

An Improved VoIP using Adaptive Multirate Encoder (AMR)

Mohammed Tee A. B.
Department of Computer Science
Kwame Nkrumah University of
Science and Technology
Kumasi, Ghana

Michael Asante, PhD
Department of Computer Science
Kwame Nkrumah University of
Science and Technology
Kumasi, Ghana

Frimpong Twum, PhD
Department of Computer Science
Kwame Nkrumah University of
Science and Technology
Kumasi, Ghana

ABSTRACT

The ability to transmit voice traffic over conventional data networks has revolutionized the way we communicate. The question is how to provide voice quality comparable to PSTN networks. For this reason, researchers have looked to understand the entire communication components and path between the source and destination. One way is to introduce QoS mechanisms that prioritize voice or latency sensitive data transmissions but most of these techniques also introduce some amount of delay whereas voice traffic is delay sensitive and a second delay can cause degradation in quality. This study describes a means for transmitting voice traffic over conventional data networks by making use of a variable bit rate encoder while anticipating network congestions in real time in order to actively mitigate delays in voice transmissions thereby achieving improved QoS.

Keywords

VoIP, Quality of Service (QoS), Congestion Control, Link Adaptation, TCP Vegas, AMR Encoder

1. INTRODUCTION

In an effort to gain wider adoption, VoIP technologies have made efforts to provide voice quality comparable to PSTN networks. The challenge now is the need to route time sensitive data over a network design that is biased towards error free transmission at the expense of time of delivery.

VoIP traffic can tolerate some amount of error but it is latency sensitive. Time is of utmost importance in any VoIP solution. The acceptable end-to-end delay (latency) should not exceed 150ms in order for users to experience toll quality audio [1].

There are many QoS techniques in use today but they are all centered on the prioritization of VoIP network traffic as well as creating virtual routes or subnets for VoIP traffic. The limitation of these techniques is that they are all implemented on the same network devices that were designed to manage the data flow on TCP/IP networks and as such are not VoIP centered.

Some of these techniques require the addition of extra bits to the VoIP frames in order to identify, classify and prioritize them from other traffic types. This in turn also increases the size of the frames at the expense of available bandwidth.

Another area of concern is the constant size of packets mainly attributed to the constant bitrates used for encoding not to forget their inability to cover the entire VoIP process from source to destination in order to create a true end-to-end QoS method.

The problem now is to design and implement a VoIP solution that is capable of adapting itself to varying network conditions

while maintaining acceptable levels of performance by providing audible audio without causing significant pressure on the available bandwidth and also the processing rate of the participating nodes (server and client).

Of course VoIP traffic will always rely on the existence of a network but it is imperative to isolate or create a layer of abstraction for the implementation of VoIP QoS that does not necessarily depend on the existing core and edge network devices.

In doing so, end to end performance improvements can be achieved. This also means that problem points and upgrades to the implementation can be isolated and resolved without looking to network vendors.

The study takes advantage of the Adaptive Multirate Encoder (AMR) as well as an adaptation of TCP Vegas congestion avoidance technique in order to achieve end-to-end QoS and effectively improve the performance of VoIP.

2. LITERATURE REVIEW

Public Switched Telephone Networks have effectively provided the platform for communication but the advent of the internet has made it possible to stay connected with even more people more so than ever. It has also proven to be a cost effective yet efficient and reliable means of communication. While most people are making the switch to internet-based voice communication, collectively termed VoIP, the need to provide more reliable and stable platforms for this service has become ever so evident.

The challenge lies in the fact that the underlying infrastructure of the internet (TCP/IP) has always been to ensure accurate delivery of packets at the expense of time. This means that to be able to route time sensitive data (voice, video), leveraging on existing infrastructure requires additional techniques that will ensure their timely delivery. According to the ITU, the average end-to-end delay should not exceed 150ms. The worst case should be 200ms before users begin to experience delays in voice communications over the network [2]. On the other hand, QoS can be used to control jitter, latency and drop precedence [3].

The study therefore looked at some of the widely used QoS techniques as well as related works in the field.

2.1 Differentiated Services/DiffServ

It is an alternative approach to IntServ. It provides better scaling options and operates by assigning resources to classes of traffic [4].

DiffServ assigns extra fields to packets to identify the service that generated them. This is termed the Type of Service (TOS) field. The TOS field serves as a reference for the router when

forwarding packets or assigning CPU resources [4].

The forwarding classes can be either Expedited Forwarding (EF) [5] which is designed for low delay, jitter and loss services by ensuring suitable intervals for packets that is independent of the load offered or Assured Forwarding (AF) [5] that uses four (4) classes to define the Per Hop Behavior (PHB) of packets which are matched against three (3) levels (Low Drop Precedence, Medium Drop Precedence, High Drop Precedence) creating twelve (12) classes.

Performance improvement is achieved by assigning the specified packets low drop precedence. This ensures that these packets have a lower drop rate than other packets when congestion is registered in the network.

2.2 Integrated Services/IntServ

It is described as an IP-based QoS mechanism designed to provide unambiguous, end-to-end QoS for packets [4][6][7][8].

It operates by allowing hosts to request resources along the network medium on a per-flow basis. IntServ as a performance improvement technique makes use of packet classification, scheduling and admission control.

It assumes that each node in the entire communication process is QoS aware. This means that all the devices must be capable of ensuring QoS for it to be effective. For VoIP or other real time based applications with strict requirements for latency, jitter and packet loss, their requirements can be addressed by guaranteed service.

IntServ requires the RSVP (Resource Reservation Protocol) as stipulated by the IETF [9].

2.3 Multi-Protocol Label Switching

Multiprotocol Label Switching is a layer three (3) routing protocol whereby packets are assigned labels by an edge router called a Label Edge Router (LER). Using a Label Switch Path (LSP) determined by a Label Switch Router (LSR), packets are forwarded solely based on the contents of the label without requiring information on the data been routed. The major differentiator is that the switch path used by MPLS is independent of any layer 2 technology.

MPLS is the single most important development in TCP/IP [10]. The mechanisms used by MPLS allow IP networks to improve security and provide a multi-level QoS through defined virtual circuits [10].

2.4 Related Works

An analysis of VoIP signal processing for performance enhancement is described by [11]. They mention that the source rate depends on the state of the network and for that matter, there is a need for estimating such state since the IP service model does not offer any congestion notification and detection of temporary congestion.

