR stop"
$ns at 4.9 "$FTP stop"

#Detach TCP and sink agents (not really necessary)
#$ns at 4.5 "$ns detach-agent $n0 $TCP ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$CBR set packet_size_]"
puts "CBR interval = [$CBR set interval_]"

#Run the simulation
$ns run
```

C. Experiment 3

```
----
|n0|--
----   \ 1.5Mb / 10ms
        \
         \----           ----
          |n2|-----------|n3|
         /----    2 Mb   ----
        /         5 ms
----   /1.5Mb/10ms
|n1|--
----
```

Basic model

```
----         ----
|s1|---------|n0|--
----   10 Mb ----  \ 1.5Mb / 10ms
        1 ms        \
                     \----       ----          ------
                      |n2|----------|e2|-----------|dest|
                     /----   2 Mb   ----   10 Mb   ------
                    /        5 ms           1 ms
----         ----  /1.5Mb/10ms
|s2|---------|n1|--
----   10 Mb ----
        1 ms
```

Model with DiffServ implementation in between n2 to e2

```
# Notes: A DS-RED script that uses CBR traffic agents and the Token Bucket policer.
# Through the arg we are controlling the individual diff and red test


#Find out if it is a DiffServ test
set diff $argc==1

if ($diff) {
	puts "Doing a DiffServ run"
}

#Create a simulator object
```

```
set ns [new Simulator]

#Giving priority to all packets
 set cir0  50000000
 set cbs0  2000000
 set cir1  100000000
 set cbs1  10000000

#Setting for a generic prority
# set cir0  500000
# set cbs0  20000
# set cir1  1000000
# set cbs1  100000

#Giving no priority at all
# set cir0  5
# set cbs0  2
# set cir1  1
# set cbs1  1
set rate0 1000000
set rate1 3000000

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Open the Trace file
set tf [open out.tr w]
$ns trace-all $tf
```

```
set packetSize 1000
# used in the diffServ queue setup


#Define a 'finish' procedure
proc finish {} {
      global ns nf
      $ns flush-trace

      #Close the NAM trace file
      close $nf

#Execute NAM on the trace file (Setup for common Uni computers)
# exec /usr/local/src/ns-allinone-2.28/nam-1.11/nam out.nam &
#Execute NAM on the trace file (Setup for my computers)
exec /home/postgrad/dasgupta/ns/ns2.28/ns-allinone-2.28/nam-1.11/nam out.nam &



#change (Sorting out data for drafting graph)
global tchan_
   set awkCode {
         {
            if ($1 == "Q" && NF>2) {
                  print $2, $3 >> "temp.q";
                  set end $2
            }
            else if ($1 == "a" && NF>2) {
               print $2, $3 >$ns> "temp.a";
            }
         }
   }

   set f [open temp.queue w]
```

```
    if { [info exists tchan_] } {
        close $tchan_
    }
    exec rm -f temp.q temp.a
    exec touch temp.a temp.q

    exec awk $awkCode all.q

    puts $f \"queue
    exec cat temp.q >@ $f
    puts $f \n\"ave_queue
    exec cat temp.a >@ $f



set awkCode1 {
        BEGIN{dropNo = 0;}
        {
            if($1 == "d" && $3 == 2 && $4 == 3) {
                dropNo = dropNo + 1;
                print $2, dropNo >> "temp.l";
            }
        }
    }
    exec rm -f temp.l
    exec touch temp.l

    exec awk $awkCode1 out.tr
    puts $f \"LOSS
    exec cat temp.l >@ $f
    close $f

#Execute XGRAPH on the trace file (Setup for common Uni computers)
#exec /usr/local/src/ns-allinone-2.28/bin/xgraph -bb -tk -x time -y queue temp.queue &
```

```
#Execute XGRAPH on the trace file (Setup for my computers)
exec /home/postgrad/dasgupta/ns/ns2.28/ns-allinone-2.28/xgraph-12.1/xgraph -bb -tk -x
time -y queue temp.queue &

#-----------------------------------------------------

exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

# create some extraa nodes for diffserv
if ($diff) {
  set s1 [$ns node]
  set s2 [$ns node]
  set e2 $n3
  set dest [$ns node]
}

#Create links between the nodes
if ($diff) {
        $ns duplex-link $s1 $n0 10Mb 1ms DropTail
        $ns duplex-link $s2 $n1 10Mb 1ms DropTail
        $ns simplex-link $n0 $n2 1.5Mb 10ms dsRED/edge
        $ns simplex-link $n2 $n0 1.5Mb 10ms dsRED/core
        $ns simplex-link $n1 $n2 1.5Mb 10ms dsRED/edge
        $ns simplex-link $n2 $n1 1.5Mb 10ms dsRED/core

