Modeling of Packet Loss and Delay Using Multi-Path Diversity

Presented By

M Asad Khan
136-CS/MS/03

Supervised By

Dr. S. Tauseef-ur-Rehman
Chairman, Department of Telecommunication
Federal Urdu University of Arts, Science & Technology
Islamabad, Pakistan

Department of Computer Science
Faculty of Applied Sciences
International Islamic University Islamabad
(2007)

To1484E2007CSMS

DATA ENTERED

@

16/4/2012

MS

MS

005.369.

KHM

To4192

1- Wireless communication systems.

2. Computer software.

To4192

To1484

multiple cor

defin C2

2 ? ? ?

بسم الله الرحمن الرحيم

International Islamic University, Islamabad

Department of Computer Science

06, 04, 2007

FINAL APPROVAL

It is certified that we have read the project report submitted by ***Mr. Muhammad Asad Khan*** and it is our judgment that this project is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the Degree of Master of Sciences in Computer Science.

COMMITTEE

External Examiner **Dr. Abdus Sattar**

Ex DG PCB

Internal Examiner **Prof Dr. Sikander Hayat Khiyal**

Head,
Department of Computer Science
International Islamic University
Islamabad.

Supervisor **Dr. S.Tauseef-ur-Rehman,**

Chairman,
Department of Telecommunication,
Federal Urdu University of Arts, Science & Technology
Islamabad.

Acknowledgments

All praise to Almighty Allah, the most merciful and compassionate, without His help and blessing i was unable to complete the project.

This project could not have come about without the help, encouragement and guidance of the following persons.

To my respected project supervisor and teacher Dr. S. Tauseef-ur-Rehman. Without his precious guidance and help i could never be able to develop and complete such a research project.

To my friends Yasir Irshad Abassi, Yasir, Bilal, Fahad and Jehanzaib their kind and really professional guidance remained with me throughout the way.

And finally to my Parents who provided me suitable support both morally and financially.

Muhammad Asad Khan

136-CS/MS/03

Dedicated

To

My Parents & My Friend Yasir Irshad Abassi

Who motivated me to achieve this position in life and helped me in all fields of life.

Declaration

I hereby declare that this research thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that i have developed this software and the accompanied report entirely on the bases of my personal efforts made under the sincere guidance of our teachers. If any part of this system is proved to be copied out i shall stand by the consequences. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Muhammad Asad Khan

136-CS/MS/03

Project In Brief

Project Title:	Modeling Of Packet Loss and Delay using Multi-Path Diversity
Undertaken By:	Mr. Muhammad Asad Khan
Supervised By:	**Dr S. Tauseef-ur-Rehman** Chairman Department of Telecommunication, Federal Urdu University of Arts, Science & Technology Islamabad.
Date Started:	10th Mar 2004
Date Completed:	05th Sep 2006
Tools:	Network Simulator-2 under the environment of Linux.
Operating System:	Linux Fedora Core2.
System Used:	Pentium IV.

Abstract

This report explains the comparison of different models for the packet loss. The Research builds on packet loss models whereby Bernoulli model and Gilbert model were compared. Based on result of the experiments the research further elaborates the use of these models for the multi-path diversity. Multi-path diversity is a better approach. Hence Gilbert model was modified to account for missing parameters in the original model. The research strengthens the fact that modeling of packet loss directly affects the performance of the FEC. The emphasis of research is on the efficiency of Forward Error Correction by the modeling of packet loss and delay in the real time communication and to make it more useable to recover the lost packets. An experimental result shows that our packet loss is much closer to real world data and gives accurate results.

Table of Contents

Chapter 1

Introduction

1. Introduction

Voice over Internet Protocol (VoIP), IP Telephony and Broadband Phone is basically the voice packets or mostly the UDP packets that are routed over the IP based network to provide the voice conversation in the form of packets to the end users through any IP-based network [1]. Since the voice connection between the end users in IP based network is not based on the circuit switching technology and only relying on packet switching technology, therefore packet loss on the node or within the certain links effect the voice quality, which sometimes become a reason for short duration calls. Therefore, the quality of voice in IP based networks is highly reliant on Packet loss and delay. The delay of packet in reaching the destination also distresses the voice quality by producing dead air or white noise. If the delayed packets exceed the time required to reach the destination, they create the packet loss within the network [1, 2].

1.1 Overview of VoIP Network

Voice over Internet protocol (VoIP) is the term used to illustrate the analog signaling conversion into digital signal communication that is voice or modem calls. These digital converted packets of data transmission are then broadcasted over the public internet or intranets. The protocols used to carry digital voice signals over the IP network are commonly known as Voice over IP or VoIP protocols [1]. VoIP technology is still under the in-depth research to find out and work on the current drawbacks which creates problem in the voice quality of the conversation using IP as medium. Fixed delays are not easy to be controlled but the variable delays can be minimized by marking voice packets as being delay-sensitive. VoIP is predisposed to network behaviors, which is referred to as packet delay and jitter, which can humiliate the voice traffic to the point that is not satisfactory to the average user. VoIP is sent in a similar manner as that which is use to send streaming audio or video over the internet [1]. The advantage of voice network over the IP based platform is that it does not need to use whole circuit for its communication, like the circuit switching network use for the TDM traffic. It only sends the voice packets and work on packet switching environment for the transportation of packets.

1.1.1 Packet Loss

Packet loss is the loss of packet that occurs when certain voice or data packets that are moving across a network fails to reach their destination [3]. The cause of Packet loss can be due to a number of factors, including poor signaling quality over the network medium, abnormal congestion on the network links, corrupted packets rejection when they reach the destination or faulty hardware [3].

Packet loss is a normal phenomenon on packet networks. Transport layers such as TCP account for loss and allow packet recovery under reasonable loss conditions [4, 5]. Audio Codec's also take into account the possibility of packet loss; especially since RTP data is transferred over the unreliable UDP layer [3, 4]. Mostly, the voice Codec is used to perform one of its several functions that make the small length of packet losses unnoticeable to the end user. For example, a CODEC may choose to use the packet received just before the lost packet instead of the lost one, or perform more sophisticated interpolation to eliminate any clicks or interruptions in the audio stream [3, 4, 5]. Most of the times, the loss of packet starts to be a major threatening problem when the loss of the continuous packets burst cross the certain percentage of the packet loss from the sent packets to the destination(roughly 5% of the packets), or when packet losses are grouped together in large packet bursts. In those situations, even the best Codec's will be unable to hide the packet loss from the user, resulting in degraded voice quality. Thus, it is important to know both the percentage of lost packets, as well as whether these losses are grouped into packet bursts [3, 4, 5].

Some network transport protocols such as TCP provide for reliable delivery of packets. In the event of packet loss, the receiver asks for retransmission or the sender automatically resends any segments that have not been acknowledged. Protocols such as UDP provide no recovery for lost packets. Applications that use UDP are designed to handle this type of packet loss [3, 4,5].

1.1.2 Delay

Delay is the time taken from point-to-point in a network. Delay can be measured in either one-way or round-trip delay [5]. One-way delay calculations require expensive sophisticated test gear and are beyond the budget and expertise of most enterprise customers. However, measuring round-trip delay is easier and requires less expensive equipment. To get a general measurement of one-way delay, measure round-trip delay and divide the result by two. VoIP typically tolerates delays up to 150 ms before the quality of the call is unacceptable [4, 5]. But the same delay creates the packet loss within the network if exceeds the play out time.

1.1.2.1 Types of Delay

There are many delay components in voice networks. In voice networks, a single component might not cause problem, but the combination of the components can cause considerable delay in the network. These components include fixed delay and variable delay. Once the components are known, delay budget is calculated to determine the effect on the voice traffic.

There are three fixed delay components. Propagation delay, Serialization delay and Processing delay. Propagation delay is the delay incurred as a packet travels between the two points. Serialization delay involves the process of placing bits on the circuit. Higher the speed, less the serialization delay. The processing delay is the delay, which occurs during the coding, compression and decompression.

1.1.3 Jitter

Jitter is the variation in delay over time from point-to-point. If the delay of transmissions varies too widely in a VoIP call, the call quality is greatly degraded. The amount of jitter tolerable on the network is affected by the depth of the jitter buffer on the network equipment in the voice path. The more jitter buffer available, the more the network can reduce the effects of jitter [4, 5].

1.1.4 Post Dial Delay (PDD)

The Post dial delay is the time between pressing the last digit of a telephone number and receiving a ring or busy signal at the receiving end. The post dial delay is highly dependent on the delay among the nodes [6]. If the network has high post dial delay, then the source user will hear the tone of the phone ringing on the destination side after very long delay, which could be more then seven to eight seconds.

1.1.5 Latency

Latency is a measure of the delay in a call. It is the measurement of both the round-trip delay between when information leaves point A and when a response is returned from point B, and the one-way delay between when something was spoken and when it was heard. The largest contributor to latency is caused by network transmission delay. Round-trip latency affects dynamics of conversation. One-way latency is used for diagnosing network problems. With round trip latencies above 300 msec or so, users may experience annoying talk-over effects [4, 5, 6].

1.1.6 Average Length of Call

The average length of a call is the sum of all the duration of all the calls divided by the number of calls made. The advantage of ALOC is that the result of this test can tells the quality of the network. If the ALOC is less then four minutes then the network is below standard to take extensive calls.

