Voice over IP: An Engineering Analysis

By

Karin Sundstrom

A Thesis
Submitted to the Faculty of Graduate Studies
In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba

© September, 1999

0-612-51804-3

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES
*****
COPYRIGHT PERMISSION PAGE

VOICE OVER IP: AN ENGINEERING ANALYSIS

BY

KARIN SUNDSTROM

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University

of Manitoba in partial fulfillment of the requirements of the degree

of

MASTER OF SCIENCE

Acknowledgements

Many thanks go to all those who were instrumental in the development of this thesis. First, I must thank my advisor, Professor Robert McLeod, for accepting me as a graduate student, and for introducing me to such an interesting subject area. Second I would like to thank Dr. Jose Rueda and Dr. Clint Gibler for their assistance in developing the project and accepting me as a student at TR*Labs*. Special thanks to Dr. Rueda for overseeing the all the details of the project at TR*Labs*.

I would also like to thank Professor Attahiru Alfa and Dr. Jeff Diamond for their assistance with developing the models. Thanks also go to the staff and students at TR*Labs*, particularly Len Dacombe, Dan Erickson, and Glenda Stark.

Finally, I would like to thank my family for their encouragement and support. In particular I would like to thank my mother and grandmother for editing the thesis and greatly improving its quality, my father for constantly nagging me to "get it done", and my fiancé for seeing me safely through another adventure.

Table of Contents

List of Figures

List of Tables

Abstract

VoIP is a rapidly emerging Internet service. It is expected that by 2004 10% of all voice calls in North America will be VoIP calls. Performance metrics and measuring methodology need to be developed so that the performance of the service can be assessed. From the customer's perspective, the service needs to be evaluated against the level of service achievable with conventional telephone networks. From a service providers' perspective, the performance of VoIP service needs to be evaluated for the purpose of network planning and eventually billing.

This thesis studies the performance of VoIP systems. Metrics for measuring the performance of VoIP systems, drawn from a combination of conventional telephone system performance metrics and Internet performance metrics, are presented. Methodologies for measuring the performance of VoIP systems, as well as expected and measured VoIP performance measurements are also developed.

Simulation is a useful tool for planning a network and gauging its performance. This thesis develops VoIP speaker activity models that can be used in a network simulator for these purposes.

Based on the measured values of the VoIP performance metrics, it will be shown that the current Internet infrastructure is not able to deliver toll quality voice service. Future architectures that may enable toll quality voice on the Internet conclude this thesis.

Index of Terms

CRTC	Canadian Radio-television and Telecommunications Commission
FCC	Federal Communications Commission
IETF	Internet Engineering Task Force
IMTC	Internet Multimedia and Teleconferencing Consortium
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPng	Internet Protocol Next Generation
ITU-T	International Telecommunication Union Telecommunication Sector
QoS	Quality of Service
RTCP	Real-Time Control Protocol
RSVP	Resource Reservation Setup Protocol
RTP	Real-time Transport Protocol
TCP	Transmission Control Protocol
TOS	Type of Service
UDP	User Datagram Protocol
VoIP	Voice over IP

1 Introduction

1.1 History of Internet Telephony

In March of 1995, an Israel-based company named VocalTec released a product that was to become the focus of much debate within both the Internet community and the telecommunications industry over the next 4 years. Internet Phone, the first commercially available product of its kind, was capable of communicating voice signals over the Internet in real-time. The technology commercialized by VocalTec is now known as Internet telephony or Voice over IP (VoIP).

The Internet telephony market has exploded since the introduction of Internet Phone, and still continues to grow to this date. It has been estimated that by the year 2004 Internet Telephony will be a $3-4 billion industry [1] and will carry 10% of the voice traffic in the US [1].

The release of VoIP software applications peaked in 1996 when, following VocalTec's lead, not fewer than 20 companies released their version of the *killer application*. These software application ran on user's PCs, MACs, or UNIX workstations and offered free voice communication between any two Internet connected terminals in the world.

1.2 Standards in Internet Telephony

The industry grew very quickly, but without a standards body in place, different vendor techniques for compression, address translation, and signaling were used in each implementation. The lack of standardization resulted in the inability of Internet telephones to inter-operate. At first users were only able to communicate if they were running the same Internet telephony application.

Communication between two computer users had limited practical potential. While growing at an exponential rate, the Internet was still not ubiquitous, unlike the conventional telephone network. What was required was the ability to reach any conventional telephone while still using the Internet as the free carrier.

Realizing that without standardization the technology would forever be relegated to hobbyist status, a group of vendors formed a coalition with the intention of developing a standard communication protocol. The requirements of the protocol were to provide inter-operability between different Internet telephone software applications and equipment, as well as the ability to inter-operate between Internet telephones and terminals on non-IP networks. The first standard signaling communication protocol was developed by the ITU-T (H.323). The IMTC has adopted H.323 as the interoperability standard for communication between VoIP application software as well as inter-terminal communication. Since H.323, several other standard signaling protocols have been proposed (SIP, MGCP, etc.), but H.323 remains the only protocol to be implemented and sold commercially on a large scale.

1.3 VoIP Services

VoIP no longer means merely software applications on end user workstations. Today a wide range of VoIP services are available. These are beneficial to both the user and the VoIP service provider. From a user's perspective, the primary service VoIP offers is low cost voice calls. There are, however, a number of other services and features that add value to VoIP.

The user interface is probably the most evident value added feature that Internet telephony can offer from a customer's perspective. The user interface on most traditional telephony handsets consists of a key pad. Liquid crystal displays are available on some phones, but these displays cannot compare to the GUI that can be offered to PC VoIP users.

Multimedia services are also commonly bundled with software based VoIP applications. For instance, VoIP applications frequently bundle a white board tool and a text-based chat tool to complement voice communication. Many applications provide video communication as well.

A service that is unique to VoIP applications is the ability to dynamically indicate the current speaker in a multiparty VoIP call. Participants in a traditional conference call can be identified, but no distinction can be made as to the identity of the current speaker. The current speaker in an Internet telephony call can easily be identified and displayed to the other conference participants.

From a service provider's perspective, the primary advantage of Internet telephony is that one network can be used to provide voice, data, multimedia, and call signaling services. This may provide operational cost savings, as it costs less to maintain one network than several. In addition, IP based networks are more efficient in terms of bandwidth utilization than traditional telephone networks. A voice call in a traditional telephone network occupies 64 kbps. A voice call made in a VoIP environment can be compressed to consume only a fraction of this bandwidth. Finally, the service provider can leverage the enhanced user interface available in VoIP systems to create advanced services not available in traditional networks (for example, integrated voice and email utilities, or detailed call display information).

1.4 Comparison Between Traditional and VoIP Networks

The differences in the features and services that can be offered to VoIP and conventional telephony users stems from differences between the two underlying networks. The intelligence in a VoIP system is located at the end-stations. In a conventional telephony network it is located within the network infrastructure. This allows enhanced services and features to exist in VoIP systems. For example, end points in a traditional telephone network are able to dynamically adjust the bit rate or compression algorithm of the codec to suit the current network characteristics. While usually high, the quality in traditional telephone calls cannot be adjusted in the event of a bad connection, or to select a lesser quality for a lower price. Moving the intelligence to the end stations also allows Internet telephony endpoints to encrypt the data to provide private and secure voice communications.

Another fundamental difference between the two networks is the switching technologies. VoIP systems are packet switched networks, while traditional telephony networks are circuit switched networks. Packet switching permits multiplexing which makes more efficient use of bandwidth than does a circuit switched network.

Signaling paths in VoIP systems also differ from those of traditional networks. In traditional networks, a separate network is often used for the signaling path (e.g. SS7). In VoIP systems, a single network is used for all services (voice, video, data, and control). This may provide cost savings to the operator.

The nature of call routing differs between conventional and VoIP systems. Routing occurs as the call is being placed. In a traditional telephony network. the path for the call is established before the call is placed and is maintained for the duration of the call. This requires state information to be maintained in each switch along the path from source to receiver.

Finally, the networks differ in terms of granting access to network resources. In a traditional telephony call. either the entire set of resources required to complete the call is allocated (i.e. 64 kbps bandwidth) or the call is not completed. In VoIP systems, however. the call is placed whether or not the network has sufficient resources.

1.5 VoIP Performance

Although the industry continues to expand, VoIP is no longer considered the *killer application* for the Internet. The success of VoIP has been limited in part due to the poor performance of IP networks.

VoIP quality metrics include latency, jitter, packet loss, and the quality of the compression algorithm. Chapter 4 examines the requirements for toll quality voice. the factors that affect voice quality in IP networks, and the level of quality that can currently be delivered by IP networks.

1.6 VoIP Service Architectures

QoS mechanisms must be established before VoIP can compete with traditional voice networks in terms of quality.

Two main classes of architectures can be employed to provide QoS in IP based networks: (i) QoS on a per packet basis, and (ii) QoS for a connection or stream of packets. Architectures that achieve QoS on a per packet basis include those based on the TOS field in the IPv4 header, those based on the *Traffic Class* field of the IPv6 header, and those based on the Differentiated Services architecture.

Architectures that provide QoS to a stream of packets fall into the category of Integrated Services architectures. Integrated services can be implemented in IPv6 by the *Flow Label* field in the IPv6 header, or by the RSVP protocol in IPv4 networks. Both QoS approaches have advantages and disadvantages in terms of Internet telephony. These are discussed in Chapter 5.

1.7 Internet Telephony Regulation

Shortly following the release of the first VoIP application, telecommunication service providers claimed that Internet telephony was a telecommunications service and should be regulated as such. After several years of debate on this issue, both the Canadian and the US telecommunications regulatory bodies, the CRTC and the FCC respectively, decided that the technology was too premature to be subjected to regulation at this point [2] [3]. The regulatory issue will be reexamined as the technology matures, however. In fact, several US states are already independently re-examining the issue to determine whether charges will be imposed at the state level [3].

1.8 Justifying the Business Case for VoIP

Telecommunications companies have viewed Internet telephony as a business threat. NetGen telcos were able to offer local and long distance voice services at a much reduced price to the consumer over traditional telephony service providers. This fear resulted in the scramble for Internet telephony regulation, somewhat justified by predictions such as 10% of all voice calls in the US will be Internet telephony calls by 2004 [1]. Internet telephony has also been viewed as a threat as it is able to add many value added data services to accompany the basic voice service.

It is becoming increasingly difficult, however, to justify Internet telephony as an alternative voice service based on cost alone. Many bulk carriers offer very competitive pricing. For instance, with 10-10-321, the cost of calling from Manhattan to San Francisco is $0.15 for the first minute and $0.065 for additional minutes (for calls over 10 minutes). All state-to-state calls using 10-10-345 are $0.10 per minute. 10-10-275 charges $0.05 per minute state-to-state and $0.09 per minute within a state. Using Net2Phone, a popular NextGen telco, US domestic calls are $0.049 per minute, and calls within Canada are $0.10 per minute.

Savings that Internet telephony can offer on domestic calls has diminished significantly from when the first Internet phone reached the market. Furthermore, when considering other factors such as quality of service and the ubiquitous nature of the traditional telephony network, Internet telephony is difficult to justify.

However, there are two business models where Internet telephony may still be able to compete: (i) the international or overseas long distance market, and (ii) inter-office communication. Table 1 shows the price per minute for long distance and overseas calls through Net2Phone [4] compared with three popular bulk long distance carriers: 10-10-321 [5], 10-10-345 [6], and 10-10-275 [7]. For this illustration, the international call originates in the US. As shown in Table 1, Internet telephony can offer savings on long distance or overseas calls in most cases. The price per minute shown for the 10-10-xxx carriers is their best case savings plan (i.e. the cost of the first minute, or the cost per minute if the total length of the call exceeds some minimum time). The cost per minute may increase with the bulk carriers after the first minute, while the cost per minute with Net2Phone remains the same regardless of time of day or length of the call.

Table 1: Cost per Minute Comparison of VoIP with Bulk Carriers

Destination	Carrier			
	Net2Phone	**10-10-321**	**10-10-345**	**10-10-275**
Canada	$0.10	$0.10	$0.10	$0.07
Mexico City	$0.25	$0.44	$0.29	$0.15
Israel	$0.17	$0.77	$0.19	$0.18
Hong Kong	$0.19	$0.39	$0.17	$0.21
North Korea	$0.94	$2.72	$3.94	NA
South Korea	$0.24	$0.36	$0.25	$0.32
Japan	$0.17	$0.34	$0.18	$0.19
Philippines	$0.36	$0.85	$0.42	$0.38
Russia	$0.47	$1.20	$0.38	$0.33
India	$0.79	$1.25	$1.05	$0.67
Malaysia	$0.27	$0.72	$1.57	$0.31

The second application of Internet telephony that may offer savings over the traditional telephone network is inter-office communications. Assuming the network infrastructure already exists for data communications, the only investment required to deliver VoIP is an upgrade to the network equipment that facilities voice communication. After the initial investment, voice may be carried for free within the data network. Architectures to deliver VoIP service are discussed in Chapter 5.

1.9 Speaker Activity Models for Internet Telephony

Traditional telephone conversation models have been found to be inappropriate for modeling Internet telephony traffic sources in terms of states, state transitions, as well as the distribution of state holding times. Based on the measurements of several Internet telephone conversations, this thesis presents models that better represent speaker activity in an Internet telephone call. The models are presented in Chapter 3.

1.10 Sending Voice over IP Networks

This section provides a brief introduction to Internet telephony technology and explains how voice is transmitted over the Internet.

The first step in the process of transmitting voice over an IP network is to convert the analogue voice signal to a digital form. All VoIP terminal equipment must have an AD/DA converter. In most home VoIP systems, the AD/DA equipment is the PC sound card. Non-desktop VoIP equipment must also have this capability, but most likely the AD/DA converter will exist as a specialized piece of equipment. An input device to the AD converter is also required. This may be a microphone in a desktop Internet phone, or a special purpose handset.

In either case, as the user speaks into the input device, the input equipment converts the sound waves into a continuos electrical voltage signal. The A/D equipment next samples and digitizes the analogue signal, converting it from analogue to digital form so it can be processed by the digital equipment.

Human voice roughly lies in the range of 300 Hz 4000 Hz [8]. According to the Nyquist Theorem, the analogue voice signal must therefore be sampled at a rate of 8000 Hz (twice the maximum frequency) in order retain the original signal information [8]. If each voice sample is 8 bits wide, sampling at a rate of 8000 samples/second requires a bandwidth of 64 Kbps.

Most dial-up Internet connections are limited to a bandwidth between 14.4 K bps and 56 K bps. To transmit the 64 kbps voice signal in this limited range, the signal is usually compressed. Even in networks with higher data rates (e.g. 10Base-T network), compression is usually performed in order to conserve network bandwidth.

In desktop VoIP applications, compression is usually performed in software. Compression is a computationally intensive operation and therefore many VoIP products now perform compression using dedicated hardware. Some vendors use their own proprietary compression scheme (e.g. Voxware), but thanks to the standards effort, most products now use a standard algorithm which facilitates interoperability.

Regardless of whether compression is performed in hardware or software, the compression algorithm submits data to the IP stack of the VoIP product. The voice data is grouped into packets, and routed over the IP network to the receiving station.

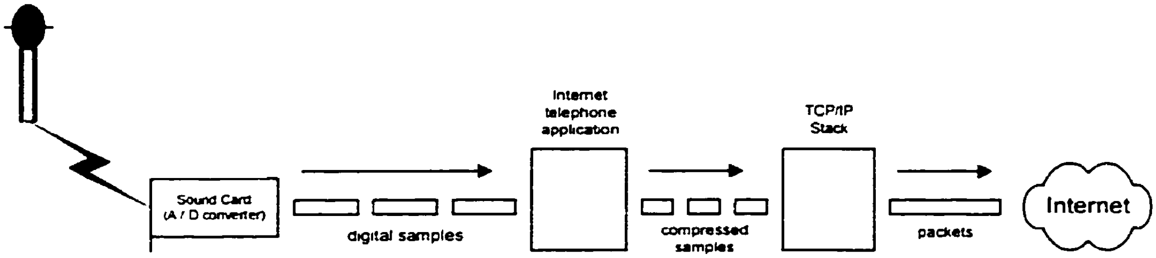


Figure 1: Sending Voice over the Internet

After a packet is received, it may be placed in a jitter-buffer before it is decompressed. The purpose of the jitter buffer is to smooth out the variable delays between packets which were incurred along the path from source to destination. Once the packet has cleared the jitter buffer it is recovered (decompressed) and passed to the D/A converter. The D/A converter returns the signal to an analogue electrical voltage, which is then input into the play out device (e.g. speakers, or handset).

1.11 Chapter Summary

This chapter has provided a brief history of Internet telephony, an overview of the technology and services, and introduced the surrounding issues. These topics are dealt with in more detail in the following chapters.

2 Internet Telephony Protocols

This chapter discusses the protocols from the TCP/IP suite that are relevant to Internet telephony. The protocols are identified. and their role or function in a VoIP system is explained.

The relationship between Internet telephony protocols is shown in the protocol stack of Figure 2.

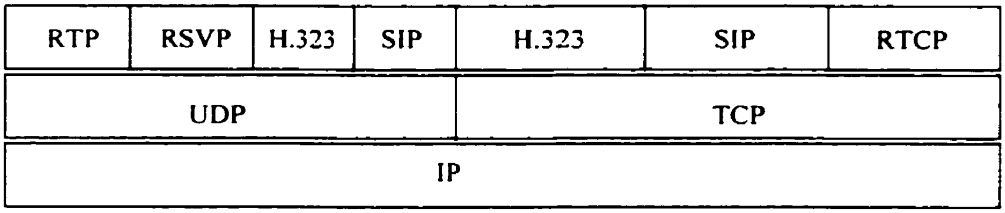


Figure 2: Internet Telephony Protocol Stack

2.1 Network Layer Protocols

Network layer protocols route datagrams from the source to the destination. The Internet Protocol (IP) is the network layer protocol in the TCP/IP protocol suite. IP is currently the most common network layer protocol used in the Internet and therefore by Voice over IP (VoIP) applications as well. In the following sections two versions of the Internet Protocol are discussed: IPv4 and IPv6.

2.1.1 IPv4

The majority of current IP based networks use version 4 of the Internet Protocol (IPv4). IP is a network layer protocol in the TCP/IP protocol stack and a layer 4 protocol in the OSI reference model. IP is an unreliable protocol that provides best effort delivery service to data. IP is a connectionless protocol; each IP datagram is handled/routed independently of the others. IP fragments transport layer protocols and routes the fragments from the source to destination. IP also provides re-assembly services to transport layer protocols.

An IP datagram consists of a header and a payload. The IPv4 header length is variable and ranges from a minimum of 20 bytes to a maximum of 60 bytes. The maximum length of an IP datagram is 65,535 bytes. Figure 3 shows the IPv4 header [9]. Fields in the IPv4 header that are of particular relevance to VoIP are discussed in the following sections.

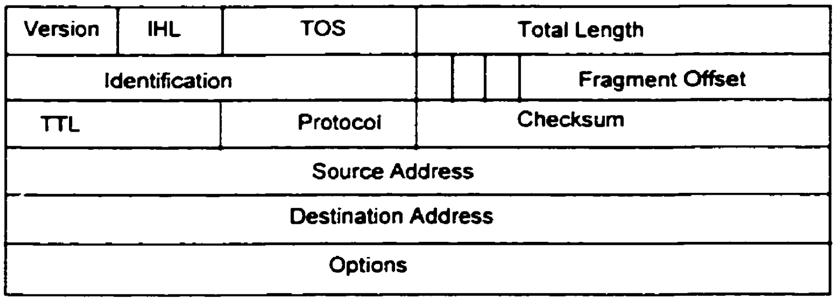


Figure 3: IPv4 Header

The Type of Service (TOS) field is a one byte field that can be used to specify the quality or class of service required by the application. This byte, shown in Figure 4, contains a 3-bit Precedence field, Delay bit (D), Throughput bit (T), Reliability bit (R), and 2 unused bits [9].

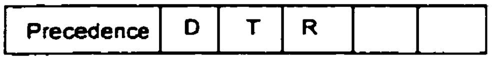


Figure 4: TOS Field in the IPv4 Header

The *Precedence* field is an integer value from 0 to 7 (decimal). This field can be set to indicate the priority with which a packet is to be handled by a router in the event of congestion. A precedence of 0 is the lowest priority.

As will be shown in Chapter 4, Internet Telephony applications have strict time constraints to achieve an acceptable level quality, and can only tolerate a low percentage of packet loss. Therefore, an Internet telephony packet should receive a higher priority marking than non real-time applications. An exact precedence value, however, has not been determined for Internet telephony applications. Current implementations of Internet telephony applications set the precedence value to 0 (low priority).

The type of service flags, D, T, and R, can each be set to 0 or 1. Table 2 lists the meanings of the D, T, and R flags in the TOS field.

Table 2: DTR Settings

Field	Value	
	0	1
D	The datagram can be delayed	The datagram cannot be delayed
T	Normal throughput required	High throughput required
R	Reliable subnetwork required	Reliable subnetworks not required

In theory, the TOS field in the IPv4 header can be used by an application to indicate a level of QoS and by routers in the network to make the best routing decisions for the application. In terms of Internet telephony applications, for example, it is more important for the application to experience low delay than reliable delivery. Therefore, an Internet telephony application would set D=1 and R=0. A router could make use of these flags when choosing a route for the packet to provide the best possible service to the application.

Until recently, the TOS field has been largely ignored by the majority of router implementations. With the revived interest in delivering QoS to IP based networks, however, router and switch vendors are beginning to make use of the field. Current implementations of Internet Telephony applications do not set any bit in the TOS field. Therefore, the TOS field should not be relied upon for guaranteeing QoS to Internet telephony applications.

The two byte *Checksum* field is used to verify the accuracy of an IP header. This field is recalculated at each hop (router) and therefore contributes to end-to-end delay experienced by VoIP applications.

The *Source* and *Destination* addresses are 32 bit network address of the host that sent the packet and the intended recipient of the packet. In terms of Internet telephony, these addresses identify the endpoints of the call.

2.1.2 IPv6

Driven largely by the need for a larger address space, the IETF began developing a new network layer protocol for the Internet in 1991. To date, IPv4 is still the dominant network layer protocol in use. Islands of IP version 6 (IPv6) networks exist in experimental and research networks, but the protocol has not been deployed in wide scale.

IPv6 is discussed in this Chapter as it (i) will be used to route VoIP packets in the future, and (ii) offers new features that may prove beneficial to VoIP once it is fully deployed.

IP Version 6 (IPv6) also known as IPng (IP Next Generation) is a network layer protocol in the TCP/IP protocol suite, and a layer 3 protocol in the OSI reference model. Like IPv4, IPng is also responsible for routing IP datagrams from source to destination. Fragmentation and re-assembly services are also provided to higher layer protocols. The main differences between IPv4 and IPv6 are that (i) IPv6 has a larger address space. and (ii) IPv6 has a simplified header.