The implementation described by [12] is focused on adaptive algorithms that have become part of most Digital Signal Processors (DSPs). They focus on LMS as an adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal. Though their study describes the role of adaptive filtering in VoIP, their study clearly mentions that work is in progress to incorporate influencing parameters for the performance improvement of the VoIP system.

The effects of packetization on the performance of VoIP is also analyzed by [13]. Their study uses packetization as a

means for rate adaptation based on a constant bitrate codec. The study shows that optimal packetization helps improve VoIP performance.

The performance of audio signals over CELP based Adaptive Multi Rate (AMR) coder with emphasis on its wideband bitrates is evaluated by [14]. There is a limitation on the use of wideband codecs because of the 4000Hz limitation on PSTN networks. This means that using WB codecs is an over stretch unless if the idea is to perform strictly VoIP to VoIP calls without any PSTN involvement.

The Brady Model is also of interest to many studies [15][16][17][18] focus on call admission control schemes based on channel load estimation and Time Between Idle Times (TBIT) as well as efficient link utilization based on on-off patterns of VoIP from the Brady Model.

Perhaps the most influential work in the field of adaptive VoIP techniques that sets the precedence for this study is the work done by [19]. The authors present an adaptive architecture for VoIP transport over wired/wireless networks. Bitrates are determined by a control mechanism based on estimation of channel congestion state. The authors present the performance of proposed framework with results indicating better voice transmission. Their research presents results associated with G.711 and G.726 ADPCM speech coding.

3. SYSTEM OVERVIEW

The study presents a framework for an improved VoIP that estimates channel congestions and based on that, determines what bitrates to use for encoding. The congestion control technique is adapted from TCP Vegas and the encoder used is the Adaptive Multi Rate (AMR) encoder which is capable of encoding at multiple bitrates on a frame by frame basis. The study aims to achieve complete end-to-end performance improvement by adapting bitrates of the encoder based on the estimated congestion on the network channel between the server and the client which ultimately influences VoIP Quality of Service parameters, latency/jitter as well as bandwidth utilization on the network channel by using Round Trip Times (RTT), the maximum allowable bandwidth and the previously used bitrate.

The framework effectively decides the most appropriate bitrate to be used for the next encoding sequence with the ultimate goal of improving bandwidth utilization as well as ensuring the effects of latency/jitter on the transmitted frames are kept at a minimum.

The transport protocol is UDP which means that there are no retransmissions of lost packets. The study adopts this because VoIP can tolerate some percentage of packet losses and as such emphasis is placed on fast delivery of packets as well as the size of encoded frames more so than on the delivery of all VoIP packets while making use of a proactive approach to identifying and estimating channel conditions which helps to mitigate packet losses even before they occur.

Figure 1 presents an architecture for the framework from audio source, which is the input to the bitrate selector that is fed through the AMR encoder for encoding and packetization on the source device and at the receiving end on the destination device. Between the source and destination as depicted in Figure 1, Round Trip Times (RTT) are measured and communicated in-between every packet transmission. RTT values obtained are fed into the congestion and bitrate selector algorithm that determines what bitrate should be used for the next encoding sequence in order to achieve improved

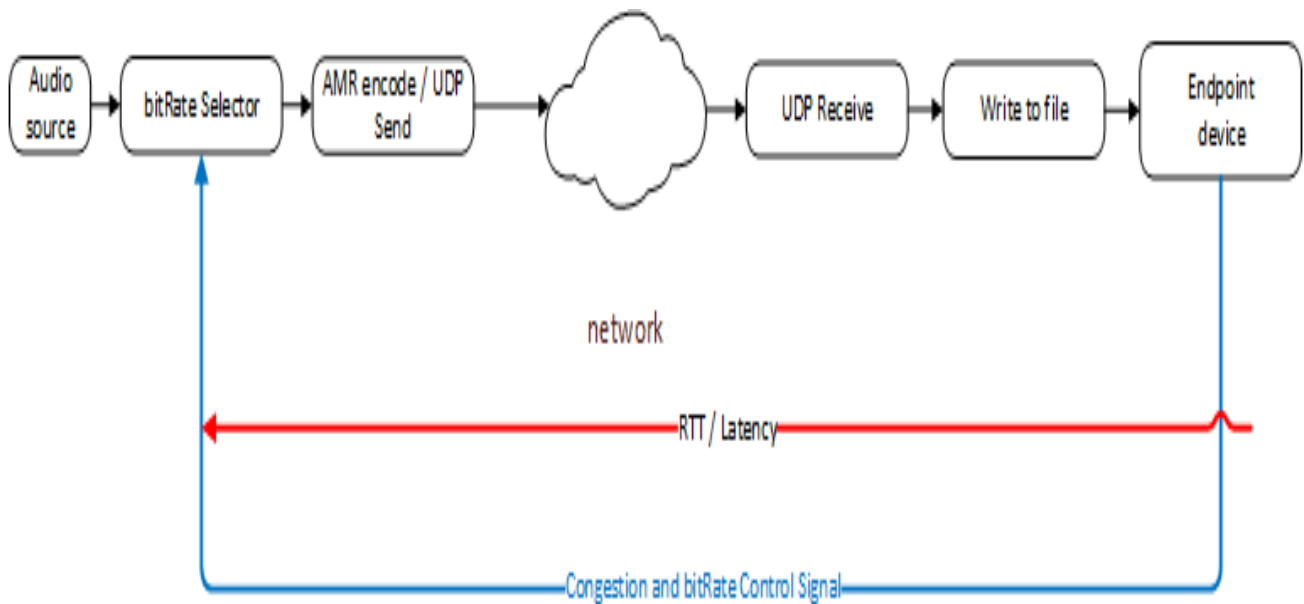


Fig 1: Overview of the Interacting Components for the Framework (Author)

Quality of Service (QoS).

The framework for Improved VoIP presented by the study uses the Adaptive Multi Rate (AMR) encoder and a congestion control and bitrate selector which serves the purpose of adaptively selecting bitrates for encoding as well as the purpose of link adaptation to improve performance of VoIP by determining the most appropriate bitrate to use for encoding given the current Round Trip Time (current RTT), the previous bit rate and the available bandwidth as factors to determine the current link utilization and ultimately anticipated congestion of the channel.

