THE EFFECT OF DYNAMIC VOICE CODEC SELECTION FOR ACTIVE CALLS ON VOICE QUALITY

A Thesis by

Jered Daniel Ast

Bachelor of Science Electrical Engineering, Wichita State University, 2005

Submitted to the Department of Electrical Engineering
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science in Electrical Engineering

May 2007

THE EFFECT OF DYNAMIC VOICE CODEC SELECTION FOR ACTIVE CALLS ON VOICE QUALITY

I have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science in Electrical Engineering.

____
Ravi Pendse, Committee Chair

We have read this thesis
and recommend its acceptance:

____
Kamesh Namuduri, Committee Member

____
Janet Twomey, Committee Member

DEDICATION

To my parents for their continued love and support.

To Colonel (Ret.) Charles H. Brogan for his wisdom.

ACKNOWLEDGMENTS

This thesis would not have been possible without the support and assistance of many people. I would like to thank my advisor, Dr. Ravi Pendse, for his continued guidance and support.

I would like to thank my committee members for their suggestions in reviewing my report.

I would also like to thank doctoral student Nagaraja Thanthry for his help, ideas, and suggestions throughout the course of my thesis work.

I would also like to thank Andrew Ossipov for his valuable input and assistance during this thesis work.

It is a great pleasure to thank my parents, brothers, and grandparents for their continued help and support as they were always there to guide and encourage me.

ABSTRACT

Converged IP networks seek to incorporate voice, data, and video on the same infrastructure. However, the integration of all types of traffic onto a single IP network has several advantages as well as disadvantages. While reducing cost and increasing mobility and functionality, VoIP may lead to reliability concerns, degraded voice quality, incompatibility, and end-user complaints due to changing network characteristics. Voice quality degrades considerably due to low bandwidth, high packet loss rates, high jitter, or if total end-to-end delay is greater than the ITU-T suggested 150ms. In order to ensure these strict requirements are met, the underlying network must deploy various schemes to ensure resource availability.

This research proposes an adaptive codec selection mechanism which changes the voice encoding scheme in the middle of an active call based on the network conditions. The mechanism is mainly proposed for H.323 based systems and is intended to cause little to no effect on voice quality. The proposed mechanism involves establishing a three-way handshake process in mid-call to renegotiate station capabilities, making the switch at a determined sequence number in an RTP packet. The proposed mechanism ensures the voice continuity while switching codecs by filling the playout buffers appropriately. The effect of these changes on voice quality is determined using the objective E-model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACELP	Algebraic Code Excited Linear Prediction
BB	Bandwidth Broker
CACA	Call Admission Control Agent
CMOS	Comparison Mean Opinion Score
CNAME	Canonical Name
COPS	Common Open Policy Service
CS-ACELP	Conjugate Structure Algebraic Code Excited Linear Prediction
CSRC	Contributing Source
DiffServ	Differentiated Services
DMOS	Degradation Mean Opinion Score
DSP	Digital Signal Processor
EIF	Equipment Impairment Factor
ETSI	European Telecommunications Standards Institute
IntServ	Integrated Services
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
LAN	Local Area Network
MCU	Multipoint Control Unit
MOS	Mean Opinion Score
MP-MLQ	Multipulse Maximum Likelihood Quantization

LIST OF ABREVIATIONS

NACK	Negative Acknowledgement
PCM	Pulse Code Modulation
PSQM	Perceptual Speech Quality Measurement
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RAS	Registration, Admission, and Status
RR	Receiver Report
RTCP	Real Time Transport Protocol
RTP	Real Time Protocol
RTT	Round-Trip Time
SDES	Source Descriptor
SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
SR	Sender Report
SSRC	Synchronization Source
TCP	Transport Control Protocol
UDP	User Datagram Protocol
VAD	Voice Activity Detection
VoIP	Voice over Internet Protocol
WAN	Wide Area Network

CHAPTER 1

INTRODUCTION

1.1 VoIP Overview

Voice over IP (VoIP) is the ability to transmit speech over packet-switched IP networks. These packet-switched networks are connectionless, whereas traditional public voice networks are connection-oriented. Traditionally, a user makes a telephone call by picking up the phone, waiting for a dial tone, dialing the number to be reached, and waiting for the other party to pick up the phone. This establishes a connection over the Public Switched Telephone Network (PSTN) which is circuit-switched. This means that each time a person makes a phone call, a dedicated path must be created in the PSTN, and the lines connecting the two parties together are "rented" and unable to be used by any other parties during the entire duration of the phone call. The use of these lines is then charged by the phone company and the user is billed.

Placing a phone call using VoIP will create a digital signal from the analog input, place those digital signals into packets with source and destination network addresses and finally send the information over the Internet or internal company IP networks, thus bypassing the need for the PSTN lines. In a traditional VoIP scenario, illustrated in Figure 1, a voice gateway connects to the Internet through an IP cloud. IP phones, fax machines, softphones (multimedia-capable PCs), and standard telephones connect to the gateway. The gateway establishes a connection between these terminal devices to terminals on other networks. When establishing a connection from a VoIP network to a traditional PSTN network, the dial peer (voice gateway) must convert analog voice signals into digitized format for transmission over the IP network. Once the digitized format is received, the voice must be decoded, allowing the original voice to be played back. This process of coding and decoding is regulated by various voice codecs. However, it is important to note that both terminals must support the particular codec for proper encoding and decoding. Two popular VoIP standards are H.323 [1] and Session Initiation Protocol (SIP) [2]. Both standards allow for direct call establishment between VoIP-capable terminals or the use of gatekeepers, which can be used to negotiate connections between endpoints. They can be used to translate between IP addresses and telephone numbers, perform registration and authentication functions, and manage bandwidth.

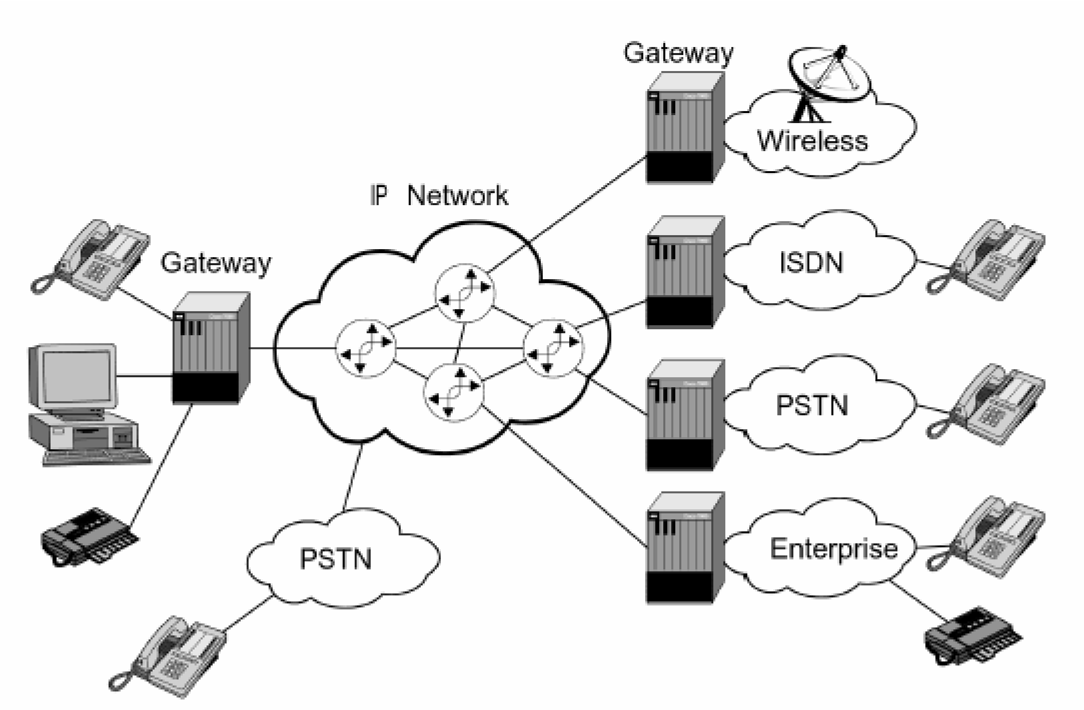

Figure 1: Basic VoIP Implementation [3].

Similar to the traditional PSTN, VoIP calls must be established prior to voice traffic being exchanged. This process for establishing a session is termed call setup. During call setup, both terminals will attempt to negotiate capabilities, agree on the codec and ports to use, and reserve resources. Voice traffic can then be transmitted between the terminals. Furthermore, control packets are routinely sent between the terminals during the call to assist in maintaining flow control, packet recovery, and time synchronization. Once the call has concluded, the devices perform a termination sequence for the call.

On both the public Internet and private intranets, the H.323 standard is used to facilitate the sending of both audio and video between endpoints. The H.323 architecture encompasses numerous protocols and specifications that interact to create successful calls. These include, but are not limited to, H.225.0, H.235, H.245, and H.450.x. H.225.0 is used for call signaling, H.235 is used for security and encryption, H.245 is the control protocol for multimedia communication, and H.450.x includes supplemental services such as call forwarding, call waiting, etc. The management of these audio and video conversations is done using two protocols, Real-Time Transport Protocol (RTP) [4] and Real-Time Control Protocol (RTCP).

1.2 Inherent Problems with VoIP

This technology has been under development for nearly a decade and has been improved to the point where many companies and institutions have either implemented this technology or are looking into the feasibility of implementing it. As with any new technology, an inherent number of opportunities and risks arise. A converged network would have IP traffic such as voice, data, and video on the same infrastructure. The integration of all types of traffic onto a single IP network creates inherent problems. While reducing cost and increasing mobility and functionality, VoIP has the potential to cause a number of risks including reliability, lack of Quality of Service (QoS), overall voice quality, security, incompatibility, end-user complaints, and integration, among others.

One of the primary factors affecting VoIP is the best-effort nature of IP [5]. Although best-effort is simple and scalable, this provides no guarantee of bandwidth for specific applications. All traffic is considered to have the same level of importance on the network.

Best-effort traffic is handled on a first-come first-serve basis, which is detrimental to the real-time requirements of VoIP traffic. According to research conducted by AT&T [6], the areas potentially overlooked by customers when implementing voice and data applications are application requirements, key network congestion points, and primary sources of network delay. These requirements, congestion points, and network delays are all detrimental to VoIP, as generally-accepted, one-way VoIP transmission must be accomplished in less than 150 ms, according to the ITU-T Recommendation G.114 [7].

There are many major causes for voice quality degradation. These include delay, jitter, and packet loss, leading to the requirement for Quality of Service. The real-time transmission of data was not an intended usage for the IP networks when first developed. However, porting real-time information onto these networks has created a need for efficient QoS. Quality of Service is necessary to minimize the effect of delay, jitter, and packet loss on the voice packets, and assure proper delivery of real-time information to the destination. Voice quality degrades considerably if the amount of bandwidth on the link is too low, the packet loss rate is too high, the amount of jitter is too high, or if the end-to-end delay is greater than the ITU-T suggested 150 ms. In order to ensure these strict requirements are met, the underlying network must deploy various schemes for quality of service and call admission controls to ensure resource availability.

1.3 Current and Future Solutions

Reducing the unfavorable effect of the best-effort nature of IP on VoIP is done through appropriate QoS implementation and call admission control mechanisms. These mechanisms contribute to maintaining lower levels of overall, cumulative delay, which is instrumental in upholding higher levels of call quality. In addition to such mechanisms, modern solutions do not attempt to modify the codec and bandwidth usage for active calls. This is apparent in various popular entities including Microsoft's Netmeeting messaging software, Speak Freely, and Cisco Systems [7] VoIP software. Changing the codec is not performed on an end-to-end basis for active calls. Unfortunately, this can reduce the number of calls allowed on the link by not converting to a lower bandwidth codec when new calls are added. Thus, the original calls retain the original bandwidth usage and additional requests for calls will be rejected by call admission controls.

Despite the problems and challenges facing VoIP today, significant strides in resolving and minimizing these issues have allowed for the increased adoption of VoIP technology in the industry. This research work analyzes the effect changing the codec during an active call has on call quality. It attempts to predict the delay associated with codec changes, control packet exchange, and buffer resynchronization. Then, for a given set of codecs, it will check if the cumulative end-to-end delay is less than accepted 150 ms. Any change in call quality will be directly affected by an increase in delay.

The rest of the thesis is organized as follows. First, a summary of related research work is covered. Then, an overview of the H.323 call setup process is described in Chapter 3, as well as the proposed codec selection scheme and the proposed handshake mechanism. Chapter 4 looks into variables affecting voice quality and the procedures for measuring voice quality. In Chapter 5 a simulation is provided to illustrate the process. It includes a comprehensive analysis and results. Finally, Chapter 6 discusses future work and conclusions.

CHAPTER 2

LITERATURE SURVEY

2.1 Voice Quality

Voice quality is the single most important aspect to any user of Voice over IP technology. Users expect and require a high level of voice quality that is comparable to the time-tested PSTN. Users have grown accustomed to the PSTN, and as such, any VoIP system is required to meet or exceed the performance criteria of the traditional PSTN. This includes reducing the number of call drops, call refusals, and maintaining an acceptable level of voice quality. Similarly, VoIP systems need to be not only comparable, but also compatible with the PSTN, and provide an inherent level of security. Therefore, the underlying IP network must reduce latency and packet loss to meet the essential requirements of VoIP calls.

The main objective of VoIP is to provide an acceptable quality of voice to its users that is comparable to the PSTN. In order to achieve this, it is imperative to understand what is detrimental to overall voice quality. Forrester [9] and Keagy [5] describe the factors affecting voice quality as delay, jitter, packet loss, compression, prioritization, and echo cancellation. Each of these factors must be addressed and resolved to ensure proper voice communication. The authors state there are several ways to contend with these factors. Compression can be used to significantly reduce the required bandwidth. Voice Activity Detection (VAD) and silence suppression can reduce bandwidth requirements as well by not sending voice packets across the link when the caller or receiver is silent. Furthermore, he states that there is a necessity for VoIP systems to employ echo cancellation techniques because call quality degrades if the end-to-end latency of a call exceeds 150 ms. Since the typical packet lengths range from 5 to 30 ms each, there are a high number of voice packets created for each call. Each IP packet header can be a maximum 40 bytes in length [10], and the use of RTP/UDP/IP encapsulation on low-speed serial links can reduce the header size to 2-4 bytes [11] saving immensely on link bandwidth usage.

In addition to the aforementioned factors affecting voice quality, Keagy also describes other variables including background noise, I/O signal level, amplitude clipping, quantization and codec distortion, multiple talkers, circuit noise, random bit errors, and burst errors (packet loss). A listener may perceive poor voice quality if the sender or receiving ends of the call experience unacceptable amounts of background noise. Under normal circumstances, audio volume remains the same when it is digitized. However, if the actual audio volume is too high or two low, the signal can become distorted. Similarly, if the amplitude level is too high, the signal is clipped off so it can be transmitted properly. Codec distortion is seen due to the lossy compression scheme for coding algorithms. With this type of coding, the receiver does not receive all of the original voice signals. Multiple talkers can have a negative effect on the perceived quality of voice because codecs generally model the voice patterns of a single talker. Furthermore, random bit errors due to hardware issues or other reasons have significant impact on the lower bit-rate codecs. Finally, in voice networks, burst errors (loss of consecutive packets) have a damaging effect on voice quality. Since voice packets are not retransmitted if lost or corrupted, any consecutive loss in voice packets is noticeable by the receiver.

Voice quality measurements are an imperative aspect required for understanding and comparing call quality. These measurements are made subjectively, objectively, or are estimated. Subjective measurements are considered the most reliable and respected approach for measuring voice quality [5], whereas objective speech quality measurements provide comparable results that are much less time-consuming and less expensive. The Mean Opinion Score (MOS) test is a widely accepted standard for subjective speech quality ratings [12]. This standard, described in ITU recommendation P.800 [13], involves users rating the perceived speech quality on a scale of 1 (poor) to 5 (excellent). However, due to the time and expense associated with subjective testing, it is necessary to use an objective method that can estimate a subjective quality. ITU recommendation G.107 [14] describes the E-model objective speech quality measurement which combines all impairment factors that affect a voice call into a single scalar value which can be correlated to a corresponding MOS value. The E-model has also been adopted by the European Telecommunications Standards Institute (ETSI) and the Telecommunications Industry Association (TIA), becoming the most widely used opinion model in the world [15].

2.2 QoS Methods

There are numerous call admission controls and QoS methods in place currently. Wang et. al. [16] presents one such scheme that is easily integrated into existing infrastructures is know as a Call Admission Control Agent (CACA). In order to effectively use this method, the current availability of resources and calls must be continuously monitored. Admission controls either accept or reject incoming calls based on the resource availability at the time of the request. As such, complete network resource monitoring must be implemented. They also state their QoS-provisioning system supports both deterministic and statistical services. Deterministic services appropriate buffers based on worst-case scenario to ensure that no packets miss their delay bound at the expense of less efficient network usage. However, statistical QoS exploits the statistical aspects of the network to allow more flows. This causes more efficient network usage at the expense of a low chance of packet loss and increased network delay. The authors state their system works in closed and semiopen (such as Internet2) networks, but due to the difficulty in modeling Internet traffic, only statistical guarantees can be provided across open networks.