The IPv6 header is shown in Figure 5 [10]. The IPv6 header is a 40 byte fixed length header. Optional extension headers may be included. however. The IPv6 header fields that are of particular relevance to VoIP are discussed below.

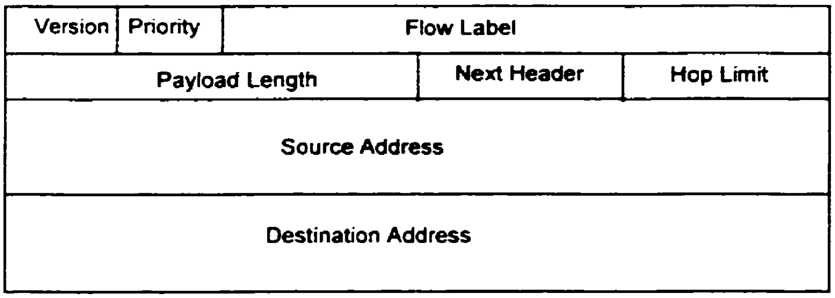


Figure 5: IPv6 Header

Similar to the Precedence field of the IPv4 header. the 4 bit *Priority* field indicates the priority of the flow. Flow priorities can take on values from 0 to 15. with a value of 0 being the lowest priority. In the event of congestion. lower priority datagrams will be discarded before higher priority datagrams. As will be shown in Chapter 4. Internet telephony applications can only tolerate a low percentage of packet loss. so this field should be set to a higher value for VoIP traffic.

The *Flow Label* field is a three byte field new to the IP header in version 6 of the protocol. The purpose of this field is to label IP packets belonging to the same flow or stream (i.e. originating from the same process) so that routers can apply the same treatment to all packets belonging to that flow. All packets belonging to the same VoIP conversation would have the same flow label, for example, so they would all have access to the same resources (e.g. reserved bandwidth).

The *Source* and *Destination* addresses are 16 byte network addresses of the sender and intended recipient of the IP datagram. respectively. These addresses will identify the endpoints of IPv6 network hosts in a VoIP call: i.e. equivalent to *telephone numbers* in conventional telephony.

2.1.3 Comparing IPv4 and IPv6

A number of changes have been made to the Internet Protocol in version 6 that are expected to improve the performance of the protocol. The performance of Internet telephony applications should also benefit from these changes.

The IPv4 *IHL* field has been omitted from the IPv6 header because the IPv6 header is fixed in length. As seen in the ATM world. processing a smaller. fixed length header improves the efficiency of end-to-end routing [11]. Processing smaller fixed length headers should also improve the efficiency of routing packets through an IP network. This will reduce the end-to-end delay experienced by an IP datagram. and by the VoIP application carried by the IP datagram. As will be shown in Chapter 4. reducing the end-to-end delay will improve the performance of VoIP applications.

The *DF. MF*, and *Fragment Offset* fields have been omitted from IPv6. All IPv6 compliant hosts must support packets of 576 bytes. which will make fragmentation less likely. This change is intended to improve efficiency of the protocol.

Another improvement of IPv6 over IPv4 is the removal of the *Checksum* field from the IPv6 header. In IPv4, the *Checksum* is recalculated at every hop along the route. It was determined that the benefits provided by this field were out weighed by the computational costs of performing the calculation; namely increased transmission delay.

Two main reasons driving the omission of the checksum were the fact that today's networks are considered to be more reliable, and therefore it is less likely that an error will occur in the first place. Second, data integrity checks are performed by higher layer protocols, therefore its is not necessary to perform them at the network layer as well. The removal of the checksum from the IP is expected to improve the overall performance of the protocol and reduce the end-to-end delay. Reducing the end-to-end delay will also improve the performance of VoIP service.

The *Flow Label* field in IPv6 identifies all packets that belong to the same flow in order to be able to provide the same treatment to all packets belonging to that flow. In terms of VoIP, all packets belonging to the same VoIP flow can make use of reserved network resources, or receive fast handling in intermediate routers or switches, for example.

The *Priority* field in IPv6 is similar to the TOS field in IPv4. The intention of this field is to mark packets with a priority that can be used to make decisions on packet handling in the event of congestion. While the TOS field was largely ignored in IPv4, the priority field in IPv6 has received much attention, as it is seen as a means to deliver QoS in the Internet.

In terms of Internet Telephony, the assignment of a high priority is expected to improve the performance of a VoIP application. Packets marked with higher priority are (i) more likely to be delivered, and (ii) more likely to be delivered in a timely fashion, both of which benefit the performance of VoIP, as will be shown in Chapter 4. Future work is required in this area to recommend a standard IPv6 priority to be used for VoIP, and to develop a behaviour mechanism to apply to packets with this marking.

2.2 Transport Layer Protocols

2.2.1 TCP

Transmission Control Protocol (TCP) is a transport layer protocol in the TCP/IP protocol suite, and classified as a layer 4 protocol in the OSI reference model. Transport layer protocols control the communication session between two network endpoints.

TCP is a connection oriented protocol; before data can be transported a connection must be established between the communicating network nodes. TCP is a reliable transport protocol. If data is lost during the connection, TCP will retransmit the packet.

The TCP header is shown in Figure 6 [12]. The minimum length of a TCP header is 20 bytes, but the header can vary in length as indicated by the *Options* field.

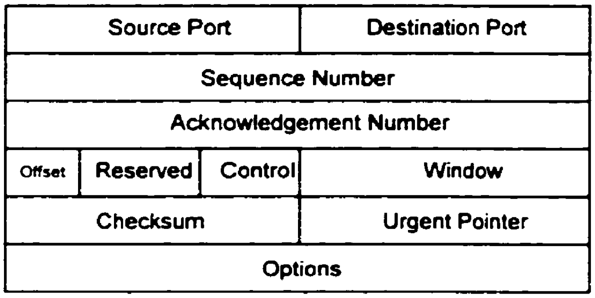


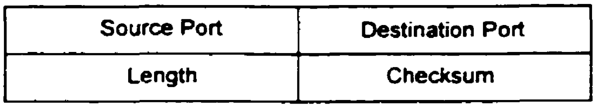
Figure 6: TCP Header

TCP is used in most Internet telephony applications for the exchange of control data. The overhead of the TCP packet, and the signaling delays associated with TCP connections make TCP inappropriate to transport voice, however.

The TCP ports used by Internet telephony applications for control data are not standardized and vary between VoIP applications. The TCP ports used by several Internet telephony applications are given in Chapter 4.

2.2.2 UDP

The User Datagram Protocol (UDP) is a transport layer protocol in the TCP/IP protocol stack, and is classified as a layer 4 protocol in the OSI reference model. UDP is a connectionless protocol; connections between a source and destination do not need to be established prior to the exchange of UDP packets. UDP is an unreliable protocol; if a packet is lost, it is not retransmitted. The UDP header is shown in Figure 7 [13]. The header is 8 bytes in length.



Source Port	Destination Port
Length	Checksum

Figure 7: UDP Header

All Internet telephony applications make use of UDP at the transport layer for the purpose of transporting the voice data. Because it is a connectionless protocol, using UDP saves call setup time, which is an important metric to consider in VoIP systems. Reliable protocols, such as TCP, only allow a window of packets, say *N*, to be transmitted without receiving an acknowledgement from the receiver. Delay is introduced into the system while waiting for the receipt of the acknowledgement before more packets can be transmitted. As UDP is an unreliable protocol, there is no windowing or delay introduced while waiting for an acknowledgement. For voice systems, it is more important that the (majority) of the voice packets be delivered within an acceptable period of time than reliable delivery of all packets.

The UDP ports used by Internet telephony applications have not been standardized and vary between applications. Refer to Chapter 4 for the UDP ports used by VoIP applications.

2.2.3 RTP

Real-time traffic, such as VoIP, has properties that require different network services than non-real time traffic. RTP (Real-time Transport Protocol) was created as TCP and UDP were unable to provide these services. There are two components to the real-time transport protocol: RTP and RTCP (Real-time Control Protocol). As will be shown in section 2.3.1, RTP and RTCP are mandatory protocols within the H.323 recommendation. Therefore, all Internet telephone applications that comply with the H.323 standard carry voice data in RTP.

Services required by real-time applications include, (i) reordering of packets as they arrive at the receiver, (ii) detection of lost packets, (iii) adjusting the play-out timing at the receiver, and (iv) identification of encoding format of the media.

Packets may take different routes of varying delay from source to destination, which cause the packets to arrive out of order at the receiver. VoIP packets must be played out in the correct order so that the speech sounds intelligible. UDP is not able to determine the correct order of packets.

In route from source to destination, a packet may be lost (dropped due to congestion in a router, for example). Packet loss results in speech quality degradation. It would be advantageous if a receiver could be made aware of loss and apply a correction mechanism to improve the quality of the speech. UDP does not provide a mechanism to determine the loss of a packet.

The network may introduce variable delay between successive packets in a stream. The correct inter-packet timing must be recreated at the receiver so the speech sounds natural. UDP does not provide timing mechanisms for this purpose.

It is possible that two endpoints in a VoIP call use different coding methods for the speech. It is necessary to covey the coding format between end points so the receiver can decode the received speech.

These services are required by real-time applications, but are not provided by TCP or UDP. RTP provides these services, and therefore it is often used to support real-time media streams, including VoIP.

Although RTP is a protocol at the transport layer, it still requires the services of an underlying transport protocol. RTP is usually used with UDP, but can be used with TCP as well. RTP does not replace TCP or UDP, but rather adds another layer of services or functionality to the services offered by the transport protocol. The format of the RTP header is shown in Figure 8 [14].

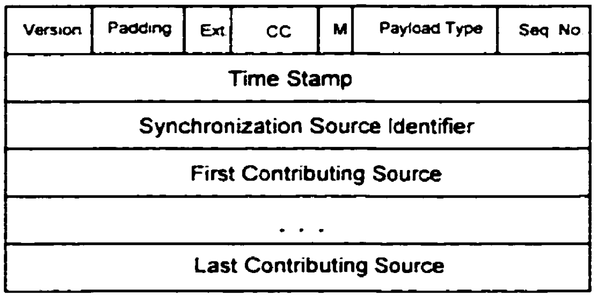


Figure 8: RTP Header

The RTP header is fixed in length and is 12 bytes long. Audio content is encapsulated in the payload portion of the RTP PDU. The RTP header and payload are transported in the payload portion of the underlying transport protocol (UDP).

2.2.4 RTCP

The Real-time Control Protocol (RTCP) is the control portion of RTP. As mentioned in the previous section, RTP and RTCP are mandatory components of H.323. RTCP is therefore used by all H.323 compliant Internet telephones.

RTCP periodically sends messages to all participants in the conference. The messages may be one of five types: *Sender Reports*, *Receiver Reports*, *Source Descriptions*, *Bye* messages, and *Application Specific* messages. Sender reports are sent by hosts that are transmitting RTP streams. The source of an RTP conference collects information about the quality of the communication from each receiver, and retransmits this information in Sender Reports. The information contained in the Sender Reports is received from each receiver in the Receiver Reports. Sender Reports contain QoS information collected from each receiver including the amount of data transmitted so far, the percentage of packets lost, and jitter measurements. Senders could use the QoS information contained in the Receiver Reports to adjust parameters of the media stream to better suit current network conditions.

2.3 Call Setup and Signaling Protocols

This section discusses two protocols that are used for call setup, call signaling, and call control in VoIP systems: H.323 and SIP. The two protocols are compared in terms of VoIP service capabilities, complexity, and vendor support.

2.3.1 H.323

H.323 is an ITU-T recommendation for multimedia communication over non-guaranteed QoS LANs. The protocol encompasses a number of ITU and IETF protocols within one standard. The Internet telephony market grew quickly and in many directions, due to the initial lack of standardization for the technology. As a result, Internet telephones were unable to inter-operate. H.323 was adopted by a number of leading VoIP vendors, as well as the IMTC (Interoperability Multimedia Telecommunications

Consortium) [16], for the purpose of standardizing VoIP call set up procedures, call control, and communication protocol support.

2.3.1.1 H.323 Architecture and Components

The H.323 architecture consists of four main components: *terminals*, *gateways*, *gatekeepers*, and *multipoint control units* (MCU) [15]. The function of each component within a VoIP system is described below.

An H.323 terminal is a LAN endpoint used for real-time multimedia communications. An H.323 terminal is the only mandatory component of an H.323 system. The terminal must be capable of sending and receiving G.711 audio (A-law and μ-law). Support for other audio codecs is optional. Video and data communications may also be supported by an H.323 terminal, but these capabilities are optional. If video capabilities are supported, then the terminal must support H.261. If data communication is supported, then the terminal must support the T.120 standard [15].

In addition to the audio, video, and data codecs, an H.323 terminal must also include support for H.245, Q.931, and RAS. H.245 is used for capabilities negotiation between terminals or between a terminal and a gatekeeper. Q.931 is used for call setup and call signaling. RAS is a protocol used to communicate between a terminal and a gatekeeper. H.323 systems must also support RTP and RTCP [15].

A gateway is an optional component of an H.323 session. A gateway is only required if the communicating endpoints are on different types of networks (e.g. IP to PSTN). The gateway provides translation functions between an H.323 terminal and another ITU compliant terminal. Translation functions include translating between call setup protocols or procedures, and translating between audio and video codecs [15].

A gatekeeper is also an optional component in an H.323 session. However, if a gatekeeper is present in the system, then terminals must make use of its services. A gatekeeper performs the following functions: address translation, call admission control, bandwidth management, and zone management. Terminals that reside in different network types will have different addressing mechanisms. The gatekeeper is the system component that performs the mapping between different address types so endpoints can locate one another. For example, suppose an IP phone wished to contact a PSTN terminal. The Gatekeeper would provide the translation between the IP address and the conventional phone number formats (among other functions) [15]. The implementation of the address translation is not subject to standardization, but is usually implemented as a database or a lookup table.

A gatekeeper also performs call admission control functions. The number of simultaneous calls that can be supported by a gatekeeper is not subject to standardization. However, a network operator may limit the number of calls using the call admission control function of the gatekeeper. A gatekeeper may also be used to perform bandwidth management. A network operator can restrict H.323 calls to occupy a percentage of the total network bandwidth. The remaining network bandwidth can be used for other functions such as file transfers. A *zone* is referred to as the collection of H.323 components that are managed by a gatekeeper. A gatekeeper can be implemented as a stand alone component, or as part of a H.323 gateway [15].

The fourth component of a H.323 system is the multipoint control unit (MCU). The MCU is also an optional component. It is only required to support conferences between three or more endpoints. The MCU contains two sub-components: a multipoint controller (MC), and multipoint processors (MP). An MC is a required component of a MCU, while an MP is optional. The MC manages the communication session but does not deal directly with the media streams. The MP, if present, performs data handling of the media streams between the endpoints in the conference. For example, functions of an MP include audio and video mixing. An MCU can be implemented as a stand alone component, or as part of other H.323 components [15].

2.3.1.2 Using H.323 for VoIP

The following discussion is an example of how H.323 could be used to place a VoIP call. This example includes a gatekeeper as part of the VoIP system.

The calling terminal requests permission from the gatekeeper to place the call. Baring any bandwidth or access restriction, the gatekeeper grants permission for the call to proceed. The calling terminal opens a TCP connection and sends the called terminal a Q.931 *setup* message. When the called terminal receives the Q.931 setup message, it requests permission from its gatekeeper to communicate with the calling terminal. (In this example, both endpoints of the communication are within the same zone, and therefore use the same gatekeeper.) Next, the called terminal's gatekeeper grants permission for the called terminal to communicate with the caller. When permission has been received, the called terminal sends a Q.931 *connect* message to the caller to indicate that it is willing to receive the call.

The calling terminal opens a TCP connection with the callee and the caller and the callee exchange *terminal capabilities* messages. H.245 is used to determine which terminal will act as master and which will assume the role of slave in the communication. After terminal capabilities have been exchanged, each terminal sends an *open logical* *audio channel request* to the other. The message includes information on which UDP port to use for sending and receiving RTP and RTCP messages. After an audio channel has been opened and acknowledged in each direction, call setup is concluded and the terminals are now ready to exchange VoIP data. The procedure for setting up a VoIP call with H.323 is shown in Figure 9.

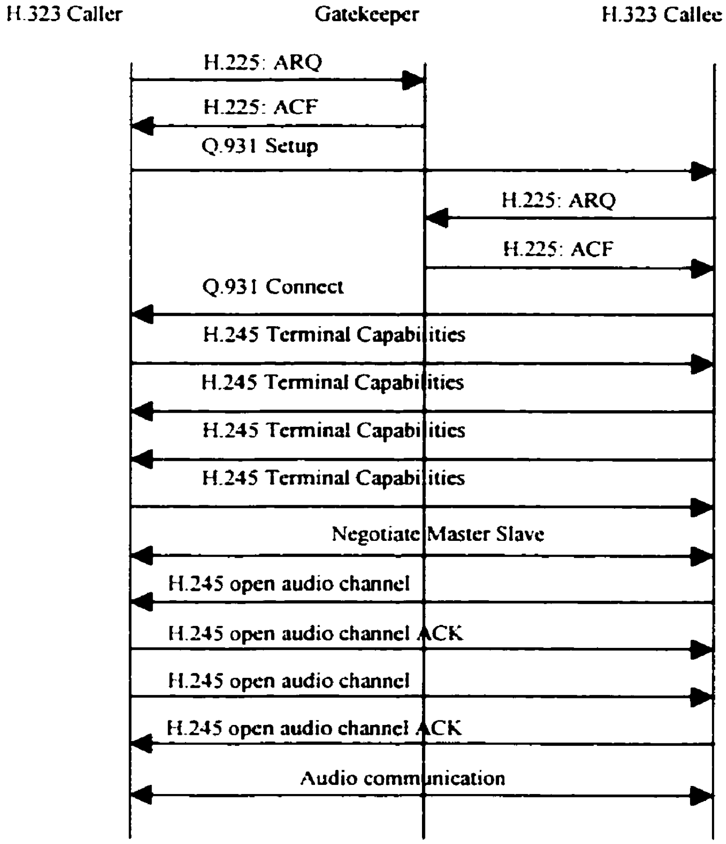


Figure 9: VoIP call setup with H.323

2.3.2 SIP

Driven by the need for a less complex protocol, the IETF set forth to develop a simpler protocol for call signaling and inter-operability for multimedia streams - Session Initiation Protocol (SIP). SIP is not limited to VoIP session control, but like H.323, may be used for call setup, call control, address translation, etc. of multimedia sessions in general. The following sections discuss the architectural components of a SIP VoIP system, and present an example of placing a VoIP call with the protocol.

2.3.2.1 SIP Architecture and Components

The SIP architecture consists of the following components: *call participants, proxy servers, redirect servers*, and *location servers*. *Call participants* are SIP-enabled terminals or end systems. Call participants are the end points in the VoIP call. SIP-enabled end systems include both a SIP client and a SIP server, and are referred to as *user agent servers*. The user interfaces with the user agent server. *Proxy servers* act as both a SIP client and server and make requests on behalf of other SIP entities. A *redirect server* is a server that provides the address of the next hop in the communication path to the client. The client uses this information to then contact the recipient directly. A *registrar* is a type of server that accepts *REGISTER* requests from clients. The registrar may be co-located with either a proxy or redirect server. A *location server* is a non-SIP component, but is required as part of the SIP architecture. A proxy or redirect server makes use of a location server to find the current location of a user. The location server may be the result of a *finger* request, an LDAP or a proprietary database [17].

2.3.2.2 SIP Messages

In a client-server protocol a client issues a request and the server responds. In SIP, client requests are referred to as *methods*. There are 6 methods in SIP: *INVITE, ACK, OPTIONS, REGISTER, BYE, and CANCEL*. INVITE is used to invite a callee into the conversation and to perform capabilities exchange of each terminal (for example which audio codec to use). ACK is used to acknowledge the INVITE method; i.e. to accept the call. ACK may also be used to signal session description parameters. The OPTIONS method is issued by an end terminal of a server to request the capabilities of the server. The REGISTER method is sent by an end user to a server to register the user's current location. The BYE method may be issued by either end terminal to terminate a connection. When a server (proxy or redirect) attempts to locate an end user, it may conduct searches along several paths in parallel. When the user is located, a CANCEL method is used to terminate the remaining searches [17].

A server responds to requests from clients by sending one of six types of response messages. These responses are summarized in Table 3 [17].

Table 3: SIP Responses

SIP Message	Meaning
1xx	Informational
2xx	Successful
3xx	Redirection
4xx	Request Failure
5xx	Server Failure
6xx	Global Failure

2.3.2.3 Using SIP for VoIP

The following section discusses the dialing procedure for placing a VoIP call with SIP. In this example, a SIP proxy server is used for user location and call setup [19].

Suppose for example Fred wishes to call Jim. Jim has published his 'phone' number as jim@ee.umanitoba.ca. Phone numbers in SIP are called *SIP URLs*. SIP URLs are in the form of user@domain. The user portion can be either a userid, a name, or an E.164-style phone number [18]. The domain portion of the address is either a domain name or an IP address. Fred would obtain Jim's SIP phone number either from a third party database (SIP phone book) or would guess his phone number from Jim's email address.

Jim, while logged into machine *ouzo* of domain ee.umanitoba.ca, registers his current location with the location server by sending a REGISTER message. When Fred places a call to Jim by dialing Jim's email address, Jim's address (jim@ee.umanitoba.ca) is resolved to the SIP server in the ee.umanitoba.ca domain (in this case sipserver.ee.umanitoba.ca).

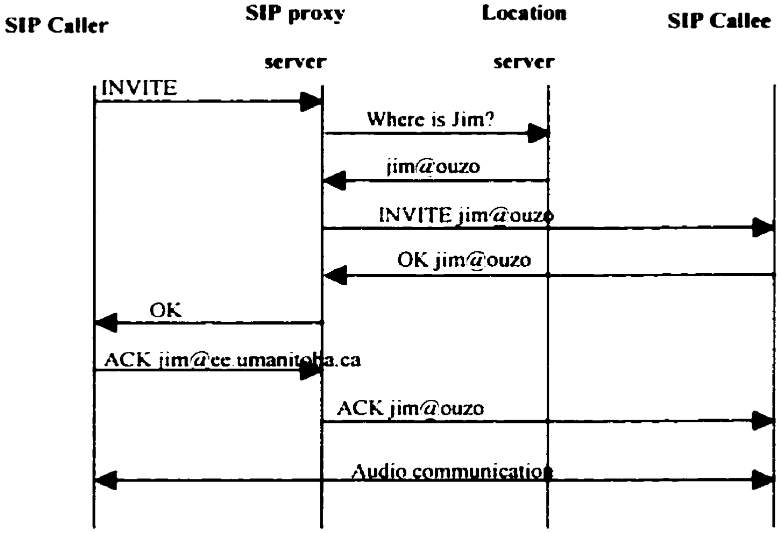


Figure 10: VoIP call setup with SIP

Once the SIP server has been located, Fred issues an INVITE message to the SIP server; INVITE jim@ee.umanitoba.ca. The SIP server consults its location server to see where Jim is currently registered ('Where is Jim?'). Since Jim has previously registered his location, the location server returns Jim's current location (jim@ouzo). Having now located Jim, the SIP server then issues an INVITE message to Jim at machine *ouzo* directly; INVITE jim@ouzo. Jim chooses to accept the call and therefore issues an OK message to the SIP server. The SIP server issues an OK message to the caller (fred@win.trlabs.ca). Fred acknowledges acceptance by issuing an ACK message to the SIP proxy server; ACK jim@ee.umanitoba.ca. The SIP proxy server issues an ACK to Jim at ouzo; ACK jim@ouzo. The call setup is now complete and Fred and Jim are able to begin their VoIP call.