The study takes advantage of the ability of the Adaptive Multi Rate (AMR) encoder to encode each frame with a different bitrate by adaptively determining the most appropriate bitrate to use based on the currently recorded congestion between the source and destination. This is achieved by implementing a ping program which provides the congestion and bitrate selector with reports in the network on congestion by returning Round Trip Times that are expressed in milliseconds (ms). These reports together with the available bandwidth and bitrate used for the previous encoding sequence is used to intelligently and adaptively determine the next appropriate bitrate such that acceptable latency/jitter is achieved as well as efficient usage of the allocated bandwidth.

The following section presents the three (3) interacting components used by the study to complete the proposed framework. The study uses the ANSI floating point c-source code specification of the Adaptive Multi Rate (AMR) encoder which uses variable bitrates as opposed to the constant bit rate used by codecs in VoIP systems.

For link adaptation and channel congestion information, the framework implements an adaptation of the TCP Vegas congestion control algorithm which is presented in section 4.2. The Round-Trip Times (RTT) information between the source (server) and the client (destination) is retrieved prior to the encoding process by passing an arbitrary number of bytes between the server and client in the form of ICMP packets. This is used as part of the control method for bit rate selection.

Finally, UDP is used as the transport medium for the encoded byte stream from the server to the client. Because the idea behind the congestion control method and bitrate selector is to anticipate congestions before they occur thereby mitigating packet loss before it occurs. UDP with its lack of support for error checking and retransmissions is ideal because there will be less time spent transporting the byte stream as opposed to TCP where retransmissions must occur.

4. SYSTEM COMPONENTS

4.1 Adaptive MultiRate Encoder

The Adaptive MultiRate codec was originally designed for circuit switched mobile radio systems but due to their flexibility and robustness, they can be used in real-time speech applications over packet switched networks [20].

The AMR codec was originally developed and standardized by the European Telecommunications Standards Institute (ETSI) for GSM cellular systems. It has since then become a mandatory codec for third generation (3G) cellular systems [21].

The AMR codec uses multiple bitrates and has support for eight (8) narrow band speech encoding modes ranging from 4.75 kbps to 12.2 kbps. It uses 8000 Hz sampling frequency and the speech coding is done on 20ms speech frames representing 160 samples.

The study leverages on the multiple bitrates of the Adaptive Multi Rate (AMR) encoder to implement the QoS method based on conditions that are likely to affect the QoS and ultimately the performance of the VoIP solution.

Table 1. Source Codec Bit-Rates for the Adaptive Multi Rate Codec[22]

Index	Bit-rate (kbits/s)	AMR Mode Operation	Total Speech Bits
0	4.75	AMR_4.75	95
1	5.15	AMR_5.15	103
2	5.90	AMR_5.90	118
3	6.70	AMR_6.70	134
4	7.40	AMR_7.40	148
5	7.95	AMR_7.95	159
6	10.2	AMR_10.20	204
7	12.2	AMR_12.20	244

Each speech frame of the encoder is made of 160 samples of the input audio using a sampling frequency of 8000 Hz on 20ms speech frames.

The framework makes use of the exact bitrates and its associated operating modes as presented in Table 1 for the entire encoding sequence and analysis of the framework. The congestion and bitrate selector described in Section 4.2 adaptively chooses from the bitrates presented in Table 1, the most appropriate to use for the next encoding sequence by passing the modes to the encoder based on the current network congestions as perceived by the adaptive congestion and bitrate control implemented by the study.

4.2 Adaptive Congestion & Bitrate Control

At the heart of the framework is the adaptive congestion and bit rate control adapted from the TCP Vegas congestion control technique. This technique was introduced as a real end-to-end congestion avoidance technique in TCP networks [23]. The fundamental concepts of the congestion control method has been adapted by this study to provide the proposed VoIP solution with a link adaption technique that enables the framework to proactively mitigate packet losses by listening to network conditions as a result of higher or lower RTT. This effectively triggers a higher or lower bitrate based on the various modes of the Adaptive Multi Rate (AMR) encoder in Table 1 and forms the heart of the framework for the VoIP solution.

Unlike Tahoe, Reno, New Reno and Sack, TCP Vegas introduces a mechanism of congestion detection before packet loss. The basic idea is that packet loss will closely follow congestion on a network. By identifying congestion early, steps to prevent packet loss and further congestion of the network can be implemented [23]. This is the driving principle behind the proposed framework whereby anticipated congestions triggers the use of a lower bitrate and vice versa.

TCP Vegas congestion control method is ideal for the study because it is based on mechanisms that adapts to network congestions before they occur.

Researchers at Caltech are adapting TCP Vegas as the basis for the new Fast TCP. Fast TCP is considered as some sort of high speed version of Vegas.

TCP Vegas achieves between 37 % and 71 % better throughput on the Internet as compared to other implementations such as Reno which is distributed with BSD

Unix [23].

Though TCP Vegas is designed for TCP networks, the study adapts the congestion avoidance mechanism of TCP Vegas to serve as a link adaptation technique and bitrate selector for when congestion is anticipated in order to adjust the bitrate of the Adaptive Multi Rate (AMR) encoder. Though the choice of congestion control for the study is based on TCP, the preferred transport protocol for the study is UDP.

The study introduces thresholds that are defined to determine the link utilization and congestion status on the channel as well as to determine what bitrate to use. Using Table 1 as a reference for the bitrate indexes used:

1. When RTT[i] is greater than or equal to 70ms
 - a. Set bitrate to Index 0. R[0]
2. When RTT[i] less than 10ms
 - a. Set bitrate to Index 7. R[7]
3. When RTT[i] is between 30ms and 40ms
 - a. Maintain the previously used bitrate. R[i]
4. Compute Link Utilization (LU)
 - a. If LU is greater than 100%, use the next lower bitrate from the previous bitrate. R[i] = R[i-1]
 - b. If LU is less than 20%, use the next higher bitrate from the previous bitrate. R[i] = R[i+1]
 - c. Else maintain the current bitrate. R[i]

The congestion and link adaptation technique are implemented programmatically in C programming language with its corresponding flowchart presented in Figure 2 and pseudo code in Section 5. The congestion and bitrate selector waits for information on channel congestion in the form of Round Trip Times (RTTs) and stored in the RTT[i] variable. The returned RTT[i] together with the information on the previously used bitrates as well as the allocated bandwidth is used to adaptively choose the next Adaptive Multi Rate (AMR) mode for the next frame.