Two QoS models known as Integrated Services (IntServ) and Differentiated Services (DiffServ) also seek to address the best-effort nature of IP. Giordano et. al. [17] discusses how DiffServ can be successfully adopted for VoIP and H.323. They propose a dynamic setting of resources to achieve the required QoS on the Internet. This is accomplished using a Common Open Policy Service (COPS) to fulfill real-time video and audio QoS requirements. This protocol provides administration, configuration, and enforcement of policies without affecting H.323 signaling and allowing dynamic resource allocation across IP. It also employs the use of a Bandwidth Broker (BB) to maintain per-aggregate traffic flows between core routers. Simulations and field trials proved this dynamic resource QoS scheme is feasible.

Koutsakis et. al. [18] discuss wireless call admission control and traffic policing mechanisms for video conferencing with the H.263 video standard. They propose an approach where only new terminals are allowed to join if the requested bandwidth is available. If there is insufficient bandwidth available for a new connection, all other terminals are requested to decrease their bandwidth usage to allow the new connection. If no terminals can reduce their bandwidth sufficiently enough to accept the new connection, the new connection is rejected. The method achieved high aggregate throughput while maintaining the necessary QoS. This could potentially be applied to VoIP by changing the parameters of voice calls to maximize network resources.

2.3 Internet Traffic Modeling

In Zhang, et. al [19], the authors examine various mathematical techniques to model Internet traffic delay and packet loss characteristics using Poissonly-distributed packet streams. Several predictors, including the Moving Average (MA), Exponentially-Weighted Moving Average, and S-Shaped Moving Average (SMA) were all shown to have similar errors in prediction of these parameters. They performed end-to-end measurements between multiple universities and research sites in the United States at different hours of the day. The authors state that 84 - 87% of the traces had less than 1% packet loss rates. Therefore, that packet loss rate will be assumed. When modeling delays, the authors state delays are highly predictable, but less steady than loss rates. The final result of their analysis yielded that the steadiness of the Internet can only be made on the time scale of minutes. Hence, packet loss and end-to-end delay may vary when traversing the Internet, but can be modeled for the purpose of the proposed mechanism.

2.4 Resource Centric Approach

In [20], Ossipov proposes a resource centric approach designed to maximize efficiency of network resource utilization. The approach continuously monitors resource availability and requirements of current calls, and it provides a mechanism that allows for dynamic change based on those parameters. The author breaks the algorithm into three sections: network monitoring, codec allocation, and gateway-level policing. Network monitoring continuously analyzes parameters such as available bandwidth, packet loss rates, and overall end-to-end delay. This can be accomplished with the Simple Network Management Protocol (SNMP) or by evaluating sender and receiver reports in RTCP. As such, the evaluation of resource availability can be obtained in both private and public IP networks. Codec allocation is therefore determined based upon available resources obtained from network monitoring procedures, thus reducing the likelihood of packet loss due to congestion. The algorithm selects the best codec option based on these available resources while attempting to maximize voice quality. The author also proposes a gateway-level policing mechanism that enforces optimal codec allocation. Furthermore, he explains this design will work best with a method that supports codec renegotiation in mid-call because of the dynamic nature of VoIP call utilization. The research proposed here works at the gateway policing level that allows for a means of performing mid-call dynamic voice codec selection to take advantage of the author's resource-centric method.

CHAPTER 3

PROPOSED ALGORITHM

This research proposes an adaptive codec selection mechanism which changes the voice encoding scheme in the middle of an active call based on the network conditions. The proposed mechanism is mainly proposed for H.323 based systems and is intended to cause little to no effect on voice quality. The proposed mechanism involves establishing a three-way handshake process in mid-call to renegotiate station capabilities, making the switch at a determined sequence number in an RTP packet. The proposed mechanism ensures the voice continuity while switching codecs by filling the playout buffers appropriately. Two options for playout buffer use are analyzed. Section 5.2 explores the creation of a new playout buffer to manage packets with the new codec. Section 5.3 examines filling the playout buffer, changing the codec, and reusing the buffer for voice packets using the new codec. The effect of these changes on voice quality is determined using the objective E-model. The simple network shown in Figure 2 will be assumed. The link connecting the two routers is 10base-T Ethernet, with a total distance of fifteen feet of cable.

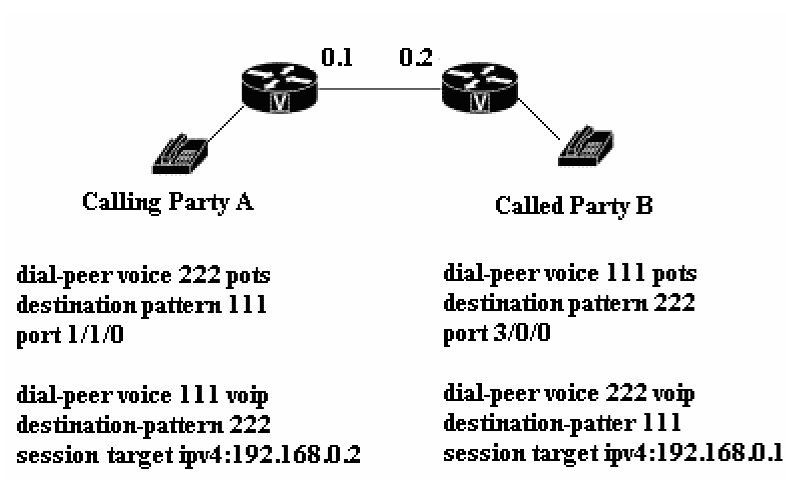

Figure 2: Sample Network Configuration.

3.1 H.323 Introduction

As previously mentioned, the popular H.323 [1] VoIP signaling standard is used to facilitate the sending of audio, video, and data between endpoints. It is composed of several software and hardware components. These components consist of terminals, gateways, gatekeepers, and multipoint control units (MCUs). Terminals are the end-points in the network and provide real-time two-way communication with other H.323 terminals or gateways. Gateways are used to connect H.323 terminals with terminals located in other networks using a different protocol, such as SIP or PSTN. Gatekeepers manage bandwidth, terminal registration, and authentications, as well as translating between IP addresses and telephone numbers. They also provide call services such as caller-id, call forwarding, etc. MCUs may be used to establish conferences with multiple parties.

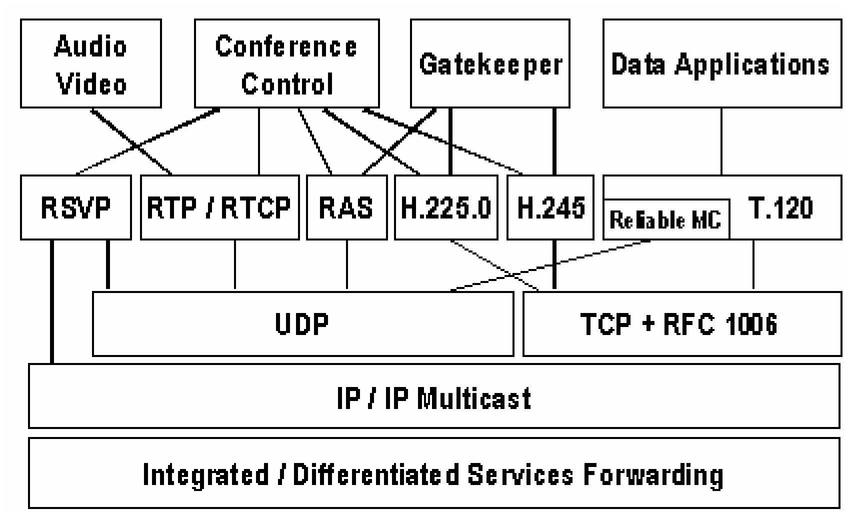

Figure 3: H.323 Protocol Architecture [20].

The H.323 protocol stack [21] incorporates numerous protocols and ITU-T standards that are required for VoIP calls. The H.225.0 standard is used for call signaling and media packetization. The H.235 standard is used for security and encryption for terminals using H.323. The H.245 protocol is provides control of media communications. The H.450.x describes supplemental services such as call waiting, call transfer, etc. The T.120 series describes data protocols for multimedia conferencing. All of these protocols interact to create VoIP calls. The H.323 protocol architecture is illustrated in Figure 3.

The H.323 protocol progresses through several phases during a call sent from the calling party (A) to the called party (B). These phases are outlined in Figure 4. If a gatekeeper is present, the terminals register with the gatekeeper and are either admitted or rejected. If no gatekeeper is present, the H.225 messages are exchanged directly between the two endpoints using the first TCP signaling channel. The two parties then go through call signaling, capability negotiation and exchange, media exchange, and potentially capability renegotiation. After the terminals have concluded media exchange, the call then enters a termination phase.

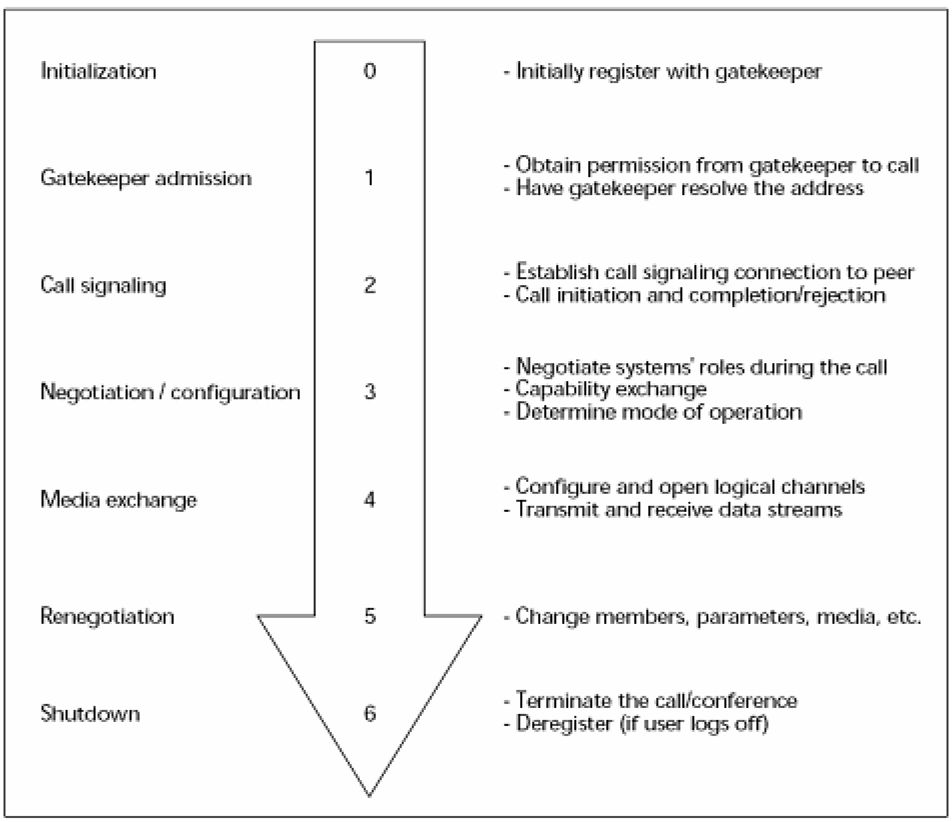

Figure 4: H.323 Protocol Phases as Outlined in [22].

3.1.1 Establishing a VoIP call between terminals (Phase A)

The H.225.0/RAS [23] standard is composed of call signaling and Registration, Admission, and Status (RAS). It manages audio, video, data, and control information on a packet-based network. Establishing a call between two IP, multimedia-capable terminals requires the use of two TCP connections. The first connection is used for call signaling, the other for call control messages utilizing the H.245 subset of standards. The first TCP connection is created with the support of Integrated Services Digital Network (ISDN) Q.931 signaling protocol, and the signaling channel makes use of TCP port 1720. In the event two terminals are making use of a gatekeeper, the first TCP signaling channel is created between the terminal and gatekeeper and it is used to exchange RAS messages. RAS performs registration, admission control, status, bandwidth changes, and call termination procedures between endpoints and gatekeepers. The called party listens for an incoming call on an open, dynamic port and uses this port to accept the call. The second TCP connection is established to the same port and is used to transfer H.245 control messages. This process is illustrated in Figure 5.

Similarly, phase A direct signaling is shown in Figure 6. In (1), Terminal A requests admission to create a call to terminal B and provides the requested bandwidth. In (2), the admission request is either confirmed with an ACF or rejected with an ARJ message. Terminal A then sends a direct call setup request to Terminal B (3). If B accepts the call, it sends back a call proceeding (4) message to A. Otherwise, it sends a Release Complete message and drops the call. B then requests admission (5) and is either confirmed or rejected (6). If B’s admission request was confirmed, it sends an Alerting message to A. Then, if B accepts the call, it sends a Connect message to A with the H.245 control channel transport address. If at any point terminal B does not wish to continue, the call is dropped with a Release Complete message.

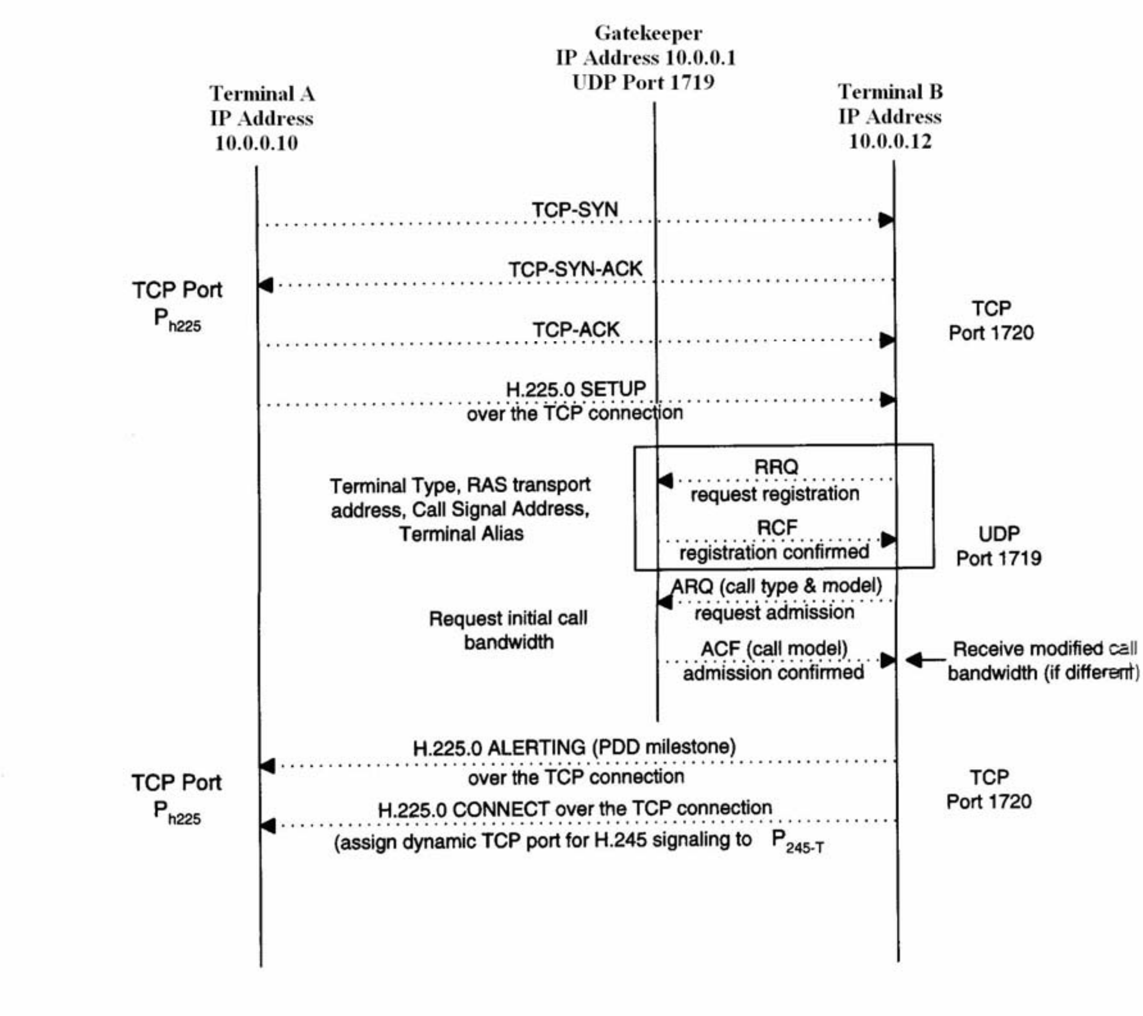

Figure 5: H.323 call setup, Phase A [24].