2.3.3 SIP versus H.323

H.323 and SIP are the two main contenders for VoIP signaling standards [20]. H.323 is an ITU standard, while SIP is an IETF standards track RFC [17]. The two protocols are roughly equivalent in terms of services provided (call set up, capabilities negotiation, address resolution, etc.).

However, SIP is considered to be the less complex protocol of the two [20], [21], which is one of its main advantages. There are several examples of the differences in complexity between the two standards. First, there are more architectural elements in an H.323 system than are required in a SIP architecture. In addition, the interactions between elements are more complex in H.323 than in SIP [21]. SIP is a text based protocol, while the H.323 protocol is binary. Implementing a parser for binary messages is considered to be more complicated than creating a text based parser [21].

Despite these differences, H.323 has clearly been established as the industry leader. H.323 was the first of the two protocols to be developed and made commercially available, and has been adopted by several VoIP industry leaders including Microsoft, Intel, and Cisco. It has been selected as the inter-operability standard for VoIP equipment by the IMTC [16].

At the time of this writing, it is difficult to find an implementation of SIP outside of academic or research labs. One non-commercial implementation of a SIP VoIP phone, *sipphone*, is available from Bell Labs. However, this product is for academic or research purposes only. At least in the near future, H.323 will remain the standard *de jour* for commercial VoIP equipment.

2.4 Quality of Service Protocols

This section discusses two protocols that could be used to deliver QoS for real-time applications such as VoIP: RSVP and RTCP.

2.4.1 RSVP

Resource reSerVation Protocol (RSVP) is an IETF protocol developed by the Integrated Services Working Group. RSVP can be used to reserve network resources for a media stream, including a VoIP call. RSVP can be either encapsulated in UDP packets or encapsulated directly in the payload of IP packets. The operation of RSVP is described below.

RSVP is a receiver oriented protocol. The receiver reserves resources along the path from receiver to source. RSVP permits resources (such as bandwidth) to be reserved at OSI layer 3 network elements (i.e. routers).

The protocol is initiated by the source sending a PATH message from the source to the receiver. At each hop along the path from source to receiver, the router adds its IP address to the PATH message. The PATH message received by the destination terminal contains the complete path from source to receiver. The PATH message serves two purposes. First, it primes the routers to expect reservation messages from the destination terminal. Second, it provides the destination terminal with the path that the media stream will be taking. The receiver uses this path message to reserve resources from the destination back to the source along the media path. It is necessary to send the path information because routes between source and destination are not necessarily symmetrical in both directions.

Once the resources have been reserved for a call, communication may begin. If, however, a router was not able to reserve the requested resources, the RSVP call setup procedure fails; resources must be reserved at each hop for the RSVP connection to be established. Resource reservations time out; i.e. RSVP is a *soft state* protocol. PATH messages are also used to periodically update reservation requests from receiver to source.

2.4.2 RTCP

RSVP could be used to reserve the required resources for a VoIP call. The complexity of RSVP and the fact that it is a state oriented protocol have led some of the Internet community to believe that RSVP is too complex to be useful in practice [19]. It has been proposed that RTCP be used as a simpler resource reservation protocol.

As discussed previously in section 2.2.4, the sender reports in RTCP contain QoS information. It is argued that this information could be used by routers to determine the required resources and to reserve these resources for a media stream, such as VoIP [19].

It is plausible to assume that most real-time multimedia calls that would require RSVP would also be making use of RTP/RTCP. By using RTCP for QoS signaling, the additional layer of signaling (RSVP) is removed. This would reduce call setup time. Each router along the path from source to destination would still be required to maintain state information for each flow. So like RSVP, the RTCP solution also has scalability issues.

2.5 Chapter Summary

This Chapter has identified the protocols used to deliver Internet telephony. To summarize, there are two main competing protocols used for VoIP call signaling: H.323 and SIP. Despite some possible advantages of SIP, H.323 has been established as the standard *de jour*.

All Internet telephony applications transport voice data with UDP. UDP is used for real-time data as it minimizes transport delays. Some applications use RTP to make use of the services RTP provides to real-time applications. In this case, voice is encapsulated in RTP which is then transported via UDP. TCP is used to transport control or signaling data for VoIP.

Several protocols within the VoIP protocol stack attempt to deliver real-time services to Internet telephony applications. IPv4 has the capability to provide QoS. However, neither the network infrastructure nor VoIP applications have been implemented to take advantage of this capability. RSVP or RTCP may be used deliver QoS in today's networks.

Building QoS into the IP network infrastructure is a current area of study in the Internet community. Future architectures that may provide the QoS required for VoIP are discussed in Chapter 5.

3 Speaker Activity Models

3.1 Introduction

Simulation is a useful technique for the purpose of network planning. As VoIP is expected to carry 10% of the voice calls in the US by the year 2004 [1], an accurate mode of VoIP traffic is required. The model could be built into a network simulator and used for the purpose of planning the network required to support this service.

In this Chapter speaker activity models are developed for three VoIP applications: Microsoft NetMeeting, Voxware Televox, and NetSpeak Webphone. Each model is compared with traditional models for conventional telephony in order to determine whether conventional models are suitable for modeling speaker activity in a VoIP call. The traffic models are also compared with one another in order to determine the effect of vendor's implementation on the resulting traffic pattern.

3.2 Previous Work

Voice communication has undergone three stages of evolution: analogue, digital, and most recently, packet voice. A considerable amount of work has been conducted in the area of modeling speaker activity in digital telephone networks. Paul Brady and Daniel Minoli are two of the primary contributors in the area of traditional telephony speaker activity models. Traditional models vary in terms of the number of states represented, but all models have all been Markovian.

In the 1960s, Brady developed a six-state speaker activity model for digital telephony. In Brady's model, two speakers (*A* and *B*) could be in one of the following states: *A* speaks while *B* is silent, *B* speaks while *A* is silent, *A* interrupts *B*, *B* interrupts *A*, mutual silence *A* spoke last, and mutual silence *B* spoke last [23]. This 6-state Markovian model is frequently referred to as the *Brady model* [24] [25] and is often used as the standard of reference.

Minoli contributed to speaker activity modeling by identifying ten events relevant to telephone conversations. These events are: talkspurt, pause, double talk, mutual silence, alternation silence, pause in isolation, solitary talkspurt, interruption, speech after interruption, and speech before interruption [24]. Minoli's events are often used to identify states of speaker activity from which speaker activity models are developed. Speaker activity models vary in complexity with the more accurate models representing a greater number of Minoli's events [24].

The Brady model is one of the most complex speaker activity models. It is considered to be the strongest model as it accurately represents Minoli's 10 states [24]. However, due to its complexity, this model is seldom implemented. Other models of lesser complexity are used more frequently, as they are able to provide fairly accurate models with reduced implementation complexity [24].

The 4-state Markov chain model represents the following states of speaker activity for two speakers (*A* and *B*): *A* is talking *B* is silent, *B* is talking *A* is silent, double talk, and mutual silence. This model is considered to represent the states of single speaker activity and mutual silence well, but does not accurately model the state of double talk [24].

The 3-state Markovian model consists of the following states for two speakers (*A* and *B*): *A* is talking *B* is silent, *B* is talking *A* is silent, and mutual silence [24]. Talk spurts are considered to be represented fairly well by this model, but this model does not accurately model the state of silence [24].

Finally, two speakers (*A* and *B*) in the 2-state Markovian model are represented by the following states: *A* is talking *B* is silent, and *B* is talking while *A* is silent. This model is frequently used because of its simplicity. However, it is not considered to be an accurate representation of the activities of telephone conversation; only the talk spurt state can be modeled, and with only fair accuracy [24].

In an experiment conducted by Deng [25], a 2-state speaker activity model was constructed for traffic generated by the following packet voice application software: *vat* (Virtual Audio Terminal), NeVot (Network Voice Terminal), and Maven (Macintosh Audio Enabler) [25]. Deng compared the collective traffic generated by these applications to the traditional 2-state Markov model for speaker activity. The result of Deng's experiment was that the exponential model for active talk state holding times was rejected 40% of the time with 95% confidence [25].

Deng's work showed that the 2-state traditional telephony model is not well suited for modeling traffic generated by packet voice applications. This result is not unexpected since fundamental differences exist between traditional voice networks and VoIP systems. As a result, some of Minoli's 10 states cannot be represented by a VoIP call. For instance, in a half-duplex VoIP call, the state of double talk cannot occur. Also, since a speaker cannot be interrupted in a half-duplex VoIP call, the states of speech before interruption and speech after interruption also cannot occur.

In this Chapter, 3 and 4-state models for speaker activity in a VoIP call are developed. In continuation of Deng's work, these models are compared against the traditional telephony models. In addition, the 3-state models are compared across three VoIP applications in order to determine the effect of implementation on the resulting traffic stream.

3.3 Model Distributions

This section introduces the distributions (and their properties) used in the speaker activity models.

3.3.1 Exponential

The probability density function, *f(t)*, of the exponential distribution is given as [26]:

$$f(t) = \lambda e^{-\lambda t}, t \geq 0$$

The parameter $\lambda > 0$. The cumulative distribution function, *F(t)*, is given as:

$$F(t) = 1 - e^{-\lambda t}, t \geq 0$$

The exponential distribution has mean, E(t), and squared coefficient of variation, c_t^2, of:

$$E(t) = \frac{1}{\lambda} \text{ and } c_t^2 = 1$$

3.3.2 Hyper-Exponential

The hyper-exponential distribution is a mixture of two exponential distributions of different means. The probability density function of the hyper-exponential distribution is given as [26]:

$$f(t) = p_1\mu_1 e^{-\mu_1 t} + p_2\mu_2 e^{-\mu_2 t}, t \geq 0$$

where $0 \leq p_1, p_2 \leq 1$ and $p_1 + p_2 = 1$

A random variable t is distributed as an exponential with mean $E(t) = \frac{1}{\mu_1}$ with probability p_1 and is distributed as an exponential with mean $E(t) = \frac{1}{\mu_2}$ with probability p_2. The squared coefficient of variation, c_t^2, of an hyper-exponential distribution is always $c_t^2 \geq 1$.

3.3.3 Erlang-k

The Erlang-k distribution is a sum of *k* exponentially distributed random variables with the same means. The probability density function, *f(t)*, of the Erlang $_{k-1,k}$ distribution is given as [26]:

$$f(t) = p\mu^{k-1}\frac{t^{k-2}}{(k-2)!}e^{-\mu t} + (1-p)\mu^k \frac{t^{k-1}}{(k-1)!}e^{-\mu t}, t \geq 0$$

where $0 \leq p \leq 1$

A random variable, t, with an Erlang $_{k-1,k}$ distribution is distributed as the sum of *k-1* exponentials each with mean $E(t) = \frac{1}{\mu}$. The squared coefficient of variation, c_t^2, of an Erlang $_{k-1,k}$ distribution is given as:

$$\frac{1}{k} \leq c_t^2 \leq \frac{1}{k-1}$$

where $p = \frac{1}{1+c_t^2}[kc_t^2 - \{k(1+c_t^2) - k^2c_t^2\}^{1/2}]$ and $\mu = \frac{k-p}{E(t)}$.

3.4 Selecting a Model for State Holding Time Distributions

The coefficient of variation is defined as:

$$c_x = \frac{\sigma(x)}{E(x)}$$

where x is a random variable with mean $E(x)$ and standard deviation $\sigma(x)$. The squared coefficient of variation, c_x^2, is a measure of the variability of the random variable x [26].

Knowledge of the value of c_x^2 for the theoretical distributions was used to select a potential model for state holding time distributions. If the value of c_x^2 was close to 1, then an exponential distribution was selected as the model for the empirical data [26]. If the value of $c_x^2 > 1$, then a hyper-exponential distribution was selected as a model for the empirical data [26]. Finally, if the value of $c_x^2 < 1$, then an Erlang distribution was selected as a model for the empirical data [26].

3.5 Kolmogorov-Smirnov Test

Having selected a theoretical distribution as a potential model for the empirical distribution, the parameters of the hypothesized distribution were calculated from the empirical data, and a goodness-of-fit test was performed between the empirical and model distributions.

The Kolmogorov-Smirnov test [27] is a goodness-of-fit test that is used to test the fit of a hypothesized or theoretical distribution to a distribution of empirically gathered data. The Kolmogorov-Smirnov test tests the null hypothesis, H_0:

H_0: $E(x) = T(x)$

where *E(x)* is the empirical cumulative distribute and *T(x)* is a theoretical cumulative distribution. The Kolmogorov-Smirnov test compares *E(x)* and *T(x)* and computes the test statistic, *D*. The statistic *D* is the largest absolute difference between the empirical and theoretical cumulative distribution functions [27].

To determine whether or not the empirical distribution fits the theoretical distribution, the test statistic *D* is compared against a critical value, D_α, for some sample size *n*. If $D < D_\alpha$, then the null hypothesis cannot be rejected with $(1-\alpha)100\%$ certainty [27].

3.6 VoIP Speaker Activity Models

This section presents the experimental network used to capture the VoIP traffic, and presents the models for speaker activity based on these traffic captures.

3.6.1 3-State Speaker Activity Model

3.6.1.1 Experiment

Traffic generated by three VoIP applications (Microsoft NetMeeting, Voware Televox, and NetSpeak Webphone) was captured using the UNIX utility *snoop*. Communication took place between a 120 MHz Pentium PC, PC_1, and a 100 MHz Pentium PC, PC_2, over a 10 Mbps Ethernet LAN. Traffic was collected over a period of several days and conversations covered a wide variety of topics. The experimental network is shown in Figure 11.

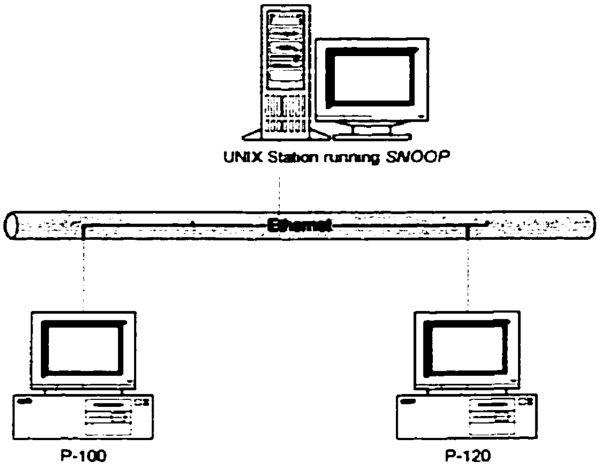


Figure 11: Experimental Network for Traffic Capture

3.6.1.2 Defining States of Speaker Activity

The snoop capture files were analyzed to extract the state holding time information for the speaker activity models. For the 3-state model, a speaker could be in one of the following states: *talk state*, *listen state*, or *mutual silence*. The *talk state* was defined to be the time during which packets were transmitted from PC_1 to PC_2, so long as the inter-packet time did not exceed the threshold, T, and the burst consisted of more than 2 packets. The *listen state* was defined to be the time during which packets were received from PC_2 by PC_1, so long as the inter-packet time did not exceed the threshold, T, and the burst consisted of more than 2 packets. Speakers were defined to be in a state of *mutual silence* in the gaps of time during which no packets were either transmitted or received. In this model, periods of mutual silence occur when the inter-packet time between successive packets traveling in the same direction exceed T, or the period of time between the change in direction of a stream of packets (alternation silence).

The threshold T is imposed in order to detect the end of a burst. A value of T = 200 ms was chosen as the threshold. Periods of silence less than 200ms are considered to be caused by stop consonants or other natural pauses during a flow of speech [23] [24] [28] and do not signify the end of a burst. Therefore, packets traveling in the same direction with inter-packet times less than 200 ms are defined to belong to the same burst. Packets traveling in the same direction with inter-packet times greater than 200 ms are defined to belong to separate bursts. Refer to Figure 12.

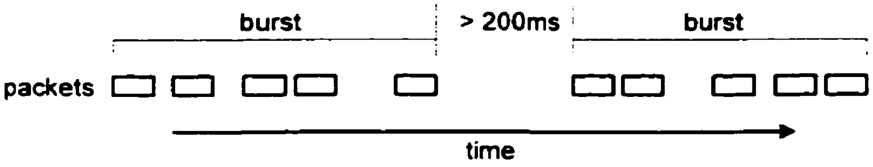


Figure 12: Definition of States

3.6.1.3 Modeling State Transitions

As shown in the 3-state speaker activity model of Figure 13, a speaker leaving the talk state always goes to a state of mutual silence, as does a speaker leaving the listening state. From a state of mutual silence, there is a probability, p_1, of moving to the listening state, and a probability, p_2, of moving to the talk state. Note that $p_1 = 1 - p_2$.

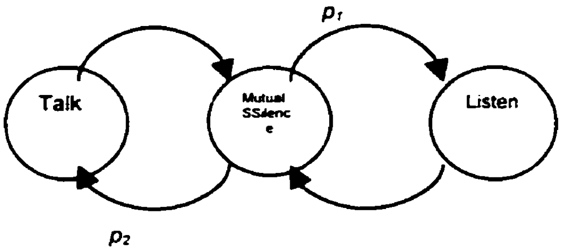


Figure 13: 3-state Speaker Activity Model for VoIP

Traffic was collected for each VoIP application (NetMeeting, TeleVox, and Internet Phone) and was analyzed in terms of the number of state transitions. The test interval for each VoIP application was 20 minutes. The tally of the number of state transitions during the test interval is given in Table 4.

Table 4: Count of State Transitions

Application	Number of State Changes: Mutual Silence to Talk	Number of State Changes: Mutual Silence to Listen	Total Left State of Mutual Silence
NetMeeting	708	739	1447
Televox	473	481	954
Webphone	611	334	945

Point estimates were used to model the probability of state transitions, p_1 and p_2. The point estimate, p, of the true proportion of a population, P, that belongs to a class is defined as [29]:

$$p = \frac{X}{n}$$

where X is the number of observations belonging to a class, and n is the total sample size.

The two-sided 100(1-α)% confidence interval on a proportion P is defined as [29]:

$$p - z_{\alpha/2}\sqrt{\frac{p(1-p)}{n}} \leq P \leq p + z_{\alpha/2}\sqrt{\frac{p(1-p)}{n}}$$

where $z_{\alpha/2}$ is defined as the upper $\alpha/2$ % of the standard normal distribution [29].

The error in estimating P by p is Err = $|p - P|$ [29], and we can be 100(1-α)% confident that Err $\leq \sqrt{\frac{p(1-p)}{n}}$

Point estimates for p_1 , 95% confidence intervals on the proportions, and error estimates are given in Table 5.

Table 5: Model for p_1 (Probability of Moving to Listen State)

Application	p_1	95% confidence interval	Err
NetMeeting	0.5107	[0.4849 , 0.5365]	2.5756 x 10-2
Televox	0.5041	[0.4724, 0.5358]	3.1728 x 10-2
Webphone	0.3534	[0.3229, 0.3839]	3.0478 x 10-2

Point estimates for p_2, 95% confidence intervals on the proportions, and Err estimates are given in Table 6.

Table 6: Model for p_2 (Probability of Moving to Talk State)

Application	p_2	95% confidence interval	Err
NetMeeting	0.4893	[0.4635, 0.5151]	2.5757 x 10-2
Televox	0.4958	[0.4641, 0.5275]	3.1728 x !0-2
Webphone	0.6466	[0.6161, 0.6771]	3.0478 x 10-2

3.6.1.4 Rejecting the Conventional State Holding Time Model

It is traditional to model state-holding times in speaker activity models as exponential distributions, as discussed in section 3.2. To test the suitability of the traditional model for VoIP applications, the following null hypothesis were tested:

H_1: The distribution of the talk state is exponential.

H_2: The distribution of the listen state is exponential.

H_3 : The distribution of the state of mutual silence is exponential.

The Kolmogorov-Smirnov test [27] was used to access the goodness-of-fit of the collective (NetMeeting, Televox, and Webphone) empirical state holding time data, to exponential distributions. Table 7 summarizes the results of the test.

Table 7: Kolmogorov-Smirnov Results for Traditional State Holding Times

State	D	$D_{\alpha=0.05}$
Talk	0.2571	0.2240
Listen	0.2579	0.2240
Mutual Silence	0.7121	0.1674

Since the Kolmogorov-Smirnov test statistic, *D*, was greater than the critical value, D_α=0.05, for each state, each null hypothesis was rejected [27] with 95% confidence. This agrees with the results obtained by Deng [25] where a two-state continuous Markov chain was rejected as a model for packet voice.

3.6.1.5 Modeling State Holding Times

3.6.1.5.1 NetMeeting

A 3-state speaker activity model for a speaker using Microsoft NetMeeting communication software was developed.

3.6.1.5.1.1 Talk and Listen States

Since $c_x^2 = 4.2288$ for talk and listen state holding times, the following null hypothesis was tested:

H_0: The distribution of talk and listen state holding times follows a hyper-exponential distribution.

The Kolmogorov-Smirnov test was used to access the goodness-of-fit of the empirical data to a hyper-exponential distribution. Table 8 summarizes the results of the test.

Table 8: Results of Kolmogorov-Smirnov Test for NetMeeting

State	D	$D_{\alpha=0.01}$
Talk and Listen	0.25	0.269

Since the Kolmogorov-Smirnov test statistic, *D*, was less than the critical value, D_α=0.01, then the null hypothesis could not be rejected.

Therefore, talk and listen state holding times for a speaker using NetMeeting is modeled as:

$$f(t) = p_1\mu_1 e^{-\mu_1 t} + p_2\mu_2 e^{-\mu_2 t}, t \geq 0$$

A two moment match was achieved with the following parameters:

$$p_1 = \frac{1}{2}(1 + \sqrt{\frac{c_x^2 - 1}{c_c^2 + 1}},\ p_2 = 1 - p_1,\ \mu_1 = \frac{2p_1}{E(x)},\ \mu_2 = \frac{2p_2}{E(x)}$$

The values of the parameters are given in Table 9.