The study uses the bitrate of 7.40 kbps or at mode AMR_7.40 for encoding of the initial frame. The idea is to use a bit rate that is specified to provide toll quality audio but should not be too high in case the network conditions are bad at the first instance of encoding. In doing so, the framework can work itself upwards towards higher bitrates when the network conditions are favorable or lower when the network conditions are bad.

Throughput of the network is based on expected and actual data transmitted using RTT times as a computing factor from the previous and current transmissions [23].

For the study, throughput is computed based on the Link Utilization Rate as a factor of the RTT of the current and previously returned RTT and bitrates.

The study expresses Link Utilization (LU) as the number of bits on the link at time (x) divided by the total channel capacity which in this case is 64kbps.

This multiplied by 100 provides the percentage Link Utilization.

$$LU = \frac{\text{bytesFlowOnLink}}{\text{Channel Capacity}} * 100 \text{ -----[expression 5]}$$

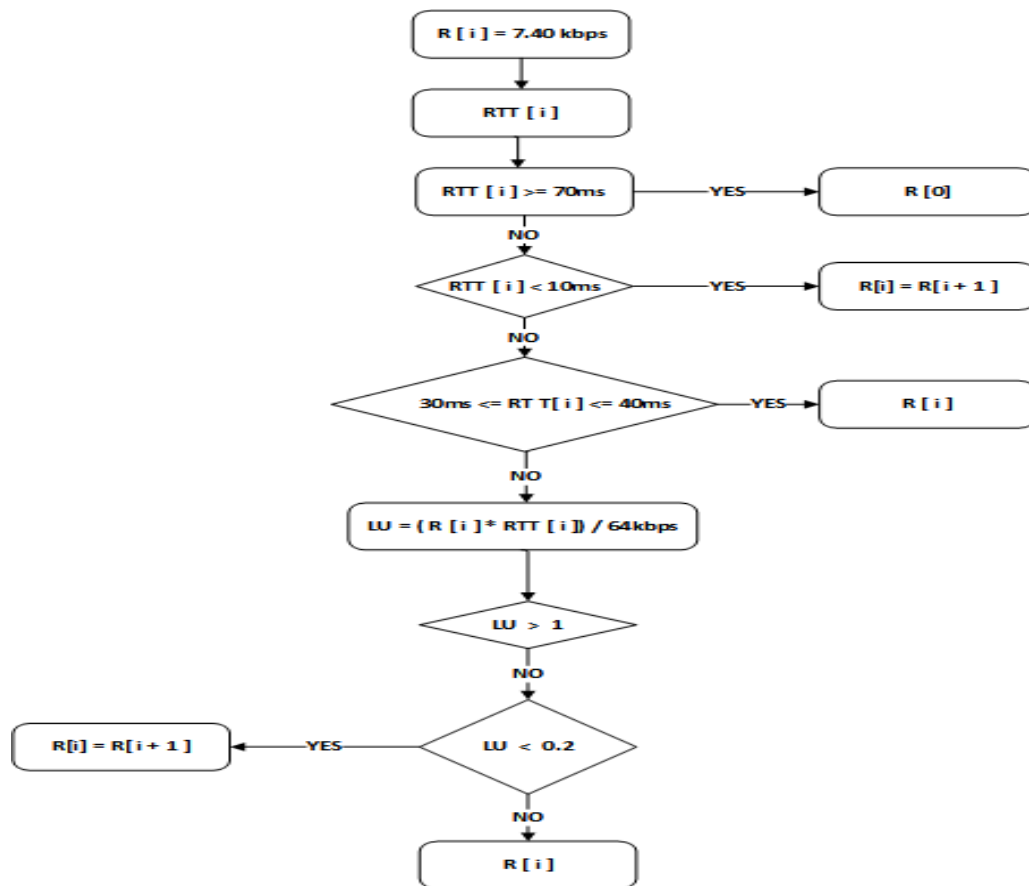


Fig 2: Flow Diagram for Congestion and Bitrate Control Selection (Author)

And flow on the link is expressed as:

$$\text{bytesFlowOnLink} = \text{currentRTT} * \text{currentBitRate} \text{ ----} \\ \text{--[expression 6]}$$

Where:

currentRTT: is the round trip time between the server and client during the transfer of the frame.

currentBitRate: is any of the bitrates of the Adaptive Multi Rate (AMR) encoder as used for the encoding of the current frame.

By multiplying the currentRTT by the currentBitRate, we can estimate the number of bits on the link.

Therefore, the link utilization (LU) using a maximum bandwidth of 64kbps can be expressed as:

$$LU = \frac{(\text{currentRTT} * \text{currentBitRate})}{64} \text{ -----[expression 7]}$$

When the link utilization in bytes exceeds, the alpha threshold, then the next lower bitrate should be used and vice versa. The alpha and beta thresholds roughly specify the upper and lower units of the utilization of the channel or available bandwidth [23].

The study ensures that while channel utilization is within the limits of alpha and beta, the encoding bitrate is maintained at 7.40 kbps. This is the bit rate at which toll quality audio is achieved. The study sets the alpha and beta thresholds to approximately 20% and 100% of the link utilization of the channel.

The Round-Trip Time (RTT) or Network Latency refers to the time it takes for a data packet to traverse a network from

source to destination. This in effect consists the propagation times between the points involved in the communication process.

The formulae for estimating RTT in TCP networks is [24].

$$RTT = (\alpha * Old_RTT) + ((1 - \alpha) * New_Round_Trip_Sample) \text{-----[expression 8]}$$

Where:

α : is a constant weighting factor ($0 \leq \alpha < 1$).

RTT: is the returned round trip time

This constant weighting factor when close to one (1) makes it immune to sudden changes whereas choosing a value close to zero (0) makes it respond to changes very quickly [2].

This algorithm for computing Round Trip Times (RTTs) was later improved by introducing standard [2].

In recording the Round Trip Time (RTT) between the server and client for the framework, the study implements a ping program that makes use of the network libraries of the linux operating system. Each Round Trip Time (RTT) is reported to the congestion and bitrate selector and used to determine the next appropriate bitrate to use for encoding the next frame and in so doing preventing congestion on the network thereby providing the VoIP framework with its own Quality of Service mechanism.