1.1.7 Flutter

The Flutter is the rapid change of the electrical signals during the flow of voice packets. The flutter really needs to be ensured during the test of any kind of voice network. If there are rapid changes occurring in the network, it affects the quality of the network and generates voice breaks at the destination side. Flutter also occurs either due to the packet loss or because of the extensive delay of packets to reach the destination.

1.1.8 ASR

Average Success Rate is the average of total number of calls which are successful. The ASR is calculated by dividing the total number of calls which are made by the total number of successful calls. The ASR is important for the analysis of the quality of any kind of network. If the ASR is below fifty percent then that means the network is below average and it will discard most of the calls that are made into the network. The ASR of a very stable network can also be less then fifty percent, if the calls coming to the network are not properly routed to the destination. In this scenario, the coming calls degrades the ASR, hence affects the analysis of network.

1.2 Modeling Algorithms

As stated earlier while designing a network we must be taken regarding modeling techniques. In this section various loss models are have been described.

1.2.1 Bernoulli Model

The Bernoulli loss model is based on a geometric distribution. It is the most widely used model and based on simple independent losses. It is very basic modeling algorithm for estimating packet loss. Therefore, often use in modeling of packet loss for IP voice and multicast systems [7]. As in the figure 1.1, the probability of packet loss in the Bernoulli model is represented by **p**. If there are large number of packets **n** to be transmitted over a network then the expected number of lost packets is **n*p**. Bernoulli loss model is a two-state process. The one state (State 0) symbolizes a packet loss, and the other state (State 1) stands for a packet reaching the destination. The mean loss probability **p** represents the probability that the current packet is lost given that the last packet was also lost. **q** is the probability that the current packet is arrived provided that the previous packet was also arrived [7].

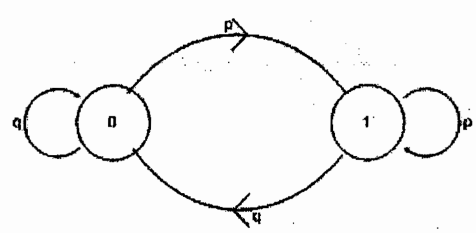


Figure 1.1 The Bernoulli Model

The Bernoulli model gives an estimation of loss probability by calculating the total number of packets that were lost and then dividing the result by the total transmitted packets. In Bernoulli model, each packet transmitted on a network has fixed and independent loss probability with constant loss rate of the link [4, 6]. Networks in which time interval between the packets is very short, the loss of packets can not be estimated properly by a Bernoulli model [7].

1.2.1.1 Bernoulli Loss Model

The estimation of Bernoulli loss can better be analyzed by designing a complex network which comprises of both data and voice traffic from different nodes to the destination nodes. Assign starting and ending time to extract relevant data during specific time. Dispense starting and ending time to all the nodes which are generating the packets. Start sending the packets and then calculate the sent time, received time and the total delay that packets required to reach the destination. Calculate the total voice packets sent by specific source to destination, the actual loss of voice packets for the above mentioned link and the actual length of bursts of packet loss. Bernoulli algorithm is then be used to calculate the mean Probability.

$$\prod = (\sum_{i=1}^{n-1} Li) / L_0$$

Where

- Li is the loss bursts numbers with length i.
- The value of i=1, 2, 3..........n-1. i=1 means that single packet loss whereas n-1 is the longest burst of packet loss.
- L_0 is the total number of packets sent from source to the destination [7].

Now as we have the Mean Probability using Bernoulli model, we need to approximate the Probability of losing the packets for the different lengths of bursts using following probability formula.

$$\prod_k = \prod * (1 - \prod)^{k-1}$$

Where

- $\prod_k$ is the actual probability for estimating the packet loss.

- k is the length of burst of packet loss. e.g. If k=1 that means $\prod_1$ is calculating the probability of single packet loss [1].

The result of probability of packet loss will lead us to the estimation of packet loss by using following algorithm.

L(est) = Total packet loss * $\prod_k$

Where

- $\prod_k$ is the probability taken from the above step.
- L(est) is the estimated packet loss using Bernoulli model of packet loss.

1.2.1.2 Bernoulli Trials

The Bernoulli trials process, named after James Bernoulli, is one of the simplest yet most important random processes in probability [7]. Essentially, the process is the mathematical abstraction of coin tossing, but because of its wide applicability, it is usually stated in terms of a sequence of generic trials that satisfy the following assumptions:

1. Each trial has two possible outcomes, generically called success and failure.
2. The trials are independent. Intuitively, the outcome of one trial has no influence over the outcome of another trial.
3. On each trial, the probability of success is p and the probability of failure is $1 - p$ [7].

Mathematically, we can describe the Bernoulli trials process with a sequence of indicator random variables: I_1, I_2, I_3...

An indicator variable is a random variable that takes only the values 1 and 0, which in this setting denote success and failure, respectively. The j'th indicator variable simply records the outcome of trial j. Thus, the indicator variables are independent and have the same density function:

$P(I_j = 1) = p$, $P(I_j = 0) = (1 - p)$

Thus, the Bernoulli trials process is characterized by a single parameter p [7]. In a sense, the most general example of Bernoulli trials occurs when an experiment is replicated. Specifically, suppose that we have a basic random experiment and an event of interest A. Suppose now that we create a compound experiment that consists of independent replications of the basic experiment. Define success on trial j to mean that event A occurred on the j'th run, and define failure on trial j to mean that event A did not occur on the j'th run. This clearly defines a Bernoulli trials process with parameter $p = P(A)$ [7].

1.2.1.3 Tests in the Bernoulli Model

Suppose that $I1, I2, \ldots, In$ is a random sample from the Bernoulli distribution with unknown parameter p in (0, 1) [7]. Thus, these are independent indicator variables taking the values 1 and 0 with probabilities p and $1 - p$ respectively. Usually, this model arises in one of the following contexts:

a. There is an event of interest in a basic experiment, with unknown probability p. We replicate the experiment n times and define $I_i = 1$ if and only if the event occurred on the i'th run.

b. We have a population of objects of several different types; p is the unknown proportion of objects of a particular type of interest. We select n objects at random from the population and let $I_i = 1$ if and only if the i'th object is of the type of interest. When the sampling is with replacement, these variables really do form a random sample from the Bernoulli distribution. When the sampling is without replacement, the variables are dependent, but the Bernoulli model may still be approximately valid.

1.2.2 Gilbert Model

The most widely known burst model is the Gilbert Model and a variant known as the Gilbert-Elliott Model. These are both two state models that transition between a "good" or gap state 0 and a "bad" or burst state 1 according to state transition probabilities P01 and P11 as mentioned below [7]. In Gilbert Model State 0 is a zero loss/error state and state 1 is a lossy state with independent loss probability Pe1. In Gilbert-Elliott Model State 0 is a low loss state with independent loss probability Pe0 and state 1 is a lossy state with independent loss probability Pe1 [7].

It is often assumed that the Gilbert Model lossy state corresponds to a "loss" state, i.e. that the probability of packet loss in state 1 is 1, however this is incorrect (it would be more proper to describe this as a 2-state Markov model). This leads to analysis of packet loss burstiness in terms solely of consecutive loss which misses the effects of longer periods of high loss density. As illustrated below [7,8], these long periods of high loss density can have significant effect on Voice over IP services.

For example, consider the following

Loss pattern 000001100101010110110000000000000000000 Correct application of Gilbert Model – burst length 15, burst density 60% Incorrect application of Gilbert Model – mean burst length 1.5 bits [7,8].

1.2.3 Extended Gilbert Model

The Bernoulli model maintains the record of all past **n** number of losses to calculate the probability of losing the next packet, where as Extended Gilbert model consider only last **n** number of consecutive loss of packets to calculate the probability of the next packet to be lost. Therefore the probability calculation and loss estimation are in the vicinity of the actual loss of packets. The extended Gilbert model needs **n+1** state to remember **n** events [7].

The Extended Gilbert Algorithm is used with a structure that maintains a counter l, which is the number of consecutive packet loss. But it is reset whenever the packet is received [7]. The extended Gilbert model is the extension of two-state Gilbert model which calculates the burst state with almost the same transition probability as of Bernoulli algorithm by considering burst state as 1 and then according to state transition probabilities P01 and P11 is the probability for the burst length of packet loss as shown in figure 1.2 [7].