Table 9: Parameters for NetMeeting Talk and Listen State Holding Time Model

Parameter	Value
μ_1 (1/s)	2.8274
μ_2 (1/s)	0.3391
p_1	0.8929
p_2	0.1071

3.6.1.5.1.2 Mutual Silence

The mutual silence state holding time for NetMeeting was also modeled. Since the squared coefficient of variation for the state of mutual silence was c_x^2 = 4.2673. then the following null hypothesis was tested:

H_0: The distribution of mutual silence state holding times is hyper-exponential.

The Kolmogorov-Smirnov test was used to access the goodness-of-fit of the empirical data to a hyper-exponential distribution. Table 10 summarizes the results of the test.

Table 10: Results of Kolmogorov-Smirnov Test for NetMeeting

State	D	$D_{\alpha=0.01}$
Mutual Silence	0.1667	0.269

Since the Kolmogorov-Smirnov test statistic. *D*. was less than the critical value. D_α=0.01. then the null hypothesis could not be rejected.

Therefore. mutual silence state holding times for a speaker using NetMeeting can be modeled as:

$$f(t) = p_1\mu_1 e^{-\mu_1 t} + p_2\mu_2 e^{-\mu_2 t}, t \geq 0$$

where the parameters of the model are given in Table 11.

Table 11: Parameters for NetMeeting Mutual Silence State Holding Time

Parameter	Value
μ_1 (1/s)	20.9693
μ_2 (1/s)	1.7401
p_1	0.7531
p_2	0.2469

3.6.1.5.2 TeleVox

In this section, a 3-state speaker activity model for Voware Televox VoIP application is developed.

3.6.1.5.2.1 Talk and Listen States

Since $c_x^2 = 3.08726$ for talk and listen state holding times, the following null hypothesis was tested:

H$_0$: The distribution of talk and listen state holding times follows a hyper-exponential distribution.

The Kolmogorov-Smirnov test was used to access the goodness-of-fit of the empirical data to a hyper-exponential distribution. Table 12 summarizes the results of the test.

Table 12: Results of Kolmogorov-Smirnov Test for Televox

State	D	$D_{\alpha=0.01}$
Talk and Listen	0.2286	0.269

Since the Kolmogorov-Smirnov test statistic, *D*, was less than the critical value, D_α=0.01, then the null hypothesis could not be rejected.

Therefore, talk and listen state holding times for a speaker in a half-duplex Televox Internet telephone conversation is modeled as follows:

$$f(t) = p_1\mu_1 e^{-\mu_1 t} + p_2\mu_2 e^{-\mu_2 t}, t \geq 0$$

A two moment match was achieved with the following parameters:

$$p_1 = \frac{1}{2}(1 + \sqrt{\frac{c_x^2 - 1}{c_c^2 + 1}}, \ p_2 = 1 - p_1, \ \mu_1 = \frac{2p_1}{E(x)}, \ \mu_2 = \frac{2p_2}{E(x)}$$

The values of the model parameters are given in Table 13.

Table 13: Parameters for Televox Talk and Listen State Holding Times

Parameter	Value
μ_1 (1/s)	3.09992
μ_2 (1/s)	0.5160
p_1	0.8573
p_2	0.1427

3.6.1.5.2.2 Mutual Silence

Mutual silence state holding times were also modeled for Televox. Since the squared coefficient of variation for this state was c_x^2 = 1.5828, then the following null hypothesis was tested:

H_0: The distribution of mutual silence state holding times is hyper-exponential.

The Kolmogorov-Smirnov test was used to access the goodness-of-fit of the empirical data to a hyper-exponential distribution. Table 14 summarizes the results of the test.

Table 14: Results of Kolmogorov-Smirnov Test for Televox

State	D	$D_{\alpha=0.01}$
Mutual Silence	0.2571	0.269

Since the Kolmogorov-Smirnov test statistic, D, was less than the critical value, D_α=0.01, then the null hypothesis could not be rejected.

Therefore, mutual silence state holding times for a speaker in a NetMeeting Internet telephone conversation may be modeled as follows:

$$f(t) = p_1\mu_1e^{-\mu_1 t} + p_2\mu_2e^{-\mu_2 t}, t \geq 0$$

where the values of the model parameters are given in Table 15.

Table 15: Parameters for Televox Mutual Silence State Holding Times

Parameter	Value
μ_1 (1/s)	2.2886
μ_2 (1/s)	0.8145
p_1	0.7375
p_2	0.2625

3.6.1.5.3 Webphone

In this section, a 3-state speaker activity model for a speaker using Netspeak Webphone communication software is developed.

3.6.1.5.3.1 Talk and Listen States

Since $c_x^2 = 1.7944$ for talk and listen state holding times, the following null hypothesis was tested:

H_0: The distribution of talk and listen state holding times follows a hyper-exponential distribution.

The Kolmogorov-Smirnov test was used to access the goodness-of-fit of the empirical data to a hyper-exponential distribution. Table 16 summarizes the results of the test.

Table 16: Results of Kolmogorov-Smirnov Test for Webphone

State	D	$D_{\alpha=0.01}$
Talk and Listen	0.08571	0.269

Since the Kolmogorov-Smirnov test statistic, D, was less than the critical value, D_α=0.01, then the null hypothesis could not be rejected.

Therefore, talk and listen state holding times for a speaker in a Webphone VoIP conversation can be modeled as follows:

$$f(t) = p_1\mu_1 e^{-\mu_1 t} + p_2\mu_2 e^{-\mu_2 t}, t \geq 0$$

A two moment match was achieved with the following parameters:

$$p_1 = \frac{1}{2}(1 + \sqrt{\frac{c_x^2 - 1}{c_c^2 + 1}},\ p_2 = 1 - p_1,\ \mu_1 = \frac{2p_1}{E(x)},\ \mu_2 = \frac{2p_2}{E(x)}$$

The values of the parameters are given in Table 17.

Table 17: Parameters for Webphone Talk and Listen State Holding Times

Parameter	Value
μ_1 (1/s)	9.1473
μ_2 (1/s)	1.3944
p_1	0.4019
p_2	0.5981

3.6.1.5.3.2 Mutual Silence

The mutual silence state holding time for Webphone was also modeled. Since the squared coefficient of variation for this state was c_x^2 = 2.2124, then the following null hypothesis was tested:

H_0: The distribution of mutual silence state holding times is hyper-exponential.

The Kolmogorov-Smirnov test was used to access the goodness-of-fit of the empirical data to a hyper-exponential distribution. Table 18 summarizes the results of the test.

Table 18: Results of Kolmogorov-Smirnov Test for Webphone

State	D	$D_{\alpha=0.01}$
Mutual Silence	0.2286	0.269

Since the Kolmogorov-Smirnov test statistic, D, was less than the critical value, D_α=0.01, then the null hypothesis could not be rejected.

Therefore, mutual silence state holding times for a speaker in a Webphone conversations can be modeled as follows:

$$f(t) = p_1\mu_1 e^{-\mu_1 t} + p_2\mu_2 e^{-\mu_2 t}, t \geq 0$$

where the parameters of the model are given in Table 19.

Table 19: Parameters for Webphone Talk and Listen State Holding Times

Parameter	Value
μ_1 (1/s)	1.9580
μ_2 (1/s)	0.4222
p_1	0.9131
p_2	0.0869

3.6.2 4-State Speaker Activity Model

In this section, a 4-state speaker activity model is developed. The model is developed for the Microsoft NetMeeting application only.

3.6.2.1 Experiment

The same traffic captures used to develop the 3-state model were reanalyzed to develop a 4-state speaker activity model. The model was developed only for the NetMeeting VoIP application.

3.6.2.2 Defining States of Speaker Activity

Examining the traffic captures, four states of speaker activity were identified: *talk state*, *listen state*, *interrupt*, and *alternation silence*. The same definitions of talk and listen state defined for the 3-state model were applied to the 4-state model.

However, within the state of mutual silence, two sub-states were identified. A speaker enters a state of *interruption* when a packet of speech from the listening party interrupts their flow of speech. A speaker is defined to be in a state of *alternation silence* in the time between speech bursts in opposite directions. The four state speaker activity model is shown in Figure 14.

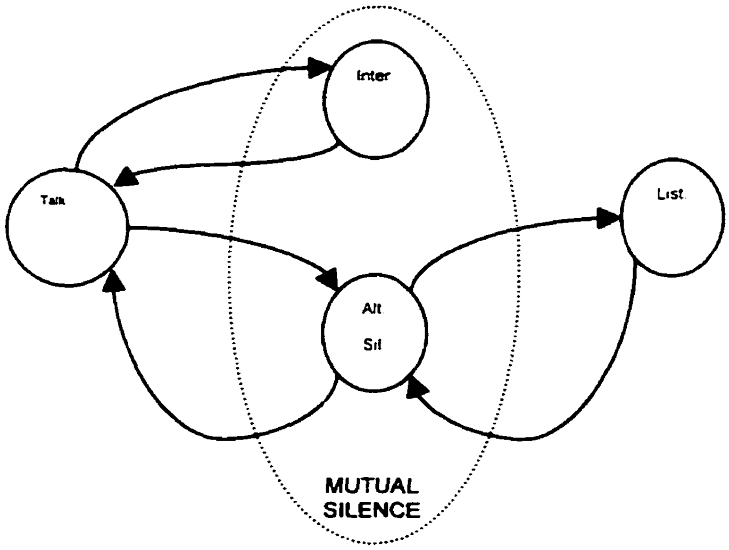


Figure 14: 4-State Speaker Activity Model

3.6.2.3 Modeling State Holding Times

A model for talk and listen state holding times was developed in section 3.6.1.5.1. This section presents models for the states of interruption and alternation silence.

3.6.2.3.1 Interrupt

Since $c_x^2 = 4.2366$ for interrupt state holding times, the following null hypothesis was tested:

H₀: The distribution of interrupt state holding times follows a hyper-exponential distribution.

The Kolmogorov-Smirnov test was used to access the goodness-of-fit of the empirical data to a hyper-exponential distribution. Table 20 summarizes the results of the test.

Table 20: Results of Kolmogorov-Smirnov Test for NetMeeting

State	D	$D_{\alpha=0.01}$
Interrupt	0.1923	0.264

Since the Kolmogorov-Smirnov test statistic, *D*, was less than the critical value, D_α=0.01, then the null hypothesis could not be rejected. Therefore, interrupt state holding times for a speaker in a NetMeeting Internet telephone conversation can be modeled as follows:

$$f(t) = p_1\mu_1 e^{-\mu_1 t} + p_2\mu_2 e^{-\mu_2 t}, t \geq 0$$

A two moment match was achieved with the following parameters:

$$p_1 = \frac{1}{2}(1 + \sqrt{\frac{c_x^2 - 1}{c_c^2 + 1}}, \quad p_2 = 1 - p_1, \quad \mu_1 = \frac{2p_1}{E(x)}, \quad \mu_2 = \frac{2p_2}{E(x)}$$

The values of the parameters are given in Table 21.

Table 21: Parameters for the NetMeeting Interrupt State Holding Times

Parameter	Value
μ_1 (1/s)	10.02798
μ_2 (1/s)	1.2004
p_1	0.8912
p_2	0.1088

3.6.2.3.2 Alternation Silence

Since c_x^2 = 2.2579 for alternation silence state holding times, the following null hypothesis was tested:

H₀: The distribution of alternation silence state holding times follows a hyper-exponential distribution.

The Kolmogorov-Smirnov test was used to access the goodness-of-fit of the empirical data to a hyper-exponential distribution. Table 22 summarizes the results of the test.

Table 22: Results of Kolmogorov-Smirnov Test for Alternation Silence

State	D	$D_{\alpha=0.05}$
Alternation Silence	0.1538	0.361

Since the Kolmogorov-Smirnov test statistic, *D*, was less than the critical value, D_α=0.01, then the null hypothesis could not be rejected. Therefore, alternation silence state holding times for a speaker in a NetMeeting Internet telephone conversation is modeled as follows:

$$f(t) = p_1\mu_1 e^{-\mu_1 t} + p_2\mu_2 e^{-\mu_2 t}, t \geq 0$$

A two moment match was achieved with the following parameters:

$$p_1 = \frac{1}{2}(1 + \sqrt{\frac{c_x^2 - 1}{c_c^2 + 1}},\ p_2 = 1 - p_1,\ \mu_1 = \frac{2p_1}{E(x)},\ \mu_2 = \frac{2p_2}{E(x)}$$

The values of the parameters are given in Table 23.

Table 23: Parameters for Alternation Silence State Holding Times

Parameter	Value
μ_1 (1/s)	4.3023
μ_2 (1/s)	1.0046
p_1	0.8107
p_2	0.1893

3.7 Chapter Summary

Traditional telephone conversation models have been found to be inappropriate for modeling VoIP applications. Three and four state models for three VoIP applications have been developed in this Chapter. These models could be used to implement a VoIP traffic simulator. The traffic simulator could be used to plan the network capable of supporting the anticipated demands of VoIP services.

4 Performance Metrics and Measurements

4.1 Introduction

This Chapter studies the performance of VoIP systems. The factors that affect voice quality are identified, and metrics to evaluate the performance of VoIP services are identified. Techniques to measure the performance of VoIP services, as well as recommended values for each performance metric, are also given.

4.2 Measuring Speech Quality

4.2.1 Category-Judgement Method

The category-judgement method is a subjective method [30] of measuring the quality of speech. The test procedure consists of two stages: a *familiarization stage* followed by an *evaluation stage*. During the familiarization stage, the quality to be associated with each category is presented to the listener. This is referred to as *anchoring*. During the evaluation stage, speech samples are presented to the listener, and the listener categorizes each sample.

The type of speech material used in the category-judgement method is recommended to be one of two types: (i) narrative, or (ii) short, phonetically balanced sentences (Harvard sentences). Recommended narrative material includes extracts from novels or newscasts. The speech material should be intelligible to the listeners, and each test should contain different speech material. A variety of talkers should be used in the test. The listener group should (i) represent the population that will be using the system (in terms of gender, age, technical background, socioeconomic background, etc.), and (ii) receive the same amount of training on the system as expected by the population that will be using the system.

4.2.2 Mean Opinion Score

A Mean Opinion Score (MOS) is a way of evaluating the results of a category-judgement test. After assigning a weight to each category, the MOS rating of a test signal is determined as follows:

$$MOS = \frac{n_i w_i}{N}$$

where:

n_i: the number of listeners that selected category i

w_i: the weight associated with category i

N: total number of listeners

It is recommended that between 10 and 50 listeners be used for each test [30].

4.3 VoIP Performance Metrics

4.3.1 Delay

Delay is one of the main causes of communication quality degradation in packet speech networks [33]. Delay is dependent on many factors, which are discussed in the following sections.

4.3.1.1 Sources and Type of Delay

Overall *end-to-end delay*, or *latency*, is defined as the time from the generation of a speech signal at the source to the time the signal is played out at the receiver. The overall end-to-end delay, D, is made up of the following components:

D = digitization + algorithmic + lookahead + processing + packetization + transmission + receiver buffer

Digitization delay is defined as the time it takes to convert the analogue signal to digital form. Digitization delay is dependent on the sampling rate and the speed of the system used to perform the A/D and D/A conversion. Digitization delay is usually small compared with the other sources of delay [32].

Speech compression algorithms operate on frames of speech. The delay introduced by buffering a frame of speech to be processed is referred to as *algorithmic delay* [32]. Algorithmic delay is a property of the particular algorithm, and will therefore vary between compression algorithms.

In addition to algorithmic delay, some compression algorithms may introduce additional delay as a result of the operation of the algorithm. For instance, the contents of one frame may be a function of the properties or contents of one or more preceding frames. Therefore an algorithm may buffer several frames of speech at a time. The delay introduced by such buffering is known as *look-ahead delay* [32]. Look-ahead delay is dependent on the particular algorithm and will therefore vary between algorithms.

Processing delay is the time required to operate on a frame of voice [32]. Processing delay results from the time required to encode the properties of the speech, as well as the time required to reconstruct the speech signal at the receiver. Processing delay is dependent on the speech processing system, in particular, the hardware used to process the speech. If implemented in software, the compression algorithm is executed using the system's resources and will therefore be limited by the system CPU speed, memory speed, operating system performance, etc.

Implementing compression algorithms in hardware on a dedicated DSP will reduce the processing delay. While hardware encoding may improve the quality of the voice call by reducing overall end-to-end delay, it also increases the cost of the system. Algorithmic complexity is a factor to consider when attempting to reduce processing delay in the system.

Packetization delay is the delay introduced by placing frames of voice data into packets for transmission over the network [32]. This delay factor is dependent on the implementation of the Internet telephone application; the application will place one or more frames of voice data into a packet. Packetization delay equals the packet time length plus protocol overhead. Therefore packet size (i.e. number of ms of seconds of encoded speech) is a factor in the end-to-end delay of a VoIP packet.

Transmission delay is defined as the time from when a packet is placed on the network for transmission until the time it is received at the destination. Transmission delay is the greatest contributor to the overall end-to-end delay in a VoIP call [32]. Transmission delay is highly variable and is dependent on a number of factors which are often not within the control of the VoIP network architect.

Two main factors that contribute to transmission delay are (i) the number of hops between source and destination, and (ii) the level of congestion in the network. As packets travel through a network, they are processed by and pass through many network elements or hops. A *hop* is defined as a layer three (or network layer) switching element (i.e. a router). As a packet passes through a router, it is placed in a queue for retransmission on the outgoing link. Factors that contribute to the transmission delay introduced by routers include: queue length, queuing algorithm, traffic congestion or arrival rate to the queue, and hardware capabilities (or processing power) of the router.

Another factor that affects transmission delay is the choice of protocol used at the transport layer. TCP is a connection oriented, reliable transport protocol. If TCP is used to transmit voice, delay will be introduced while the connection is being established. In addition the window size of a TCP connection will also contribute to the end-to-end delay. TCP can only transmit a certain number of packets, say x, without receiving an acknowledgement. After x packets have been transmitted, the source must stop and wait for the receive message. During this time, delay would be introduced into the stream from source to receiver.

UDP is a connectionless, unreliable protocol. No connection setup time is incurred in UDP based packet voice system. In addition, there is no transmission window. For these reasons, UDP is used in all VoIP systems to transport the voice signal.

Note also that transmission delay is also introduced by network elements at layers below the network layer; i.e. repeaters, bridges, and switches. However these devices generally introduce lower levels of delay as they involve less processing [34]. Delay is also introduced by gateways. Gateways may translate between protocols above the network layer. This is a computationally intensive procedure. If gateways are present in the VoIP call path, they will usually be a significant contributor to overall end-to-end delay [32].

4.3.1.2 Measuring Delay

The Internet Engineering Task Force (IETF) IP Performance Metrics (IPPM) Working Group (WG) identifies the end-to-end or one-way delay as a performance metric for IP networks. The IPPM WG defines one way delay as the time from when the first bit of a packet leaves the source until the last bit of the packet is received. The IETF definition of one-way delay is equivalent to the definition of transmission delay presented in the previous section.

Delay has been identified as a general metric for measuring IP performance, but it can be applied specifically to measuring the performance of an Internet telephony call. It has been shown for conventional telephony that communication begins to break down when the one-way delay exceeds 150 to 400 ms [33]. With respect to an acceptable delay threshold, note that a higher degree of delay may be tolerated for Internet telephony calls than for conventional telephony calls. It has been shown in other voice systems that the performance tolerance is higher when expectation of quality is reduced [33]. No studies on reduced expectations of VoIP quality are known to have been conducted; this is an open area left for future study. However it is reasonable to assume that similar results would be obtained in VoIP systems.

There are several issues involved when measuring the one-way delay of an IP stream: synchronization, accuracy, resolution, and skew. The IETF defines synchronization as the extent to which two clocks agree on the time [35]. Accuracy is defined as the extent to which clocks are in agreement with a reference source such as UTC [35]. Resolution is defined as the precision of a clock [35]. Skew is defined as a measure of the change of accuracy with time [35].

A methodology for measuring the transmission delay in an IP network is outlined by the IETF. The methodology is summarized below [35].

1. Synchronize the source and destination clock.

2. Transmit a packet from the source which includes a timestamp.
3. Timestamp the arrival of the packet.
4. Subtract the transmit time from the arrival time to estimate the one-way delay.
5. If the one-way delay time exceeds a threshold, report the delay as infinite.

This general measurement procedure could be adapted to measure the delay in VoIP systems. There are several characteristics that could affect the transmission delay of a packet as identified by the IETF [35]: the packet length, transport protocol used, port number, and QoS information. These factors should be taken into consideration when developing the test packet.

To measure the one-way delay within a VoIP system, the size of the packet should be chosen to reflect the typical packet size of the application that will be using the connection. Packet sizes are a factor of (i) the compression algorithm used, and (ii) how many frames the application groups into one packet. Neither of these factors are standardized for VoIP, therefore packet sizes vary between applications. Refer to section 4.4.1 for packet sizes used by Internet telephony applications.

The IETF [35] specifies that the contents of the packet should contain randomized data. This is to ensure that the packet will not undergo compression along the path from source to destination, which would alter the packet length and affect the delay measurement. For the purpose of measuring delay for Internet telephony, the contents of the packets should also contain randomized data. This can be accomplished by either filling packets with randomized or compressed data.

Port number may affect the end-to-end delay of a packet. For instance, firewalls filter packets based on their port number, which would subject packets to queuing delay based on the port number used in the application. The test packet should therefore use the same port number as the Internet telephony application in order to get an accurate estimate of the end-to-end delay of a particular path. The use of port number is not standardized in for VoIP applications, and varies between vendor. Refer to section 4.4.2 for port numbers used by VoIP applications.

The selection of transport protocol will affect the one-way delay of a packet. Refer to section 4.4.3 for transport protocols used in Internet telephony.

QoS marking will affect the delay of a packet. Therefore, the test packet should receive the same QoS marking as the VoIP application under test.

Finally, the IETF delay methodology gives an infinite delay to packets that are received outside the allowed threshold. This attribute exists because for time sensitive applications such as VoIP, receiving a packet outside a useful time interval is equivalent to not receiving the packet at all. The acceptable threshold was not defined, however, by the IPPM WG. For conventional telephony, if the one way delay exceeds 150 ms [33], then the communication begins to break down. Therefore, for VoIP, if a packet is not received within 150 – 400 ms of being transmitted, the packet should be counted as lost.

To summarize, an end-to-end or one-way delay metric has been defined by the IETF IPPM WG. Using the measured characteristics of VoIP applications (such as packet length, transport protocol, port number, etc.), this general methodology could be applied to measure the delay performance of a VoIP system by constructing a network test tool or simulator. The delay metric could be used by either the customer or the service provider to measure the performance of the Internet connection used to carry the voice call. If measured before the call, the metric could be used to negotiate for a different grade of service. If measured during the call. the metric could be used to dynamically adjust the quality of the call; i.e. give packets a higher QoS marking. or select an alternate route of lower delay.