3.1.2 Initial Communication and Capability Exchange (Phase B)

In phase B, the H.245 control channel is established, and all future aspects of the call are managed through this channel. Terminal A sends a Terminal Capability Set (1) message to Terminal B, sending both its capabilities and transport address to be used for the control channel. Once B receives these capabilities, the original TCP connection for H.225 signaling can be closed, as it is no longer required. B then replies with an acknowledgement (2) for the received capabilities and transport address. B then sends its own capabilities to A (3). Once A responds with an acknowledgement (4), the H.245 control channel has been established correctly. Both parties then enter into Master/Slave determination mode (5) and (6), where they negotiate who will be the master and who will be the slave. The capability exchange is illustrated in Figure 7.

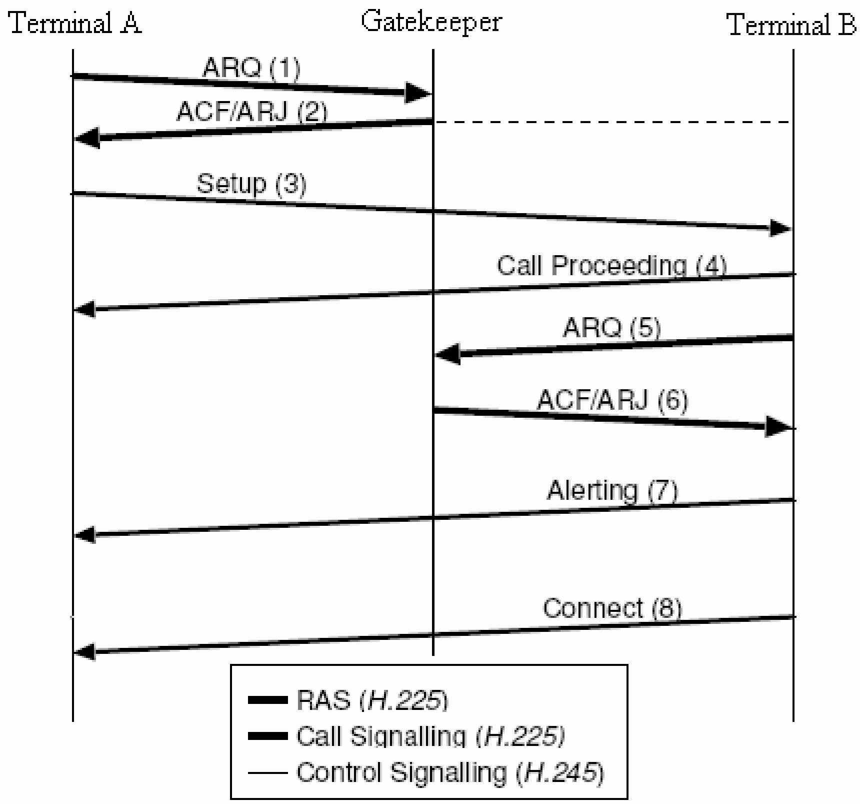

Figure 6: H.323 Call Setup Phase A with Direct Signaling [24].

3.1.3 Establishment of Audio/Video Communication (Phase C)

During phase C, both terminals open logical channels to exchange audio, video, and data streams. Terminal A sends an Open Logical Channel request (1) to B, B sends an Open Logical Channel request to A (2) and then acknowledges A's request (3). A then acknowledges B's request and the logical channel for a media stream has been opened. This process is repeated for the number of media channels required, as shown in Figure 8.

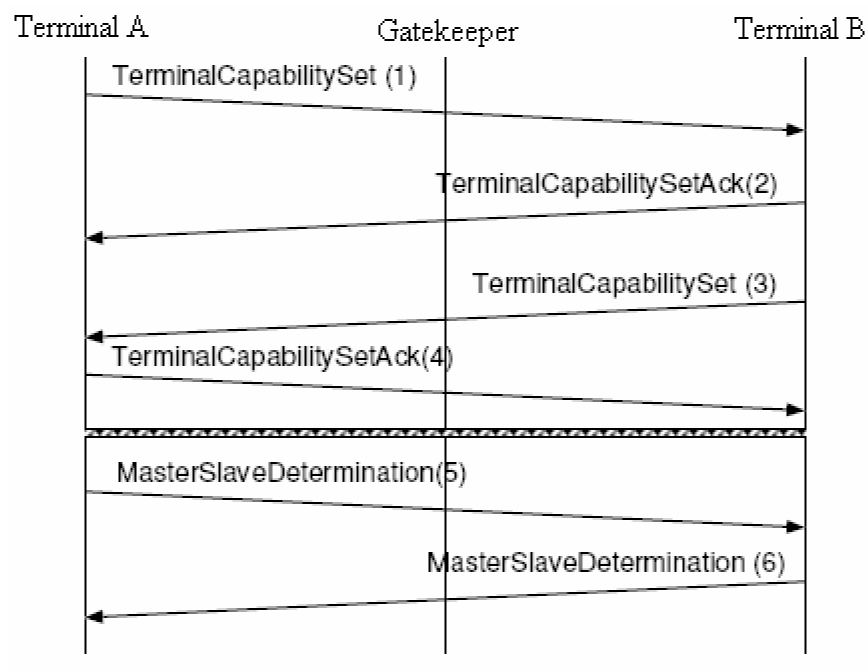

Figure 7: H.323 Phase B Initial Communication and Capability Exchange [24].

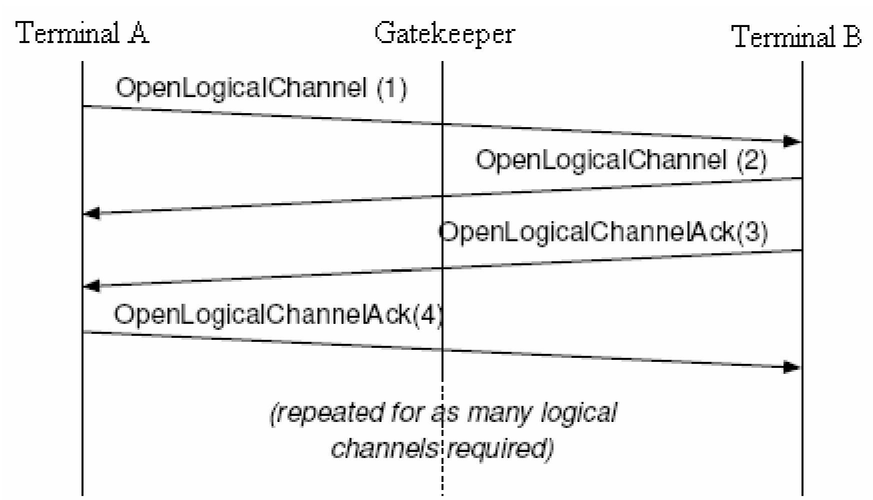

Figure 8: H.323 Phase C Establishment of Audio/Video Communication [24].

3.1.4 Call Services (Phase D)

H.323 is not directly involved with the actual transmission of media packets in Phase D. As mentioned earlier, it utilizes the RTP protocol. However, if any other services are required during this phase, such as bandwidth change requests, status updates, conference expansion, or the use of other supplementary services in H.450.x, H.245 messages are used between endpoints and RAS messages are used when talking to the gatekeeper.

3.1.5 Call Termination (Phase E)

A call can be terminated at any point in the sequence. The call termination sequence is illustrated in Figure 9. All media exchange logical channels that were previously created must be closed. If terminal A ends the call, it sends a CloseLogicalChannel (1) request to B. B then responds with a CloseLogicalChannel (2) request where it stops listening on the channel, stops sending receiver reports, and sends an acknowledgement (3) for A’s request. A then acknowledges (4) B’s request to close the channel, in which A and B stop listening for reports. This process is repeated until all open logical channels have been terminated.

After all logical media exchange channels have been closed, terminal A sends an EndSession (5) command to B, which initializes the termination of the unidirectional H.245 channel from A to B. B responds with an EndSession (6) command to A, which terminates the H.245 channel from B to A. A then terminates its side of the H.245 channel and sends a ReleaseComplete (7) message to B. The ReleaseComplete message will also tear down the H.225 signaling channel, if it had not been torn down previously. Both parties then unregister from the gatekeeper, which notifies the gatekeeper the call resources are no longer needed.

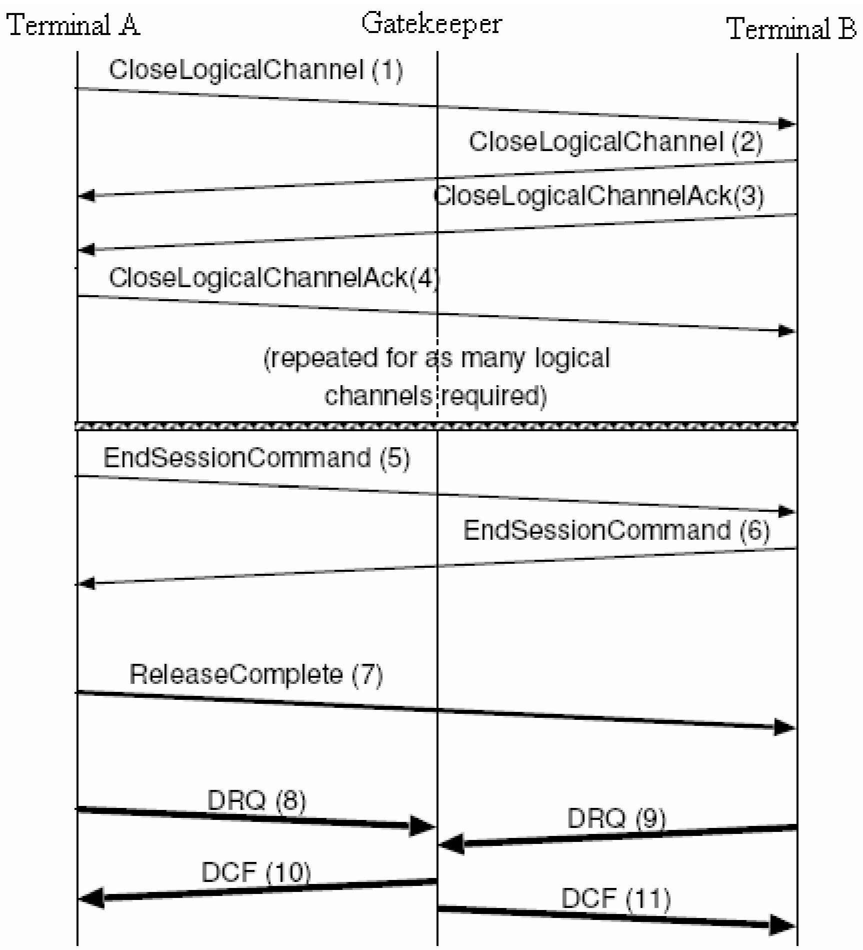

Figure 9: H.323 Phase E Call Termination [24].

3.2 RTP Packet Format

Audio, video, and multimedia services require the use of RTP, which provides the necessary end-to-end delivery requirements of time sensitive data. Both RTP and RTCP were designed to run independently of the underlying transport and network layers. RTP often runs in unison with the User Datagram Protocol (UDP), which supports multiplexing and checksums. UDP is exploited by RTP because the retransmission nature of TCP is detrimental to VoIP. If a packet is not received at the destination, TCP will attempt to retransmit the lost packet. However, due to the strict time constraints for VoIP, the retransmission of the packet will be of no use due to the time sensitivity of the data.

A typical RTP [4] packet includes a sequence number that allows the receiver to reconstruct the data the sender has sent in the appropriate order, as shown in Figure 10. Furthermore, it allows for time synchronization through the use of timestamps and can differentiate between multiple senders in a multicast stream. However, it does not ensure the packets are received in order, guarantee the delivery of the packets, nor address resource reservation. It relies on lower-layer services to provide the expected quality of service. The sequence number and timestamp assist in reordering the packets in the jitter buffer. Each time a packet arrives at the receiver, the buffer will place the packet in the appropriate order based on these values. If a packet is not available when it needs to be played out, the last frame to leave the buffer is copied and played out repeatedly until the timestamp of the next available packet is arrived at. This RTP sequence number will be used to determine which RTP packet will contain the new codec.

<table>
<tr><td>00</td><td>02</td><td>04</td><td>06</td><td>08</td><td>10</td><td>12</td><td>14</td><td>16</td><td>18</td><td>20</td><td>22</td><td>24</td><td>26</td><td>28</td><td>30</td></tr>
<tr><td>Ver</td><td>P/X</td><td colspan="2">CC</td><td>M</td><td colspan="3">PT</td><td colspan="8">Sequence Number</td></tr>
<tr><td colspan="16">Timestamp</td></tr>
<tr><td colspan="16">SSRC</td></tr>
<tr><td colspan="16">CSRC</td></tr>
</table>

Figure 10: RTP Packet Format [4].

Ver, Version – Two bits specifying the RTP version number (usually version 2).

P, Padding – If this padding bit is set, it indicates padding bytes are present that are not part of the payload.

X, Extension – A one bit value that indicates only one header extension follows.

CC, CSRC Count – A four bit value detailing the number of CSRC identifiers that follow the fixed header.

CSRC, Contributing Source – The contributing source is a thirty-two bit value. It identifies the sources that contributed to the packet's payload. Up to fifteen elements can be represented.

M, Marker – The marker is a one bit value defined by a given profile that allows events such as frame boundaries to be marked in the packet stream.

PT, Payload Type – The payload type is a seven bit value that distinguishes what format the RTP payload includes and determines how the application is to interpret it.

Sequence Number – The sequence number is a sixteen bit value that increments for each RTP packet and can be used by the receiver to re-establish proper packet sequence or detect lost packets. The initial number is random, therefore making it more difficult for known-plaintext attacks. Sequence numbers are unidirectional. An H.323 terminal A could send an RTP sequence number starting with 579 for the first packet sent to B. All subsequent packets sent in the direction of B would have sequence numbers incrementing by one from 579. However, packets flowing from B to A could start with an RTP sequence number of 2509, and each packet flowing in that direction would increment the sequence number by one.

Timestamp – The timestamp is a thirty-two bit value that allows for synchronization and jitter calculations. It signals the instant the RTP packet is sampled. It is derived from a clock that increments linearly in time.

SSRC, Synchronization Source – This field is a thirty-two bit value that identifies the synchronization source. This number should be a unique value so no two sources within the same RTP session have the same SSRC.

3.3 Real Time Control Protocol

RTCP allows network administrators to monitor network conditions and provides minimal control and identification functionality [4]. It operates in unison with RTP, but does not transport data. It utilizes the periodic sending of control packets to active terminals. These control packets provide feedback on the quality of data reception, which is directly related to the transport protocol flow and congestion. Each user sends control packets to every other user, which allows the sender to observe each user independently. This information enables the sender to modify the encoding and transmission rates at any time during the session. The multiplexing of these control packets and data packets is done using separate port number with UDP.

There are several types of RTCP formats specified in RFC3550. These include sender report packet (SR), receiver report packet (RR), source description RTCP packet (SDES), goodbye RTCP packet (BYE), and application specific RTCP packets (APP), which are explained below:

SR – The sender report consists of transmission and reception statistics from active senders. This packet format is shown in Figure 11.

<table>
<tr><th>00</th><th>02</th><th>04</th><th>06</th><th>08</th><th>10</th><th>12</th><th>14</th><th>16</th><th>18</th><th>20</th><th>22</th><th>24</th><th>26</th><th>28</th><th>30</th></tr>
<tr><td>Ver</td><td>P</td><td colspan="2">RC</td><td colspan="4">PT=SR=200</td><td colspan="8">Length</td></tr>
<tr><td colspan="16">SSRC of packet sender</td></tr>
<tr><td colspan="16">NTP timestamp, most significant word</td></tr>
<tr><td colspan="16">NTP timestamp, least significant word</td></tr>
<tr><td colspan="16">RTP timestamp</td></tr>
<tr><td colspan="16">Sender's packet count</td></tr>
<tr><td colspan="16">Sender's octet count</td></tr>
<tr><td colspan="16">SSRC_1 (SSRC of first source)</td></tr>
<tr><td colspan="4">Fraction lost</td><td colspan="12">Cumulative number of packets lost</td></tr>
<tr><td colspan="16">Extended highest sequence number received</td></tr>
<tr><td colspan="16">Inter-arrival jitter</td></tr>
<tr><td colspan="16">Last SR (LSR)</td></tr>
<tr><td colspan="16">Delay since last SR (DLSR)</td></tr>
</table>

Figure 11: RTCP Sender Report Packet Format [4].

RR – The receiver report include reception statistics from non-active senders. This packet format is shown in Figure 12.