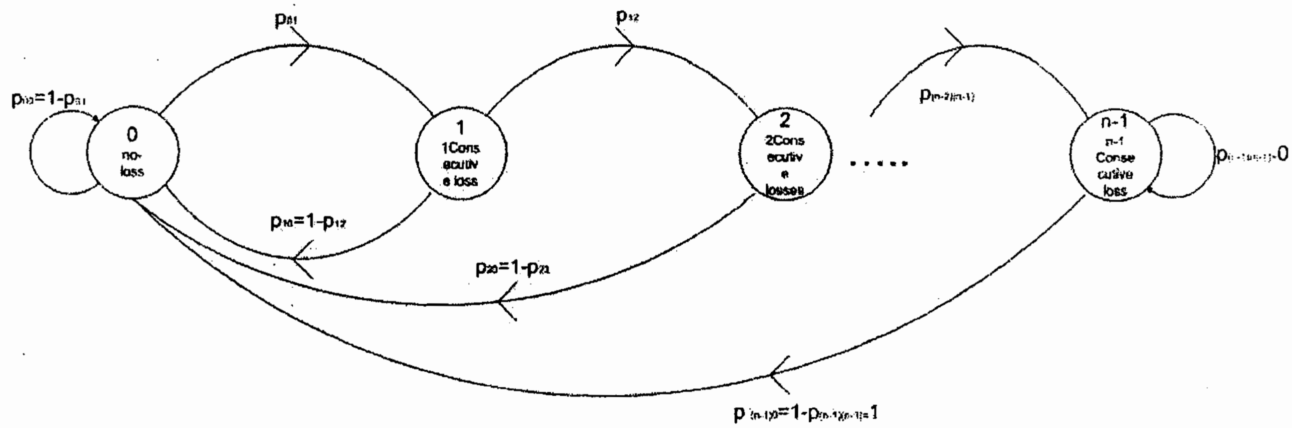


Figure 1.2 The Extended Gilbert Model

1.2.3.1 Extended Gilbert Loss Model

The estimation of Extended Gilbert loss can better be analyzed by designing a complex network which comprises of both data and voice traffic from different nodes to the destination nodes. Assign starting and ending time to extract relevant data during specific time. Dispense starting and ending time to all the nodes which are generating the packets. Start sending the packets and then calculate the sent time, received time and the total delay that packets required to reach the destination. Calculate the total voice packets sent by specific source to destination, the actual loss of voice packets for the above mentioned link and the actual length of bursts of packet loss. Extended Gilbert algorithm is then be used to calculate the mean Probability.

$$\mu = 1 - \left(\sum_{j=1}^{n-1} L_j * (j-1)\right) / \left(\sum_{j=1}^{n-1} L_{(j-1)} * (j-1)\right)$$

Where

- Lj is the loss bursts numbers with length j.

- The value of j=1, 2, 3..........n-1. j=1 means that single packet loss whereas n-1 is the longest burst of packet loss.

- Now as we have the Mean Probability using Extended Gilbert model, we need to approximate the Probability of losing the packets for the different lengths of bursts using following probability formula.

$$\mu_k = (1- \mu_{(k-1)(k)})^{k-1} * \mu$$

Where

- μk is the actual probability for estimating the packet loss.

- k is the length of burst of packet loss. e.g. If k=1 that means μ_1 is calculating the probability of single packet loss [7].

The result of probability of packet loss will lead us to the estimation of packet loss by using following algorithm of extended Gilbert Model.

$$L(est)k = \sum_{j=1}^{k-1} Lj * \mu_k$$

Where

- μ_k is the probability taken from the above step.
- $L(est)_k$ is the estimated packet loss using Extended Gilbert model of packet loss.
- k is the length of packet loss for which estimation has been calculated.

1.2.4 Markov Model

A Markov model is a general multi-state model in which a system switches between states i and j with some transition probability p(i, j). A 2-state Markov model has some merit in that it is able to capture very short term dependencies between lost packets, i.e. consecutive losses [8]. These are generally very short duration events (say 1-3 packets in length) but occasional link failures can result in very long loss sequences extending to tens of seconds. By combining the 2-state model with a Gilbert-Elliott model it is possible to capture both very short duration consecutive loss events and longer lower density events [8].

1.3 Multi-Path Diversity

In Multi Path diversity, the copies of identical packets are sent over a network to achieve the advantage of uncorrelated packet loss and delay [9]. The advantage of multi-path diversity over single path is that there is less probability of losing the packet. If certain packet is lost from one path, still it has positive probability of reaching the destination from the second path. Another advantage of multi-path diversity is that the extent of the packet loss burst is relatively diminutive. The major shortcoming of multi-path diversity is that it needs superfluous bandwidth on both source and the destination links. In addition the destination node needs further processing to discard the already received packets. Therefore, it creates some extra delay in the arrival of voice packet to the destination, but it is better then losing the packets [9].

The destination node needs to either check that the arrived packet is coming first time towards the destination and it is not already been received. If the arrived packet is already received then the destination node discards the arrived packet. Another very interesting scenario in the multi-path diversity is that one stream has less congestion and the packets are arriving at the destination at very high speed, while the other stream has either long path or have congestion among the nodes. All of a sudden certain packet is lost from the faster stream and again the packets start arriving. The lost packet is later received by the other slower stream. Now the destination node will ensure that the arrived packet is within the playing time, if the packet crosses the play out time, the destination node discards it. If the arrived packet reaches at the destination before the play out time then the destination node does extra processing to place the packet in a buffer at its appropriate location.

1.4 Forward Error Correction

FEC is accomplished by adding redundancy to the transmitted information using a predetermined algorithm. Each redundant bit is invariably a complex function of many original information bits. The original information may or may not appear in the encoded output; codes that include the unmodified input in the output are systematic, while those that do not are nonsystematic. There are two main categories of FEC, which are block coding and convolution coding [15].

Block codes work on fixed-size blocks (packets) of bits or symbols of predetermined size. Convolutional codes work on bit or symbol streams of arbitrary length. A convolutional code can be turned into a block code, if desired. Convolutional codes are most often decoded with the Viterbi algorithm, though other algorithms are sometimes used. There are many types of block codes, but the most important by far is Reed-Solomon coding because of its widespread use on the Compact disc, the DVD, and in computer hard drives [15]. Block and convolutional codes are frequently combined in concatenated coding schemes in which the convolutional code does most of the work and the block code (usually Reed-Solomon) "mops up" any errors made by the convolutional decoder.

1.5 Issues in Single Path

The different issues regarding scalability and performance in single path are as follows:

1.5.1 Reliability

The single path phenomenon does not provide reliability of packets to reach the destination. As if certain packet is lost from the stream, it is very difficult to recover within play-out time. Although there are few algorithms like Forward Error Correction to recover the lost packets, but the voice packets are time dependent. So FEC needs some time to recover the packets in time, which creates delay in delivery of packets to the destination.

1.5.2 Loss Burst

In single path, the length of the packet loss burst increases because of the unavailability of any other path. This creates difficulty to forward error correction in the recovery of the packets. Recovery in forward error correction is highly dependent of the next coming packet. Therefore, if consecutive packets are lost from the stream, forward error correction algorithm does not have the solution to recover the consecutive lost packets. Because the forward error correction mechanism is based on the recovery of packets from the next coming packet. The information in lying in the header of the packet, so it sends the request to the sender to resend the lost packet. But is is not possible in very bursty network.

1.5.3 Voice Quality

In the single path, voice quality of the network is one of the major issues. Because if the delay exceeds the limit then the listener can only have white noise or dead air. Also, if the packets are lost or reach the destination after the playout time then there is break in voice for both the sender and the listener. This deficiency is covered in the multi-path diversity phenomenon because of the multiple packets streams availability for the receiver.

1.5.4 Jitter Buffer

Another disadvantage in the single path is that if the network is very congested at the destination node then the Jitter buffer remains bottle neck on most of the occasions. The Jitter buffer is use to store the arrived packets into some buffer and then play the packets before the play out time. The advantage of Jitter buffer is that the voice distortion and voice gap within the communication becomes very less. But if the Jitter buffer is full most of the time because of sudden arrival of the packets then the use of buffer becomes minimal. As the newly arrived packets start dropping due to unavailability of their place in the buffer.

1.6 Issues in Multi-Path

The different issues regarding scalability and performance in multi path are as follows:

1.6.1 Overhead

In the multi-path diversity phenomenon, the major disadvantage is the overhead in the packets. As the packets are moved from the source to the destination, they have to keep track of the packets from source to the destination to discard the redundant packets if arrived at the destination. This produces an extra overhead at the destination node.

1.6.2 Processing Delay

The other major disadvantage in the path diversity is that the destination node has to do some extra processing. The processing needs to be done at the receiving node to observe any packet that is either earlier received or the earlier lost packet is recovered from some other stream. If the arrived packet is already been received then the destination node discard the arrived packet. If the arrived packet is the recovery of earlier lost packet then the packet is placed in its appropriate place in the playing queue. Therefore, this creates an extra processing delay at the destination. But still this processing delay at the destination is better then losing most of the packets at the destination.

1.7 Objective

The main objective of thesis is to provide better quality of service in the IP based network by introducing better model in multi-path diversity. As in multi-path diversity, if certain packet is lost from one link, it has fair chance to reach the destination from some other stream. Our goal is to provide some advantage to Forward Error Correction. The first goal is to get better model for packet loss and delay among Bernoulli model and Extended Gilbert Model for the single path. We then evaluate the better loss model from the results of single path architecture, which will then be used for the modeling of packet loss for multi-path diversity phenomenon. The present research aims to increase the efficiency of Forward Error Correction (FEC) by the modeling of packet loss and delay in the real time communication.

1.8 Motivation

The reason for selecting the research on the modeling of packet loss and delay is that in very near future the whole communication structure will be transferred from TDM traffic to the IP communication. Although the VoIP is already very advance and completely deployed in most part of the world. But the implementation of VoIP over Mobile wireless like Wimax is still under standardization. Once the Standards will be set for the VoIP will be fully implemented on Wimax. At that point of time, they need to estimate and find the probability of the traveling packets to reach the destination to achieve the better quality of the service to end user. Also for the end to end voice communication using the VPN with controlled and better quality they need to send the voice packets from the two streams.

The Aim was to the provide the research based schema to the next coming service provider so that they will not need extra efforts to get the better modeling of algorithm to provide services like Voice over the IP with wireless medium.