4.3 Transport leg (UDP)

UDP is a connectionless protocol. It does not have any inbuilt mechanisms for retransmissions and error recovery but is ideal for real time applications that are capable of tolerating

some amount of loss with little tolerance for delay. Transmissions that involve large amounts of data also take advantage of UDP especially when bandwidth is guaranteed.

The study chooses UDP as its preferred transport protocol because it is the first-choice protocol for most real time applications. Also, the added overhead of retransmissions and acknowledgements from TCP transmissions is also not required by the proposed VoIP implementation.

Since the ultimate goal of the framework is to mitigate losses before they occur, it is only appropriate to use UDP since it does not provide any error corrections and retransmissions.

During the transmission session between the server and the client, Round Trip Times (RTTs) are computed and together with the current and previous bitrates, they are used to determine the next appropriate bitrate in order to ensure adaptive and efficient use of the bandwidth while ensuring improved performance. In doing so, the decision to use higher or lower bitrates is entirely dependent on the VoIP solution itself.

5. PSEUDOCODE FOR ADAPTIVE CONGESTION AND BITRATES SELECTION

1. bitRateOptions = {4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.2, 12.2};
2. currentBitRate = 0.0;
3. bitRateIndex = 4;
4. availableBandWidth = 64.0;

//This is because 7.40 is the lowest bit rate for toll quality audio

5. currentBitRate = bitRateOptions[bitRateIndex];

6. While 1

//encode (audio, currentBitRate); // Encoding of the audio file takes place here

- a. currentBitRate = bitRateOptions[getNewBitRateIndex(bitRateIndex, computeRoundTripTime());]
- b. print (currentBitRate);
7. getNewBitRateIndex (oldIndex, computedRoundTripTime)
 - a. newIndex = 0;
 - i. **if** computedRoundTripTime >= 70.0
 1. newIndex = 0;
 - ii. **else if** computedRoundTripTime >= 30.0 && computedRoundTripTime <= 40
 1. newIndex = oldIndex;

- iii. **else if** computedRoundTripTime < 10.0
 1. newIndex = 7;
- iv. **else**
 1. newIndex = computeLinkUtilizationForAppropriateIndex(oldIndex, computedRoundTripTime);
 - b. **return** newIndex;
 8. computeLinkUtilizationForAppropriateIndex (oldIndex, computedRoundTripTime)
 - a. newIndex = 0;
 - b. linkUtilization = (bitRateOptions[oldIndex] * computedRoundTripTime) / availableBandWidth;
 - i. **if** linkUtilization > 1
 1. newIndex = oldIndex - 1;
 - ii. **else if** linkUtilization < 0.2
 1. newIndex = oldIndex + 1;
 - iii. **else**
 1. newIndex = oldIndex;
 - c. **return** newIndex;

6. RESULTS AND ANALYSIS

The proposed framework is implemented in C on a Linux platform. Eclipse is used for compiling and running the C modules that makes up the framework and network analysis is done using Wireshark. The input file is an 8000Khz 16 bits audio wav file and the output is the encoded file using one of the eight (8) bitrates of the AMR encoder which is then transmitted using UDP from source computer to the destination computer.

6.1 Bitrate Distribution

The output when the framework is run is presented in Figure 3. It shows the number of frames encoded and its corresponding number of bytes together with the bit rates distribution as used for the entire encoding sequence for the multiple channel implementation when the framework is used. The higher bit rates of the Adaptive Multi Rate (AMR) encoder denoted by MR122/12.2 kbps is used more often in the encoding sequence than the lowest bitrate denoted by MR475/4.75 kbps.

Out of the 1962 frame encoded during the simulation, MR122 is used for encoding 1810 of the frames. The lowest bitrate of MR475 is only used for encoding 49 frames only. Out of the remaining six (6) bitrates of the Adaptive Multi Rate (AMR) encoder, MR670 is used for encoding 93 frames out of 1962 frames and MR740 is used for encoding 10 frames. The remaining bitrate modes of MR515, MR 590, MR795 and MR102 are never used during the encoding sequence as per the network congestion conditions recorded during the simulation period.



Fig 3: Frequency of bitrates for the framework encoding sequence (Author)

The VoIP solution is able to adaptively choose between the different bitrates of the AMR encoder. Using higher bitrates for when congestion is low and lower bitrates for when congestion is anticipated.

highest bitrates is used more often for the encoding sequence rather than maintaining a lower constant bitrate with the view to not congest the network.

Better voice quality is achieved based on the fact that the

6.2 Bandwidth Utilization

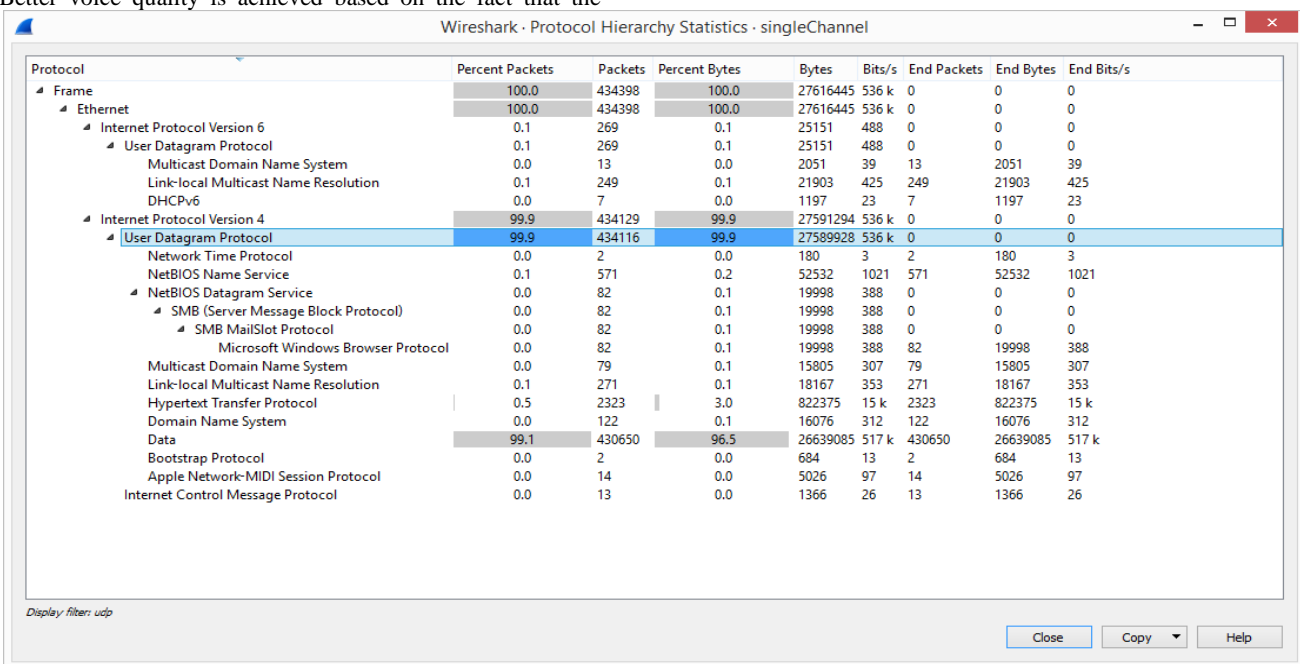


Fig 4: Single channel transmission bandwidth utilization (Author)