<table>
<tr><td>00</td><td>02</td><td>04</td><td>06</td><td>08</td><td>10</td><td>12</td><td>14</td><td>16</td><td>18</td><td>20</td><td>22</td><td>24</td><td>26</td><td>28</td><td>30</td></tr>
<tr><td>Ver</td><td>P</td><td colspan="2">RC</td><td colspan="4">PT=RR=201</td><td colspan="8">Length</td></tr>
<tr><td colspan="16">SSRC of packet sender</td></tr>
<tr><td colspan="16">SSRC_1 (SSRC of first source)</td></tr>
<tr><td colspan="4">Fraction lost</td><td colspan="12">Cumulative number of packets lost</td></tr>
<tr><td colspan="16">Extended highest sequence number received</td></tr>
<tr><td colspan="16">Inter-arrival jitter</td></tr>
<tr><td colspan="16">Last SR (LSR)</td></tr>
<tr><td colspan="16">Delay since last SR (DLSR)</td></tr>
</table>

Figure 12: RTCP Receiver Report Packet Format [4].

SDES – The source description packet contains source descriptor items such as a canonical name (CNAME) that is used as a transport-level identifier for an RTP sender.

BYE – The goodbye packet indicates the end of participation.

APP – The application specific packet describes application-specific functions.

Both the RTCP sender and receiver reports provide feedback on reception quality. If a station is only a receiver, the receiver report is generated. However, if the receiver is also a sender, then the sender report is generated. The only difference between the sender and receiver report packet formats is the Packet Type (PT) and the five words of sender information in the sender report. The RTCP fields are explained in detail below.

Ver, Version – Two bits specifying the RTP version number (usually version 2).

P, Padding – If this padding bit is set, it indicates padding bytes are present that are not part of the payload. The number of padding bytes to ignore is contained in the last byte of padding.

RC, Reception Report Count - This five bit value represents the number of reception report blocks in the packet.

PT, Packet Type – This eight bit value is set to 200 for sender reports and 201 to identify receiver reports.

Length – A sixteen bit value representing a 32-bit word, minus one, which allows for a valid zero length and avoiding potential loops in scanning of RTCP packets.

SSRC, Synchronization Source – Identifies the originator of the report packet.

NTP timestamp – A thirty-two bit value that assists in the calculation of RTT propagation times. This is ideally selected to use a common wall-clock that can be used as a reference for all members.

RTP timestamp – A thirty-two bit value that corresponds to the NTP timestamp, but reflects the random offset as the RTP timestamp in data packets.

Sender's Packet Count – A thirty-two bit value that signifies the total number of RTP data packets sent by the sender from initial transmission to the current report packet being generated.

Sender's Octet Count – A thirty-two bit value that indicates the total number of payload octets sent by the sender from initial transmission to the current report packet being generated.

SSRC_n, Synchronization Source – A thirty-two bit SSRC identifier of the source that the information included in the report belongs to.

Fraction lost – An eight bit value that indicates the fraction of RTP data packets that have been lost since the previous sender or receiver report was generated. It is represented as the number of packets lost divided by the number of packets expected since the last report.

Cumulative Number of Packets Lost – A twenty-four bit value that represents the total number of RTP data packets that have been lost since communication began.

Extended Highest Sequence Number Received – A thirty-two bit value that corresponds to the highest RTP sequence number received from a given source, SSRC_n, with the lower sixteen bits representing that value and the most significant sixteen bits signifying the number of sequence number cycles.

Inter-arrival Jitter – Is a thirty-two bit value that characterizes the statistical variance of RTP data packet inter-arrival times. It is measure in timestamp units.

LSR, Last SR Timestamp – A thirty-two bit value which indicates the middle thirty-two bits of the total sixty-four bit NTP timestamp.

DLSR, Delay Since Last SR – A thirty-two bit value that denotes the elapsed time between receiving the last SR and sending a RR. It is expressed in units of 1/65536 seconds.

3.4 Selecting the Appropriate Codec

Coders are devices that encode or decode signals. Encoding a signal provides a more efficient form for transmission or storage, and decoding restores the signal to the original form. This compression and decompression of signals negatively impacts voice quality. Therefore, selecting the appropriate codec is necessary to obtain best quality of voice with the lowest bandwidth requirements. Codec selection itself can make an enormous difference in voice quality. There are many codecs specified by the ITU-T. Among these are G.711, G.723.1, and G.729. The G.711 codec uses Pulse Code Modulation (PCM) and transmits at 64 Kbps. This is equivalent to the traditional PSTN. Consequently, this codec uses high bandwidth, but has voice quality similar to PSTN. G.723.1 uses Algebraic Code Excited Linear Prediction (ACELP) techniques and encodes or decodes at 6.3 Kbps with a coding delay of 37.5 ms. G.723.1 can also decode at 5.3 Kbps using Multipulse Maximum Likelihood Quantization (MP-

MLQ) with similar delay. This codec uses significantly less bandwidth than G.711, but has a lower quality of voice and higher delay. Another codec, G.729, uses Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP) methods and consumes 8 Kbps with a coding delay of 15 ms. Table 1 describes briefly the differences between some popular codecs in delay and MOS scores.

Codec	Bit Rate (Kbps)	Delay (ms)	MOS Score
G.711	64	0.125	4.10
G.723.1 MP-MLQ	6.3	37.5	3.9
G.723.1 ACELP	5.3	37.5	3.65
G.729 CS-ACELP	8	15	3.92

Table 1: Values from "Integrating Voice and Data Networks" [5].

Table 2 illustrates the impairment attributed to various codecs. For G.723.1 ACELP, the impairment of this codec subtracts 19 points from the R factor theoretical maximum of 94.2. Therefore, a tradeoff exists between high-quality, high-bandwidth codecs and lower-quality, lower-bandwidth codecs that support more simultaneous calls while reducing clarity and increasing delay.

Codec	Bit rate (kbps)	Frame time (ms)	Look ahead (ms)	Codec impairment
G.711	64.0	10	0	0
G.723.1-MPMLQ	6.3	30	7.5	15
G.723.1-ACELP	5.3	30	7.5	19
G.729	8.0	10	5	11

Table 2: Various Codec Impairments [25].

.

Keep in mind that one way delay must be around 150 ms to maintain an acceptable voice quality for end users. A MOS score of 4 out of 5 is considered PSTN toll quality. Therefore, selecting the appropriate codec requires a tradeoff of bandwidth for delay. Using higher bandwidth codecs will trigger higher cost, while using a lower bandwidth will lower the quality of voice. Over the LAN, a codec such as G.711 at 64Kbps could be used with ease, as these local networks have much more bandwidth and are subsequently less expensive as a result. When utilizing Wide Area Network (WAN) resources, however, one may want to consider a codec such as G.729 that has a relatively high voice quality and low coding delay over the expensive WAN link. This will reduce the necessity of large bandwidth costs over the WAN.

T1 lines, one of the more popular transport lines used by companies, can support up to 24 simultaneous telephone calls under the traditional PSTN circuit-switched networks. These calls are encoded at 8 KHz with 8000 samples, or 64 Kbps. When using VoIP, more than 24 simultaneous calls can be supported as well as the existing data traffic due to the selection of voice codecs that will reduce the bandwidth requirement, yet preserve call quality. Therefore, upgrading the T1 lines may not be required, depending on current network usage. Avaya's [26] research states that “dedicated network transports supporting computer data on traditional telephony systems are generally about 30 percent utilized.” Only data network analysis will determine the current utilization of the link, but with an average of 70% utilization free on the T1 line, multiple VoIP calls can be made without requiring an increase in bandwidth.

3.4.1 Proposed Algorithm Codec Selection

Codec selection for this research will be achieved by employing a similar method to that proposed by Ossipov [20]. The proposed codec allocation scheme takes into account the dynamic network conditions as obtained from RTCP sender and/or receiver reports and expected call requirements, and it subsequently makes a codec selection based on those statistics. If current network conditions permit the use of higher bandwidth codecs, the codec selection scheme will select a codec appropriately. However, if insufficient bandwidth is available or the number of calls is too great, the codec selection scheme will only permit lower-bandwidth, lower quality codecs. Since this method adjusts utilization according to current network parameters, packet loss is likely to be reduced due to not oversubscribing the links. This is advantageous to the lower bandwidth codecs that have voice qualities that are more susceptible to packet loss.

The reception quality feedback provided in the RTCP reports was designed to be useful for senders, receivers, and third-party monitors. Network administrators can use applications that evaluate network performance based on these RTCP reports. Similarly, receivers can perceive problems with links across a wide spectrum. In addition, senders can adjust media capabilities based on the parameters including packet loss rates, round-trip delays, and jitter obtained from RTCP reports. These parameters are maintained cumulatively for overall performance measurements or at intervals between reports. This allows for measurements at shorter intervals that represent recent quality of the distribution or over longer intervals that represent overall reception quality.

With respect to packet loss rates, the fraction lost and cumulative number of packets lost RTCP fields in the reports provide useful measurements of loss rates. The fraction lost field indicates the percentage of packets lost between the two most recent reports. The cumulative number of packets lost field indicates the total number of packets lost over the duration of the call. Provided packet loss and delay parameters approximately follow a normal distribution, 95% of all future values will lie within two standard deviations of the loss rate or delay averages. The interval between reports varies based upon the number of participants. The interval is calculated in RFC 3550, section 6.3.1 Computing the RTCP Transmission Interval. As the number of members increases, reception information might not be kept for all receivers or the interval between reports could become exceptionally long. It calculates the interval between RTCP reports to be:

$$\max(\mathrm{Tmin}, n * C) \tag{3.1}$$

T_{min} is set to 2.5 seconds if the participant has not sent an RTCP packet or 5 seconds if it has already sent an RTCP packet. N corresponds to the number of members, and C is a constant that represents the average RTCP packet size divided by 75% of the RTCP bandwidth. For purposes of this research, T_{min} will be taken as the maximum, which will be 5 second intervals. The worst case scenario for obtaining two consecutive reports would be 10 seconds. Then, checking for trends over the course of one minute would contain at least 6 reports (60/10). Consequently, the fraction lost field will be exploited in the proposed algorithm to provide the most recent packet loss values for an interval that would include at least 6 consecutive reports.

Since it is desirable to keep one-way delay less than 150 ms, the RTCP report can be used to help select the appropriate codec to keep the delay to a minimum. The source SSRC_n can compute the Round-Trip Time (RTT) propagation delay to the receiver, SSRC_r, by noting the time, T, when the reception report is received and is local to the router. The sum of the last SR timestamp (LSR) and the delay since last SR (DLSR) is subtracted from the time T. The round-trip propagation time is therefore represented as:

$$RTT = T - LSR - DLSR \tag{3.2}$$

This resulting difference in time approximates the round-trip time delay, as some links experience exceptionally asymmetric delays. The RTT is divided by two to approximate the one-way delay. Provided RTT/2 is less than 150 ms, no codec change is required.

The inter-arrival jitter field is used to indicate congestion prior to packet loss. It is a short-term measure of network congestion, whereas the fraction lost or cumulative packet loss fields indicate more persistent levels of congestion. The inter-arrival jitter field provides a brief look at the jitter currently experienced. RFC 3550 explains that the jitter field should be used for comparison purposes across a number of reports in one network from one receiver. If the jitter buffer values in successive reports are continuously increasing, it could indicate an increase in network congestion. However, it also states that due to the variation in transmission delay of the packets reduces the accuracy of the jitter calculation. If the jitter calculation is used strictly for comparison purposes, the variation of the delay subtracts out. If the change is relatively small, then the increase in network congestion levels are likely insignificant.

The dynamic codec selection scheme allows for the most optimal codec to be used for the call based on the feedback provided by the RTCP reports that represent current network conditions. In traditional implementations, calls are configured to begin using a predefined codec. If the network normally has considerable bandwidth available for calls, the predefined codec could be G.711. If the network normally experiences a great deal of congestion, the codec could be G.729 or either of the G.723.1 codecs. In the first case, without the ability to renegotiate codecs mid-call, the call participants would be subject to worse call quality as network conditions worsen since it is using a high bandwidth codec. In the other case, calls using only the lower quality codecs are not making full use of the available bandwidth of the system, as at times the network may be able to support a codec of higher bandwidth and quality. The proposed scheme is designed to ensure calls use the highest quality codec based on network conditions.

The values for the reduction of I_e shown in Chapter 5, Table 8 correspond to an equivalent reduction in R factor. Using current implementations, the codec used to initialize the call must be used throughout the entire call, as no change in codec is permitted. Take for instance using the G.711 codec for a call - the codec exhibits extremely high call quality when packet loss is minimal. However, once packet loss nears 3% or higher, the I_e factor is higher than that of the G.729 codec with 0% packet loss. Recall G.711 uses 64 Kbps and G.729 uses only 8 Kbps. If the call is not allowed to renegotiate to a codec that requires less bandwidth, packet loss rates will remain at high levels or continue to rise, thus affecting the quality of the call more severely. However, the dynamic codec selection algorithm proposed in this research allows a change in codec to maximize the call quality by using the optimal codec. In the example above, at 3% packet loss, the system would request a change from G.711, which has an 11.15% reduction in I_e or R factor, to the G.729 codec, which only has a 10% reduction in I_e or R factor at 0% packet loss. The reduction in bandwidth from 64 Kbps to 8 Kbps will relieve congestion and improve the call quality.

Similarly, if the call is currently using the G.729 codec, and packet loss rates are in excess of 1%, it can be seen from Table 8 that a change to the G.723.1 6.3 Kbps codec would increase the call quality by an R factor of 17.91 – 15, or 2.91. The same situation follows for calls using the G.723.1 6.3 Kbps that experience packet loss rates in excess of 1%. In this event, a switch to the G.723.1 5.3 Kbps codec would ensure call quality remains as high as possible based on the network conditions.

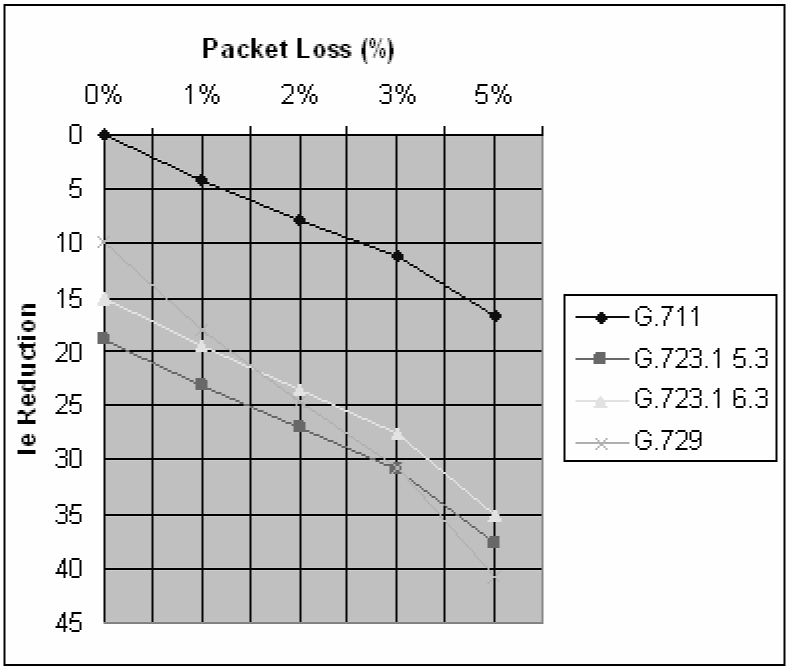

Figure 13: Reduction in I_e due to Varying Packet Loss Rates.

The situations expressed above represent conditions where the network is experiencing high levels of congestion which in turn causes high levels of packet loss. The dynamic codec selection scheme also provides for codecs to renegotiate to higher quality codecs if the network conditions permit. For example, if a call started with the G.729 codec, and network conditions were expressing congestion, the codec would have been renegotiated to G.723.1 6.3 Kbps. But, in the event network conditions improve, it would be beneficial for the codec to renegotiate to a higher quality codec, rather than remain at lower call quality.

The proposed algorithm will initiate a codec change request based on feedback received from RTCP reports, as shown by known results for codec with various packet loss rates in Figure 13. G.711 is used as the default codec to begin the call, as it requires the highest amount of bandwidth. Essentially, each codec represents a state, S, and each state has three future states, S_F, that can be reached based on the feedback received. The future states can either be an increase to a higher bandwidth codec, a decrease to a lower bandwidth codec, or no change.

Using the values from Table 8, G.711 can remain as the codec provided packet loss rates are not greater than 3%. However, if loss rates are higher than 3% and the one-way delays are within the bounds of the other codecs, it is possible that using either G.729 or G.723.1 6.3 Kbps codec can free up bandwidth to reduce packet loss. The use of either codec varies based on the one-way delay, as shown in Figure 14.

Similarly, if the current codec is G.729, the codec has three future states. It can remain as the codec if the packet loss rates are less than or equal to 1% or if the flexibility in delay does not permit the use of a lower bandwidth codec with higher delay. On the other hand, if loss rates are greater than the tolerable 1%, the codec will migrate to G.723.1 6.3 Kbps codec, provided the one-way delay is less than 82.5 ms (150 – 67.5). If packet loss rates exhibit relief, or if the one-way delay begins to exceed 150 ms, G.711 will be selected as the new codec.