Chapter 2

Literature Survey

2. Literature Survey

2.1 Modeling Algorithms

It is proposed that the real time communication is very sensitive as far as packet loss and delay is concerned. Generally this delay is handled and modeled using different algorithms.

2.1.1 Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia service quality

Wenyu Jiang and *Henning Schulzrinne* gave a unique concept of modeling of packet loss and delay in real time with an integration of Inter-loss distance. Their main emphasis and concern was the loss of voice packets in lengthy bursts. According to them, if n packet is lost then n+1 can also be lost; therefore this creates loss of packets in bursts hence really effect the quality of the IP based network. They proposed and worked on the joint use of extended Gilbert Model and Inter-loss distance. They used inter-loss distance scheme for the temporal loss dependency. The integration models and flow of packets with FEC recovery is shown in figure 2.1. These models were then applied on the internet packets and they have used traces to identify the effectiveness of different modeling algorithms. The modeling of packet loss is done using Bernoulli model, Gilbert model and extended Gilbert model. The comparative results have shown that extended Gilbert model is much better then the Bernoulli and 2-state Gilbert model [7] for the modeling of packet loss. They have not proposed anything regarding the introduction and usage of forward error correction with the multi-path diversity phenomenon.

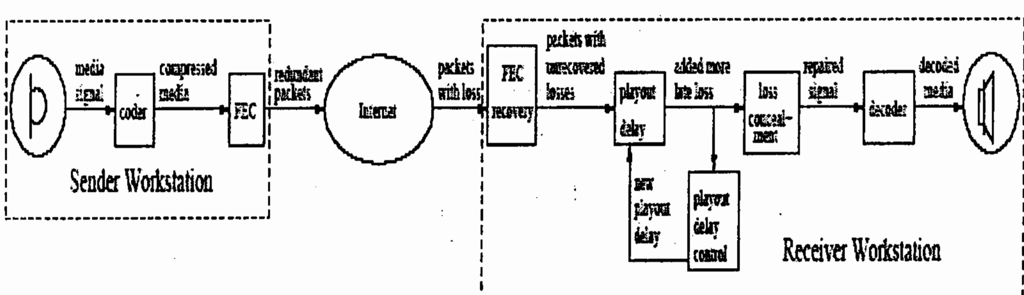


Fig 2.1 Flow of voice packets with FEC [7]

They have also discussed the basic flow of the voice packet over the internet along with the coding and decoding of the packets and the implementation of forward error correction. After the voice codec are applied to the packets, they are sent with the implementation of forward error correction technique by sending the redundant packets form the same path. At the destination, there is recovery mechanism ready for the recovery of the packets, and try to recover the lost packets before delivering to the end user. But the FEC fails to recover the lost packets, if the packets are lost in lengthy bursts. The recovery of the packets in the forward error correction is highly dependent on the next coming packet. Therefore, if the packets are lost in continuous sequence, FEC is unable to recover those packets [7].

The drawbacks in the research of this paper are that they have used the trace having very less packet losses and the network was not very bursty. The packet losses were very minute, therefore they were unable to identify and show the importance of the Bernoulli model. The Bernoulli model has an advantage of over estimation in most of the burst lengths and loss burst cases. Therefore, they were able to identify the over estimation of the Bernoulli model but were unable to identify the advantage of Bernoulli model in case of estimation for modeling of packet loss in bursty network [7].

Secondly, the modeling of packet loss using different algorithms is done using single path. They have not introduced any concept of modeling of packet loss using multi-path diversity, which really effects positively in routing of packet loss [7].

2.1.2 Authentication Streamed data in the Presence of Random Packet Loss

Philippe Golle and *Nagendra Modadugu* worked on the authentication of packets in the presence of random and consecutive packet loss forming a lengthy burst. Their research was also based on the assuring of UDP packets to reach the destination in real-time. Their major emphasis was on the efficiency of packets authentications, so that the overhead of authentication does not affect the packets to reach the destination by adding extra delay. They formed signature scheme that allows sending and receiving parties to exchange voice and video streams with guarantees of integrity and no repudiation [10].

This paper also explains the importance of robustness during the assuring of packet delivery by making the signature scheme, robust enough that authentication remains possible even if some packets are lost. Existing authentication schemes that resist packet loss have been designed to resist worst-case packet loss. Any number of packets may be lost anywhere in the sequence, without interfering with the receiver's ability to authenticate the packets that arrived. Studies conducted on packet loss in UDP suggest that resisting worst-case packet loss is overkill. The focus should be instead on resisting random packet loss. They proved that their construction is optimally resistant to bursty packet loss given the resources available to the sender and the receiver, and has the lowest possible communication overhead [10].

2.1.3 Reducing Packet-Loss by Taking Long-Range Dependencies into Account

J. Ignacio Alvarez-Hamelin and *Pierre Fraigniaud* worked on the reduction of packet loss in the real-time. They have shown that the "fractal" behavior of Internet traffic can be efficiently and practically employed to significantly reduce in packet loss. They defined the probabilistic congestion of a link, based on an estimated computation of the packet-loss probability over the specific link. This congestion parameter allows valid predictions on the future behavior of the network, on which the sending and receiving ends can form efficient routing strategies. They have shown in their research how to implement the computation of the probabilistic congestion, and illustrated several applications for improving unicast and multicast protocols [11]. The packet flow and the loss of packets on specific node are shown in figure 2.2.

This research paper also explains that the Packet-loss is the cause of important performance degradations in packet networks, for it is fundamentally related to all "standard" Quality of Services (QoS) measures: latency, bandwidth, jitter, etc. Using TCP, it is necessary to resends packets, with time-consuming effects. Because in the TCP network, the packets are mostly data packets which are not time dependent, whereas, the packets coming in the UDP are mostly the voice packets, which are time dependent and we can not afford to have the delay in sending the packets to the destination. Using UDP, packet-loss has a significant impact on the user-perception for certain applications. IPv4 allows the use of a Type-of-Service (ToS) field in the IP-headers, for choosing the route with minimum packet-loss [11].

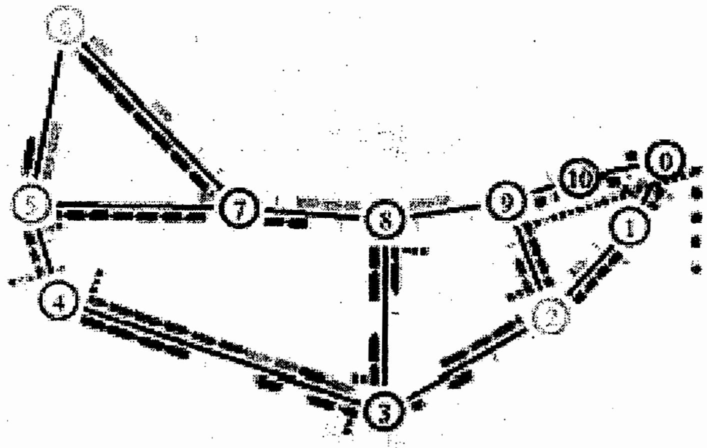


Fig 2.2 Reducing Packet-Loss by Taking Long-Range Dependences into Account [11]

The major drawback in their research paper is that they have not discussed about the type of service for choosing the route with minimum packet loss. Also, through the research paper, it is still unclear how much the global traffic would improve in terms of packet-loss. Indeed, the behavior of the network traffic depends heavily on TCP which dynamically adapts to the congestion of the routes. They have left this for the future enhancement and research [11].

2.1.4 Modeling End-to-End Packet Delay Dynamics of the Internet using System Identification

Hiroyuki Ohsaki, *Masayuki Murata* and *Hideo Miyahara* worked on the modeling of end-to-end delay of the packets for both real time communication and non-real time communication. They also elucidated that the delay of the packets to reach the destination directly affects the quality of service [12].

They have proposed the modeling of packet delay using Novel approach. The approach for the modeling of end to end delay is by sending the packets from specific source to specific destination to find out the exact delay dynamics for the internet packets using system identification. The end to end delay is modeled as single input single output model (SISO). The input to the system is the packet inter-departure time from the source host. They have calculated inter-departure time of the packet from network layer of source host to the network layer of destination host, which is also shown in figure 2.3. Then they calculate the variation of end-to-end delay to making several experiments. The output from the system is the calculated variation in the delay of the packet, while traveling from source to the destination.. They have also used ARX (Auto-Regressive eXogenous) model and its coefficients are determined using system identification [12].

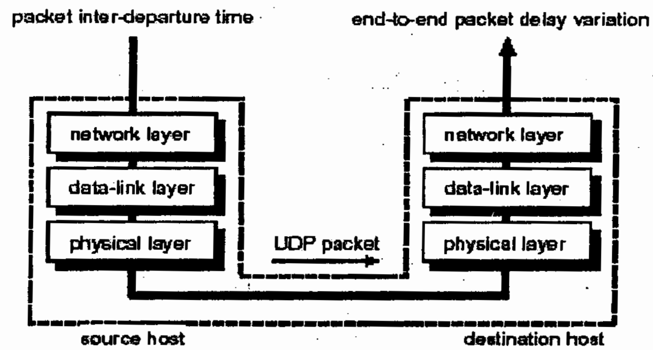


Fig 2.3 Modeling end-to-end packet delay dynamics as SISO [12]

Their major emphasis is to make a mathematical model that can be used, in particular, to design a delay-based congestion control mechanism. Their idea was to initially, develop a suitable model to estimate the end to end delay and then apply the optimal control theory to design an efficient delay based congestion control mechanism. The implementation is based on the simulation of TCP and UDP packets from the source to the destination. The inter-departure time is randomized using exponential distribution. The host is either sending UDP packets only or both TCP and UDP packets simultaneously. But in either case, the destination host is only calculating the end to end delay of the UDP packets as shown in figure 2.3 [12].