   $ns simplex-link $n2 $e2 2Mb 20ms dsRED/core
```

```
    $ns simplex-link $e2 $n2 2Mb 20ms dsRED/edge

#Setup for higher bandwidth in bottleneck
#   $ns simplex-link $n2 $e2 3Mb 20ms dsRED/core
#   $ns simplex-link $e2 $n2 3Mb 20ms dsRED/edge

    $ns duplex-link $e2 $dest 10Mb 1ms DropTail


    #Set Queue Size of link (s1-n0) etc to 25
    $ns queue-limit $s1 $n0 25
    $ns queue-limit $s2 $n1 25
    $ns queue-limit $dest $e2 25

    set qn2e2 [[$ns link $n2 $e2] queue]
    set qe2n2 [[$ns link $e2 $n2] queue]

    set qn0n2 [[$ns link $n0 $n2] queue]
    set qn1n2 [[$ns link $n1 $n2] queue]
    set qn2n0 [[$ns link $n2 $n0] queue]
    set qn2n1 [[$ns link $n2 $n1] queue]

puts "Packet size is: "
puts $packetSize

# Set DS RED parameters from n0 to Core n2:
$qn0n2 meanPktSize $packetSize
$qn0n2 set numQueues_ 1
$qn0n2 setNumPrec 2
$qn0n2 addPolicyEntry [$s1 id] [$dest id] TokenBucket 10 $cir0 $cbs0
$qn0n2 addPolicyEntry [$s2 id] [$dest id] TokenBucket 10 $cir1 $cbs1
$qn0n2 addPolicerEntry TokenBucket 10 11
$qn0n2 addPHBEntry 10 0 0
$qn0n2 addPHBEntry 11 0 1
```

```
$qn0n2 configQ 0 0 25 50 0.1
$qn0n2 configQ 0 1 25 50 0.1
$qn0n2 set bytes_ false
$qn0n2 set queue_in_bytes_ false

# Set DS RED parameters from n1 to Core n2:
$qn1n2 meanPktSize $packetSize
$qn1n2 set numQueues_ 1
$qn1n2 setNumPrec 2
$qn1n2 addPolicyEntry [$s1 id] [$dest id] TokenBucket 10 $cir0 $cbs0
$qn1n2 addPolicyEntry [$s2 id] [$dest id] TokenBucket 10 $cir1 $cbs1
$qn1n2 addPolicerEntry TokenBucket 10 11
$qn1n2 addPHBEntry 10 0 0
$qn1n2 addPHBEntry 11 0 1
$qn1n2 configQ 0 0 25 50 0.1
$qn1n2 configQ 0 1 25 50 0.1
$qn1n2 set bytes_ false
$qn1n2 set queue_in_bytes_ false

# Set DS RED parameters from e2 to Core n2:
$qe2n2 meanPktSize $packetSize
$qe2n2 set numQueues_ 1
$qe2n2 setNumPrec 2
$qe2n2 addPolicyEntry [$s1 id] [$dest id] TokenBucket 10 $cir0 $cbs0
$qe2n2 addPolicyEntry [$s2 id] [$dest id] TokenBucket 10 $cir1 $cbs1
$qe2n2 addPolicyEntry [$dest id] [$s1 id] TokenBucket 10 $cir0 $cbs0
$qe2n2 addPolicyEntry [$dest id] [$s2 id] TokenBucket 10 $cir1 $cbs1
$qe2n2 addPolicerEntry TokenBucket 10 11
$qe2n2 addPHBEntry 10 0 0
$qe2n2 addPHBEntry 11 0 1
$qe2n2 configQ 0 0 25 50 0.1
$qe2n2 configQ 0 1 25 50 0.1
$qe2n2 set bytes_ false
$qe2n2 set queue_in_bytes_ false
```

```
# Set DS RED parameters from n2 to e2:
$qn2e2 meanPktSize $packetSize
$qn2e2 set numQueues_ 1
$qn2e2 setNumPrec 2
$qn2e2 addPHBEntry 10 0 0
$qn2e2 addPHBEntry 11 0 1
$qn2e2 configQ 0 0 5 15 0.1
#$qn2e2 configQ 0 0 15 25 0.05
$qn2e2 configQ 0 1 5 15 0.1
$qn2e2 set bytes_ false
$qn2e2 set queue_in_bytes_ false

# Set DS RED parameters from n2 to n0:
$qn2n0 meanPktSize $packetSize
$qn2n0 set numQueues_ 1
$qn2n0 setNumPrec 2
$qn2n0 addPHBEntry 10 0 0
$qn2n0 addPHBEntry 11 0 1
$qn2n0 configQ 0 0 5 15 0.1
$qn2n0 configQ 0 1 5 15 0.1
$qn2n0 set bytes_ false
$qn2n0 set queue_in_bytes_ false