The study compares the bandwidth utilization for two states of the proposed VoIP solution. Firstly, the entire encoding sequence is done with a constant bitrate. The results is presented in Figure 4 and subsequently done with the framework implementation that adapts the bitrates to the anticipated network congestion effectively creating a multiple bitrate VoIP solution. The bandwidth utilization for when the Multiple bitrate is used for the encoding sequence i.e. via the framework is presented in Figure 5.

The encoded frames are captured in real time using Wireshark and filtered based on the I.P.'s of the source and destination. By analyzing the Protocol Hierarchy Statistics of the captured data during the transmission flow for both the single and multi-channel implementations, the study is able to present transmission analysis of the frames during the encoding and transport phase.

The multi-channel solution presented in Figure 5 makes use of 19kbps bandwidth during the encoding sequence. This is

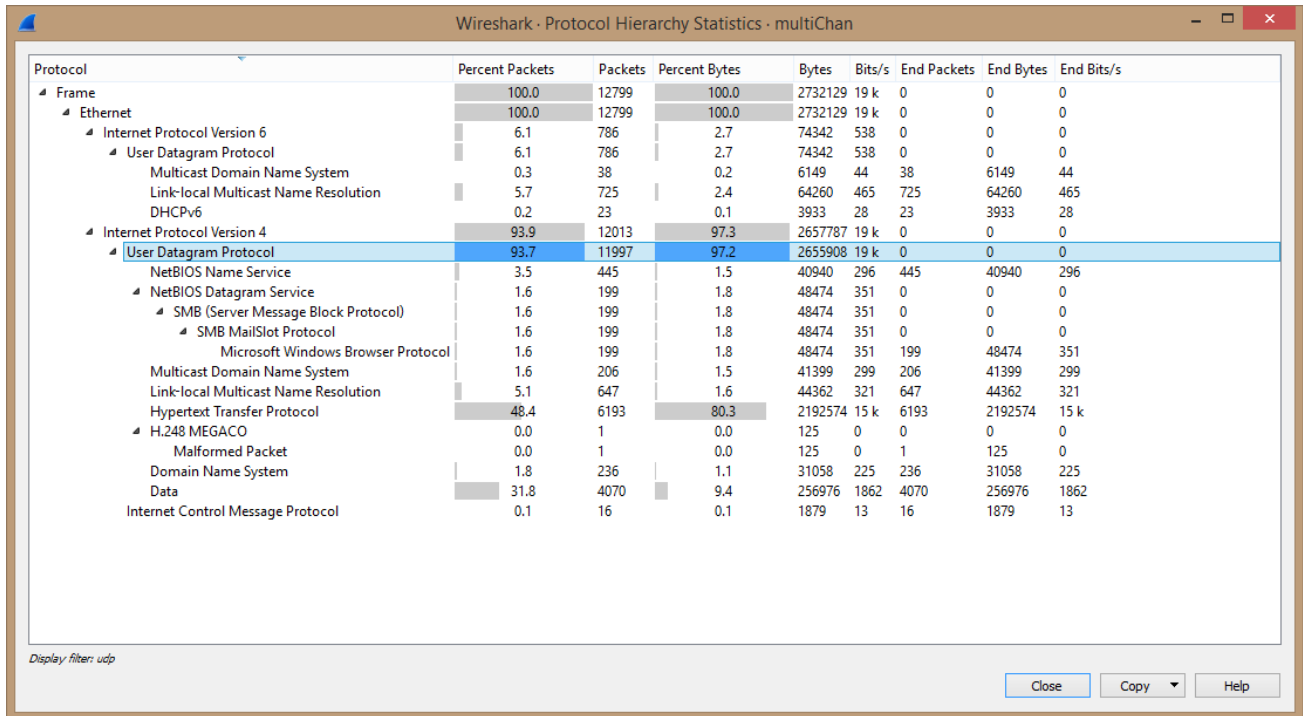


Fig 5: Multi channel transmission bandwidth utilization (Author)

relatively an efficient utilization of the channel as opposed to the single channel implementation presented in Figure 4 that makes use of 536kbps. This is the case even though the multichannel implementation uses a higher bit rate for encoding most of the frames as described in Section 6.1.

The reason for the efficient utilization of the bandwidth can be attributed to the nature of the framework to determine when to use lower bitrates in order not to congest the network. In doing so, the VoIP solution is almost always adaptive to the available network bandwidth.

It also means the multichannel implementation using the congestion and bit rate selector performs better when it comes to bandwidth managements than the single bitrate implementation that uses 7.40 kbps (MR740). This is the bitrate at which the AMR encoder begins to achieve toll quality audio.

6.3 Latency/Jitter

Using the IO Graphs feature of Wireshark under the Statistics option, the study is able to effectively analyze the latency/jitter as it occurred between the server and client by generating a graph based on the relationship between the bytes transmitted and time intervals recorded during the encoding sequence.

The analysis is based on two different states.

- i. Single channel capture when the bit rate is set to 7.40 kbps and presented in Figure 6.
- ii. Multi-channel capture using the framework's congestion and bit rate selector and presented in Figure 7.

Figure 6 and 7 show the frame inter-arrival times for the framework implementation during the encoding sequencing for a single bitrate implementation and subsequently when the framework is used in a multiple bitrate implementation.

The Y-Axis show the bytes of data as against the X-Axis that displays the time intervals for the frame transmission on the network.