In this research paper, they did not calculate the delay of those packets which reach the destination after the play-out time. This creates the loss of the packet ate the destination. In this specific case, the packet does reach the destination, but the delay is so huge that the reached packet is of no use. The research has also missed the modeling of packet loss using any modeling algorithm, therefore the dynamic delayed packets which turns into the loss of packets can not be modeled for the better quality of service [12].

2.2 Multi-Path Schemes

The multi-path phenomenon or the packet path diversity is the major and very sensitive implementation for the better quality of service. The multi-path diversity is important and increases the efficiency of voice and video packets to reach the destination by providing extra path.

2.2.1 Real-Time Voice communication over the Internet using Packet Path Diversity

Yi J.Liang and *G.Steinbach* worked on the quality of the network, specifically voice network, because voice networks are more sensitive as compare to pure data networks. They discussed that the receiving side mostly have different kinds of buffers to store the packets in case of congestion at the delivery point. But these buffers sometimes take too much time to store and process the buffered data that they create late loss of the packets. They propose to improve the transaction among delay, late loss rate, and speech quality using multi-stream transmission of real-time voice over the internet. They have introduced the phenomenon of sending the redundant packets over the internet using different independent network path with the comparison of FEC stream also shown in figure 2.4 [13].

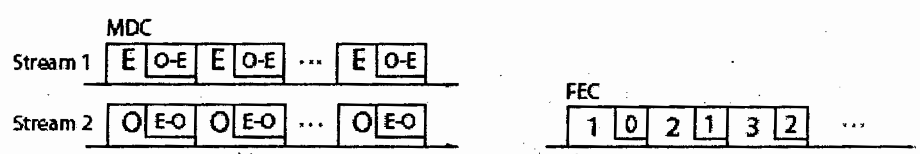


Fig 2.4 Comparison of MDC and FEC [13]

They have used the Lagrangian cost function to operate delay versus loss. The results from the conducted experiments over Internet suggest largely uncorrelated packet erasure and delay jitter characteristics for different network paths which leads to a noticeable path diversity gain. Their results have also shown significant reductions in mean end-to-end latency and loss rates as well as improved speech quality when compared to FEC protected single-path transmission at the same data rate. In addition to Internet measurements, they scrutinize the performance of the proposed multi-path voice communication scheme using the ns network simulator for different network topologies [13].

This research is based on the use of the Lagrangian cost function for the multi-streaming using packet-path diversity. They did not work on the modeling of packet loss and delay using different modeling algorithms. This research was based only on the cost analysis for quality of service in real time communication using Lagrangian cost function. They have not analyzed that which modeling algorithm is better for the modeling of packet loss in single path phenomenon and multi-path phenomenon in the real time communication [13].

2.2.2 Path Diversity and Multiple Descriptions with Rate Dependent Packet Losses

Jagadeesh Balam and *Jerry D. Gibson* researched on the path diversity and the integration of multiple descriptions for the packet loss which are rate dependent. In their research, they have heavily considered the wireless standard 802.11 which is mostly use for the wireless LAN and ad hoc networks. The packet loss phenomenon use in this research work is that if the single bit of a packet is having any error, as a result the packet loss probability is directly proportional to the packet size [14].

They have compared the average packet distortion per symbol achieved at the receiver for simple path diversity methods against MD coding. The pattern of encoding and decoding of packets are shown in Figure 2.5. Their accent is on the overheads of the packets that sometimes dominates the packet size in the case of small packets. Firstly, they have compared the performance of the packets that consists of only source information and then they have shown the effect of packet overheads on the performance of each of the methods. [14]

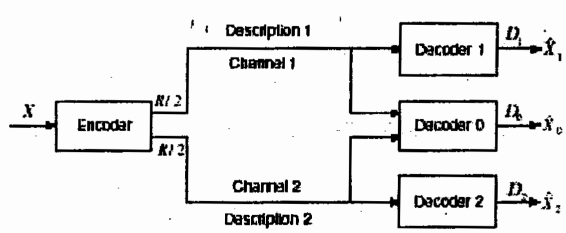


Fig 2.5 A two-description coder having two independent channels [14]

The research implementation is based on different scenarios to get the better results for the comparison. The specific scenario of their work is based on on-off parallel channels. They have assumed that in this kind of scenario, the MD coding should outperform a (single description) SD method because the packet size in each description for the multiple description is smaller then the single description packets. They have proved that when the payload is small (Which is mostly in the voice communication) then the packet header can dominate the packet size and the bandwidth efficiency that is achieved in the multi description coding is insignificant [14].

2.2.3 Path Diversity Based Techniques for Resilient Overlay Multimedia Multicast

Matulya Bansal and *Avideh Zakhor* worked and researched on the congestion at the node or failure/drop of a link. They have analyzed and reported that the packet loss mostly occurs either due to link failure among the two nodes or the congestion of the packets at any node. The congestion mostly occurs due to the sudden arrival of the packets at specific node. In this paper, they have proposed and evaluated two new techniques. In the first scheme, each receiver streams an independent multimedia stream, such as the Forward Error Corrected (FEC) bit-stream or a Multiple Description Coded (MDC) description, from a parent, depending upon the existing error conditions in the network. This simplifies the receiver implementation as no packet partitioning is required. In the second scheme, they have constructed multiple interior-node disjoint k-DAGs, and stream mutually exclusive stripes of content on each k-DAG. This improves tolerance to node failures as any node is an interior node in at most one of the k-DAGs. Our results show that the proposed techniques are very effective in dealing with packet losses in the network, improving FEC good put by 20-35% and MDC connectivity by 15-20% [16].

In this paper, they have presented three overlay multicast techniques for improving the performance of Forward Error Corrected (FEC) and Multiple Description Coded (MDC) media under conditions of congestion and node failures in the network. All these proposed techniques are based on the K-DAGs (Directed Acyclic Graphs) as shown in figure 2.6. In the first scheme that is called ROM, the receiver streamed simultaneously from multiple different parents, thereby decor relating losses and mitigating the effect of node failures. In the second technique that is called SPS (Simple Parent Selection), the receiver streams an independent multimedia stream. Such as the Forward error correction coded bit stream or multiple description coding from the parent. The third technique known as Multiple Interior Node disjoint k-DAGs (MINK), constructs multiple interior node disjoint k-DAGs, and streams mutually exclusive stripes of content on each [16].

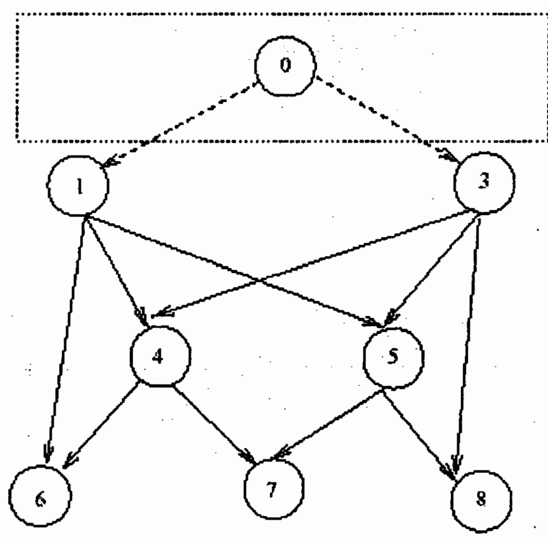


Figure 2.6 A 2-DAG. Unlike a traditional multicast tree [16]

2.3 Network Architecture

The importance of network architecture for the modeling of any kind of algorithm cannot be negligible. The best results can be get only when the architecture is really strong.

2.3.1 Path Diversity with Forward Error Correction (PDF) System for Packet Switched Networks

Thinh Nguyen and *Avideh Zakhor* worked on the end to end delay and packet loss over the packet switch network that is internet. They have described that the Delay sensitive applications such as video streaming and conferencing are challenging to deploy over the Internet due to a number of factors such as high bit rates, delay, and loss sensitivity. They have also enlightened that the Transport protocols such as TCP are not suitable for delay sensitive applications since they retransmit lost packets resulting in a long delay. They have also referred to their earlier research for proving that the packet loss can be controlled by sending the packets at specific rate and sending those packets from different hosts and through disjoint paths [17].

In this paper they have proposed to use the multi-path diversity with the forward error correction system for delay sensitive applications over the internet in which the packets re sent over the network with disjoint paths using the collection of relay nodes. In the following figure 2.7, they have also introduced the redundant paths between the sender and the receiver and achieved significant reduction in the packet loss by differentiating actual path and the redundant path [17].