4.3.1.3 Expected Delay in the Internet

How much delay is typically present in Internet connections? Andover.Net conducts measurements on various metrics relating to Internet performance. including *response time*. Response time is defined as a measure of the round trip delay; the transmission delay from point A to point B and back to point A [36].

One-way transmission delay could roughly be approximated as one half of response time. Graphs, re-reproduced with the permission of Andover.Net. showing long term response time measurements in North America, Asia. Australia, and South America are found in Appendix A. Using the measurements conducted by Andover.Net, an approximate 30 day average one-way transmission delay for each continent is given in Table 24. The one-way delay is approximated as one-half the response time. An approximate MOS rating for voice quality corresponding to each delay measurement. is also shown. based on an experiment documented in ITU-T G.114 [33].

Table 24: One-Way Delay Measurements and MOS Rating

Continent	**30 day average one-way transmission delay (ms)**
North America	93
Asia	253
Australia	300
South America	380

4.3.1.4 Effect of Delay on Voice Quality

The ITU-T Recommendation G.114 recommends values for one-way transmission time. ITU-T G.114 defines transmission time as the sum of propagation delay and processing delay. This definition is equivalent to our definition of one way end-to-end delay. or latency. The ITU-T makes the following recommendations for end-to-end delay [33]:

Table 25: ITU-T Conditional Delay Acceptability

End-to-end delay (ms)	Acceptability
0 – 150	Acceptable for most applications
150 – 400	Conditionally acceptable
Above 400	Generally unacceptable

The ITU-T G.114 recommendation states that depending on the application, delay not exceeding 150ms (one-way) is usually acceptable. Delay in the 150 to 400 ms range may or may not be acceptable; if users expectations are lower (i.e. satellite connection), a higher delay may be tolerated [33]. However, if one-way delay exceeds 400ms, the delay is too severe [33]. The ITU-T recommends that the processing delay component of end-to-end delay not exceed 50ms [33].

Several studies have been conducted on the effect of delay on the quality of voice communications. In a study reported in the ITU-T G.114 Recommendation, the effect of pure delay on the perceived quality of telephone connections was presented. In the study, delay of 0, 250, and 500 ms were injected into the end-to-end delay of a telephone connection. The perceived quality of the connection was measured by a MOS survey. The results of the experiment are summarized in Table 26 below.

Table 26: MOS Rating of Delay

One-way end-to-end delay (ms)	MOS rating
0	"good" (above 3.5)
250	"fair" (approximately 2.8)
500	"poor" (below 2.5)

As shown in Table 26, delay degrades the quality of the voice communication. Using the results of this experiment, Figure 15 shows the approximate correlation between delay measurements conducted by Andover.Net and the ITU-T corresponding MOS rating.

Continent	Corresponding MOS Rating
North America	Good
Asia	Fair
Australia	Fair
South America	Fair

Figure 15: Expected MOS Rating due to Delay

In addition to the challenges faced by traditional voice systems, delay introduces additional problems in packet switched voice systems.

Large end-to-end delay may allow parties to hear *speaker echo*. Speaker echo is reflected signal energy caused by an impedance mismatch in analogue telephony equipment. Components of a VoIP system may provide the opportunity to reflect signal energy. For instance, in a hybrid VoIP network (i.e. IP-to-PSTN), the voice call will pass through different types of networks along the path from source to destination. If analogue telephony equipment is present along the path, there may be an opportunity to produce speaker echo. It has been shown that if end-to-end delay exceeds 45-50 ms [37], speaker echo may be heard.

Speaker overlap is defined as both parties speaking at the same time. Speaker overlap degrades the quality of communication. The situation may occur in an Internet telephony call if, after having spoken, the speaker doesn't get a response within the expected time limit. In this event, one speaker may reiterate at the same time as the other party begins to respond. At this point, both parties are speaking at the same time, causing speaker overlap.

In the case of half duplex Internet telephony, speaker overlap would have the effect of clipping one another's speech. In the case of full duplex Internet telephony, both parties would be speaking and listening at the same time; i.e. interrupting each other. In either case, communication would break down as a result. It has been shown that if end-to-end delay exceeds 250 ms [37], speaker overlap may occur.

Conversational silences also convey information. If a non-natural period of silence is injected into a conversation, the context may be changed. Therefore, it is possible that end-to-end network delay may result in miscommunication.

4.3.2 Delay Variation

For VoIP, variation in delay, or *jitter*, is defined as the difference in one-way delay, or transmission delay, between two consecutive packets in a stream [38]. Jitter reduces the quality of voice communication [39].

4.3.2.1 Sources of Delay Variation

Jitter may be introduced by network elements between the source and destination that queue packets for transmission (e.g. routers). Jitter may also result from routing decisions. For instance, if two packets in a stream take different paths from source to destination, there may be a variation in the end-to-end delay between the two packets.

4.3.2.2 Measuring Delay Variation

Jitter, as defined above, is one of the metrics proposed by the IPPM WG as a general measure of IP performance. A methodology for measuring jitter (or inter-packet delay variation as termed by the IETF) is summarized below [38].

1. Measure the transmission delay for 2 consecutive packets
2. Calculate the IPDV by subtracting the transmission delay of the second packet from the transmission delay of the first packet.
3. If one or both of the packets fail to arrive within a reasonable time, then the IPDV is undefined.

This metric could be used to measure QoS for a VoIP connection by correlating statistics of the IPDV with a particular MOS rating. The metric could be used by either the customer or the VoIP service provider. The customer could use the metric to negotiate for a level of service. The service provider could use the metric to measure the level of quality experienced for billing purposes.

In terms of Internet telephony, correlating between jitter and MOS ratings (and identifying an acceptable level of jitter for a VoIP application) is left for future study. To perform such a study, the definition of *reasonable time* between packets (item 3 above) must be defined.

4.3.2.3 Effect of Delay Variation on Voice Quality

Jitter contributes to voice quality degradation in a number of ways. If the inter-packet time is variable but packets arrive within a reasonable interval, packets may be re-played, but the speech will sound jerky and unnatural. If the inter-packet time is excessive, the receiver may replay the sequence but drop the overdue packet. This event will have the same effect as a lost packet on the quality of the received speech; there will be gaps or holes in the speech signal which will degrade the quality and intelligibility of the communication. Refer to section 4.3.3 on the effect of packet loss on voice quality.

4.3.2.4 Combating Delay Variation

Jitter may be combated in VoIP systems by inserting a *jitter buffer* at the receiver, such as the Receive Path Delay component of an H.323 terminal [15]. A jitter buffer stores or buffers speech in order to remove the variable delay between packets so that speech can be replayed smoothly. However, this technique introduces receiver buffer delay which increases the overall end-to-end delay of the system.

4.3.3 Packet Loss

The IETF IPPM WG identifies packet loss as a performance metric for IP connections. *Packet loss* is defined as a packet not being received within a reasonable period of time [40]. Packet loss should also be included as a performance metric for VoIP. The metric is significant to the performance of Internet telephony connections as well as general IP connections. The amount of packet loss during a conversation directly affects the MOS rating of the communication.

This metric could be applied to Internet telephony in two ways. First, it could be used by customers to measure the quality of an expected connection before a call is made in order to assess whether or not to negotiate for better than best effort service. Second, the metric could be used to measure the actual quality of an Internet telephone call to determine if service contracts have been met.

4.3.3.1 Measuring Packet Loss

The IETF outlines a methodology for measuring packet loss. This methodology is summarized below [40].

1. Synchronize the source and destination clocks
2. Send a packet from source to destination that contains the departure time
3. Timestamp the packet upon its arrival at the destination.
4. Subtract the departure time from the arrival time.
5. If the packet arrives within the allowed threshold, count the packet as received.
6. If the packet fails to arrive within the allowed threshold, count the packet as discarded.

The methodology presented by the IETF for measuring packet loss is very similar to that presented for measuring end-to-end delay. In fact the only difference is what is reported as the value of the metric. For both tests the requirement is for a packet to arrive at the destination within an acceptable threshold.

The *threshold* should be the same as that which was identified for packet delay: 150 – 400 ms [33]. If the packet does not arrive within the tolerable threshold, then it is too late to be used. The following sections show the relationship between packet loss and MOS ratings.

4.3.3.2 Expected Packet Loss in the Internet

The graphs, shown in Appendix A, were reproduced with the permission of Andover.Net. These graphs show long term packet loss (intervals of 30 days) in North America, South America, Asia, and Australia. The average packet loss for each continent is summarized in Table 27.

Table 27: Approximate Average Packet Loss in the Internet

Continent	Average packet loss rate (%)
North America	2
Asia	7
Australia	5.5
South America	5.5

4.3.3.3 Effect of Packet Loss on Voice Quality

Packet loss results in the loss of information. As will be shown in the following sections, the amount of packet loss suffered by a VoIP application is proportional to the quality of the communication. In general as the amount of loss increases the quality of the communication decreases.

A popping or cracking sound results when a packet is dropped from a stream of speech packets. The artifact is a result of the discontinuity in amplitude between one segment of speech and the missing segment.

A packet that is small in terms of number of bytes may actually contain a large amount of information; i.e. contains compressed speech. Therefore it is not enough to express packet loss thresholds in terms of numbers of packets; knowing the amount of information per packet is also required.

4.3.3.4 Packet Loss Threshold for Acceptable VoIP Quality

An experiment was conducted to determine the acceptable packet loss threshold for VoIP. In the following sections the model for packet loss, the experimental procedure, and the results of the experiment are discussed.

4.3.3.4.1 Model for Packet Discard

It has been shown that the average number of consecutively lost audio packets is approximately 1 while network load is less than 80% of capacity [41]. This experiment modeled this level of network load (80%), and therefore this model of audio packet loss was assumed.

4.3.3.4.2 Listener Group

The listener group consisted of 16 people with no previous training on the system prior to the test. A survey on the background of the participants was conducted, and the results are given in Table 28.

Table 28: Background Survey

Background Item	Percentage of Listener Group
Age Group: 25-30 years	35.7
Age Group: over 35 years	64.3
Male	100
High School Education	14.3
Bachelor Degree	35.7
Graduate Degree	50
Technical Expertise: Novice	0
Technical Expertise: Intermediate	42.9
Technical Expertise: Expert	57.1

The majority of the participants were over 35 years of age, and had high levels of technical expertise and training. All participants were male. The characteristics of the listener group were consistent with the expected user population.

4.3.3.4.3 Implementation of System Model

To model the system, speech samples from books on tape were encoded in PCM format. The samples were encoded using the Windows 98 sound recorder, and saved in .wav format.

Packets were then selected for discard as described in section 4.3.3.4.1. The complete speech segment, including the lost packet, was then written to the test signal .wav file. Survey participants subjectively rated each signal via a category-judgement test and the results were analyzed with a MOS survey.

4.3.3.4.4 Optimal Packet Size at Average and Worst Case Loss Rates

Twenty 8-bit test signals were recorded at a sampling rate of 8000 Hz in PCM format. The samples were saved as .wav files. Ten samples were recorded in female voice and 10 were recorded in a male voice. The speech material in the test signals consisted of narrative material read from a novel. Each test signal consisted of different narrative material. Two rates of packet loss were applied to the test signals (2% and 6%) to represent average and worst case packet loss conditions expected in North America [36]. The results of the MOS survey for a loss rate of 2% are shown in Table 29.

Table 29: Optimal Packet Size at 2% Loss Rate

Test Signal	Voice	Packet Size (ms)	Packet Size (bytes)	MOS Rating
1	Male	10	80	2.5714
2	Female	20	160	3.2857

3	Male	30	240	2.6429
4	Female	40	320	2.7857
5	Male	50	400	3.5714
6	Female	60	480	3.2143
7	Male	70	560	3.5
8	Female	80	640	2.8571
9	Male	90	720	3.2857
10	Female	100	800	2.5714
11	Male	110	880	2.8571
12	Female	120	960	2.3571
13	Male	130	1040	2.5714
14	Female	140	1120	2.8571
15	Male	150	1200	2.7143
16	Female	160	1280	3.0
17	Male	170	1360	2.7857
18	Female	180	1440	2.8571
19	Male	190	1520	2.7143
20	Female	200	1600	2.5

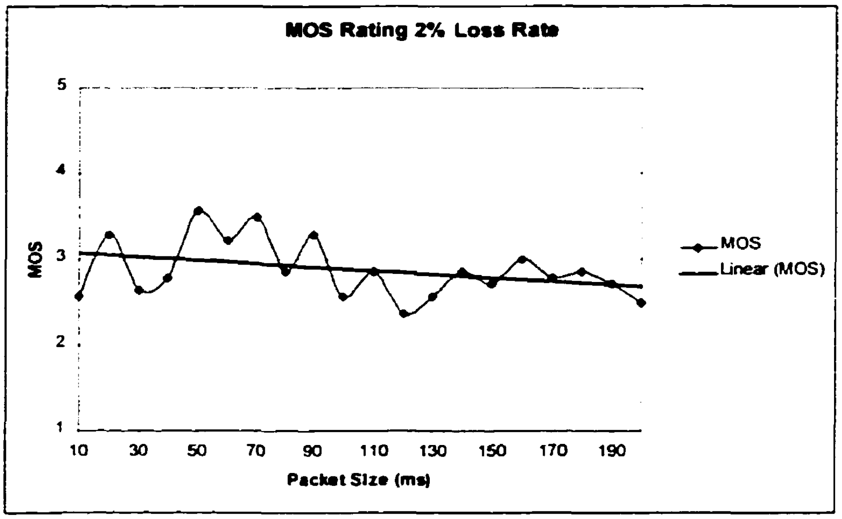


Figure 16: Results of MOS Survey for 2% Packet Loss

The results of the MOS survey for 2% packet loss are shown in Figure 16. A linear trend line is also shown in the graph. At the average rate of packet loss in North America, toll quality voice (i.e. MOS of 4.0) was not achieved. In general, as shown by the trend line, the larger the packet the lower the MOS rating. The highest MOS rating, 3.5714, was given to a packet size of 50 ms (or 400 bytes).

The experiment was repeated for loss rates of 6%; the worst case long term loss rate in North America. The results of the MOS survey are given in Table 30

Table 30: Optimal Packet Size at 6% Loss Rate

Test Signal	Voice	Packet Size (ms)	Packet Size (bytes)	MOS Rating
1	Male	10	80	2.4286
2	Female	20	160	2.2143
3	Male	30	240	1.9286
4	Female	40	320	2.5
5	Male	50	400	2.5
6	Female	60	480	2.4286
7	Male	70	560	2.8571
8	Female	80	640	2.7857
9	Male	90	720	2.0714
10	Female	100	800	3.0714
11	Male	110	880	2.2143
12	Female	120	960	2.3571
13	Male	130	1040	2.3571
14	Female	140	1120	2.1429
15	Male	150	1200	2.4286
16	Female	160	1280	2.7143
17	Male	170	1360	2.6429
18	Female	180	1440	2.3571
19	Male	190	1520	2.0714
20	Female	200	1600	1.9286

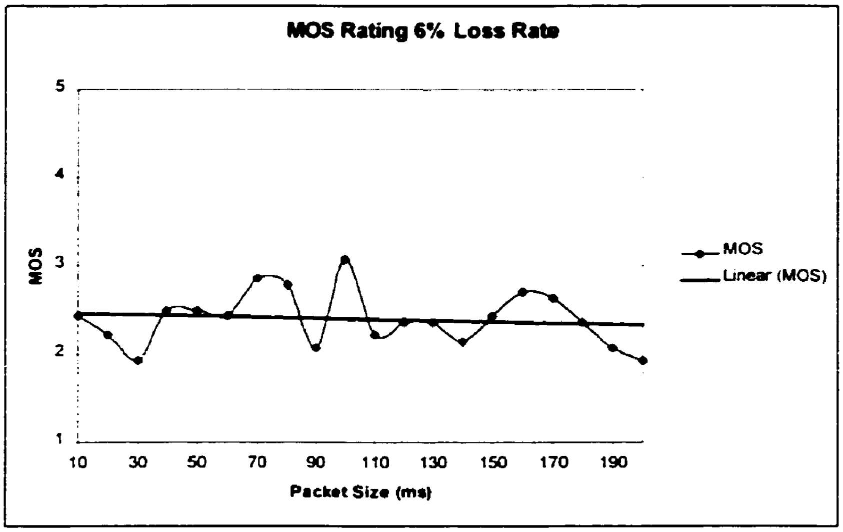


Figure 17: Results of MOS Survey for 6% Loss Rate

The results of the MOS survey for 6% packet loss are shown in Figure 17. A linear trend line is also shown in the graph. Under the worst case conditions of Internet packet loss in North America, toll quality voice (i.e. MOS of 4.0) was not achieved. Lower quality voice was achieved than at 2% loss rates. In general, as shown by the trend line, the larger the packet, the lower the MOS rating. The highest MOS rating, 3.0714, was given to a packet size of 100ms or 800 bytes.

4.3.3.4.5 Performance Under different Levels of Loss

An experiment was conducted to determine the MOS rating of speech under different levels of packet loss for constant packet size. Twenty test signals were recorded in PCM format (8000 Hz, 8 bit, mono). The signals were saved as .wav files. Ten of the signals were recorded in a male voice and 10 were recorded in a female voice. The speech material consisted of narrative material read from a novel, and each of the 20 signals consisted of different narrative material.

Test signals were divided into packets 120 ms (960 bytes) in length. To each test signal, packets were selected for discard following the model described in section 4.3.3.4.1. A MOS survey was conducted to measure the quality of each test signal, and the results are shown in Table 31.

Table 31: MOS Rating Under Different Levels of Packet Loss

Test Signal	Voice	Loss Rate (%)	MOS Rating
1	Male	1	3.5
2	Female	2	3.2857
3	Male	3	2.6429
4	Female	4	3.1429
5	Male	5	2.5
6	Female	6	3.0
7	Male	7	2.1429
8	Female	8	2.1429
9	Male	9	1.7857
10	Female	10	2.2857
11	Male	11	1.5714
12	Female	12	2.4286
13	Male	13	1.6429
14	Female	14	1.6429
15	Male	15	1.4286
16	Female	16	1.9286
17	Male	17	1.9286
18	Female	18	1.4286
19	Male	19	1.1429
20	Female	20	1.4286

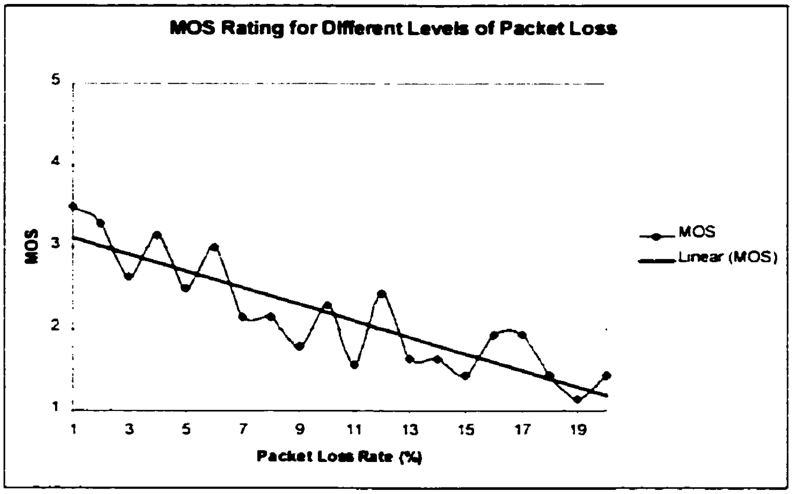


Figure 18: Results of MOS Survey for Different Levels of Loss

As expected, MOS ratings decreased with increasing rates of packet loss, as shown by the trend line in Figure 18. Even at only 1% packet loss, toll quality voice (MOS of 4.0) was not achieved. Based on the packet loss measurements shown in Table 27 and the results given in Table 31, Table 32 gives the expected MOS ratings for VoIP for different continents.

Table 32: Expected MOS Rating for Different Continents

Continent	Average Packet Loss (%)	Expected MOS Rating
North America	2	3 – 3.5
Asia	7	2 – 2.5
Australia	5.5	2 – 3
South America	5.5	2 – 3

4.3.3.5 Improving the Quality of Speech – Packet Recovery Techniques

As has been shown in the previous sections, packet loss degrades the quality of voice communication. In order to improve the quality of speech due to packet loss, the information contained in the missing packet must be replaced. Packet recovery techniques are a current area of study. This section reviews several techniques for improving the quality of speech in the event of packet loss.

The simplest method of replacing the information lost in the missing packet is to replace the packet with noise. This has been shown to be an improvement in quality compared to systems that merely play out the silent interval [37].

Another method of recovering the information is to replay the last correctly received packet in the place of the missing packet. This technique has been shown to be an improvement over replacing the packet with noise. This technique can be effective if packet loss is infrequent and occurs in non-continuous blocks [41]. While this technique fills in the missing time, it does not replace the missing information.

Another recovery technique is to carry redundant information about the n^{th} packet along with the $n+1$ packet. Several variations of this technique have been proposed. They range from carrying the entire preceding packet along with the next packet to carrying only properties of the n^{th} packet along with the $n+1$ packet. The redundant properties carried by the $n+1$ packet are usually created by a vocoder. The advantage of this technique is improved communication quality even under high degrees of packet loss [41]. However, end-to-end delay increases by the time required to encode the redundant information, which in turn affects the VoIP performance. In addition, the application bandwidth increases, which may contribute to congestion and eventually packet loss. Packet recovery techniques are currently an open area of research in VoIP.

4.3.4 Connectivity

Connectivity is defined as the ability for packets to be communicated between hosts within a meaningful interval of time [43]. This metric is significant to the performance of Internet telephony service as a conversation cannot occur unless the two participants are able to (i) exchange packets, and (ii) communicate packets during a time interval that is meaningful.

This section discusses how the general connectivity measurement procedure, outlined by the IETF, can be adapted to measure connectivity in VoIP systems. The metric can be used by service providers or customers to measure the performance of a VoIP connection.

4.3.4.1 Measuring Connectivity

The IETF IPPM WG gives a methodology for measuring the connectivity of two Internet hosts: host *A* and host B. The general methodology is summarized below [43].

1. Compute N sending times that are uniformly distributed in the interval of [T, T + dT – W] where N is the number of packets to be sent, T is the sending time of a packet, dT is the inter-departure time between packets and W is the waiting time. The waiting time is the time the receiver can wait for a packet before the packet becomes useless.
2. At each sending time, transmit a packet from address A to B.
3. Timestamp the arrival time of the response from B to A.
4. If no response is received by T+dT then conclude that the hosts are not connected.

The IETF recommends the following values for the algorithm: N = 20 packets. W = 10 seconds. and dt = 60 seconds. Using these values, a pair of internet hosts are considered to be connected if a response to a packet is received within 50 seconds.