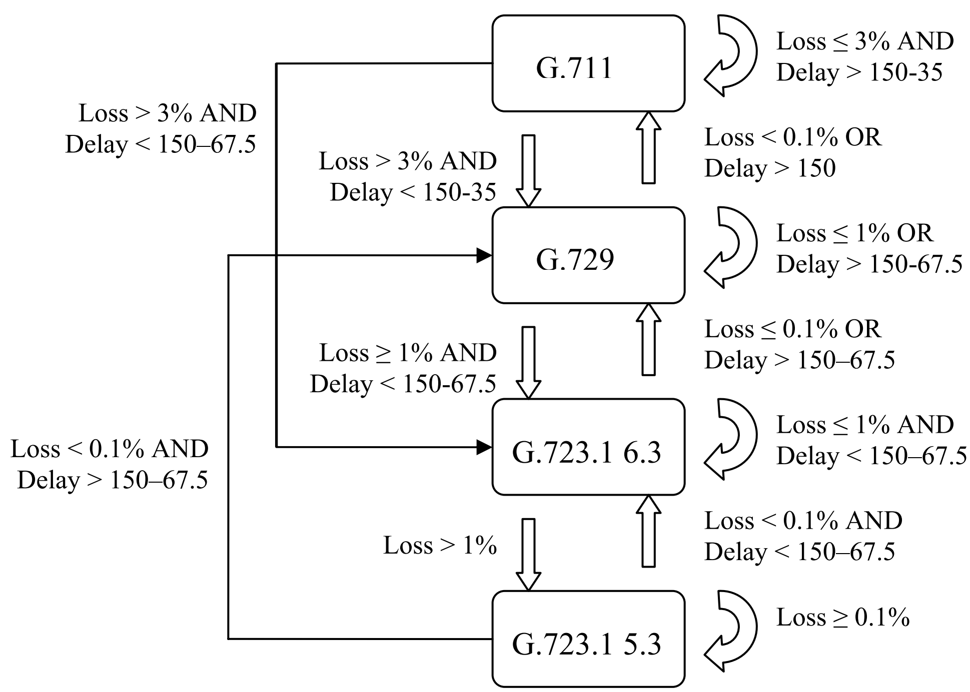

Figure 14: Codec Selection Flow Chart.

If the current state of the system is using the G.723.1 6.3 Kbps codec, there are also three possibilities for future states. If the packet loss rates remain relatively low, less than 1%, and the one-way delay does not exceed the 82.5 ms allowed, the system will remain in its current state. However, if loss rates begin to ease and the delay increase above 82.5 ms, the system will request a codec change to G.729. Finally, if the loss rates are greater than 1%, the system will request a change to the lower bandwidth G.723.1 5.3 Kbps codec.

If the system is currently using the lowest possible bandwidth codec, G.723.1 5.3 Kbps, and the loss rates are negligible, or less than 0.1%, the system can attempt to increase the quality of the codec used. If the one-way delay is less than 82.5 ms, the new requested codec will be G.723.1 6.3 Kbps, whereas if the one-way delay is higher than 82.5 ms, the new codec will be G.729. If loss rates remain high, the G.723.1 5.3 Kbps codec will continue to be used.

3.5 Jitter Buffer

Jitter buffers are necessary when using voice traffic. They are placed on the gateway at the receiving end to buffer voice packets as shown in Figure 15. Figure 16 illustrates how packets arrive at the de-jitter buffer. The progression from left to right in the figure shows packets may arrive out of sequence and at different times. In a fixed mode jitter buffer, the size is fixed, but can be manually configured as a certain size. Packets are then placed in the jitter buffer in the proper order required for playout. The Digital Signal Processor (DSP) receives these voice packets from the jitter buffer at evenly spaced intervals.

In an adaptive-mode jitter buffer, the size of the buffer can be adjusted depending on network conditions. The DSP algorithms in the codec adjust the average value of delay as network circumstances vary. The minimum and maximum values for the adaptive jitter buffer can be manually configured in Cisco voice-capable routers based on network needs. The algorithms are capable of decreasing jitter slowly and increasing delay quickly so voice quality is not impacted to a great extent.

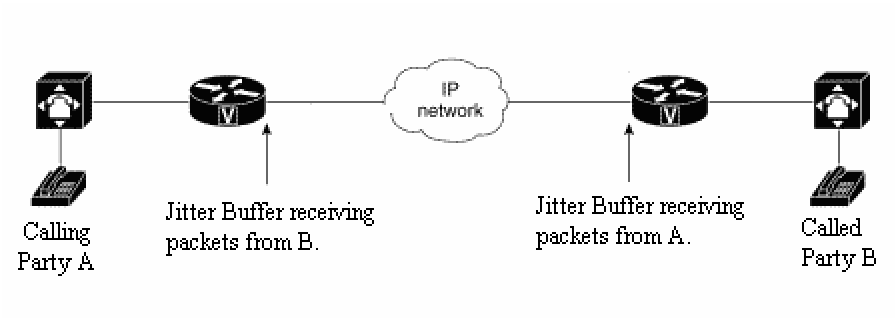

Figure 15: Cisco Jitter Buffer Placement [27].

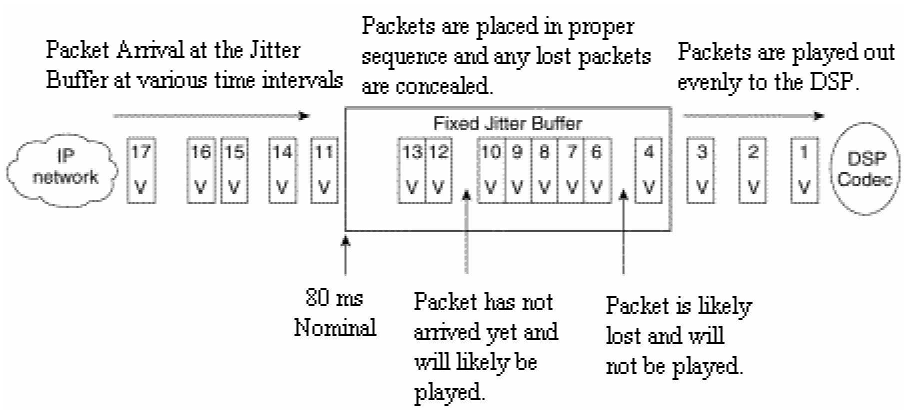

Figure 16: Cisco Fixed Mode Jitter Buffer [27].

3.6 Proposed Handshake Mechanism

The proposed scheme is designed to take advantage of the resource-centric method. A codec change request is initiated based upon the parameters determined from that approach. The approach continuously monitors resource availability and the requirements of current calls by analyzing parameters such as available bandwidth, packet loss rates, and overall end-to-end delay. These parameters can be obtained either with the use of the Simple Network Management Protocol (SNMP) or by evaluating sender and receiver reports in RTCP, which provide feedback on current network statistics. Codec allocation is therefore determined based upon available resources obtained from network monitoring procedures. The algorithm selects the best codec option based on these available resources while attempting to maximize voice quality. Once the current network resources have been determined and a codec selected, the proposed algorithm provides a means for changing to the appropriate codec in the middle of the call.

Take the following situation for example. There are two H.323-capable stations, A and B. Assume the sender always initiates the codec change request and that both the sender and receiver use different codecs for sending. A three-way handshake is designed to accomplish mid-call codec renegotiation as illustrated in Figure 17 and explained in the following steps:

1) Station A determines that a codec change is needed. It will send a control packet to B indicating which codec it would like to use and continues to send RTP packets with the current codec.

2) Station B receives the change request from A and checks whether is supports the requested codec. If the requested codec is not supported by station B, it replies with a negative acknowledgement (NACK). However, if the new codec is supported, station B replies to A with a message indicating which RTP sequence number will begin the new codec transmission. The RTP sequence number must be calculated at a minimum of 3 round-trip times (RTT) from the current point in time, allowing station A to acknowledge. A minimum of 1.5 RTT's are required to allow the packet to reach the destination and an acknowledgement to return. Therefore, $2 \times$RTT is used to compensate for any unforeseen delays due to the inconsistency of delays over the Internet [19].

3) Station A receives the response from B and replies with an ACK to confirm the sequence number or send a more realistic sequence number back. If the response

from B is never received, perhaps because of packet loss, it re-tries the request as in (1).

4) Station B receives the ACK from station A and becomes ready to accept packets with the new codec starting at the previously determined RTP sequence number. All of the packets using the new codec will then be placed into the same jitter buffer used for the original codec. Packets in the jitter buffer prior to the codec switch will continue to be played out, followed by packets using the new codec. If station B never receives the ACK from A, it re-calculates the RTP sequence number to be used and sends a new ACK with that sequence number back to A as in (2).

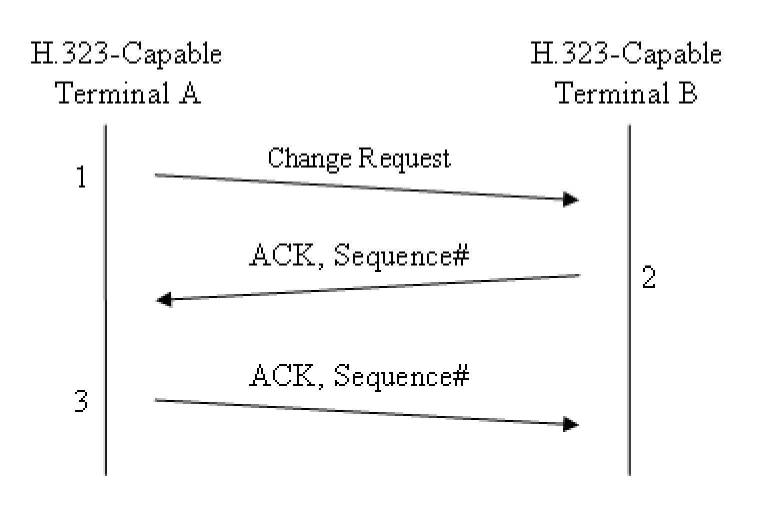

Figure 17: Proposed Three-Way Handshake Mechanism for Codec Change.

3.7 Technical Description

The proposed method in 4.1.1 employs H.323 station A sending a control message to B indicating a change in codec is requested. This is achieved by sending a modified H.245 message. Step one in Figure 17 sends a modified terminalCapabilitySet message to B. ITU H.245 ANNEX A specification [28] describes setup messages that include PDU types for terminalCapabilitySet.object, where objects include transmitMultipointCapability, receiveAndTransmitMultipointCapability, mcCapability, among others. It is unnecessary to retransmit all of these objects because this process is only changing the audio codec. As a result, a modified packet can be send that strips all objects, with the exception of the transmitAudioCapability object that specifies the codec to be used. The reduced packet size will reduce delay by not adding needless data transfer.

The original H.245 capability exchange packet is 169 bytes on the wire. The proposed protocol results in a new packet size that will include 14 bytes for Ethernet header, 20 bytes for IP header, 20 bytes for TCP header, and 4 bytes for TPKT header size. Then, there will be 11 bytes for the terminalCapabilitySet portion and capabilityTable Item 0 which depicts the codec to be used. Therefore, the packet size will be only 69 bytes on the wire, as opposed to the original 169. The packet layout is shown in Figure 18.

```
⊞ Ethernet II
⊞ Internet Protocol
⊞ Transmission Control Protocol
⊟ TPKT, Version: 3, Length: 11
    Version: 3
    Reserved: 188
    Length:11
⊟ H.245
  ⊟ PDU Type: request (0)
    ⊟ request: terminalCapabilitySet (2)
      ⊟ terminalCapabilitySet
          sequenceNumber: 1
          protocolIdentifier: SOTAOT
        ⊞ multiplexCapability: h2250Capability (4)
        ⊟ capabilityTable: 5 items
          ⊞ Item 0
            ⊟ Item
                capabilityTableEntryNumber: 2
              ⊟ capability: receiveAudioCapability (4)
                ⊟ receiveAudioCapability: g711Alaw64k (1)
                    g711Alaw64k: 20
```

Figure 18: Packet Format for Step 1.

Step two in the process involves station B determining if the new codec is supported. If not, the request for codec renegotiation is denied with a NACK. However, if the codec is supported, B responds with a modified packet that includes terminalCapabilitySet.receive-AudioCapability. The receiveAudioCapability object indicates B is accepting the requested codec. Furthermore, it is assumed the router calculates an RTP sequence number a minimum of two RTT's in advance. This RTP sequence number must be sent in the ACK back to station A so it knows which RTP sequence number will contain payload with the newly negotiated codec. This 16-bit RTP sequence number will be appended onto the end of the ACK packet. The router will automatically interpret the last 16-bits of the ACK as the sequence number making use of the new codec. The packet format is shown in Figure 19.

```
⊞ Ethernet II
⊞ Internet Protocol
⊞ Transmission Control Protocol
⊟ TPKT, Version: 3, Length: 17
    Version: 3
    Reserved: 188
    Length:17
⊟ H.245
  ⊟ PDU Type: request (0)
    ⊟ request: terminalCapabilitySet (2)
      ⊟ terminalCapabilitySet
          sequenceNumber: 1
          protocolIdentifier: SOTAOT
        ⊞ multiplexCapability: h2250Capability (4)
        ⊟ capabilityTable: 5 items
          ⊞ Item 0
            ⊟ Item
                capabilityTableEntryNumber: 2
              ⊟ capability: receiveAudioCapability (4)
                ⊟ receiveAudioCapability: g711Alaw64k (1)
                    g711Alaw64k: 20
   ⊟ RTP Sequence Number
     ⊟ sequenceNumber: 16-bit Value
```

Figure 19: Packet Format for Step 2.

The new packet size will include 14 bytes for Ethernet header, 20 bytes for IP header, 20 bytes for TCP header, and 4 bytes for TPKT header size. Then there will be 11 bytes for the terminalCapabilitySet.receiveAudioCapability object. Then, the 2 bytes of RTP sequence number will be appended. Consequently, the packet will be 71 bytes in size.

Step three involves station A confirming the RTP sequence that will contain the new codec. Station A responds to B with a terminalCapabilitySetAck message and the appended sequence number it will anticipate containing the new codec payload. As mentioned previously, A could reply to B with a more realistic sequence number. To demonstrate the reason for this function, assume B requests the change in a packet with RTP sequence number 10, but A missed and sent the new codec in a packet with sequence number 11. B would be expecting A to send the new codec in 10, but it would instead interpret packet 10 to have the new codec, when in reality it was still using the old codec. The new packet size will include 14 bytes for Ethernet header, 20 bytes for IP header, 20 bytes for TCP header, 4 bytes for TPKT header size, 3 bytes for terminalCapabilitySetAck, and 2 bytes for the RTP sequence number. This yields a new packet size of 63 bytes. This packet layout is portrayed in Figure 20.

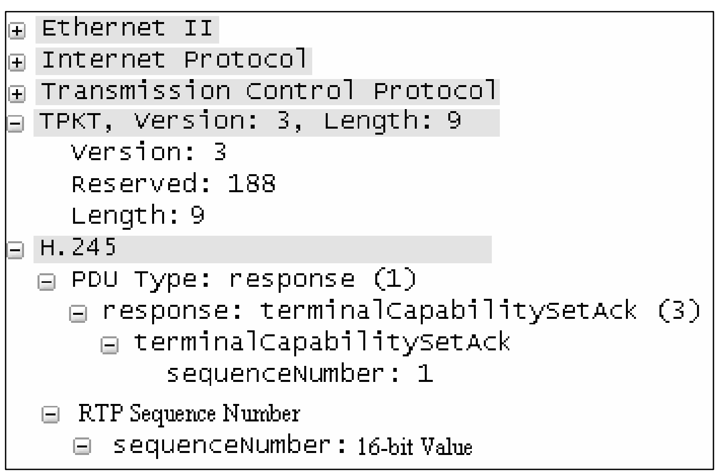

Figure 20: Packet Format for Step 3.

CHAPTER 4

METHODOLOGY

4.1 Overview

Ensuring proper quality of service is the biggest technical challenge in transitioning to a voice over IP environment for both LANs and WANs, according to AT&T [29]. Therefore, quality of service is an integral part of any network planning that involves the integration of voice or any real-time related traffic. Voice, video, audio, and other real-time traffic should be given the highest priority so that these packets experience less delay. If real-time traffic is not prioritized and is given the same quality of service level as data, as congestion in the network increases voice conversations will become erratic. Non real-time traffic is often unaffected and tasks such as visiting a website or sending an email go unnoticed by the user because the delays are insignificant to the operation of that application. This is not the case for VoIP or other real-time data. To maintain acceptable call quality, it is imperative to provide this high level of service to the information that needs it the most.

Reliability can be improved with appropriate QoS configurations. Ideally, voice networks were designed with the objective of a 99.999% reliability. This equates to only two hours of downtime for every forty years of operation. Although this is typically the case with the traditional PSTN, this is a much more difficult level of reliability to obtain in data networks. Some additional considerations to improve QoS on the network are to remove hub devices from the LAN and replace them with 100Mbps switches. This will reduce the chance of excessive collisions, which create additional packet loss and jitter as the traffic on the LAN increases. Furthermore, switches and routers should be connected at full-duplex, or at minimum the same duplex on each end to avoid large numbers of corrupt packets.