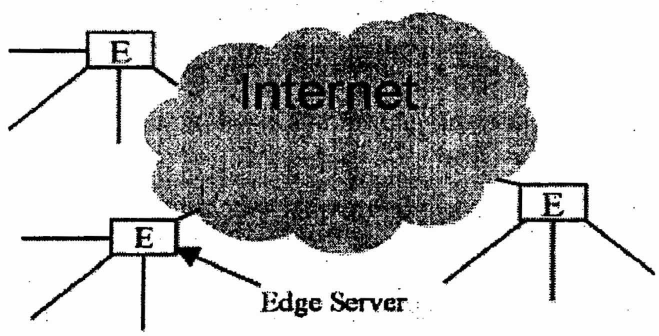


Fig 2.7 Edge Server Architecture.

They have described that the RON dynamically determines the best path for the packets to be sent on, while the PDF system sends the packets simultaneously on different multiple paths. Detour is similar to the PDF, but there is small difference between them and that is Detour selects the path ate the network layer that is having a router, whereas the multi-path works over the application layer of the OSI layer as shown in figure 2.8 [17].

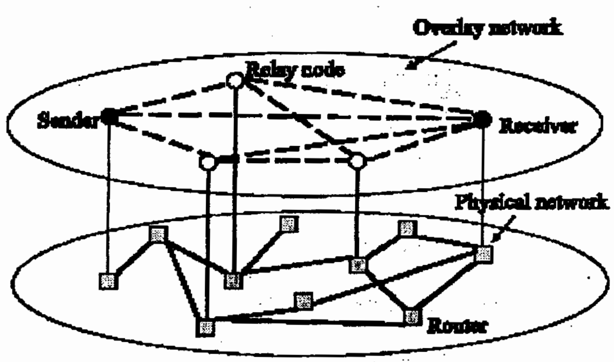


Fig 2.8 System Architecture [17].

2.4 Problem Domain

In the earlier research, most of the algorithms are used to calculate the packet loss and delay for the single path. They worked mostly to calculate the delays using different algorithms, but no body has compared them to find the better among them.

If we summarize the researches, which we had discussed earlier in the research, the researchers examined applications using the old method for the calculation of adaptive quality of service. The algorithm used was Markov two state models. In one research they used the Bernoulli model for the modeling of packet loss and delay. They implemented it only on the single path; the concept of multi-path was not introduced by them in this research. They did the modeling of packet loss to estimate the get the better quality of service. In another research which was based on the multi-path diversity phenomenon, the researchers worked on finding the cost function of the network using Lagrangian cost function. In another extensive research, the authors examined the link failure between the two networks. The problem was based on the passive measure that in passive measurements alone cannot provide results on the quality of the voice calls.

The Problem in all these research is that no body has ever aimed to get the better algorithm for the single path and multi-path phenomenon by comparing the algorithms. They have either used single path for the modeling of packet loss or they have used the multi-path structure for calculating the cost function. So in all these research they never seek to use the multi-path to calculate the packet loss in comparison with the single path. So the efficiency of Forward Error Correction was not compared and its better work efficiency was not used for the multi-path diversity phenomenon.

2.5 Proposed Solution

As we have discussed about the certain problems in the real time communication. So we have proposed a solution to have better quality of service using the multiple paths. The other major factor for the packet loss and delay during the transmission is the burst ness of the packets, which leads to the congestion. So this delay and packet loss degrades the quality of service for the real time communication. Therefore, it requires a technique to recover the lost data. The technique mostly use is, Forward Error Correction (FEC) to recover the packets for the single path. The problem in FEC for the single path is that, if one packet is lost then all the interlinked packets will be lost.

The model for the packet loss and delay and their effect on the real time communication is done by the extended Gilbert model. We are going to compare the extended Gilbert model with Bernoulli model for the single path. The best is going to be implemented on the multi path diversity. The multi path diversity is use to send the same packets from different paths, so if the certain packet is lost on the one stream, the second packet will still be able to reach the destination. We are going to calculate the better algorithm for the single path and then implement it on the multi path diversity phenomenon to model the packet loss and delay. Earlier, the communication was mostly done on the wired network, so there was less chance of losing the packet, because in the wired network, the dedicated stream for the packets is set before the transmission. Now in the wireless communication, the packets are sent without the determination of the stream, so there is more chance for losing the packets. We have introduced the FEC for the recovery of the lost packets for the better wireless real time communication. For the recovery of lost packets, few extra bits are added for the encoding of the packets. The encoding technique will be the soft decision encoding technique. In the soft decision encoding/decoding, the packets are not send on the predefined path, rather the packets are send on the best possible path. So we will be able to overcome the problem in the hard decision encoding technique. The problem in the hard decision encoding/decoding is that the packets are sent on the predefined path. So any link failure creates the loss in the packets. The whole emphasis is on the efficiency of Forward error correction by the modeling of packet loss and delay in the real time communication and to make it more useable to recover the lost packets. By the implementation, we are also going to check that if there is any change in the efficiency of forward Error Correction (FEC) by the introduction of multi path diversity.

Chapter 3

System Analysis and Design

3. System Analysis and Design

The design phase is the first step moving towards the problem solution. In the system design phase all the processes of the proposed system are designed. This means how these processes will be implemented and interacts with each other. The design of the system is the best if a system built from it, completely satisfies the requirements specified in the requirement specification document

This chapter describes the design and implementation of object oriented programming and aspect oriented programming based applications, and comparison of both applications. The application is basically a simulation of packets to get the appropriate results. The idea about this application is that an opportunity to practice analyzing designing and implementing using network simulator.

3.1 The Basic version

This section describes the actual object-oriented software development process that solves the problem domain of the network simulation of packet loss and delay system requirements.

When designing an OOP application, the following steps should be taken:

3.1.1 Basic Analysis

Find the basic business objects needed to get the basic functionality running, in accordance with the problem domain. Detect any sub-objects that are needed by the basic business objects, for instance a connection object will probably need the header files, packet type and some operation objects. One easy way to do is to write down what the system must do and turn the nouns into entity classes and the verbs into operations on the entity classes. It will not make a perfect model, but a good starting point. Objects are added to class diagrams in UML. Once this has been completed, it is extremely important to review the class diagram to see if the existing objects are cleanly encapsulated. If not, make the proper changes until there is a modular, clean-cut architecture. For now, focus should not be on object interaction, rather the types of objects needed and what basic services they should offer.

The analysis phase started with identifying the basic objects and functionality of the system, based on the problem domain. The basic idea is to make the application that can simulate the packets from the source to the destination and the application should store the starting, ending time and the delay of the packets. An extra processing is required to get the estimated packet loss and delay for the specific algorithm and specific scenario (Single Path and Multi-Path).

The TclClass is a general purpose class and it is a public class that holds a information regarding the network objects, which are basically provided by the network simulation nodes and data type. TclClass corresponds to the base class for all library objects; it creates the Tcl object, and returns new class which is created by the Tcl base class. This construction satisfies the assignment requirements, and also promotes reuse. Note that TclObject is the object that holds a Tcl object reference, and creates an object of type Tcl as shown in figure 3.1. The sendmsg is virtual function which interacts with the object of network simulator. It then gives the information to Tcl.

The general purpose, simple common header foundation was already designed and implemented, and therefore reused. The actual construction of this part was not included in measurements of packet header information, as it was already available according to the hdr_cmn as shown in figure 3.1. Therefore, analysis of the packet header itself was not needed; but an analysis was required on how to use it. The common header file also stores the type of packet within the UDP packet, which is RTP in this case. It also stores the sequence number of the packet, the flag status and the time stamp of each packet. The packet type from specific node which connects to the Tcl object and then sends information to the Tcl object was analyzed as below:

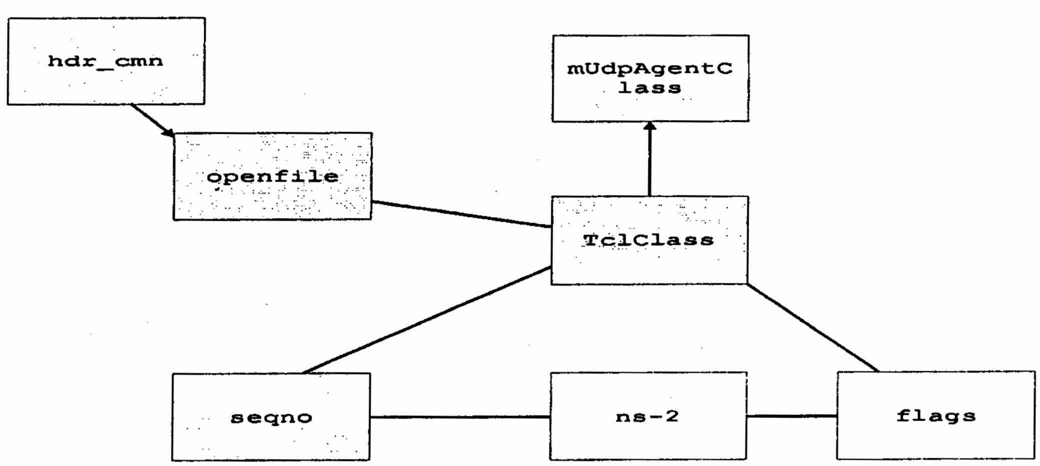


Fig 3.1 The Tcl Objects

The bind is a function that is use to bind the packet size of the UDP packet with the already feed size in the IP header file of the Tcl. The `mUdpAgentClass` is the base class for the calculation and analysis of the simulated packets.