A waiting time of 10s is too great for VoIP connections. Knowing that communication begins to break down in voice conversations when the one way delay exceeds 150-400 ms [33], the following algorithm could be applied to determine if an acceptable communication channel can be established between two Internet telephony hosts.

1. Send N packets from the caller to the callee.
2. Upon receipt of a packet, return the packet to the sender.
3. If a response has not arrived at the sender within 2x[150-400ms], conclude that the hosts cannot be connected in a manner that will provide an acceptable level of communication quality (in terms of delay).

The IETF test was designed to be a general test for connectivity of two Internet hosts. The following factors should be taken into consideration when designing a test or measurement tool for VoIP.

Packet lengths in Internet telephony calls vary in length depending on the compression algorithm used. The packet length should be chosen to reflect the typical packet length of the application. Refer to section 4.4.1 for packet lengths of VoIP applications.

Internet telephony packets usually contain compressed data. The contents of the packets used in the test should contain either compressed or randomized data, as previously discussed in the section on delay metric, so as not to affect the delay measurement.

As Internet telephony voice is transferred using UDP, the test data should also utilize UDP as the transport protocol.

The waiting time is defined as the time in which receiving a reply to a packet is still meaningful. The waiting time recommended by the IETF is W=10 s. Ten seconds is too long to wait for reply during a telephone conversation. At most, a reply to a packet should be received with 400ms of being transmitted.

A significant omission of the IETF connectivity test is interpreting the results. If a response to a packet is received within the window of significance, then the hosts are connected. If the response to a packet is not received within the window of significance, the value of the connectivity metric is false. If one packet out of 20 is marked as true then are the two hosts connected? If only 19 of 20 packets are received, are the hosts disconnected? VoIP can tolerate a certain amount of packet loss and still achieve acceptable levels of quality, as discussed in section 4.3.3. The interpretation of the metric should be redefined for VoIP, taking into account the percentage of packet loss and the amount of information contained in each packet.

In summary, the connectivity metric is an important metric to be considered for measuring the quality of an Internet telephony connection. However, the methodology for measuring the value of this metric, as defined by the IETF, is too general to be applied to Internet telephony. The connectivity metric could be used by either the customer or the service provider to determine a suitable path that for the VoIP call.

4.3.5 Call Set Up and Tear Down Time

Call setup time is the time required to initiate the communication between caller and callee. Call setup includes the dialing stage. capabilities exchange, and algorithm negotiation. Call tear down time is the time required to perform the signaling necessary to release the voice connection. The majority of Internet telephone applications exchange control packets before voice data is transmitted and after one of the applications releases the connection.

Call setup time should be included as a performance metric for VoIP systems. This section presents the results from an experiment in which the call setup and tear down times for several VoIP applications were measured.

4.3.5.1 Measuring Call Set Up Time

An experiment was conducted to measure Internet telephone call setup and tear down times. Call setup time shall be defined as the time from when a source application dials the destination address (i.e. transmits the first control packet) until the time the first voice data packet is received by the destination application. Call tear down time shall be defined as the time from when the first tear down packet is sent from one host to the other until no further packets are exchanged between hosts.

4.3.5.1.1 Experiment

Call setup and tear down measurements were taken for three Internet telephone applications: VocalTec Internet Phone, Microsoft NetMeeting, and NetSpeak Webphone. The experiment was conducted between two PCs on a 10Base-T LAN. For each application ten connections were established between source and destination. Voice traffic was generated using books on tape as the input signal. All traffic was captured using the UNIX utility *snoop*. The source application both initiated and released the connection between peer Internet telephones.

Before the Internet phone connection was established, *snoop* was set to capture all traffic exchanged between the source and destination PCs. It was observed that no traffic was exchanged between the two IP hosts before the connection was established.

Once the Internet phone connection had been established, but before voice signals were injected into the system, Microsoft NetMeeting and NetSpeak Webphone exchanged traffic. It was also observed that during connection tear down time, NetMeeting and Webphone also exchanged traffic. This indicates that these two applications have a call set up and tear down phase during which control data is exchanged. It was observed that the control data exchanged during these periods was transported using TCP.

VocalTec Internet Phone, however, has neither a connection setup or tear down phase. No traffic exchange between the source and the destination was observed other than when voice signals were input into the system.

4.3.5.1.1.1 Webphone

It was observed that Webphone exchanges six 512 byte TCP packets between the source and destination during the call setup phase of the connection. It was also observed that four packets were exchanged during the call tear down phase. Table 33 and Table 34 present call setup and tear down measurements, respectively.

Table 33: Webphone Call Setup Time

Call Setup Statistic	Value (s)
Minimum	1.9416
Maximum	2.0672
Mean	2.0124
Standard Deviation	0.0441

Table 34: Webphone Call Tear Down Time

Call Tear Down Statistic	Value (s)
Minimum	0.03404
Maximum	0.05971
Mean	0.03747
Standard Deviation	0.00785

4.3.5.1.1.2 Microsoft NetMeeting

NetMeeting exchanged a greater number of control packets than did Webphone during the call setup phase. Unlike Webphone, the number of packets exchanged was not constant between calls. Table 35 and Table 36 give the call set up and tear down statistics, respectively.

Table 35: NetMeeting Call Setup Time

Call Setup Statistic	Value (s)
Minimum	2.4744
Maximum	3.7923
Mean	3.2021
Standard Deviation	0.3588

Table 36: NetMeeting Call Tear Down Time

Call Tear Down Statistic	Value (s)
Minimum	0.4386
Maximum	1.3853
Mean	0.8438
Standard Deviation	0.2606

4.3.5.1.1.3 VocalTec Internet Phone

No explicit exchange of call setup or tear down control traffic was observed for Vocaltec Internet Phone clients. If control information is exchanged during an Internet Phone call. it is embedded within the voice packet.

4.3.5.2 Effect of Call Setup Time on VoIP Quality

From the customer's perspective. call setup time affects the communication process as voice communication is delayed for this period of time. Excessive call setup delay may be considered as degraded system performance by users.

The time required to setup and tear down a VoIP call are also of importance to the service provider. Resources are used during these intervals that cannot be used to carry other voice calls, which will affect the billing system as well as network capacity planning.

4.3.6 Voice Coding Algorithms

The Internet is a digital transmission medium. Before voice can be transmitted over the Internet, the analogue signal must be converted to digital form. The process is referred to as *digitization* and was discussed previously in Chapter 1. There are two main classes of voice coding algorithms: waveform encoders, and vocoders. The algorithm used to encode the voice signal affects the quality of VoIP communication and should therefore be included as a performance metric for VoIP systems. This section discusses the affect of codec on voice quality and presents the algorithms used by VoIP applications.

4.3.6.1 Waveform Encoding

Waveform encoding attempts to reproduce the shape or amplitude of the voice signal in time. There are two main factors that contribute to the quality of waveform sampled speech: sampling rate and number of quantization levels. In waveform encoding, the analogue signal is sampled at discrete time intervals. The closer the samples in time, the better the representation of the original waveform [44].

The other main factor affecting waveform encoded speech is the number of quantization levels. Although the original signal contains an infinite array of amplitudes, only a finite number of these can be represented digitally. The difference between the amplitude of the original analogue signal at some time and its digital representation introduces an artifact known as *quantization error* or *quantization noise*. In other words, the shape of the digitized signal will not be exactly the shape of the original. Quantization noise gives the speech a raspy sound; the speaker's voice sounds hoarse [44]. These artifacts are introduced at the source in a VoIP system.

Quantization error can be reduced by representing the original signal by a larger number of discrete levels. However, this increases the bit rate required to represent the signal.

Waveform encoders are more tolerant to transmission errors than other classes of algorithms [8]. VoIP systems that transmit speech using other techniques should expect reduced quality at the same error rates, or lower packet loss thresholds to achieve the same level of quality. Waveform encoding methods that are subject to these types of artifacts include PCM, DPCM, Delta Modulation.

4.3.6.1.1 Vocoder Methods

Vocoders attempt to represent the original waveform by its properties rather than its shape in time. The shape of the waveform in time is reconstructed at the receiver based on these properties. In general, vocoder methods produce speech that is of lower perceived quality than waveform encoders [44]. However, vocoders typically require lower data rate or bandwidth to represent the speech signal [44]. For this reason, vocoders are frequency used in Internet telephony applications.

4.3.6.2 Internet Telephony Compression Algorithms

During the installation of an Internet telephone application, the user is typically required to indicate their Internet connection speed. Unless manually overridden by the user, the application then selects an encoding algorithm to meet the user's bandwidth limitations. Algorithms commonly used in Internet telephony applications, and their bit rates, are given in Table 37 [45].

Table 37: Properties of Internet Telephony Compression Algorithms

Algorithm	Type	Bandwidth (k bps)	Category
G.711	PCM	48, 56, 64	Waveform encoder
G.722	ADPCM	48, 56, 64	Waveform encoder
A-law	Compander	64	Waveform encoder
μ-law	Compander	64	Waveform encoder
G.723.1	ACELP	5.3, 6.3	Vocoder
G.728	LD-CELP	16	Vocoder
G.729	ACELP	8	Vocoder
G.729A	ACELP	8	Vocoder
GSM	RPE-LTP	13	Vocoder
TrueSpeech	DSP Group G.723.1	5.3, 6.3, 8.5	Vocoder
VSC 5	VocalTec proprietary	5	Vocoder
VSC 8	VocalTec proprietary	8	Vocoder

Home users with dial-up connections to their ISPs are typically limited to bandwidth in the range of 14.4 to 28.8 Kbps. For these users, an Internet phone application would likely select one of the vocoder algorithms. In an office environment, the user likely has access to a LAN bandwidth in the 10 to 100 Mbps range. In the case LAN users, the VoIP application may make use of a waveform encoder to take advantage of the higher available bandwidth and offer better quality voice. The algorithms supported by four Internet telephone applications are listed in Table 38.

Table 38: Algorithms Supported by VoIP Applications

Algorithm	Microsoft NetMeeting	VocalTec Internet Phone	NetSpeak Webphone	Voxware Televox
G.711 (A-law)	√	√		
G.711 (μ-law)	√	√		
G.723.1	√	√		
G.722	√			
GSM			√	
TrueSpeech			√	
VSC 8		√		
VSC 5		√		
MetaVoice				√

4.3.6.3 Affect of Compression Algorithm on Voice Quality

Speech quality is related to the bit rate of the signal. In general, the lower the bit rate the lower the perceived quality [8] [44]. Table 39 shows the relationship between speech compression bit rate and sound quality [46].

Table 39: Relationship between bit rate and speech quality

Bit rate (k bps)	Speech Quality
64 (or greater)	Broadcast
64 to 12	Toll
12 to 6	Communications
Below 6	Synthetic

The highest quality of speech is referred to as broadcast quality. An example of broadcast quality is the quality of a recording on a CD. Toll quality is the quality achieved in conventional telephone systems. Communications quality is characterized as being intelligible, but noticeably lower quality than toll due to distortion. Synthetic quality speech is intelligible, but sounds unnatural. [46]

An experiment was conducted to measure the perceived sound quality of audio codecs commonly used by Internet Telephony applications. The experiment isolated the effect of compression algorithm on voice quality before other network induced impairments were introduced into the system.

The quality of the signals were rated by a group of 16 participants (refer to section 4.3.3.4.2 for the background of the listener group). The test signals and results of the survey are shown in Table 40.

Table 40: MOS Rating of Internet Telephony Compression Algorithms

Test Signal	Voice	Audio Codec	MOS
1	Male	PCM	4.0714
2	Female	PCM	4.0
3	Male	True Speech	3.9286
4	Female	True Speech	3.7857
5	Male	GSM	4.0
6	Female	GSM	4.0
7	Male	A-law	3.7143
8	Female	A-law	4.4286
9	Male	u-law	4.4286
10	Female	u-law	4.0714

The results of the MOS survey show that the algorithms used in VoIP systems are capable of achieving toll quality voice. Quality limitations in VoIP systems are therefore due to network induced artifacts, such as delay and packet loss.

4.4 Measured Properties of VoIP Applications

This section presents measured properties of VoIP applications. These include packet length. source and destination ports. and transport protocols. While not performance metrics themselves, these measurements provide information required to conduct the performance measurements described in the previous sections.

4.4.1 Packet Length

An experiment was conducted to determine packet length distributions of Internet telephony applications as a function of compression algorithm. Packet lengths generated by three Internet telephones (VocalTec Internet Phone. Microsoft NetMeeting. and Netspeak Webphone) were measured.

Internet telephone connections were established between two PCs on a 10Base-T LAN. Books on tape were used as the source of voice signals. Traffic generated by the Internet phone applications was captured using the Unix utility *snoop*. For each of the Internet phones. three sessions of 20 minute recordings were captured. *Snoop* captured all IP traffic exchanged between the two PCs. The number of bytes of voice data contained in each packet was the object of the measurement; the protocol header lengths are not included in the report.

4.4.1.1 VocalTec Internet Phone

VocalTec Internet Phone allows the user to select the compression algorithm used by the application. There are five compression algorithms to choose from: GSM, TrueSpeech, µ-law, VSC 5 and VSC 8. VSC 5 and VSC 8 are VocalTec proprietary compression algorithms. Table 41 gives the results of the measurements.

Table 41: Internet Phone Packet Lengths

Compression Algorithm	Data Length (bytes)
GSM	147
True Speech	145
μ-law	145
VSC 5	129
VSC 8	129

Packet lengths generated by VocalTec Internet Phone are constant. The packet length does vary, however, between compression algorithms.

4.4.1.2 Microsoft NetMeeting

Microsoft NetMeeting also allows the user to select the compression algorithm used in the conversation. Packet length measurements were conducted while NetMeeting was configured with the following compression algorithms: CELP, G.723.1 (5.3 k bps), G.723.1 (6 k bps), G.711 (A-law), and G.711 (μ-law). The results of the measurements are shown in Table 42.

Table 42: NetMeeting Packet Lengths

Compression Algorithm	Packet Length (bytes)
CELP (4.8 kbps)	68
G.723.1 (5.3 k bps)	80
G.723.1 (6.4 k bps)	92
G.711 (A-law) (64 kbps)	276
G.711 (μ-law) (64 kpbs)	500

NetMeeting also generates packets of constant length. However, the length of the packet varies between compression algorithms.

4.4.1.3 NetSpeak's Webphone

Webphone does not allow users to manually configure the compression algorithm. The compression algorithm is automatically selected by the application as a function of processor speed. If a Pentium processor is detected, Webphone uses the TrueSpeech compression algorithm. When lesser processing power is detected, Webphone uses GSM to compress speech.

Since the measurement was conducted between two Pentium PCs, Table 43 shows the voice packet data length distribution of the Webphone voice packet lengths while configured to use TrueSpeech.

Table 43: Webphone Packet Length

Statistic	Value
Minimum packet length	177 bytes
Maximum packet length	189 bytes
Mean packet length	182.9492 bytes
Variation in packet length	2.7605 bytes^2

Unlike Internet Phone and NetMeeting, Webphone voice packet lengths are variable in length. Webphone may make use of some form of VAD (Voice Activity Detection) algorithm, and cut off the packet when silence is detected.

4.4.2 Source and Destination Ports

An experiment was conducted to determine which ports are used by Internet telephone applications. Observations were made of two Internet phone applications: VocalTec Internet Phone, and Microsoft NetMeeting.

For each application a connection was established between a source and destination phone 25 times. VocalTec Internet Phone used a consistent port number (port

22555) for both source and destination. NetMeeting used a variety of source and destination port numbers, all in the 49000 range, to communicate voice data. Observed port numbers used by NetMeeting were: 49604, 49605, 49606, 49608, 49609. None of the ports used by these applications are well defined ports [48].

4.4.3 Transport Protocol

An experiment was conducted to observe which transport protocols are used by Internet telephone applications for carrying voice and control data. The experiment was conducted on three Internet phone applications: VocalTec Internet Phone, Microsoft NetMeeting, and NetSpeak's Webphone.

It was observed that traffic exchanged before voice was injected into the system was transported using TCP. Once voice was input into the system, packets exchanged were transported using UDP. It was therefore concluded that control data is carried in TCP packets, while voice data is carried in UDP packets. Applications compliant with H.323 further encapsulate the voice data in RTP packets [15], but this was not explicitly observed with *snoop*.

4.5 Chapter Summary

In this chapter, metrics that can be used to measure the performance of VoIP systems were identified. To summarize, these metrics are: end-to-end delay, delay variation, packet loss, call setup time, and voice coding algorithm. Methodology for measuring each metric, the effect the artifact has on voice quality, performance recommendations for the metric were also presented. To summarize, it is recommended that end-to-end delay not exceed 150-400 ms, and that packet loss be less than 1% to achieve toll quality voice. Many of the compression algorithms used in VoIP are capable of achieving toll quality voice. It is recommended that the highest quality voice codec (in terms of MOS rating) that can be supported by the bandwidth of the connection be used in the VoIP system.

Measurements of the characteristics of VoIP applications were also presented in this Chapter. These characteristics included: packet size, transport protocol, port number, and compression algorithm.

The metrics and characteristics presented in this Chapter have application for both the customer as well as the VoIP service provider. The customer could use these metrics to assess the quality of the service in order to select a service provider. The service provider could use these metrics during call setup time to determine the path for the VoIP call that has the lowest packet loss, the shortest delay, etc. These metrics could be used in real-time by the VoIP equipment vendor to dynamically adjust the quality of the communication. Finally, these metrics could be used by either customer or carrier to communicate the quality of the communication experienced for the purpose of billing.

5 VoIP Network Architectures

This Chapter reviews the network architecture and components required to deliver VoIP services. Three main architectures and the components required to deliver VoIP service are presented.

The current IP network infrastructure does not provide QoS mechanisms required for real-time applications, such as VoIP. This Chapter presents QoS architectures that may be capable of providing the level of service required by VoIP applications.

5.1 VoIP Service Architectures

There are three basic VoIP service architectures. This section discusses each configuration, and identifies the network components required to deliver the service.

5.1.1 IP Host-to-IP Host Architecture

The original VoIP service architecture, and the simplest, is one in which two IP hosts communicate across an IP network. The IP host or workstation can be either a PC, MAC, or UNIX terminal. The terminal is equipped as follows: soundcard, speakers, microphone (or handset), network card, and a VoIP software application.

A *directory server* is also often a component of this basic architecture. The directory server performs two functions: first it allows users of the VoIP application to find one another, and second it performs address translation and call setup services. This VoIP service architecture is shown in Figure 19.

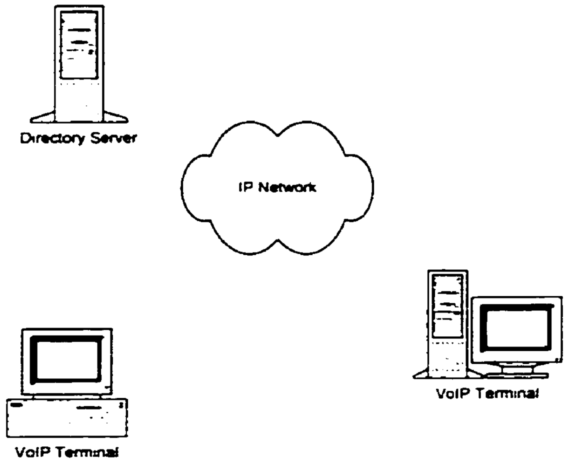


Figure 19: IP Host-to-IP Host VoIP Service Architecture

Home VoIP users who access the Internet through a dial-up connection to their ISP are assigned an IP address from a pool of addresses owned by their ISP. Dial-up users have no guarantee that their IP address will be the same between sessions. From the perspective of VoIP, this is the same as having a phone number that always changes. This makes it difficult for users to locate one another.

The directory server is a component of a VoIP system that provides a solution to this problem. The directory server provides the services of a conventional telephone book; it publishes the names and contact information of users. A directory server is typically provided by each VoIP application vendor.

Before placing or receiving a call, the application registers the user's current contact information with the director server. This is done when the application initializes. When one user attempts to place a call to another, the VoIP software uses the directory server to look up the current address of the callee, and then returns the contact information to the calling application. The caller uses this information to establish communications with the callee directly. The role of the directory server in VoIP architecture is shown in Figure 20 below.

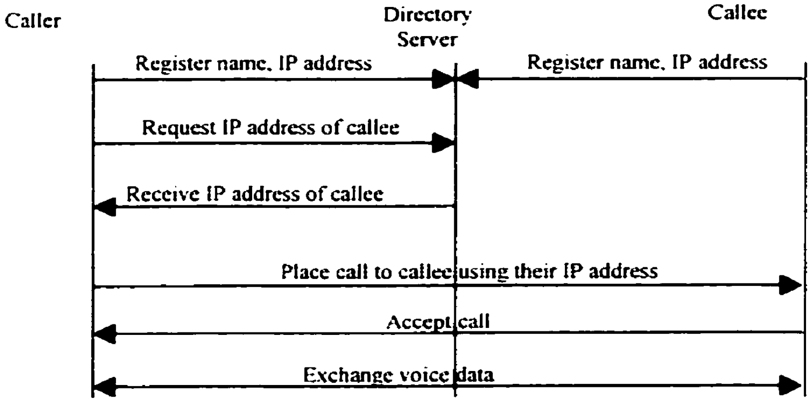

Caller
Directory Server
Callee
Register name, IP address
Register name, IP address
Request IP address of callee
Receive IP address of callee
Place call to callee using their IP address
Accept call
Exchange voice data

Figure 20: Role of the Directory Server

Some vendors publicize their database allowing users to look up the contact information of other users, similar to the conventional telephone book. Other vendors maintain a database, but do not publish it in order to protect the privacy of their customers. In the latter case, the only way a user can receive a call is if they have given their address to another user, analogous to unlisted telephone numbers.

The IP host-to-IP host VoIP architecture is common. A few of the more popular VoIP software applications which make use of this architecture include: Microsoft NetMeeting, VocalTec Internet Phone, and NetSpeak Webphone. NetMeeting and Internet Phone provide directory services and publish the names of their clients in their public servers. Webphone also maintains database of contact information, but does not publish the database. Webphone users are required to know the name or alias of the peer Webphone client they wish to contact.

5.1.2 IP Host-to-Conventional Phone Architecture

The second generation of Internet telephony service architectures is one in which a user. running an Internet telephony application (or dedicated VoIP hardware) on a workstation. is able to place calls to a conventional telephone. The architectural components required to deliver this service include: an IP host with VoIP software or hardware. conventional telephone. IP network. conventional telephone network. and an IP-to-PSTN gateway (or network of gateways).