4.2 Variables Affecting Voice Quality

As previously mentioned, there are numerous variables that affect voice quality [5]. Background noise, such as talking in a noisy restaurant, can negatively affect the perceived quality of voice. Signal levels, or audio volume, generally remain constant when converted from analog to digital. However, the signal becomes distorted if the signal levels are too high or too low. If a signal is too large, the signal is clipped which distorts the original signal. This is known as amplitude clipping. Codec distortion results due to lossy compression schemes in low bit-rate speech-coding algorithms. The receiver does not receive the entire original signal due to this lossy compression scheme. Temporal clipping can be seen in voice activity detection (VAD) systems that eliminate background noise and conserve bandwidth when silence is detected. When the system returns to speech transmission mode from silence suppression mode, beginnings of words may be clipped and lost. Since low bit-rate codecs model voice patterns of a single talker, when multiple talkers speak on the line, perhaps in a conference, quality suffers.

Further variables affecting voice quality include circuit noise, jitter, echo, packet loss, and delay. Analog circuits can introduce unwanted signals such random electrical noise. Jitter is a variation in the inter-arrival times of packets. The higher the variation the more likely the voice quality will degrade. The National Institute of Standards and Technology (NIST) [30] recommends jitter of less than 40ms. If packets are received out of order, they need to be reassembled at the destination. If these packets are received too far apart, the entire section of speech may be useless. An echo is the result of speech to the receiver reflecting back to the speaker, causing the speaker to hear an echo of his or her own voice. Packet loss can adversely affect the quality of voice calls, especially if multiple packets are lost in succession, known as burst loss. If a large number of these packets do not reach the destination, the end user will likely hear choppy voice quality due to a lack of speech information. NIST also recommends below one and three percent packet drop to maintain a higher level of call quality. Delay is the time elapsed from when a packet is sent to the time it is received at the other end. One way delay over 150 ms can negatively impact the perceived voice quality by the end user.

According to Network General [31], there are numerous components to total end-to-end delay. Of these, signal encoding and decoding typically run between 15 ms and 37.5 ms at the source and destination. Protocol overheads including RTP, UDP, IP information, and echo cancellation typically contribute less than 5 ms. Loading packets onto the LAN and WAN can introduce framing and queuing delays in a range of 5ms to 25 ms depending on the transmission rate. Routing, propagation, and queuing delays across the WAN typically incur 10 ms to 40 ms of delay due to the transmission media and distance traversed. As mentioned earlier, jitter may present between 20 ms and 40 ms or more due to reorganizing the packets that have varying arrival times. Many of these factors are fixed, and cannot be changed. However, selecting the appropriate codec can influence overall network performance.

Wide Area Network connections are often congestion points for networks due to their support of less bandwidth. Merging this traffic onto a single network could cause congestion and over-utilize the existing WAN connections. Therefore, understanding the current constraints on the WAN connections to ensure convergence does not negatively affect the amount of available bandwidth that is necessary. Two major sources of WAN delays are caused by serialization and propagation delay. Serialization delay is the time required to place a packet onto the WAN access line. Serialization delay can be calculated using the following equation:

$$S_D = \frac{(PacketSizeinBytes) * \left(\frac{8bits}{byte}\right)}{LinkSpeedinBitsPerSecond} \tag{4.1}$$

For example, a 1500 byte packet inserted onto a T1 1.544Mbps line will have a serialization delay of

$$S_D = \frac{(1500)*\left(\frac{8bits}{byte}\right)}{1.544x10^6} = 7.77ms$$

Propagation delay is the delay associated with the actual physical characteristics of end-to-end transportation of the packets across fiber-optic-based or copper-based medium. Propagation delay approximates to 10 ms of delay for every 1000 miles on the wire. Therefore, calls across the continental U.S. could have upwards of 30 ms of propagation delay for simply transmitting the information. These two sources of WAN delay must be taken into consideration in order to keep the one-way voice transmission delay under the recommended 150 ms. If this delay is not kept to a minimum, users will experience low voice quality.

4.3 Measuring Voice Quality

Measuring speech quality can be accomplished through many means. Measurements can be done subjectively, objectively, or can be estimated. As mentioned previously, subjective measurements are considered the most reliable and respected approach for measuring voice quality. Three popular methods for subjective speech quality measurement include Mean Opinion Score (MOS), Comparison Mean Opinion Score (CMOS), and Degradation Mean Opinion Score (DMOS). In MOS, subjects listen to samples of speech and rank the perceived quality on a scale as shown in Table 3.

In CMOS, subjects listen to pairs of samples and rank the samples by which has best quality sound and by how much the sound is better. One sample is a control or reference sample, such as the G.711 codec, which produces good quality sound. The other sample is the variable being tested. These ratings are shown in Table 4.

MOS Score	Listening Quality Scale	Listening Effort Scale
5	Excellent	Complete relaxation possible; no effort required
4	Good	Attention necessary; no appreciable effort required
3	Fair	Moderate effort required
2	Poor	Considerable effort required
1	Bad	No meaning understood with any feasible effort

Table 3: Rating Scale for MOS.

CMOS Score	Quality of Second Sample Compared to the Control
3	Much Better
2	Better
1	Slightly Better
0	About the Same
-1	Slightly Worse
-2	Worse
-3	Much Worse

Table 4: Rating Scale for CMOS.

Objective Speech Quality Measurement is much less time-consuming and less expensive. The intent of objective measurement is to have a machine-based, automatic assessment that provides comparable results to subjective tests, without incurring the extensive time and expense associated with testing. A popular objective method is known as Perceptual Speech Quality Measurement, or PSQM. This standard by the ITU is specified in recommendation P.862 [32]. This method provides similar results to that of subjective methods, but is not as accurate. PSQM is tailored more for laboratory settings, but is difficult to implement in live networks [5].

The Equipment Impairment Factor Method (EIF) is popular for situations when an existing network does not exist. It assists in predicting what voice quality could be expected when the network is in the design and planning stages. This method is specified by the ITU in recommendation G.113 [33]. The value attained by the EIF method represents an additive value representing all of the impairment factors. This value can then potentially be converted to a subjective test score such as MOS.

The E-model is an extension of the EIF model that considers additional impairments not covered by the EIF. The E-model uses a scalar value, known as the R factor, to represent call quality on a scale of 0 (poor) to 100 (excellent), with any value below 60 considered as not recommended [14]. The R factor is calculated using the following equation:

$$R = R_0 - I_s - I_e - I_d + A \tag{4.2}$$

R_0 represents the signal to noise ratio. Is depicts simultaneous factors including excessive loudness, sidetone (signal providing feedback from mouthpiece to earphone), and quantization distortion. I_d takes into account the end-to-end delay factors such as talker echo, listener echo, and absolute delay. I_e comprises impairments induced by equipment such as non-waveform low bit-rate codecs and packet network effects, among others. 'A' represents the advantage, or expectation factor, gained by reducing the expectation of end-users due to some other benefit. It accounts for lowered expectations of quality of the user in exchange for the service provided, such as the trade-off of lower quality voice in cellular phone usage while achieving higher mobility. The resultant, R, indicates the expected value of user satisfaction, and can be directly correlated to a MOS value. The R factor equation can be reduced further because only I_d and I_e are typically considered variable in a VoIP network [34]:

$$R = R_0 - I_e - I_d \tag{4.3}$$

The theoretical value of R_0 is reduced to 94.2 [35] due to inherent degradation that results when voice is converted from analog to digital and back. The equation is therefore simplified to:

$$R = 94.2 - I_e - I_d \tag{4.4}$$

Estimating the impact of loss due to equipment, I_e, is accomplished by analyzing transmission and encoding losses, including playout buffer and network losses. In [35] and [36], it has been shown that I_e can express the overall packet loss rate e as follows:

$$I_e = \gamma 1 + \gamma 2 * \ln(1 + \gamma 3e) \tag{4.5}$$

Encoding voice quality impairment is represented as γ_1, while γ_2 and γ_3 describe the impact of codec loss on perceived voice quality as calibrated using subjective and objective simulations for several codecs in [36]. Application factors such as codec performance and packet loss concealment (PLC) are also taken into account. These codecs and respective values are summarized in Table 5.

Codec	**Frames/pkt**	**PLC**	γ_1	γ_2	γ_3
G.711	2	none	0	30.0	15
G.723.1.B-5.3	1	silence	19	71.38	6
G.723.1.B-6.3	1	silence	15	90.00	5
G.729	1	silence	10	47.82	18
G.723.1.A+VAD-6.3	1	none	15	30.50	17
G.729A+VAD	2	none	11	30.00	16

Table 5: Values of $\gamma 1$, $\gamma 2$, and $\gamma 3$ Calibrated in [36].

Estimating the impact of end-to-end delay, d, depends on various network delay factors. $D_{network}$ is the sum of the propagation and queuing delays leading to jitter. D_{codec} represents codec-related delays caused by packetization and encoding, including processing and look-ahead. $D_{playout}$ is denoted as the playout buffer delay. To determine values for D_{codec}, it is necessary to calculate the packetization delay which is derived as follows:

$$D_{codec} = \frac{frames}{packet} \times framesize + lookahead + process \tag{4.6}$$

Aside from selecting a codec based on bit-rate, it is also important to note algorithmic delays such as look-ahead and processing complexity. For example, if two G.729 frames are encapsulated into a single RTP packet, there is a packetization delay of 20 ms, a 5 ms look-ahead delay, and a 10 ms processing delay. Table 6 illustrates varying D_{codec} values for different codecs.

Codec	Frame size	Frames/pkt	Packetization	Encoding	D_{codec}
G.711	0.125 ms	2	< 1 ms	< 1 ms	2.25 ms
G.723.1	30 ms	1	30 ms	37.5 ms	67.5 ms
G.729	10 ms	2	20 ms	15 ms	35 ms

Table 6: Codec-Related Delays for Different Codecs [5].

Due to varying network conditions, the delivery of voice packets has an irregular nature. As such, different voice packets in the same call can arrive at the destination at different times, thus arriving out of sequence. To compensate for this effect, a jitter buffer is placed near the receiver. The buffer receives voice packets at irregular intervals, holds the packets temporarily, reorders them in the proper sequence if necessary, and plays them to the decoder at regular intervals. $D_{playout}$ characterizes the delay produced by these playout buffers.

Cisco [26] routers have configurable playout buffer configurations. Under voice dial-peers, playout delay modes can be set for either adaptive or fixed. The adaptive playout mode allows the jitter buffer size and amount of playout delay to be adjusted during a call based on current network conditions, whereas the fixed playout mode does not allow the buffer size to adapt to network conditions and implements a constant playout delay. Playout values can be specified for maximum, nominal, and minimum delays, provided the maximum is greater than or equal to the nominal value, and the nominal value is greater than or equal to the minimum value. Default playout values are 200 ms for maximum, 200 ms for nominal, and 40 ms for minimum playout delay. Minimum playout values may also be specified for low (10 ms) for low jitter network conditions or (80 ms) for high jitter network conditions.

This overall end-to-end delay is represented as the sum of the three components:

$$d = D_{network} + D_{codec} + D_{playout} \tag{4.7}$$

It is noted in [35] that voice quality degrades more rapidly as end-to-end delay exceeds 177.3 ms. This accelerated degradation is illustrated in Figure 21. This figure illustrates the requirement to keep one-way delay under 150 ms.

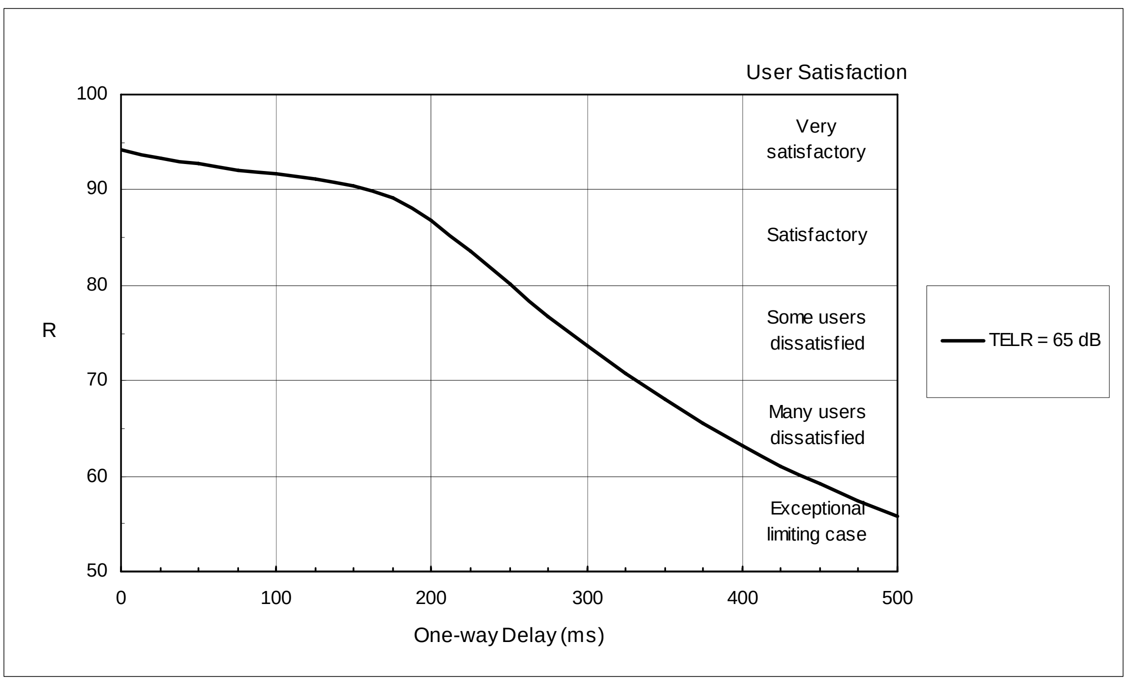

Figure 21: R Factor Reduction due to One-way Delay [37].

The authors also provide values of I_d for selected, one-way delays as shown in Table 7. The resultant equation modeling this effect is represented by Equation 4.8:

$$I_d = 0.024\,d + 0.11\,(d - 177.3)\,I(d - 177.3) \tag{4.8}$$

$$\text{Where } \mathbf{I}(x) = \begin{cases} 0, & x < 0 \\ 1, & \text{otherwise} \end{cases}$$

One-way Delay (msec)	I_d
0	0
25	0.9
50	1.5
75	2.1
100	2.6
125	3.1
150	3.7
175	5.0
200	7.4

Table 7: Values of Delay Impairment for Selected, One-Way Delay.

Once an R factor value has been obtained, it is desired to translate it into a relative MOS score. The R factor is related to MOS in the following equations [14]:

$$\text{MOS} = \begin{cases} 1 & R < 0 \\ 1 + 0.035 * R + 7.10^{-6} * R * (R - 60) * (100 - R) & 0 < R < 100 \\ 4.5 & R > 100 \end{cases} \tag{4.9}$$

User satisfaction levels are shown in Figure 22 and a graphical representation of the relationship between R factor and MOS is illustrated in Figure 23.

Figure 22: User Satisfaction Rankings [25].

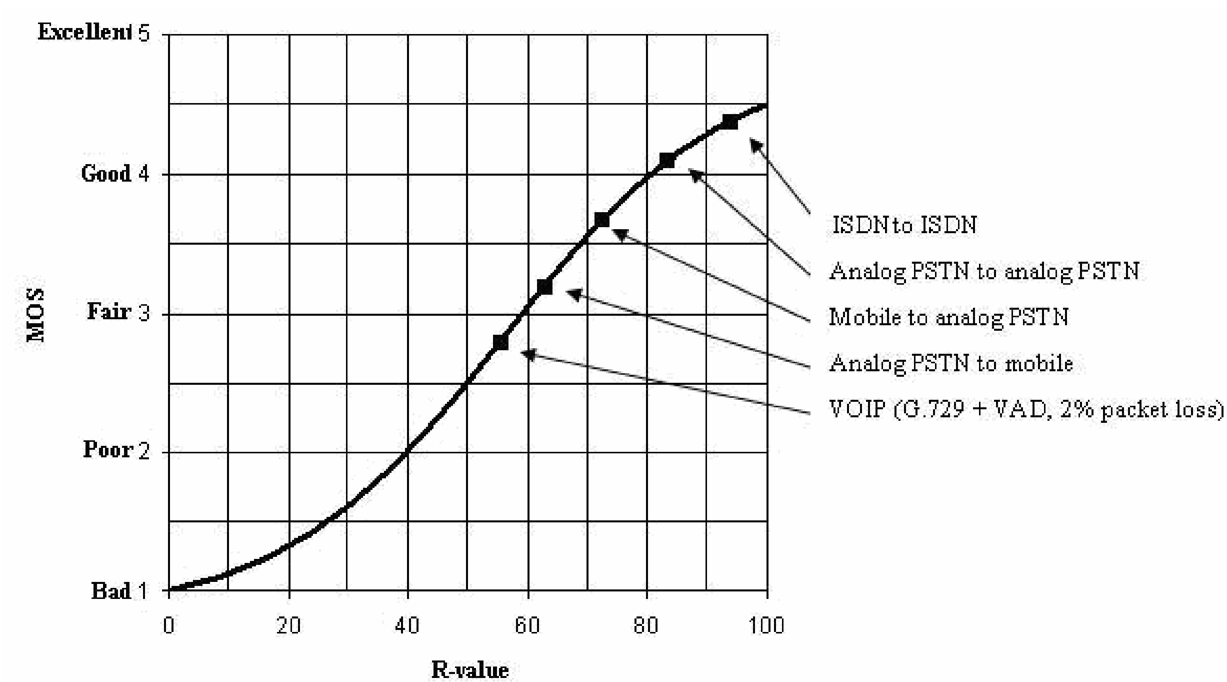

Figure 23: Relationship between R factor and MOS [5].