# Set DS RED parameters from n2 to n1:
$qn2n1 meanPktSize $packetSize
$qn2n1 set numQueues_ 1
$qn2n1 setNumPrec 2
$qn2n1 addPHBEntry 10 0 0
$qn2n1 addPHBEntry 11 0 1
$qn2n1 configQ 0 0 5 15 0.1
$qn2n1 configQ 0 1 5 15 0.1
$qn2n1 set bytes_ false
$qn2n1 set queue_in_bytes_ false
```

```
    $ns queue-limit $n0 $n2 50
    $ns queue-limit $n1 $n2 50
    $ns queue-limit $n2 $e2 25
    $ns queue-limit $e2 $n2 50

#Give node position (for NAM)
$ns duplex-link-op $s1 $n0 orient right
$ns duplex-link-op $s2 $n1 orient right
$ns duplex-link-op $e2 $dest orient right
} else {
        $ns duplex-link $n0 $n2 1.5Mb 10ms DropTail
        $ns duplex-link $n1 $n2 1.5Mb 10ms DropTail

        #change
        $ns duplex-link $n2 $n3 2Mb 20ms RED
        #$ns duplex-link $n2 $n3 3Mb 20ms RED

    #Set Queue Size of link (n2-n3) to 10
    $ns queue-limit $n2 $n3 25
    $ns queue-limit $n3 $n2 25
    $ns queue-limit $n0 $n2 25
    $ns queue-limit $n1 $n2 25
    set redq [[$ns link $n2 $n3] queue]
    $redq set gentle_ false
    $redq set bytes_ false
    $redq set queue_in_bytes_ false
}

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
```

```
if ($diff) {} else {
   #Monitor the queue for link (n2-n3). (for NAM)
   $ns duplex-link-op $n2 $n3 queuePos 0.5
   #change
   $ns duplex-link-op $n3 $n2 queuePos 0.5
}

puts "Positions for NAM are set"

set lastDrops 0

proc queueTrace interval {
         global tchan_ ns n0 n1 n2 e2 lastDrops
         set redq [[$ns link $n2 $e2] queue]
         puts $tchan_ "Q [$ns now] [$redq getCurrent 0 ] "
         puts $tchan_ "a  [$ns now] [$redq getAverage 0 ] "
         puts $tchan_ "d  [$ns now] [expr [$redq getStat drops ] - $lastDrops] "
      set lastDrops [$redq getStat drops ]
         # puts $tchan_ "Q [$ns_ now] [$redq getCurrentV 0 0 ] "
         # puts $tchan_ "a  [$ns_ now] [$redq getAverageV 0 2 ] "
         $ns at [expr [$ns now] + $interval] "queueTrace $interval"
}

# Tracing a queue
   set redq [[$ns link $n2 $n3] queue]
   set tchan_ [open all.q w]
if ($diff) {
   queueTrace 0.002
} else {
   $redq trace curq_
   $redq trace ave_
   $redq attach $tchan_
}
```

```
#Setup a TCP connection
set TCP [new Agent/TCP]
$TCP set class_ 2
set sink [new Agent/TCPSink]
if ($diff) {
   $ns attach-agent $s1 $TCP
   $ns attach-agent $dest $sink
} else {
   $ns attach-agent $n0 $TCP
   $ns attach-agent $n3 $sink
}
$ns connect $TCP $sink
$TCP set fid_ 1

#Setup a FTP over TCP connection
set FTP [new Application/FTP]
$FTP attach-agent $TCP
$FTP set type_ FTP

#Setup a UDP connection
set UDP [new Agent/UDP]
if ($diff) {
   $ns attach-agent $s2 $UDP
} else {
   $ns attach-agent $n1 $UDP
}
set null [new Agent/Null]
if ($diff) {
   $ns attach-agent $dest $null
} else {
   $ns attach-agent $n3 $null
}
$ns connect $UDP $null
$UDP set fid_ 2
```

```
#Setup a CBR over UDP connection
set CBR [new Application/Traffic/CBR]
$CBR attach-agent $UDP
$CBR set type_ CBR
$CBR set packet_size_ 128
$CBR set rate_ 1mb
$CBR set random_ false

puts "About to schedule the traffic start and stop events"

#Schedule events for the CBR and FTP agents
$ns at 0.1 "$FTP start"
$ns at 1.0 "$CBR start"
$ns at 3.0 "$CBR stop"
$ns at 3.5 "$CBR start"
$ns at 4.0 "$CBR stop"
$ns at 4.9 "$FTP stop"

#Detach TCP and sink agents (not really necessary)
#$ns at 4.5 "$ns detach-agent $n0 $TCP ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$CBR set packet_size_]"
puts "CBR interval = [$CBR set interval_]"

#Run the simulation
$ns run
```