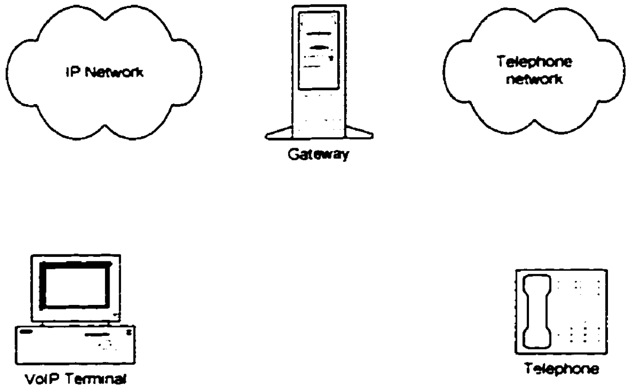


Figure 21: IP Host-to-Conventional Phone VoIP Components

A gateway is required in this service architecture since the endpoints of the communication reside on different network types. The gateway provides an interface to each type of component.

This VoIP service architecture provides a form of toll bypass, and companies that provide this service are referred to as NextGen telcos. NextGen telcos offer cost savings to customers of long distance telephone services. Two of the most well known NextGen telcos are The FreeWorld Dial-up Project and Net2Phone. The FreeWorld Dial-Up

Project was the first example of this form of VoIP service, and is offered as a free service world wide. Net2Phone is a commercial NextGen Telco.

In order to deliver this form of VoIP service, a network of gateways must be deployed. The NextGen telco places a gateway in each location that a connection to the conventional telephone network is required. The VoIP call initiated in one location is routed over the Internet to the gateway in the nearest vicinity of the destination of the call. The gateway then interfaces with the public telephone network in that area and completes the last leg of the call on the local phone network.

From the customer's perspective, this service can offer substantial savings to long distance customers over the cost of placing the same call using a conventional long distance carrier end-to-end. However, the caller must initiate the call from their desktop VoIP application, which may not be as convenient as a conventional handset.

5.1.3 Phone-to-Phone over IP Service Architectures

The third class of VoIP service architecture is one in which communication occurs between two conventional telephones, but the call is carried by an IP network rather than a conventional, circuit switched telephone network. Since the endpoints of the call are conventional telephone handsets, this is the ideal architecture for delivering VoIP service from the customer's perspective; the user interface is familiar and ubiquitous. There are two variations to this service architecture. One architecture uses a private IP network to carry the call, and the other architecture uses the public Internet as the carrier.

5.1.3.1 Private IP Carrier Network

The private IP network carrier architecture would be typically used between two sites of a corporation using the existing IP data network. An interface between the corporation`s private phone network and IP data network is provided by an IP-to-phone gateway. In this case, a gateway would be required at each end of the connection. A call originating from one conventional telephone would hop off the private phone network to the IP network through the gateway. The call would then be routed to the peer gateway at the destination location over the private IP network. At the destination location, the second gateway would provide the interface between the IP network and the private phone network and would complete the last leg of the call.

To build a private phone-to-phone over IP network, the following architectural components are required: a phone-to-IP gateway at each end location, IP network, and a corporate database mapping phone numbers to gateways. The database could be located in the gateway, or as a stand-alone component. This architecture is shown in Figure 22.

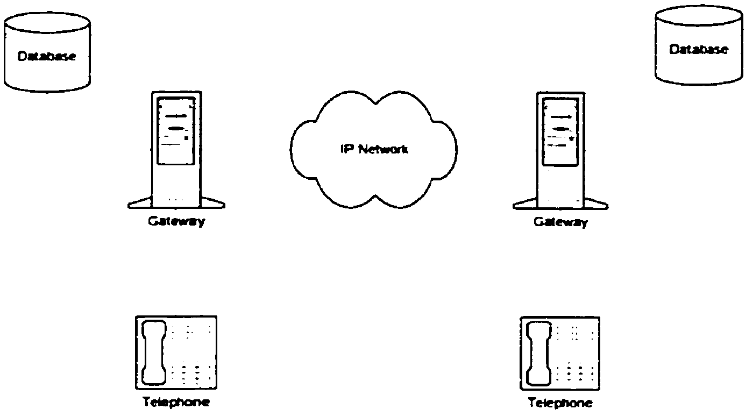


Figure 22: Private Phone-to-Phone over IP Service Architecture

Integrating voice services into a corporation's IP data network may provide cost savings, as only one network needs to be built and maintained. Another advantage of this service architecture is that since the IP network that is interconnecting the sites is private, some degree of control can be exercised with respect to the quality of the communication.

5.1.3.2 Public IP Carrier Network

This architecture is equivalent to the private network described in the previous section, except that the phone terminals reside on the public telephone network rather than a private network, and the IP network used to carry the calls is the public Internet.

This service architecture is another variant of a NextGen telco. The components of this VoIP service architecture would include IP-to-phone gateways, a database mapping phone numbers to gateway locations, and billing infrastructure. The database and billing components may exist in the gateway. Figure 23 below shows the service architecture and components.

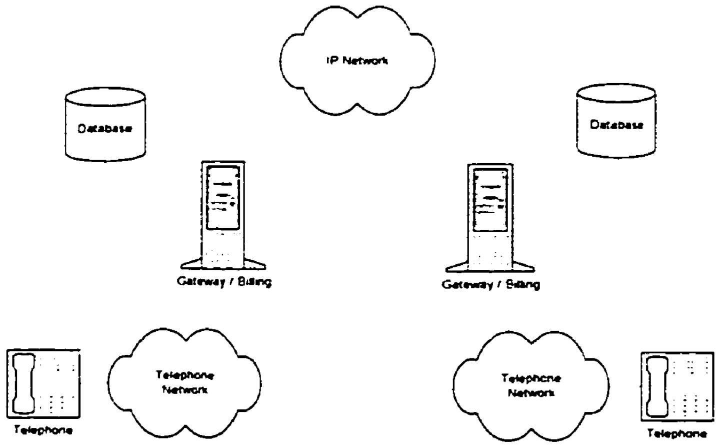


Figure 23: Public VoIP Service Architecture

Like the private phone-to-phone network, this VoIP service architecture has the advantage of using the familiar and ubiquitous telephone handset as the interface to the service. With the proper deployment of gateways, the service can be offered to anyone with a conventional telephone. Since the IP network used to carry the calls is publicly owned. limited startup and maintenance cost is required of the service provider. The service provider can in turn pass on the saving to the customer in the form of low rate long distance service.

A disadvantage of using the public IP network as the carrier is that the current infrastructure provides only best effort QoS. The quality of the call cannot be controlled by the service provider, which results in lesser quality than can be offered by the conventional telephone network.

5.2 QoS Architectures

Currently traffic in an IP network is provided with best effort service. Best effort service is acceptable for services that do not have real-time requirements. such as FTP transfers. As shown in Chapter 4, however, the quality of VoIP service degrades with latency, jitter, and packet loss, none of which can be controlled by best effort service.

In order to provide VoIP packets the special handling required by the service. IP networks must perform three functions: mark packets for special handling, recognize packets requiring special handling, and deliver preferential service to marked packets.

This section discusses alternative architectures for building QoS mechanisms into an IP infrastructure. There are two main classes of service architectures: those that provide service on a packet by packet basis (packet marking QoS), and those that provide service to a stream of packets (flow marking QoS).

5.2.1 Packet Marking QoS Mechanisms

5.2.1.1 QoS in an IPv4 Network

As discussed in Chapter 2, the IPv4 header contains a field for QoS marking: the *TOS octet*. The TOS octet, as defined in RFC 791, contains six fields: a 3-bit precedence field, a Delay (D) bit, a Throughput (T) bit, a Reliability (R) bit, and two unused bits. A cost (C) field was later added, as shown in Figure 24.

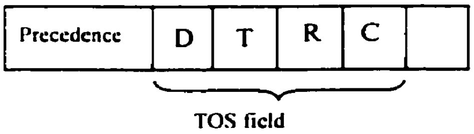


Figure 24: Original TOS Definition

The original definition of the TOS octet was later refined in RFC 1349 so that it contained a 3-bit precedence field, one 4-bit TOS field, and one unused bit. The revised TOS field is shown in Figure 25.

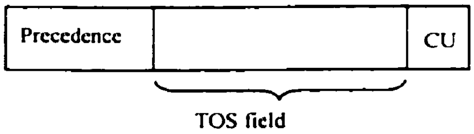


Figure 25: Revised TOS Definition

A *service marking model* is a quality of service model in which packets are assigned an absolute grade of service. The service marking model is implemented in IPv4 as the Type of Service (TOS) field. The TOS field allows packets to specify a level or grade of service. As defined in RFC 1349, the 4-bit TOS field value is interpreted as one value, rather than by the individual bits. Table 21 shows standard TOS values and their interpretations [47].

Table 44: Standard TOS Values

TOS Value	Interpretation
1111	Maximize Security
1000	Minimize delay
0100	Maximize throughput
0010	Maximum reliability
0001	Minimize monetary cost
0000	Normal service

RFC 1700 recommends TOS values for some common application and control protocols [48]. However, no recommendation has been made for VoIP service. Of the TOS values currently defined, packets belonging to VoIP streams should receive '1000' TOS markings to request low delay service.

The 3-bit precedence field in the IPv4 TOS field is an implementation of a *relative priority model*. The precedence field can take on values from 0 to 7 and is used mainly for congestion control mechanisms. In the event of congestion, lower priority packets are discarded before higher priority packets. As shown in Chapter 4, VoIP can tolerate minimal packet loss. Therefore, VoIP packets should receive a higher priority marking to avoid being discarded in the event of congestion.

The components required to provide QoS in an IPv4 architecture include IP hosts, markers, and QoS capable routers. In this architecture, it is the host node's responsibility to request a level of service (i.e. mark the TOS octet in the IPv4 header). A network administrator may therefore also wish to include a policer at the ingress to the domain to ensure that traffic contracts are not violated. The service architecture is shown in Figure 26.

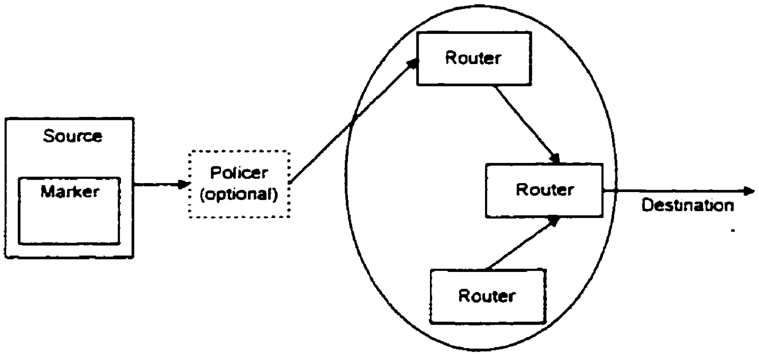


Figure 26: IPv4 Service Architecture

5.2.1.1.1 Implementations of IPv4 QoS

The TOS octet has been part of the IP standard since the definition of IPv4. Although not commonly used in the past, this service model is beginning to be implemented in current routing products.

Cisco products support the relative priority, service marking, and Integrated Services QoS architectures [49]. Using these products, network operators have control over marking or remarking the IP precedence and/or TOS fields. Cisco supports weighted Fair Queuing (WFQ) and Priority Queuing, each of which bases queuing decisions (at least in part) on the IPv4 precedence and TOS fields.

Despite QoS mechanisms starting to become availabe within the network infrastructure, IP packets carrying voice do not make use of this mechanism. Re-examining the capture files of Chapter 3, it was determined that each VoIP application marked the TOS octet as '00000000', or requested only best effort service.

One advantage of basing a QoS mechanism on IPv4 is that the architecture is realizable in terms of state-of-the art networking technologies. One disadvantage of this type of QoS mechanism is that QoS marking is left to the end user's discretion. Unless policing is implemented by the service provider at the ingress to the network, the service provider is at the mercy of the customer to not abuse QoS mechanisms or violate service level agreements.

5.2.1.2 QoS in an IPv6 Network

IPv6 contains an 8-bit *traffic class* field. As discussed in Chapter 2, the traffic class field serves a similar purpose to the TOS field in the IPv4 header; it can be used to distinguish between different classes of packets [10]. The Traffic Class field can be set by both source nodes and intermediate routers. The architectural components required to deliver QoS in an IPv6 network include hosts or nodes capable of traffic class marking, IPv6 network, and intermediate nodes capable of traffic class marking and forwarding. Like IPv4, since customer's have the ability to mark packets, a service provider may also wish to include a policer at the ingress to an IPv6 network to ensure that customers do not violate traffic contracts.

IPv6 networks do not currently exist outside research or academic communities. Once in place, however, the traffic class field could be used to deliver QoS to VoIP applications.

5.2.1.3 Differentiated Services

Differentiated Services (DS) is an emerging service architecture developed by the IETF [50], intended to replace the TOS field in the IPv4 header. Differentiated Services was designed primarily to (i) provide a QoS mechanism for IP networks, and (ii) to provide QoS in a scalable manner. This section introduces the DS architecture, and discusses how the architecture can be applied to deliver QoS to VoIP.

5.2.1.3.1 Architecture

The DS architecture is composed of the following components: Per Hop Behaviors (PHB), packet classifiers, traffic conditioners, boundary nodes, and an IP network. Traffic conditioners may contain traffic meters, traffic markers, traffic shapers, and traffic policers [50].

The Differentiated Services architecture changes the IP header. The 8-bit DS field is intended to replace the TOS octet in the IPv4 header, and the traffic class octet in the IPv6 header. The proposed DS field is shown in Figure 27 below [51].

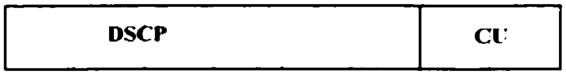


Figure 27: DS field

The DS field contains two sub-fields: (i) a 6-bit differentiated services code point (DSCP) field, and (ii) a 2-bit currently used (CU) field. The DSCP value is used to identify the type of forwarding behaviour that is to be applied to packets. Currently only one type of forwarding behaviour has been defined for the DS architecture: default or best effort forwarding [51]. Two other forwarding behaviors have been proposed (expedited forwarding, and assured forwarding), but these have not yet been standardized.

Expedited forwarding (EF) is a proposed PHB that provides at least the minimum specified departure rate to EF marked packets [52]. EF PHB can be implemented with several queuing algorithms: priority queuing (PQ), weighted round robin (WRR), and class based queuing (CBQ) [52]. In initial experimentation conducted by the IETF, it was determined that PQ provides the lowest level of jitter [52]. VoIP packets should receive an EF marking in a DS domain.

The assured forwarding (AF) PHB is a proposed forwarding behaviour that provides different levels of assurance that packets will be forwarded within a DS domain [53]. Currently four levels of forwarding assurance have been defined [53]. Within each of the four levels, a packet is also marked with a priority level. In the event of congestion, lower priority packets are discarded before higher priority packets.

As traffic enters a DS domain, the boundary nodes classify each packet and assign the packet a type of forwarding behaviour indicated by the DSCP value in the IP header. Each node within a DS domain maintains a table, which provides a mapping between DSCP values and the PHB to be applied to a packet with that marking. Each DS domain must contain the recommended default DSCP-to-PHB mappings, and may contain local DSCP-to-PHB mappings.

The Differentiated Services architecture achieves scalability by (i) not requiring per-flow information to be maintained at each intermediate node in a network, and (ii) requiring that the majority of the marking and processing to be done at the boundary nodes to a DS domain. It is valid for sources to mark their own packets and for interior nodes to have some marking and conditioning functionality. However, moving this complex functionality away from the boundary nodes limits the scalability of the service architecture.

5.2.1.3.2 Issues with Differentiated Services

Currently only one PHB has been defined: default forwarding behaviour. In order to provide IP QoS in a DS architecture, defining PHBs that provide better than best effort service is required.

It is not desirable to rigidly map PHBs to DSCP values for two reasons. First, the Internet community currently has a limited understanding of PHBs, therefore it is too early to provide a standardized mapping. Second, providers may want to define their own code points for PHBs. Therefore, the IETF proposes local DSCP to PHB mapping within each domain [50].

However, not rigidly providing PHB-to-DSCP mappings raises the issue of how to communicate the requested PHB between adjacent domains. Each domain may associate different PHBs with DSCP values.

There are currently two proposed methods for mapping DSCP values between DS domains. The first proposal is to have adjacent domains agree on a static mapping of PHB to DSCP code point values ahead of time [50]. This solution has scalability problems. However, it may be necessary to use static mappings between domains in the first generation of the DS architecture, but this should not be adopted as a long term solution. A second proposal is to develop a dynamic protocol that will perform the mapping [50]. No such protocol currently exists.

Interoperability with other QoS mechanisms is another open issue in IP QoS [50]. First, in order to provide end-to-end QoS, DS domains may need to communicate with domains which implement a different QoS mechanism (flow based QoS for example). This mapping has not yet been defined.

5.2.2 Flow Marking QoS Mechanisms

5.2.2.1 Integrated Services

The Integrated Services architecture is a QoS architecture that is based on the ability to identify flows of packets. Special handling is applied to packets belonging to a particular flow. The behaviour to be applied to a flow is signaled ahead of time by a reservation protocol. For example, in setting up a VoIP connection, a reservation protocol might signal to the intermediate routers to reserve 64 kbps bandwidth for all packets belonging to the flow marked *x*. As the VoIP packets belonging to flow *x* arrive at the router, the router would provide them with preferential handling.

There is no field in the IPv4 header to mark packets as belonging to a flow. RSVP is used in IPv4 network to identify a flow, and to signal resource requirements for that flow to the routers along the path from source to destination [54]. RSVP uses a combination of IP address and port number to identify a flow to a router.

IPv6, however, contains a *flow label* field [10]. The flow label is a 20-bit field that is used to identify all packets that belong to the same flow or stream. The behavior to apply or the resources to deliver to a packet is not conveyed by the flow label: expected service information must be conveyed with an additional protocol, such as a resource reservation or control protocol.

The architectural components of an Integrated Services network include end hosts, intermediate nodes capable of maintaining resource reservation information for each flow. flow marking components, and a resource reservation signaling protocol (for example. RSVP).

5.2.2.2 Issues with Integrated Services

The main issue with adopting an Integrated Services QoS architecture is that the architecture does not scale well. For each flow that requests special handling. that handling information must be stored in each intermediate node from source to destination. Routers are only capable of maintaining state information for a finite number of flows.

Specifically in terms of VoIP applications, another disadvantage of this QoS architecture is that it requires signaling to set up the channel from source to destination. This increases call setup time for VoIP applications. In addition, resource reservation information is stored in *soft states* in the intermediate routers. This means that QoS information must be updated periodically which requires additional signaling. There is neither call setup time or maintenance signaling required in a DS type architecture; QoS information is carried within each packet independently. In this respect. DS based architectures are better suited for delivering QoS to VoIP services.

5.3 Chapter Summary

There are three main architectures that can be used to deliver VoIP service: IP host-to-IP host, IP host-to-phone, and phone-to-phone service across an IP network. IP host-to-IP host was the original VoIP service architecture, and remains the simplest and least expensive to implement. This architecture has limitations, however. First, the end-terminal is not a ubiquitous interface, and therefore accessibility to the service may be limited. Second, unless specialized hardware is present in the IP host terminal, this interface may not be capable of providing acceptable voice quality.

The IP host-to-phone architecture was the second evolution of VoIP services. With this architecture, a VoIP user is able to call a conventional telephone user, which expands the service domain of VoIP. This architecture is currently used by NextGen telcos to offer cost-reduced long distance service.

The third evolution of VoIP service is the phone-to-phone architecture using an IP network as the carrier. This architecture combines the best of both worlds: familiar, ubiquitous interface as well as reduced toll charges. This service architecture may be become subject to regulation in the future.

Despite the service architecture, each VoIP network requires an underlying QoS infrastructure, which currently is not in place. Two main classes of QoS architectures have been proposed by the Internet community: Differentiated Services type mechanisms, and flow based Integrated Services mechanisms. Differentiated Services mechanisms are better suited for VoIP as soft states do not have to be maintained, and call set up is not required. Which mechanism to implement is currently an open area in the Internet community, but will likely be driven by vendors of IP networking equipment.

6 Conclusions

Internet telephony has risen from hobbyist stature to an anticipated 3-4 billion dollar industry. The technology is capable of offering many of the traditional telephony services, as well as value added multimedia services. Although it is becoming increasingly difficult to justify VoIP from a cost savings perspective alone, the technology is still competitive in the international or overseas long distance markets.

VoIP service is delivered with the TCP/IP protocol suite. Two main camps for VoIP signaling exist: H.323 and SIP. Despite some advantages of SIP, H.323 has been established as the standard *de jour*.

Traditional telephone conversation models have been found to be inappropriate for modeling VoIP applications. Three and four state models for three VoIP applications were developed in this thesis. These models could be implemented as a VoIP traffic simulator for the purpose of network planning to support the anticipated demands of VoIP.

Metrics for measuring the performance of VoIP systems, drawn from a combination of conventional telephone and Internet performance metrics, were reviewed in this thesis. It is recommended that metrics for measuring VoIP performance should include: end-to-end delay, delay variation, packet loss, call setup time, and voice coding algorithm. The methodology for measuring each metric, the effect the metric has on voice quality, and the performance goals were presented. To summarize, it is recommended that end-to-end delay not exceed 150 ms, and that packet loss be less than 1% to achieve toll quality VoIP service.

There are three main architectures that can be used to deliver VoIP service: IP host-to-IP host. IP host-to-phone. and phone-to-phone service across an IP network. IP host-to-IP host was the original VoIP service architecture, and remains the simplest and least expensive to implement. This architecture has limitations, however. First. the end-terminal is not a ubiquitous interface. and therefore accessibility to the service may be limited. Second. unless specialized hardware is present in the IP host terminal. this interface may not be capable of providing acceptable voice quality.

The IP host-to-phone architecture was the second evolution of VoIP services. With this architecture, a VoIP user is able to call a conventional telephone user, which expands the service domain of VoIP. This architecture is currently used by NextGen telcos to offer cost-reduced long distance service.

The third stage in the evolution of VoIP services is the phone-to-phone architecture using an IP network as the carrier. This architecture combines the best of both worlds: familiar. ubiquitous interface as well as reduced toll charges.

Current IP networks are not capable of delivering toll quality voice service as shown by the measurements of the performance metrics. To be successful, VoIP networks require an underlying QoS infrastructure, which currently is not in place. Two main classes of QoS architectures have been proposed by the Internet community: Differentiated Services type mechanisms, and flow based Integrated Services mechanisms. Differentiated Services mechanisms may be better suited for VoIP services as no call setup nor per state information is required. Future work is required in order to develop a QoS mechanism suitable for VoIP services.

7 References

[1] Internet Telephony, "Internet Telephony All Grown Up", Volume 2, Number 3, March 1999.

[2] Telecommunications Act, CRTC, October 22, 1998.

[3] Internet Telephony, "A Call To Action on Regulation", Volume 2, Number 5, May 1999.