The `mUdpAgentClass` is analyzed and described below in figure 3.2:

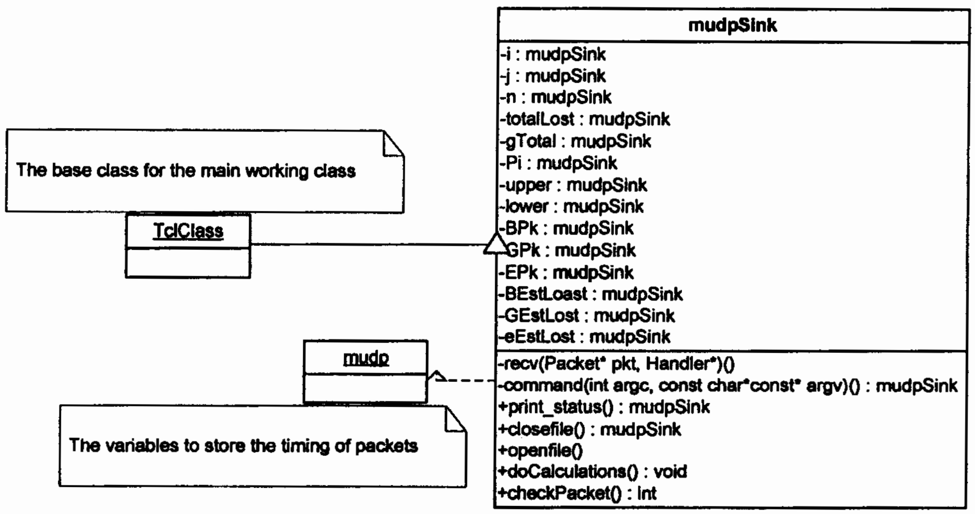


Fig 3.2 The Base Class Description

The system and the interaction of classes with the ns-2 simulated packets should be in a way that the corresponding object oriented file get the packet information whenever any node send the packet from the source to the destination. The information that needs to be store in the file for the further analysis and should be extracted from the header of the packet is the sending time that is the time when certain packet is originated from the source. The destination time needs to be stored in the file along with the round trip delay. These all information is added using the created object with the type File.

3.1.2 Basic Design

This is where the idealized base model from the base analysis is made more concrete. The design phase is divided into following phases and mentioned in figure 3.3:

i. System design, where decisions are taken about the qualities of the system as a whole, to a certain level of detail (variable according to taste).

ii. Object design, which is about adding details to the objects of the analysis phase.

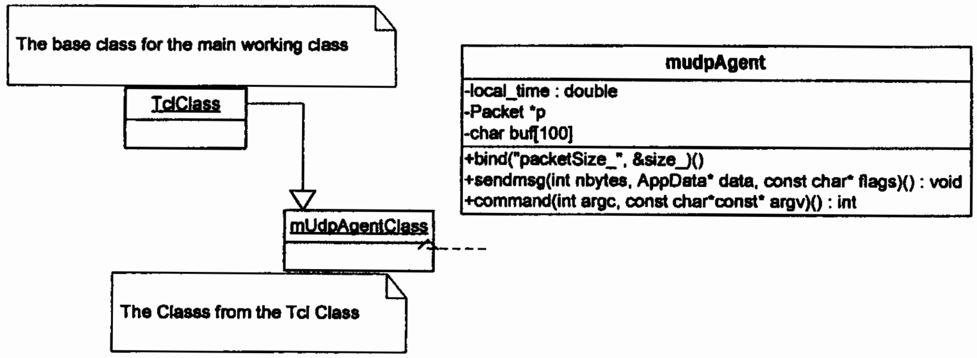


Fig 3.3 The mUdpAgent Class Description

3.1.2.1 Object design

The basic view of the packet modeling design is shown in figure 3.4:

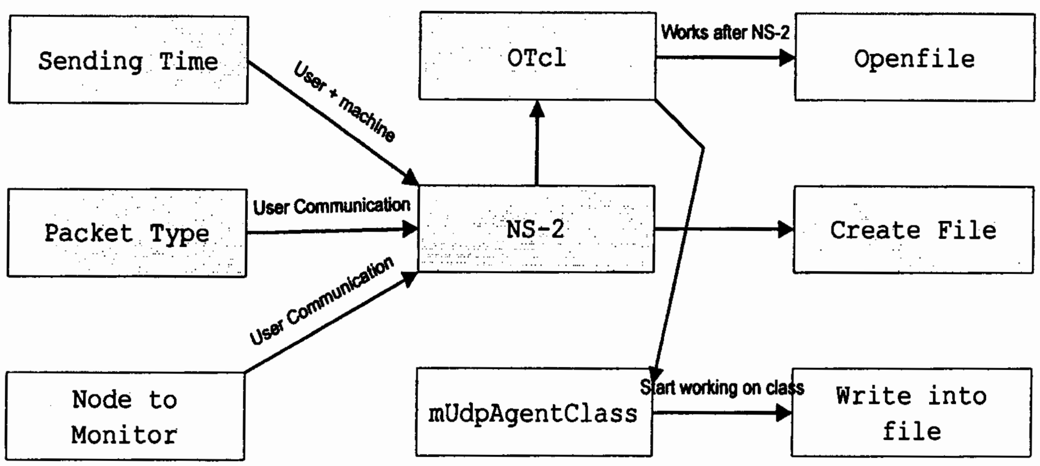


Fig 3.4 Analysis model of the Simulation

The rd is a packet information object which stores the data coming from the mUdp class. The storage of the packet information is in a file that holds the record of the packets that are sent over the network. The function of ns `set_filename` is use to first create the file and then get the information from the integrated file through openfile command for opening the file first. The file then stores all the events that packet generates during flow from source to destination. When the last packet is arrived or lost at the destination the file close itself using command closefile and we can view the results of the file accordingly.

3.1.2.2 System design

The application is divided into three sub-systems; the ns-2 simulated file, the UDP application, the Udp Sink application. The latter is merely to sink the udp calculation file with the network simulated packets, such as string to make connection, store the packet information in file, and the proper calculation of packet loss estimation using different algorithms.

3.2 Network Architecture

The network architecture consists of different network scenarios to get the better results for the estimation of packet loss and delay. It is mainly based on two types of network phenomenon.

- Single Path Phenomenon
- Multi-path Phenomenon

3.2.1 Analysis of Network for Single Path

The network should be in a way that it should give the best possible ways of getting the information for different kinds of networks. The network should be able to analyze the best case and the worst case that is needed to formalize the better algorithm for any kind of network architecture.

The source of the network should be able to generate TCP and UDP traffic. As the network mostly have data traffic and voice traffic most of the time. So the network should be designed having complete simulation of TCP and UDP packets from most of the nodes to the specific or non-specific destination. The routing of the packets must be static to formalize and get the complete path of the packet in observation. At the application layer for both TCP and UDP packets, we have to formulate certain application. It could be CBR for the UDP packets and FTP for the TCP packets. The network delay among the nodes needs to be set almost equivalent to the real network link delay. The capacity of node has to set for getting the results. The queues that are used in the links could be SFQ, the Stochastic Fair Queue or Drop tail queue. If we will use the stochastic fair queue, we will have an advantage that the stochastic fair queue treats every packet very fairly and then put or drops the packet in or out of the queue. While the drop tail queue does not treat every coming packet fairly.

Now during the packets communication from the source to the destination within different nodes and links, we will have all different types of the packets. But we need to monitor only the UDP packets that are traveling between specific source and destination. Because there will be both tcp and udp packets coming in and out of the network. It should be make sue that unnecessary packets will not be monitored as they will effect the results. Therefore, to make sure that only specific packets are monitored, the file which is created and storing all the records of the packets should only be created in the udp specified node.

3.2.2 Design of Network for Single Path

We designed a network which was based on static routing. The sources of the nodes were generating either TCP packets or TCP packets. The TCP based network was having FTP at the application layer, whereas the CBR was running at the application layer of the UDP based sources. The delay among the nodes was approximately identical to the real communication networks. The links were all having the capacity of E1 that is 2MB. The capacity of the links was capable enough of handling huge voice and data traffic.

The queues used in the links were SFQ, the Stochastic Fair Queue. The advantage of SFQ over DropTail Queuing is that it does fair queuing therefore there will be similar number of TCP and UDP dropped packets. The size of the packets of UDP is similar to the original voice packets. The packets were sent in random but with stable rate.

The file will be created which will only store the packets which are firstly the udp packets and they are traveling within the specific source and the destination. The file will now store the sending time of the packet and the receiving time of the packet at the destination. The udp type of the packet will be attached to the node, so that it will only send the udp packet. In the end, the starting and ending time of the packet will be set for every node, so that the node will start sending the packet at that specific time and stops it when the limit time for ending will come. The nodes having attached source type will also make sure that the rate of sending the packets will be according to the user specified time.