CHAPTER 5

ANALYSIS AND RESULTS

5.1. Calculate R Factor Equipment Impairment (I_e)

To calculate the E-model R factor values, recall Equation 4.4. Equation 4.5 will provide values for I_e necessary to calculate the R factor. These values differ based on the codec used, and can be calculated using the values in Table 5. Using the G.723.1 5.3 Kbps codec and assuming a 2% packet loss, it can be seen that Ie is determined below. Values of I_e with various packet loss rates are represented in Table 8 and illustrated in Figure 13. These equipment impairment values will remain the same for either buffer option.

$$I_e = \gamma 1 + \gamma 2 * \ln(1 + \gamma 3e)$$

Where $\gamma_1 = 19$, $\gamma_2 = 71.38$, $\gamma_3 = 6$, $e = 0.02$

$$I_e = 19 + 71.38 * \ln(1 + 6 * 0.02) = 10.91$$

Codec	Packet Loss Rates					
	0%	1%	2%	3%	5%	10%
G.711	0	4.19	7.87	11.15	16.79	27.49
G.723.1.B-5.3	19	23.16	27.09	30.81	37.73	52.55
G.723.1.B-6.3	15	19.39	23.58	27.57	35.08	51.49
G.729	10	17.91	24.70	30.64	40.69	59.24

Table 8: Values for I_e With Variable Packet Loss Rates.

5.2 New Buffer Delay Impairment (I_D)

In the new buffer model, packets with the newly negotiated codec will be placed into a new playout buffer which is created to accommodate packets using the new codec. The same three-way handshake process will be followed for either buffer model. The sender will continue to send RTP packets to the original playout buffer until the agreed-upon sequence number is reached, and immediately after switching the codec, all of the future packets will be placed into the new playout buffer. Figure 24 demonstrates how packets using the new codec, which are packets beginning with RTP sequence number 11, proceed to fill up the new jitter buffer. Packets from the initial playout buffer will be played until it is empty, then switch to buffer two and play from there. The original buffer would then be de-allocated to free up space.

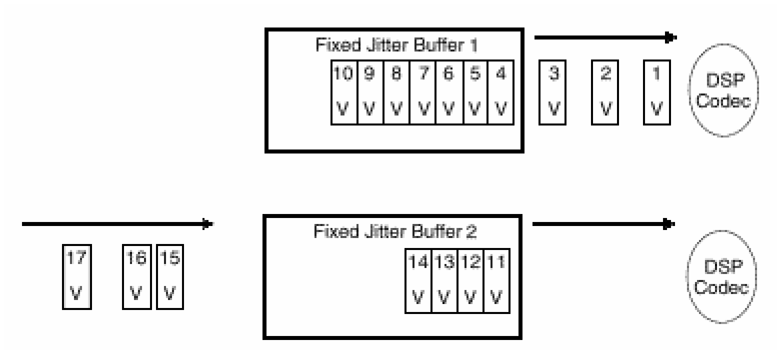

Figure 24: Creation of a Second Playout Buffer after Codec Renegotiation.

The value of I_d must now be calculated to determine the R factor. Recall from Equation 4.7 that the delay, d, is the sum of three components: $D_{network}$, D_{codec}, and $D_{playout}$. As mentioned previously, $D_{network}$ is the sum of the one-way propagation, serialization, and queuing delays as shown in Equation 5.1. Propagation and queuing delays will remain static in the buffer reuse model. Only serialization delays will change due to the negotiation of a new codec.

$$D_{network} = D_{propagation} + D_{serialization} + D_{queuing} \tag{5.1}$$

Propagation and serialization are calculated below using Equations 5.2 and 4.1. Serialization delays (S_D) for various codecs are shown in Table 9.

$$\Pr opagationDelay = \frac{10ms}{\frac{5280\,ft}{mile} \times 1000 miles} = \frac{Dms}{14\,ft} = 26.5ns \tag{5.2}$$

$$S_D = \frac{CodecPacketSize(bytes) \times 8 \frac{bits}{byte}}{10 \times 10^6} \quad (4.1)$$

Codec	**Size on Wire (bytes)**	**S_D (ms)**
G.711	214	0.1712
G.723.1-5.3	74	0.0592
G.723.1-6.3	78	0.0624
G.729	74	0.0592

Table 9: Serialization Delay Values for Various Codecs.

Serialization delay is the only delay that will affect $D_{network}$ with in the new buffer creation model. Table 9 displays a range of packet sizes for various codecs. Take for instance changing from a lower payload codec, g.729, to a higher payload codec, g.711. There is a difference of 140 (214-74) bytes in size of packet on the wire. The serialization delay associated with this difference will momentarily add to the overall network delay while the original buffer is filled. Moving from a lower payload codec to a higher payload codec increases the serialization delay. The relationship is demonstrated in Equation 5.3 below. On the other hand, if going from a high payload codec, g.711, to a lower payload codec, g.729, this value would not increase overall network delay. Queuing and propagation delays remain static for this network but will change significantly across greater distances or the Internet.

$$S_{Dincrease} = \frac{NewCodec(bytes) \times 8 \frac{bits}{byte}}{10 \times 10^6} - \frac{OldCodec(bytes) \times 8 \frac{bits}{byte}}{10 \times 10^6} \quad (5.3)$$

$$S_{Dincrease} = \begin{cases} 0, & new < old \\ S_{Dnew} - S_{Dold}, & new > old \end{cases}$$

Packet traces indicate this sample network has a queuing delay of 654.5 μs. Both propagation and queuing delays are assumed static. D_{codec} represents codec-related delays caused by packetization and encoding, including processing and look-ahead. Values for D_{codec} can be obtained from Table 6. $D_{playout}$ is denoted as the playout buffer delay. Playout buffer delay for this network is configured to be a fixed 80 ms. Once the values of $D_{network}$ have been determined, and taking the appropriate value for D_{codec} from the table, Equation 4.8 provides the appropriate estimate for I_d. For this network, overall delay, d, is simplified to:

$$d = D_{network} + D_{codec} + 80\,\text{ms} \tag{5.4}$$

These values can subsequently be used in Equation 4.8 to determine the delay impairment.

5.3 Buffer Reuse Delay Impairment (I_D)

An alternative option to creating a new buffer involves the reuse of the original playout buffer, which is illustrated in Figure 254. Voice packets one through ten encoded with the original codec, and voice packet eleven has the specified RTP sequence number, thus it carries the new voice codec. Packets one through ten continue to play out to the receiver. Packet eleven will enter the jitter buffer behind packet ten, and all subsequent packets will follow.

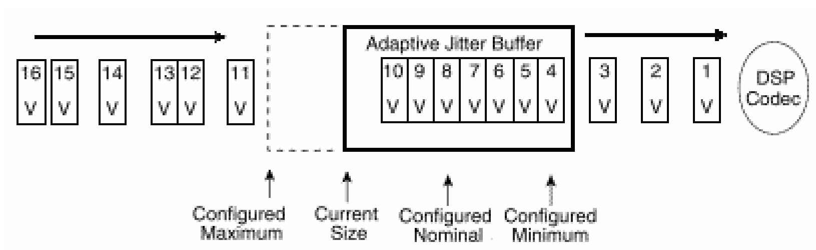

Figure 25: Reuse of Original Playout Buffer.

As noted in section 5.2, only serialization delays will change due to the negotiation of a new codec for this network. Propagation and queuing delays will remain constant. Values for codec delay will also remain the same. Serialization delays will be affected identically as shown in Equation 5.3 in both buffer models. Unlike the fixed buffer size used in the creation of a new playout buffer, reuse of the original playout buffer will require a flexible buffer size. As mentioned previously, moving from a lower payload codec to a higher payload codec increases the serialization delay. Therefore, it may be necessary for the jitter buffer to vary in size to accommodate this increase in delay. Consequently, the buffer reuse model will require an adaptive jitter buffer.

The adaptive jitter buffer can be manually configured with minimum, nominal, and maximum values as explained in section 4.3. The network used in this research was manually configured with a nominal value of 80 ms, with the default minimum of 10 ms. The maximum value to configure for the adaptive jitter buffer will depend on the increase in delay associated with the new codec. For this network, since propagation and queuing delays are static, serialization delay will dictate the maximum. In networks where propagation and queuing delays are dynamic, such as traversal of VoIP calls across the internet, the maximum jitter buffer size will be higher. As such, for a worst case scenario $D_{playout}$ would be equivalent to the maximum of the adaptive jitter buffer size, or $D_{playoutMax}$. This mathematical representation for delay, d, in this case is shown below:

$$D = D_{network} + D_{codec} + D_{playoutMax} \tag{5.5}$$

5.4 Example

To illustrate the calculation of an R factor value for the codec renegotiation method, take this situation as an example. H.323-capable station A is using G.723.1-ACELP, a 5.3 Kbps codec in RTP voice packets sent to H.323-capable station B. Station B is transmitting voice packets to A with the G.711 codec, 64 Kbps. Station A sends a control packet with terminalCapabilitySet to station B requesting a codec change so it is also using G.711. Station B recognizes G.711 as a valid codec, and calculates an RTP sequence number that will initiate the codec change. B accepts the request for codec change and acknowledges to A with a receiveCapabilitySet ACK and the RTP sequence number where the change should occur. Station A receives the ACK from B and determines if the RTP sequence number is reasonable. If so, it sends a terminalCapabilitySetAck packet back to B with the same RTP sequence number.

To calculate the R factor for this situation, recall the R factor is the sum of I_e and I_d. Assuming a 1% packet loss rate, Table 8 provides an I_e of 4.19 for G.711. Calculating the delay in the network involves the sum of $D_{network}$, D_{codec}, and $D_{playout}$. The fixed delays in $D_{network}$ are propagation and queuing delays, which are 26.5 ns and 654.5 μs respectively. With G.711 as the new codec and G.723.1-ACELP as the old, $S_{Dincrease}$ from Equation 5.3 yields 0.112 ms. Therefore, Dnetwork is calculated as:

$$\text{Dnetwork} = \text{Propagation} + \text{Queuing} + S_D + S_{Dincrease} \tag{5.6}$$

$$\text{Dnetwork} = 26.5\text{ ns} + 654.5\ \mu\text{s} + 0.1712\text{ ms} + 0.112\text{ ms} = 0.938\text{ ms}$$

The value of D_{codec} can be selected from Table 6. For G.711, D_{codec} is 2.25 ms. Assuming a fixed playout buffer taken from the method for creating a new playout buffer to accept packets with the new codec from section 4.2.2, the value for $D_{playout}$ is 80 ms. The end-to-end delay, d, is then calculated with Equation 4.7 as shown:

$$d = 0.938\text{ ms} + 2.25\text{ ms} + 80\text{ ms} = 83.188\text{ ms}$$

Then, Id is calculated using Equation 4.8 as shown:

$$I_d = 0.024 * d + 0.11 * (d - 177.3) * \mathbf{I}(d - 177.3) = 1.997$$

As such, the R factor value for this example is calculated with Equation 4.3:

$$R = 94.2 - 4.19 - 1.997 = 88.013$$

This R factor value can be correlated to a MOS score with Equation 4.9. This results in a MOS value of approximately 4.24, in which users are satisfied with the call quality. It is important to keep in mind this is only a momentary drop in call quality that occurs simultaneously with the codec switch. Therefore, comparing the call quality at that point in time to the call quality that would normally be expected is required. The serialization delay is the only delay that is affected in this example. Therefore, performing the calculations again with a serialization delay shown in Table 9 for G.711 provides a serialization delay of 0.1712 ms. Substituting this serialization delay for the serialization increase in the above equations yields a new $D_{network}$:

$$Dnetwork = Propagation + Queuing + S_D \tag{5.7}$$

$$Dnetwork = 26.5\text{ ns} + 654.5\ \mu\text{s} + 0.1712\text{ ms} = 0.8257\text{ ms}$$

A new delay value d is calculated as:

$$d = 0.8257\text{ ms} + 2.25\text{ ms} + 80\text{ ms} = 83.076\text{ ms}$$

Which yields a new delay impairment value of:

$$I_d = 0.024 * d + 0.11 * (d - 177.3) * \mathbf{I}(d - 177.3) = 1.994$$

Consequently, the new R factor is:

$$R = 94.2 - 4.19 - 1.994 = 88.016$$

It is clear that voice quality drops momentarily during the codec renegotiation. The R factor for a codec not undergoing a renegotiation bears a factor of 88.016, while a codec undergoing a renegotiation endures an R factor of 88.013. Although this difference is extremely small and users would not audibly notice the difference, it is quite possible that networks having higher propagation and queuing delays will likely have a larger gap between R factor values. On the contrary, the proposed scheme works well in this small LAN environment, as exhibited by only a small difference in R factor, which is not audible from the user's perspective.

In the case of the public Internet, to provide acceptable call quality, the one-way delay cannot exceed 150 ms. Hence, a 300 ms round trip delay would be considered a worst case scenario. Internet conditions will vary at different times of the day and consequently, end-to-end delays will fluctuate accordingly. As mentioned in the research survey section, these Internet conditions can be modeled, and the codec renegotiation scheme can adjust accordingly. The proposed algorithm will work with minimal negative effect, regardless of Internet delays. The actual renegotiation of codecs does not come into picture until the three-way exchange is done. As such, the serialization delay increase associated with the algorithm shown in Equation 5.3 is the only factor leading to an increase in overall one-way delay. As long as the total delay is kept below 150 ms, voice quality will not be severely affected.

It can be deduced from Table 9 and from Equation 5.3 that there will only be an increase in serialization delay from the codec renegotiation in three instances, since $S_{Dincrease}$ results in zero or subzero values for all other situations, or the particular instance is not allowed. The proposed protocol design does not permit instances of codec migration directly from either of the two G.723.1 codecs to the G.711 codec, as shown in Figure 14. The first instance of an increase in serialization delay involves codec renegotiation from G.723.1 5.3 Kbps to G.723.1 6.3 Kbps. This renegotiation has an increase in serialization delay from 0.0592 ms to 0.0624 ms, or 3.2 μs. Then when it is used to calculate I_D for the change, the result is only 76.8e-6, as shown in Table

10. Since this value is subtractive in nature to the R factor equation, the result is an R factor depreciation of 76.8e-6.

CurrentCodec	NewCodec	Change in I_D (in 10^-3)	Impact on R factor
G.723.1 5.3	G.723.1 6.3	76.8e-3	negative
	G.729	0	no change
	G.711	2.688	not supported
G.723.1 6.3	G.723.1 5.3	-76.8e-3	positive
	G.729	-76.8e-3	positive
	G.711	2.611	negative
G.729	G.723.1 5.3	0	no change
	G.723.1 6.3	76.8e-3	negative
	G.711	2.688	negative
G.711	G.723.1 5.3	-2.688	not supported
	G.723.1 6.3	-2.611	positive
	G.729	-2.688	positive

Table 10: Change in I_D and Effect on R Factor.

The second instance resulting in an increase in serialization delay is migrating from the G.729 codec to the G.723.1 6.3 Kbps codec. It can be seen that the increase in serialization delay has a negative effect on the R factor value by 76.8e-6, which is identical to the first instance. The change in I_D directly corresponds to a decrease in the R factor if the change is positive, which in turn has a negative impact on voice quality. However, if the change in ID is negative, the R factor value is actually improved, leading to a positive impact on voice quality. The final instance of an increase in serialization delay is seen when migrating from the G.729 codec to the G.711 codec. This situation experiences the highest decrease in R factor value due to having the largest difference in serialization delays, which leads to a decrease in R factor of 2.688e-3. It is important to note, however, that the effect these codec changes have on R factor values, and thus corresponding MOS values, is negligible. The R factor decrease or increase in quality is only instantaneous with respect to the change, and then immediately moves to the ideal case. This instantaneous effect on call quality is inaudible to the user, as R factor values are only affected on the scale of milli or micro. Consequently, this scheme can be implemented without negatively affecting call quality during the switch, but retaining the highest possible call quality based on ever-changing network conditions.