[4] www.net2phone.com

[5] www.1010321.com

[6] www.1010345.com

[7] www.saveonld.com

[8] J. Bellamy, *Digital Telephony*. John Wiley & Sons Inc., 1991.

[9] J. Postel, "Internet Protocol", RFC 791, September 1981.

[10] S. Deering, R. Hinden, "Internet Protocol Version (IPv6) Specification", RFC 2460, December 1998.

[11] W. J. Goralski, *Introduction to ATM Networking*. McGraw-Hill, Inc., 1995.

[12] J. Postel, "Transmission Control Protocol", RFC 793, September 1981.

[13] J. Postel, "User Datagram Protocol", RFC 768, August 1980.

[14] H. Schulzrinne , S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[15] ITU-T Recommendation H.323 (1996), *Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service*.

[16] "IMTC Voice over IP Forum Service Interoperability Implementation Agreement 1.0", IMTC Voice over IP Forum Technical Committee, December 1997.

[17] Handley, Schulzrinne, Scholler, Rosenberg, "SIP: Session Initiation Protocol", Internet Draft, August 1998 – February 1999.

[18] CCITT Recommendation E.164 (1991), *Numbering Plan for the ISDN Era*.

[19] H. Schulzrinne, J. Rosenberg, "Internet Telephony: Architectures and Protocols an IETF Perspective", Columbia University, Department of Computer Science, Technical Report, February, 1998.

[20] H. Schulzrinne, J. Rosenberg, "Signaling for Internet Telephony". Columbia University, Department of Computer Science, Technical Report CUCS-005-98, February, 1998.

[21] H. Schulzrinne, J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony", Columbia University, Department of Computer Science, Technical Report, February, 1998.

[22] www.bell-labs.com/project/sip/

[23] P. Brady, "A Model for Generating On-Off Speech Patterns in Two-Way Conversation", Bell System Technical Journal, Volume 48, 1969.

[24] D. Minoli, "Issues in packet voice communication", Proceedings of IEE, Vol. 126, No. 8, August 1979.

[25] S. Deng, "Traffic Characteristics of Packet Voice", IEEE, 1995.

[26] H. Tijms, *Stochastic Models An Algorithmic Approach*. John Wiley & Sons, Inc., 1995.

[27] Weisstein, E.W., "Kolmogorov-Smirnov Test", http://www.astro.virginia.edu/~eww6n/math/Kolmogorov-SmirnovTest.html

[28] A. Siegman, S. Feldstein, Chapter 8: "Speaking and Not Speaking: Processes for Translating Ideas into Speech by Patricia Brotherton", *Of Speech and Time Temporal Speech Patterns in Interpersonal Contexts*. John Wiley & Sons Inc., New York, 1979.

[29] D. Montgomery, G. Runger, *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., 1994.

[30] "IEEE Recommended Practice for Speech Quality Measurements", IEEE Standard 297-1969.

[31] G. Kranzler, J. Moursund, *Statistics for the Terrified*. Prentice-Hall, Inc. New Jersey, 1995.

[32] D. Miloli, E. Minoli, *Delivering Voice over IP Networks*. John Wiley & Sons, Inc., 1998.

[33] ITU-T Recommendation G.114 (1996), *One Way Transmission Time*.

[34] B. Graham, *TCP/IP Addressing*. Academic Press, 1997.

[35] G. Almes. S. Kalidindi. M. Zekauskas, "A One-way Delay Metric for IPPM". Internet Draft. August 1998.

[36] www.andover.net

[37] www.telology.com

[38] C. Demichelis, "Instantaneous Packet Delay Variation Metric for IPPM", Internet Draft. July 1998.

[39] *Digital Network Notes*, Telecom Canada, 1983.

[40] G. Almes, S. Kalidindi. M. Zekauskas, "A Packet Loss Metric for IPPM". Internet Draft. August 1998.

[41] J. Bolot. H. Crepin, A. Vega-Garcia, "Analysis of Audio Packet Loss in the Internet". www.inria.fr/rodeo/personnel/avega/papers/nossdav95/nossdav.html, 1995.

[42] V. Hardman, M. Sasse. M. Handley, A. Watson. "Reliable Audio for Use over the Internet". www-mice.cs.ulc.uk/mice/publications/inet95_paper/, 1995.

[43] J. Mahdavi. V. Paxton, "IPPM Metrics for Measuring Connectivity". Internet Draft. August 1998.

[44] J. Bayless. S. Campanella, A. Goldberg. "Voice signals: Bit-by-bit". *IEEE Spectrum*. October 1973.

[45] K. Sayood. *Introduction to Data Compression*. Morgan Kaufmann Publishers. Inc.. 1996.

[46] N. Kitawaki, H. Nagabuchi, "Quality Assessment of Speech Coding and Speech Synthesis Systems". *IEEE Communications Magazine*. October 1988.

[47] P. Almquist. "Type of Service in the Internet Protocol Suite". RFC 1349, July 1992.

[48] J. Reynolds, J. Postel. "Assigned Numbers", RFC 1700.

[49] www.cisco.com

[50] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services". Internet Draft, October 1998.

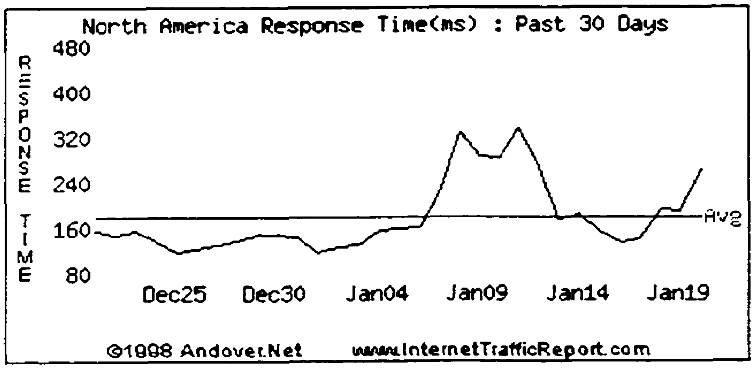
[51] K. Nichols. S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", Internet Draft, October 1998.

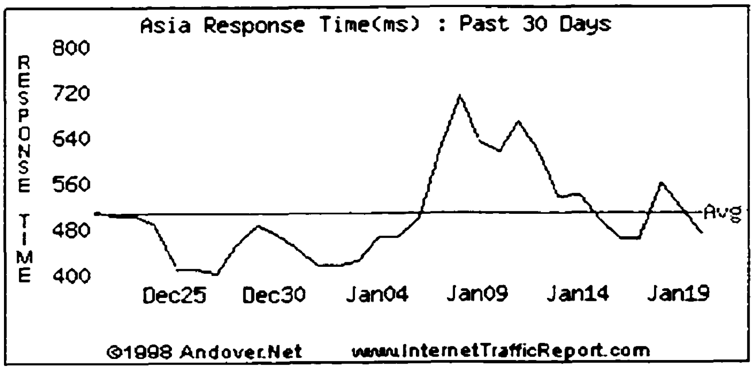
[52] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", Internet Draft. November 1998.

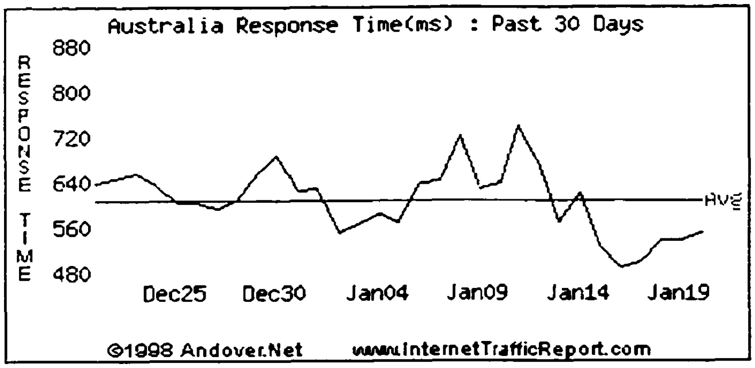
[53] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group". Internet-Draft. November 1998.

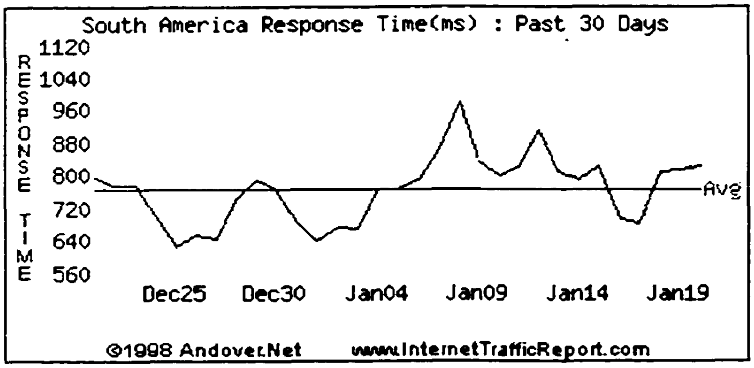
[54] S. Thomas. *IPng and the TCP/IP Protocols*. John Wiley & Sons. Inc. 1996.

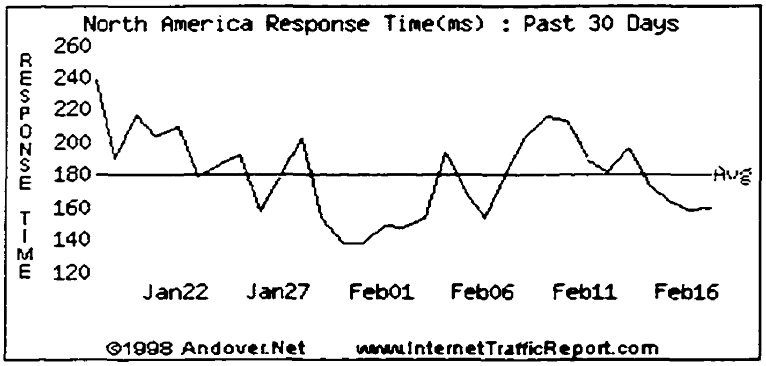
Appendix A

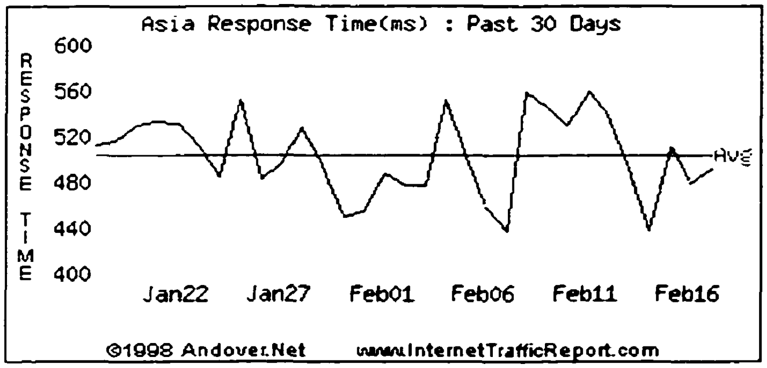

North America Response Time(ms) : Past 30 Days
RESPONSE TIME
480
400
320
240
160
80
Avg
Dec25
Dec30
Jan04
Jan09
Jan14
Jan19
©1998 Andover.Net
www.InternetTrafficReport.com

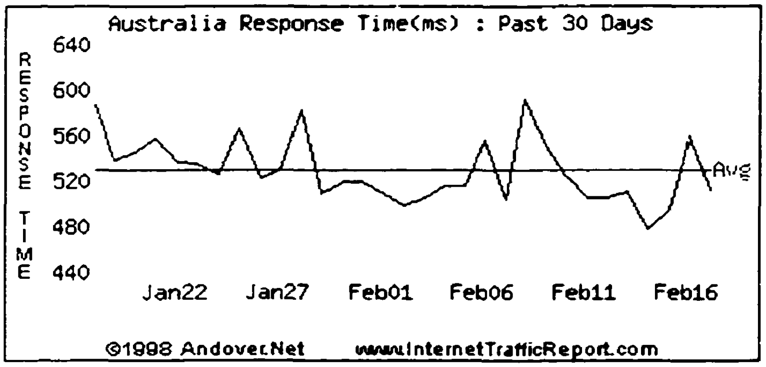

Asia Response Time(ms) : Past 30 Days
RESPONSE TIME
800
720
640
560
480
400
Avg
Dec25
Dec30
Jan04
Jan09
Jan14
Jan19
©1998 Andover.Net
www.InternetTrafficReport.com

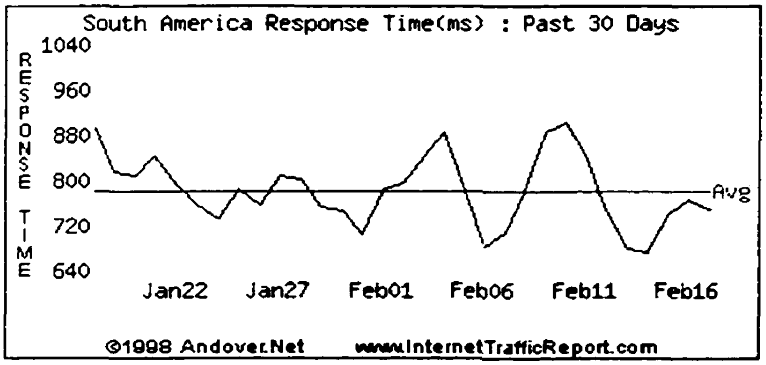

Australia Response Time(ms) : Past 30 Days
RESPONSE TIME
880
800
720
640
560
480
Avg
Dec25
Dec30
Jan04
Jan09
Jan14
Jan19
©1998 Andover.Net
www.InternetTrafficReport.com

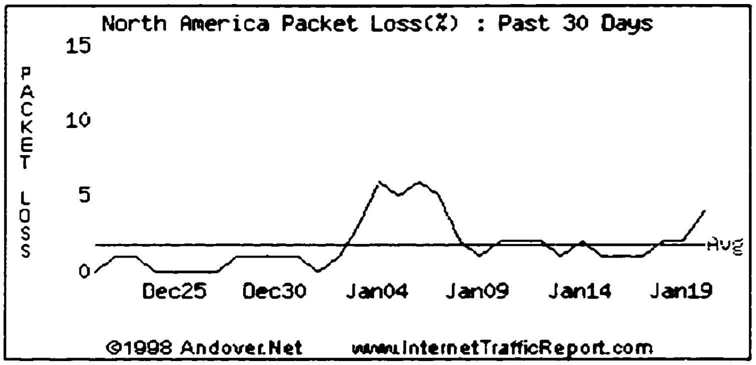

South America Response Time(ms) : Past 30 Days
RESPONSE TIME
1120
1040
960
880
800
720
640
560
Avg
Dec25
Dec30
Jan04
Jan09
Jan14
Jan19
©1998 Andover.Net
www.InternetTrafficReport.com


North America Response Time(ms) : Past 30 Days
RESPONSE TIME
260
240
220
200
180
160
140
120
Avg
Jan22
Jan27
Feb01
Feb06
Feb11
Feb16
©1998 Andover.Net
www.InternetTrafficReport.com

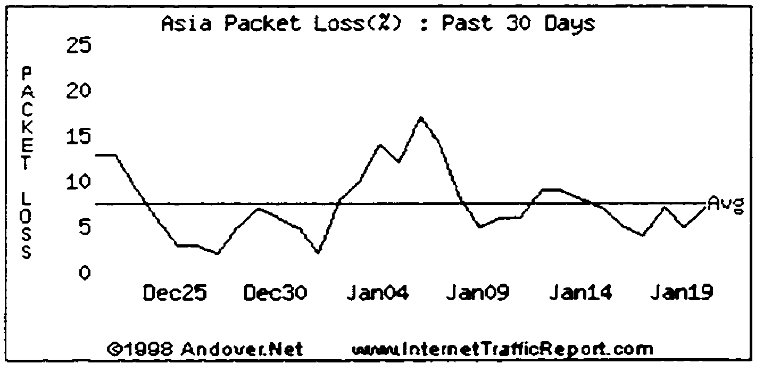

Asia Response Time(ms) : Past 30 Days
RESPONSE TIME
600
560
520
480
440
400
Avg
Jan22
Jan27
Feb01
Feb06
Feb11
Feb16
©1998 Andover.Net
www.InternetTrafficReport.com


Australia Response Time(ms) : Past 30 Days
RESPONSE TIME
640
600
560
520
480
440
Avg
Jan22
Jan27
Feb01
Feb06
Feb11
Feb16
©1998 Andover.Net
www.InternetTrafficReport.com

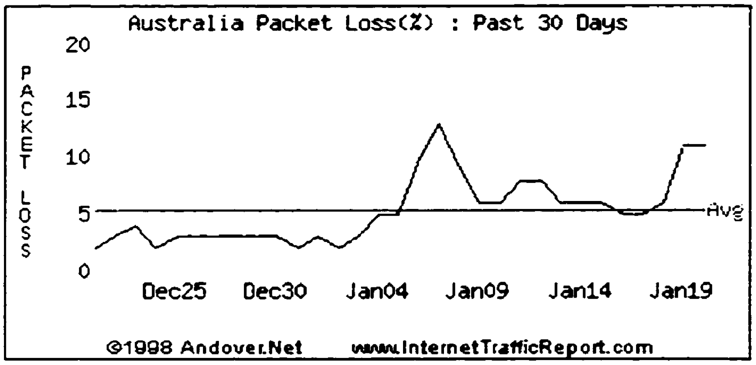

South America Response Time(ms) : Past 30 Days
RESPONSE TIME
1040
960
880
800
720
640
Avg
Jan22
Jan27
Feb01
Feb06
Feb11
Feb16
©1998 Andover.Net
www.InternetTrafficReport.com



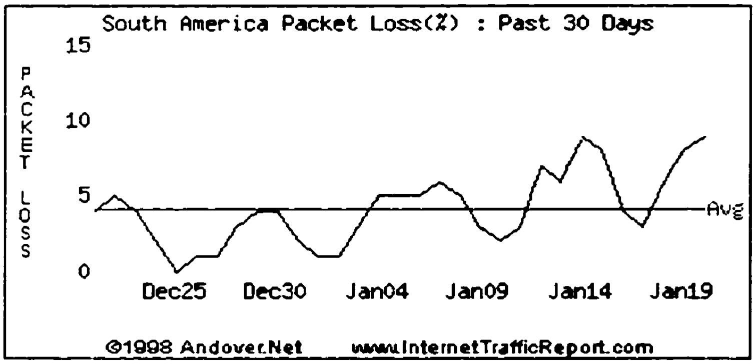
©1998 Andover.Net www.InternetTrafficReport.com



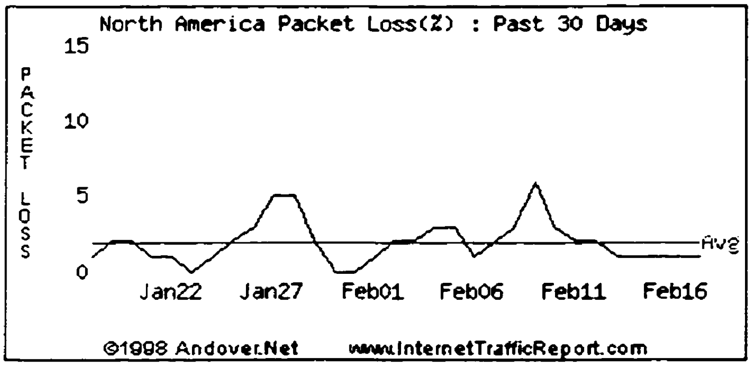
©1998 Andover.Net www.InternetTrafficReport.com



©1998 Andover.Net www.InternetTrafficReport.com

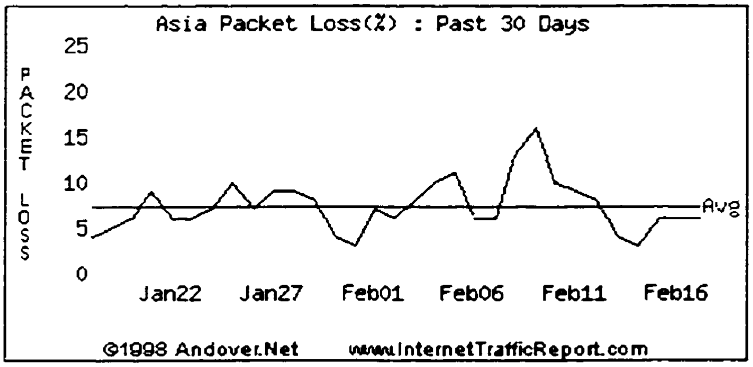


©1998 Andover.Net www.InternetTrafficReport.com



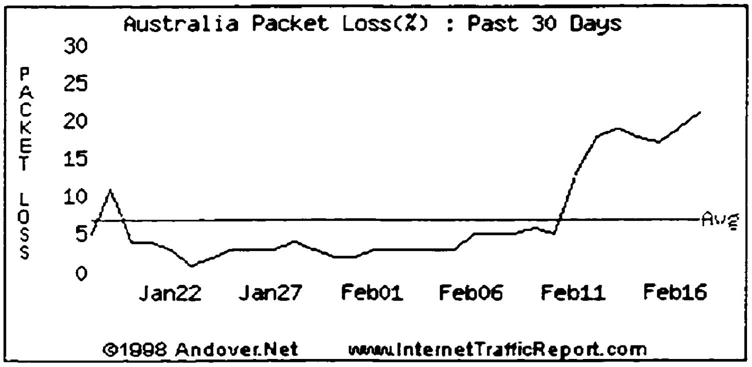
North America Packet Loss(%) : Past 30 Days

©1998 Andover.Net www.InternetTrafficReport.com



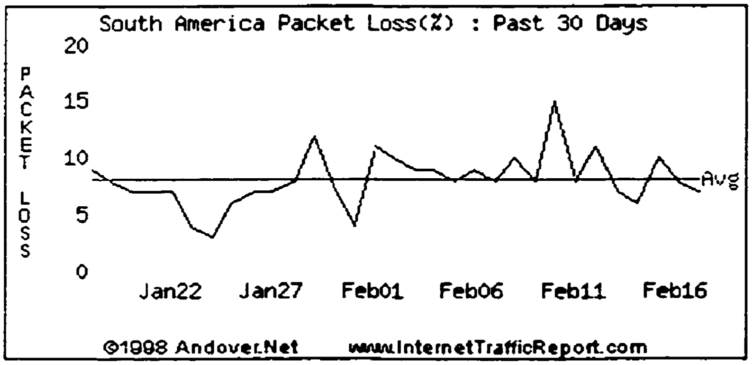
Asia Packet Loss(%) : Past 30 Days

©1998 Andover.Net www.InternetTrafficReport.com



Australia Packet Loss(%) : Past 30 Days

©1998 Andover.Net www.InternetTrafficReport.com



South America Packet Loss(%) : Past 30 Days

©1998 Andover.Net www.InternetTrafficReport.com