When an audio file comprising 1962 frames totaling 25512 bytes is transmitted using the framework, the study records an even flow of traffic as per the latency/jitter graph in Figure 7 though there are varying bitrates throughout the entire encoding sequence as is the case for the framework implementation for a multiple bitrate VoIP. This is attributed to the fact that the framework effectively identifies periods of low and high latencies signifying anticipated congestion states on the network in order to adaptively determine the most appropriate bitrate to use.

The latency/jitter graph for a single channel implementation in Figure 6 displays a strong contrast in results as compared to

the framework implementation presented in Figure 7. The spikes are more random and not uniform. It is interesting to note that the single channel implementation which is the case of current VoIP solutions has no internal mechanisms for congestion detection and avoidance. The erratic nature of the

time intervals is evident from the graphical representation of the time distribution per the transmitted bytes.

The spikes in the graphs represent the latency and jitter as recorded during the transfer.

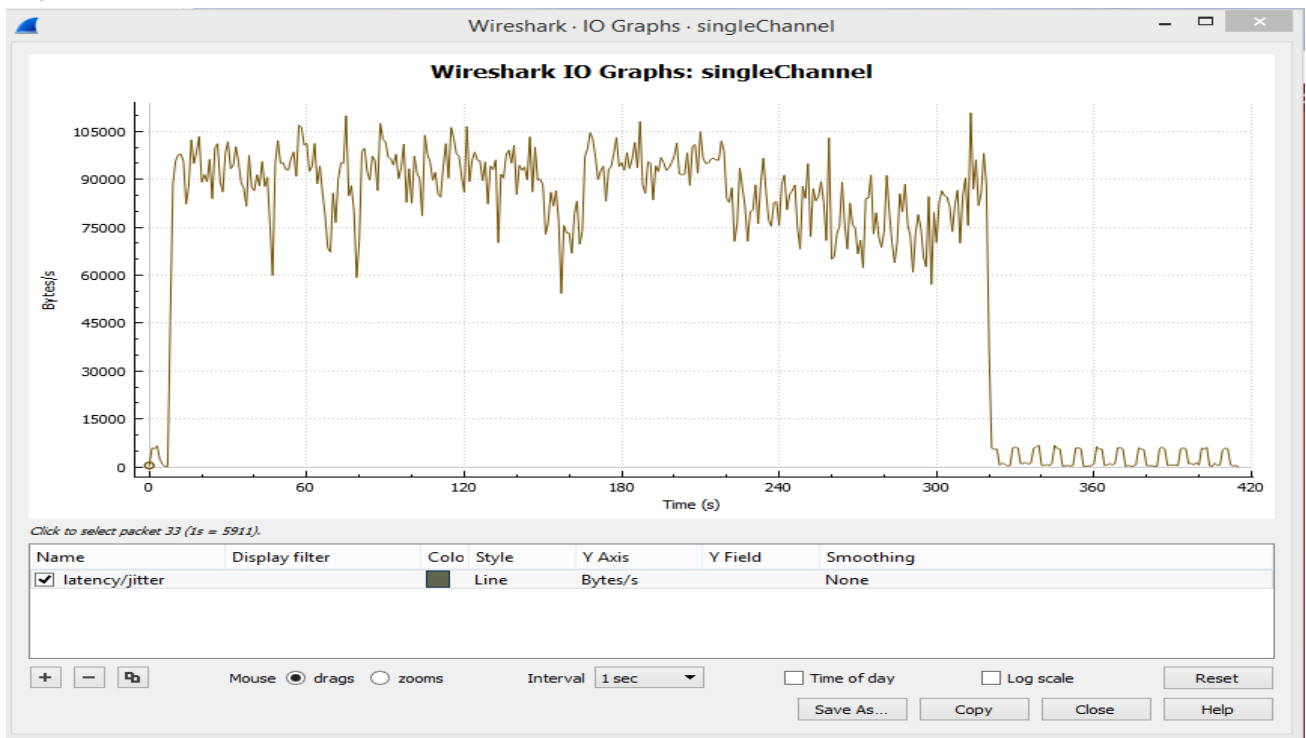


Fig 6: Single channel graph analysis of latency/jitter (Author)