CHAPTER 5

DISCUSSION

5.1 Implications

The proposed algorithm in this research has many implications. If available link bandwidth is in short supply, current calls could switch from a higher bandwidth codec mid-call to a lower bandwidth codec to reduce link usage. However, moving this may decrease voice quality due to lower MOS scores of the lower bandwidth codec. Alternatively, if link bandwidth is ample, current calls could switch from a lower bandwidth codec to a higher bandwidth codec. This would, in effect, increase the voice quality for each call that issues the codec renegotiation. If the link bandwidth was limited with the current calls, and all users were operating with high bandwidth codecs, those users could renegotiate to a lower codec in order to accommodate more calls on the link. This would require some means of appropriate bandwidth monitoring and an application that could initiate a codec renegotiation for calls based on bandwidth availability.

Tailoring a codec renegotiation scheme to a widely used standard such as H.323 has the potential to promote and ease wide implementation. The algorithm uses a modified H.245 capability set phase to accomplish the renegotiation procedure. In addition, the reuse of the playout buffer would be beneficial in situations where there are many VoIP calls being made on the gateway and buffer memory was limited. Assuming there were many calls currently active on the gateway, and it was desired for each of those calls to renegotiate to a new codec simultaneously, buffer reuse would be a proper choice. On the other hand, if available buffer memory was in ample supply based on the number of active calls, creating a new buffer to accept voice packets with the new codec would be acceptable. Assuming all calls were renegotiated simultaneously, the option of creating a new buffer would require twice the total amount of current buffer space in use.

5.2 Future Work

With a single link, terminals are essentially using per-destination load balancing (one link only). Packets using a single link will not arrive at the destination out of order. However, packets arriving using per source load balancing, or multiple paths, could be analyzed. Voice packets may arrive at the destination out of order and require sorting which could affect voice quality and increase the jitter buffer delay. Another aspect that could be explored is synchronizing a codec change for both VoIP directions. This research explored changing a codec uni-directionally. Bidirectional codec renegotiation can be investigated as it would require synchronization between both endpoints. As mentioned previously, the scheme proposed in this research was modified for use with the traditional H.323 call setup process. A similar scheme could be proposed to interoperate with the Session Initiation Protocol call setup process.

Furthermore, this research analyzed a local Ethernet network connection. Serialization delays for WAN links will have higher delays as opposed to Ethernet, Fast Ethernet, and Gigabit Ethernet. As such, WAN links would be more susceptible to voice quality degradation. In addition, propagation delays for cross-continental U.S. VoIP calls can be upwards of 30ms or more. These serialization and propagation delays can severely affect overall voice quality. For these situations, if the gateway can recognize a new codec will increase the payload size, a method of introducing a proportionally divided delay could prove beneficial. Since there will be a gap while waiting for the packet with the new codec, that delay could be divided amongst the packets currently in the playout buffer. Rather than continuing to play out packets from the jitter buffer at the current rate, the voice quality impact could be made less audible if there was a proportional increase in playout of each packet.

The proposed dynamic codec renegotiation scheme has immense potential for use in call centers. Call centers experience variable call volumes [38] based on many factors such as time of day, day of the week, or even seasonal variations, among others. A look into the benefits and disadvantages of implementing the proposed scheme in a call center could potentially yield improved voice quality and provide significant cost savings.

5.3 Conclusion

This research proposes an adaptive codec selection mechanism which changes the voice encoding scheme in the middle of an active call based on the network conditions. The proposed mechanism was proposed for H.323 based systems and caused little to no effect on voice quality. The mechanism ensures voice continuity while switching codecs by filling the playout buffers appropriately. The proposed idea appears theoretically feasible, as shown by simulations. While this research evaluated the effect of this mechanism on a simple, local network, it needs to be practically implemented and tested in a production-level environment.

REFERENCES

LIST OF REFERENCES

[1] “H.323 Standard,” ITU-T Recommendation H.323, http://www.itu.int/ITU-T/

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: Session Initiation Protocol,” IETF RFC 3261, June 2002, http://www.ietf.org/rfc/rfc3261.txt

[3] D. Hardman, “Noise and Voice Quality in VoIP Environments,” Agilent Technologies, April 23, 2003, pg 3.

[4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” IETF RFC 3550, July 2003, http://www.ietf.org/rfc/rfc3550.txt

[5] Scott Keagy, *Integrating Voice and Data Networks*, Cisco Press, Aug. 2001.

[6] “Managing and Maintaining VoIP and Enterprise Data Applications: Understanding Key Requirements,” AT&T, January 16, 2006.

[7] “Recommendation G.114 - One-way Transmission Time,” International Telecommunications Union ITU-T G.114, http://www.itu.int/itudoc/itu-t/aap/sg12aap/history/g.114/

[8] Cisco IOS Voice Command Reference, Release 12.3, Cisco Systems, 2003, http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122tcr/122tvr/vrgbook.pdf , p. VR-381

[9] N. Forrester, “Measuring Performance and Quality of Packet Voice Systems,” Motorola Whitepaper, January 2003.

[10] Information Sciences Institute, “Internet Protocol,” IETF RFC 791, September 1981, http://www.ietf.org/rfc/rfc0791.txt?number=791

[11] S. Casner, and V. Jacobson, “Compressing IP/UDP/RTP Headers for Low-Speed Serial Links,” IETF RFC 2508, February 1999, http://www.ietf.org/rfc/rfc2508.txt?number=2508

[12] L. Carvalho, E. Mota, R. Aguiar, A. Lima, J. Souza, and A. Barreto, “An E-Model Implementation for Speech Quality Evaluation in VoIP Systems,” 10th IEEE Symposium on Computers and Communications (ISCC 2005). Volume 00. p. 933-938.

[13] “P.800 Methods for Subjective Determination of Transmission Quality,” International Telecommunications Union ITU-T P.800. http://www.itu.int/rec/T-REC-P.800/en

[14] “The E-Model, a Computational Model for Use in Transmission Planning,” International Telecommunications Union ITU-T Recommendation G.107, March 2005.

[15] A. Takahashi, H. Yoshino, and N. Kitawaki, “Perceptual QoS Assessment Technologies for VoIP”, IEEE Communications Magazine, July 2004.

[16] S. Wang, Z. Mai, D. Xuan, and W. Zhao, "Design and Implementation of QoS-Provisioning System for Voice over IP," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 17, No. 3, March 2006.

[17] S. Giordano, M. Mancino, A. Martucci, and S. Niccolini, "VoIP Dynamic Resource Allocation in IP DiffServe Domain: H.323 vs. COPS Interworking," Terena Networking Conference, June 2002, Limerick, Ireland.

[18] P. Koutsakis, M. Paterakis, and S. Psychis, "Call Admission Control and Traffic Policing Mechanisms for the Wireless Transmission of Layered Videoconference Traffic from MPEG-4 and H.323 Video Coders," The 13th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications, Vol. 5, 15-18 Sept. 2002, pp. 2155-2159.

[19] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the Constancry of Internet Path Properties," ACM SIGCOMM Internet Measurement Workshop, November 1-2, 2001, San Francisco, CA.

[20] A. Ossipov, "Dynamic Voice Codec Determination Mechanism for VOIP," MS Thesis. 2006. http://libcat.wichita.edu/cgi-bin/Pwebrecon.cgi?Search%5FArg=Osipov&Search%5FCode=NAME%5F&CNT=30&PID=sD?[t%60OJtT?[l@?[nlsHx]OHoXOH&BROWSE=3&HC=1&SID=2

[21] S. Niccolini, et. al., "IP Telephony Cookbook," http://www.informatik.uni-bremen.de/~prelle/terena/cookbook/main/index.html

[22] B. Li, M. Hamdi, D. Jiang, X. Cao, and Y. Hou, "QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges," IEEE Communications Magazine, April 2000.

[23] "Call Signaling Protocols and Media Stream Packetization for Packet-Based Multimedia Communication Systems," International Telecommunication Union ITU-T Recommendation H.225.0, May 2006. http://www.itu.int/rec/T-REC-H.225.0/en

[24] Yuan. "H.323 and SIP", http://www.cs.fsu.edu/~xyuan/cis6930_ipqos/voip3.ppt#256,1,H.323

[25] "Assessing VoIP Call Quality Using the E-Model," IXIA, 2005. http://www.ixiacom.com/pdfs/library/white_papers/voip_quality.pdf

[26] "Understanding VoIP," Avaya, October 2005. http://www.lantanacom.com/images/Avaya_Understanding%20VoIP.pdf

[27] "Playout Delay Enhancements," Cisco Systems. http://www.cisco.com/en/US/products/sw/iosswrel/ps1834/products_feature_guide09186a008008033c.html#1001633

[28] "H.245 Series H: Audiovisual and Multimedia Systems - Control Protocol for Multimedia Communication," Telecommunication Standardization Sector of ITU. May 2006. http://www.itu.int/rec/T-REC-H.245/en

[29] "Critical Steps for a Successful VoIP Deployment," AT&T, 10/04/2005. http://www.business.att.com/nx_resource.jsp?repoid=Topic&repoitem=voip_telephony&rtype=Whitepaper&rvalue=critical_steps_for_a_successful_voip_deployment_attsolctr&segment=ent_biz

[30] D. Kuhn, T. Walsh, and S. Fries, "Security Considerations for Voice Over IP Systems," National Institute of Standards and Technology. Special Publication 800-58. January 2005. http://csrc.nist.gov/publications/nistpubs/800-58/SP800-58-final.pdf

[31] M. Miller, "Implementing the VoIP Network, A Technical Briefing Series on VoIP and Converged Networks," Network General. http://www.networkgeneral.com/Uploads/WhitePapers/20059263564933/wp_Implement_VoIP3-crpd.pdf

[32] "Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs," Telecommunications Union ITU-T Recommendation P.862. http://www.itu.int/rec/T-REC-P/en

[33] "Transmission Impairments Due to Speech Processing," International Telecommunications Union ITU-T Recommendation G.113. 2002. http://www.itu.int/rec/T-REC-G.113/en

[34] S. Tao, K. Xu, et. al. "Improving VoIP Quality Through Path Switching," INFOCOM 2005. http://repository.upenn.edu/cgi/viewcontent.cgi?article=1139&context=ese_papers

[35] R. Cole, and J. Rosenbluth, "Voice over IP Performance Monitoring," AT&T Laboratories, ACM SIGCOMM Computer Communications Review, vol. 31, no. 2, 2001, pp. 9–24, April 2001.

[36] L. Ding, and R. Goubran, "Speech Quality Prediction in VoIP Using the Extended E-Model," GLOBECOM 2003 - IEEE Global Telecommunications Conference, v. 7, 2003, p 3974-3978.

[37] R. Britt, "Voice Quality Recommendations for IP Telephony," Nortel Networks Standards Project PN-4689, August 15, 2000. http://ftp.tiaonline.org/tr-41/Tr4112inactive/Public/2000-08-Vancouver/TR4112_00-08-03_PN4689v5.doc

[38] S.G Henderson, S.G Steckley, and V. Mehrota, "Call Centers and Poisson's Equation," 31st Conference on the Mathematics of Operations Research, January 17-19, 2006, Lunteren, The Netherlands.

APPENDIX

APPENDIX

CODEC SELECTION CODE

```
// "WhyBoomTalk" Dynamic Codec Selection Program
// Jered (JD) Ast

#include "stdafx.h"
#include "math.h"
#define programname "WhyBoomTalk"

double stdDeviation(int numValues, double values[])
{
  int i;
  double stdDev = 0;
  double PreSD = 0;
  double total = 0;
  double mean;
  double SDbm;
  double SD;

  // Get the total of all values
  for(i=0;i<numValues;i++)
    total += values[i];

  // Calculate the mean
  mean = total/numValues;

  // SUM[(value - mean)^2]
  for(i=0;i<numValues;i++)
    PreSD += pow((values[i]-mean),2);

  SDbm = (PreSD/(numValues-1));

  // Standard Deviation
  SD = sqrt(SDbm);
  // printf("-----STDDEV-----\n\nStandard Deviation: %.3f\n\n", SD);
  return SD;
}
// It also computes the maximum assumed value, which lies within
// two standard deviations from the mean. Both Loss Rate and RTT
// parameters are calculated using this function.

double maxValue (int numValues, double values[], double mean)
{
  double value;
  value = mean + (2 * stdDeviation(numValues,values));
  return value;
}

// Computes the mean value for either Loss Rate or RTT
double meanValue(int numValues, double values[])
{
  int i;
  double total = 0;
```

```
  double mean;

  // Sum total of all values
  for( i = 0; i < numValues; i++ )
    total += values[i];

  // Find the mean of the values
  mean = total / numValues;

  return mean;
}

int main()
{
  int i, j, k;
  int RTT;
  int oneWayDelay;
  int numValues = 6;
  int currentCodec = 0;
  int newCodec;
  //g.711 = 0, which is default
  //g.729 = 1
  //g.723.1 6.3 = 2
  //g.723.1 5.3 = 3
  char codec[4][15] = {"G.711","G.729","G.723.1 6.3","G.723.1 5.3"};
  double valuesLoss[6];
  double valuesRTT[6];
  double stdDev, mean = 0;
  double lossRate = 0, lossRateAvg, rttAvg;

  // Define the information from the packet - {LossRate, T, LSR, DLSR}
  int packetInfo[16][4] =
{{7,1000,900,35},{6,1200,1100,35},{5,1400,1300,65},{7,1600,1500,80},{5,1800,1700,80},{3,2000,1900,80},{5,2
400,2300,80},{1,2800,2700,90},{2,3200,3000,43},{0,3600,3500,23},{0,3800,3700,17},{0,4500,4300,50},{0,4700,
4500,35},{0,5000,4900,12},{0,5500,5400,74}};

  // Loop through all packets, gathering a running average of the last 6
  // packets (packets 1-6, 2-7, ...)
  for( i = 0; i < (16-numValues); i++)
  {
    // Move the current 6 packets into an array for processing
    for( k = 0; k < numValues; k++ ) {
      valuesLoss[k] = (double) packetInfo[k + i][0];
      valuesRTT[k] = packetInfo[k+i][1] - packetInfo[k+i][2] - packetInfo[k+i][3];
    }
    // Averages
    lossRateAvg = meanValue(numValues,valuesLoss);
    rttAvg = meanValue(numValues,valuesRTT);

    // Operational values
    lossRate = maxValue (numValues, valuesLoss, lossRateAvg);
    RTT = (int) maxValue (numValues, valuesRTT, rttAvg);
    oneWayDelay = RTT / 2;

    printf("\nSample %d: Loss Rate %f  One-Way Delay %dm\n\nCurrent Codec: %s\n\n", k, lossRate,
oneWayDelay, codec[currentCodec]);
```

```
  // Display packet contents
  for( j = i; j < (i + numValues); j++)
  {
    printf("Loss: %d\t  T: %d\t  LSR: %d\t  DLSR: %d\t\n",
      packetInfo[j][0], packetInfo[j][1], packetInfo[j][2], packetInfo[j][3]);
  }
  printf("------------------------------------------------------------------\n\n\n");

// HERE WE DETERMINE WHAT TO CHANGE THE CODEC TO:
if ( currentCodec == 0 && lossRate > 3 )  //if currently G.711
  {
    if ( oneWayDelay < ( 150 - 35 ) )
      newCodec = 1;
    if ( oneWayDelay < ( 150 - 67.5 ) )
      newCodec = 2;
  }
if ( currentCodec == 0 && lossRate <= 3)  //if currently G.711
    newCodec = 0;
if ( currentCodec == 1 )  //if currently G.729
  {
    if ( lossRate >= 1 && oneWayDelay < ( 150 - 67.5 ) )
      newCodec = 2;
    if ( lossRate < 0.1 || oneWayDelay > 150 )
      newCodec = 0;
    if (lossRate <= 1 || oneWayDelay > ( 150 - 67.5 ) )
      newCodec = currentCodec;
  }
if ( currentCodec == 2 )
  {
    if ( lossRate > 1 )  //if currently G.723.1 6.3
      newCodec = 3;
    if ( lossRate <= 1 && oneWayDelay < (150 - 67.5 ) )
      newCodec = currentCodec;
    if ( lossRate <= 1 && oneWayDelay > (150 - 67.5 ) )
      newCodec = 1;
  }
if ( currentCodec == 3 )  //if currently G.723.1 5.3
  {
    if ( lossRate < 0.1 && oneWayDelay < ( 150 - 67.5 ) )
      newCodec = 2;
    if ( lossRate < 0.1 && oneWayDelay > ( 150 - 67.5 ) )
      newCodec = 1;
    if ( lossRate >= 0.1 )
      newCodec = currentCodec;
  }
// then send packet to initiate codec change request
if (newCodec != currentCodec)
  printf("********** Change Codec to: %s **********\n\n", codec[newCodec]);

currentCodec = newCodec;
}
  printf("#\n#\n#\n");
return 0; }
```