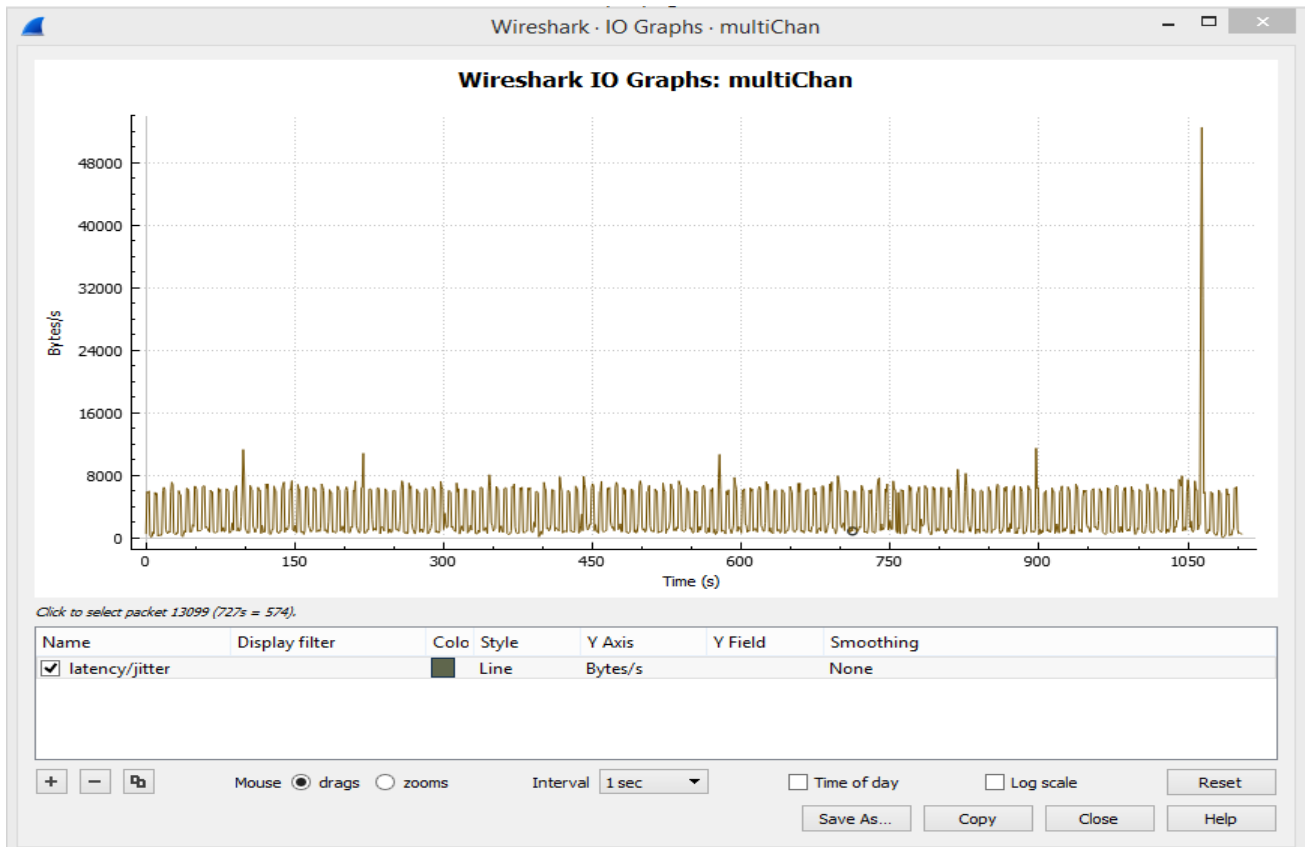


Fig 7: Mutli channel graph analysis of latency/jitter (Author)

It is clear that there is coordination and uniformity when the multi-channel solution is operating based on the presented framework though there are different levels of complexity and delay which can be attributed to source and destination devices such as Central Processing Unit (CPU) capabilities.

7. CONCLUSION

The study presented a framework for using multiple bitrates to ensure end-to-end Quality of Service in VoIP. This creates a VoIP solution that has its own internal mechanism for ensuring QoS without relying on layer 2 and 3 QoS techniques. By making use of the Adaptive Multirate Encoder and a Congestion and Bitrate Selector, the study is able to improve performance in terms latency/jitter recorded between the source and destination as well as maximize the use of bandwidth as opposed to when a single bitrate is used.

8. FUTURE SCOPE

The framework though efficient in tackling QoS also presents some delay. The researcher will like to analyze and explore the effects of this delay on the quality of voice. The researcher will also explore the effects of encryption technologies on securing VoIP traffic. This is necessary in securing IVR and Self Service based VoIP systems.

9. REFERENCES

- [1] Telecommunication Standardization Sector of ITU Rec. G.114. (2003). Series G: Transmission Systems and Media, Digital Systems and Networks.
- [2] Jacobson V. & Karels M.J. (1988). Congestion Avoidance Control.
- [3] Gonia K. (2004). Latency and QoS for Voice over IP. SANS Institute.
- [4] Blake S., Black D., Carlson M., Davies E., Wang Z., & Weiss W. (1998). An Architecture for Differentiated Services: RFC 2475.
- [5] Heinanen J., Baker F., Weiss W., & Wroclawski J. (1999). Assured Forwarding PHB Group. RFC 2597.
- [6] Ford P., Yavatkar R., Baker F., Zhang L., Speer M., Braden R., Davie B., Wroclawski J., & Felstaine E. (2000). A Framework for Integrated Services Operation over Diffserv Networks RFC 2998.
- [7] Wroclawski J. (1997). The Use of RSVP with IETF Integrated Services. RFC 2210
- [8] Shenker S., Partridge C., & Guerin R. (1997). Specification of Guaranteed Quality of Service. RFC 2212.
- [9] Braden R., Ed., Zhang L., Berson S., Herzog S., & Jamin S. (1997). Resource ReSerVation Protocol (RSVP) - RFC 2205.
- [10] Finneran M. F. (2006). Designing MPLS Networks for VoIP.
- [11] Singh D. H., Mian M., & Singh J. (2010). Analysis of VoIP Signal Processing for Performance Enhancement.
- [12] Horney C. (2000). Quality of Service and Multi-Protocol Label Switching
- [13] Ngamwongwattana B. (2008). Effects of Packetization on VoIP Performance. National Electronics and Computer Technology Center (NECTEC).
- [14] Gajjar P., Bhatt N., & Kosta Y. (2012). Overall Performance Evaluation of Adaptive Multi Rate 06.90 Speech Codec Based on Code Exited Linear Prediction Algorithm using MATLAB. International Journal of Speech Technology 15 (2), 119-129
- [15] Ganiga R., Muniyal B., & Pradeep. (2012). Characteristic Analysis of VoIP Traffic for Wireless Networks in Comparison with CBR using QualNet Network Simulator. International Journal of Computer Applications, Volume 50, No.11, PP 25-31.
- [16] Yasukawa K., Forte A. G., & Schulzrinne H. (2007). Distributed Delay Estimation and Call Admission Control in IEEE 802.11 WLANs. IEEE International Conference on Network Protocols, PP 334-335.
- [17] Dini P., Baldo N., & Nin-Guerrero J. (2010). Distributed Call Admission Control for VoIP over 802.11 LANs based on Channel Load Estimation. IEEE International Conference on Communications journal, PP 1-6.
- [18] Ji-Young J., Dong-Yoon S., & Jung-Ryun L. (2013). VoIP Call Admission Control Scheme Considering Voip on-off Patterns. International Conference on Information Networking (ICOIN), pp. 371 – 374.
- [19] Trad A., Ni Q., & Afifi H. (2004). Adaptive VoIP Transmission over Heterogeneous Wired/Wireless. International Workshop on Multimedia Interactive Protocols and Systems.
- [20] J. Sjorberg, M. Westerlund, A. Lakaniemi & Q. Xie. (2007). RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs.
- [21] 3GPP TS 26.090, "Adaptive Multi-Rate (AMR) speech transcoding", version 4.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).
- [22] 3rd Generation Partnership Project (3GPP) TS 26.071 v11.0.0. (2012). Mandatory Speech Codec Speech Processing Functions.
- [23] Brakmo L. & Peterson L. (1995). TCP Vegas: End to End Congestion Avoidance on a Global Internet. IEEE Journal on Selected Areas in Communication, Vol 13, no. 8, pg. 1465-1480.
- [24] Comer & Douglas. (2000). Internetworking with TCP/IP. Page 226. Upper Saddle River, N.J.: Prentice Hall.