3.2.2.1 Network Example1

The design of the network is based on average type of network, comprises of data rate of the packets up to 1Mb. The packet size of this network is 1000 bytes. The total nodes in this network are 15. The total links among these nodes are also 15 and only 14 links are up. We have used only specified links just to make it similar to the real network environment, where few links are up and some of the links are down either administratively or physically. The delay among the links varies from 10ms to 30ms. The delay among the node varies on the basis of the load on specific link and the bandwidth available to handle certain amount of packet at the same time. The bandwidth is 2MB. The simulated packets are as below in figure 3.5:

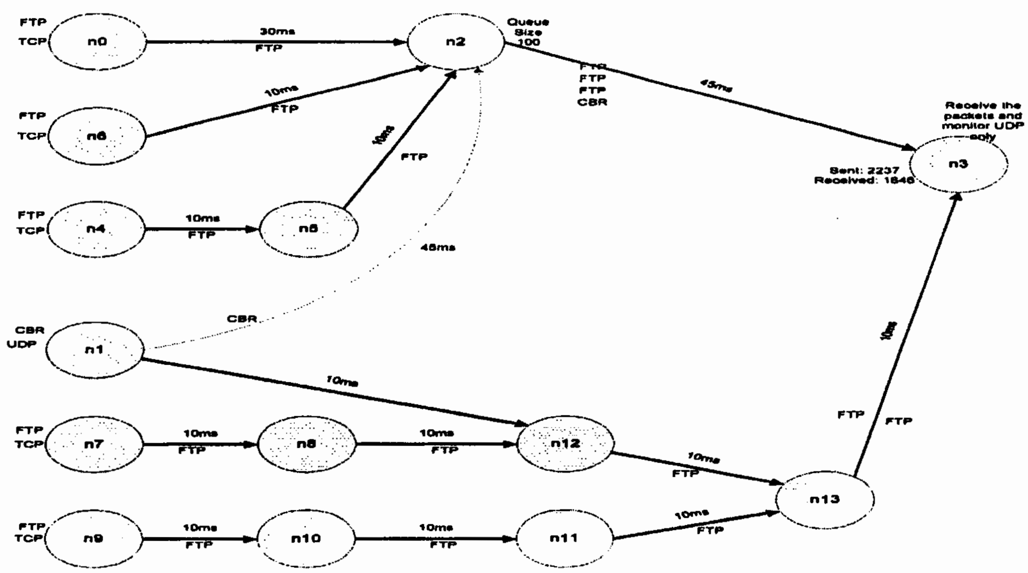


Fig 3.5 Network Architecture Design Data rate 1MB

3.2.2.2 Network Example2

The design of the network is based on very bursty type of network, comprises of data rate of the packets up to 2Mb. The packet size of this network is 1000 bytes. The total nodes in this network are 15. The total links among these nodes are also 15 and only 14 links are up. The limited numbers of links that will be used are to make it similar to the real network environment, where few links are up and some of the links are down either administratively or physically. The delay among the links varies from 10ms to 30ms. The delay among the node varies on the basis of the load on specific link and the bandwidth available to handle certain amount of packet at the same time. The bandwidth is 2MB. The simulated packets are shown in figure 3.6:

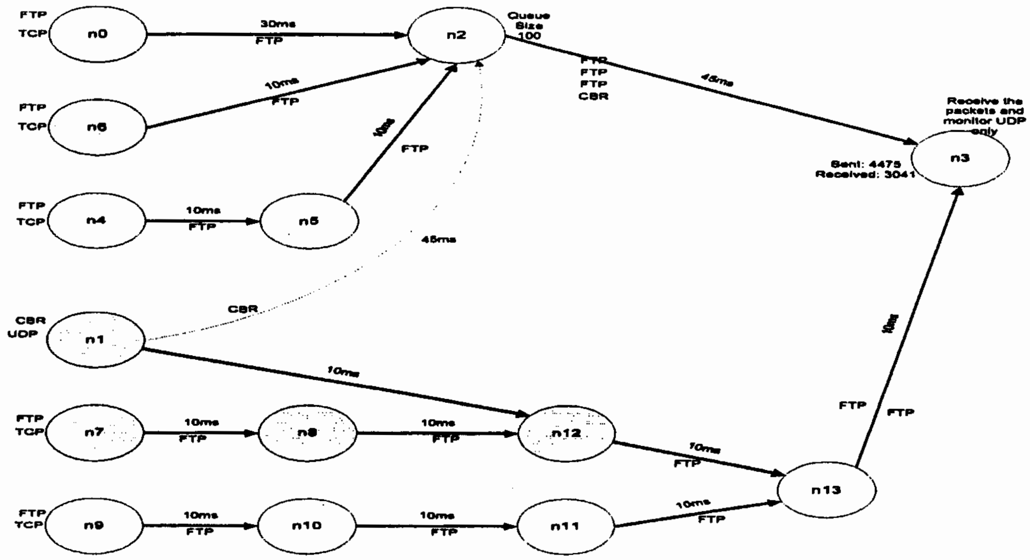


Fig 3.6 Network Architecture Design Data rate 2MB

3.2.3 Analysis of Network for Multi Path

The multi-path phenomenon or the multi-path diversity is the procedure to send the identical packets from two different streams. Therefore, utilizing the advantage of receiving identical packets from two different paths at same time. Therefore the network should also be designed in a way that there will be two different paths from the source to the destination for the specific voice communicating nodes. The identical voice packets stream from the specific node should start at the same time, so that the departure or the sending time of the identical packets should be same. The fid of the packet should be same as that of its identical packet.

The above analysis for the network architecture determines that if there will be no packet loss during the flow of packets from the source to the destination. Then the destination node will receive two packets with same fid, which means that the destination node will receive every next packet as identical packet of the last arrived packet. This needs to work out during the design of the network that the receiving node should never pass on the identical packets.

The network should be in away that the node which is going to be monitored for the udp packets need to generate the duplicate packets. The packets should follow two different and separate streams from the source to the destination to achieve the advantage of multi-path phenomenon. The network needs to be designed to have same sequence number for the duplicate packets, so that the destination node knows that the coming packets are the same packets.

3.2.4 Design of Network for Multi Path

The major change in designing the network architecture for the multi-path diversity over single path was that we designed a network in which there were two streams from the source to the destination for the specific voice communicating link. Now as soon as the network gets up it sends the flooding packets to exchange the information among the nodes. At the end of flooding all the nodes are having their routing table updated with the routing information to route the packets and if there will be congestion in the link or there will be some link failure the nodes will automatically find the other route to transport the packets to the destination. The sending time and the packet fid of duplicate packets are same. Therefore, if there will be no packet loss in the two streams that means the destination node receives every packet twice. So there is a need of discarding the packets which are already arrived from the other stream. When any packet is arrived to the destination, its unique id will be compared with all earlier received packets. If the id of the packet is matched with any of the earlier received packet's id that means it is already received and there is no need to receive that packet again.

The network should be in away that the node which is going to be monitored for the udp packets need to generate the duplicate packets. The packets should follow two different and separate streams from the source to the destination to achieve the advantage of multi-path phenomenon. To achieve the above mentioned statement, the fid of the packets will be same for both the sources coming from the same node. That means, the source node and the destination node will be one. But the source attached to the initializing node will be having double packets, having same data and header information.

The following code shows the function that is use for the packet recovery or duplication. It is the responsibility of this function that if the packet coming is duplicate packet and already received then discard it. If the coming packet is recovery of some earlier lost packet and its playout time is not expired then uses it for recovery.

```
//*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*//
//FUNCTION TO CHECK THE PKT RECIEVED FOR DUPLICATION OR
RECOVERY//
//*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*//
int mUdpSink :: checkPacket()
{
	int status = 1;// STATUS OF CURRENT PKT, 1 = ACCEPT THE PKT, 0 =
DISCARD THE PKT
	int status1 = 0;
	// IF NOT THE FIRST PKT
	if (i > 0)
	{	// CHECK THOUGH ALL THE BRUSTINESSES
		for (j = 1; j < 21; j++)
		{	// LOST PKT RECIEVED
			// TOO OLD PKT
			if (j == 1 && (tempID < que[j]))
			{	status = 0;	// DISCARD THE PKT
				break;		// TERMINATE THE CHECKING
PROCEDURE
			}
			// IF PKT ID ALREADY IN THE HISTORY QUEUE
			if (tempID == que[j])
			{		// PRINT THE STATUS TO THE OUTPUT
FILE
				fprintf(tFile, "Duplicate Packet: %d\n", tempID);
				status = 0;	// DISCARD THE PKT
				break;		// TERMINATE THE CHECKING
PROCEDURE			}
			// A LOST PKT HAS BEEN RECOVERED
			// IF THERE IS A DIFFERENCE OF MORE THAN ONE IN
HISTORY QUEUE PKT IDS
			// AND CURRENT PKT ID WAS LOST PREVIOUSLY
			if ( (j > 1) && (que[j]-1 != que[j-1]) && (tempID > que[j-1])
&& (tempID < que[j]) )
			{
				// INSERT THE RECOVERED PKT INTO THE
HISTORY QUEUE
				insertIntoQueue(tempID, j);
				// INCREMENT THE BRUSTNESS LENGTH
				pktLost[que[j-1]-que[j-2]-1] += 1;
				pktLost[que[j]-que[j-1]-1] += 1;
				status = 0;	// DISCARD THE PKT
				break;		// TERMINATE THE CHECKING
PROCEDURE

			}
```

```
            }
        }
        return status;                    // RETURN THE STATUS OF THE
RESPECTIVE PKT
}
```

3.2.4.1 Network Example1

The design of the network is based on average type of network, comprises of data rate of the packets up to 1Mb. The packet size of this network is 1000 bytes. The total nodes in this network are 15. The total links among these nodes are also 15 and only 14 links are up. The delay among the links varies from 10ms to 30ms. The bandwidth is 2MB. There are two paths for the communication of voice packets from the source to the destination for the communication of identical packet flow. The simulated packets as per above scenario are shown in figure 3.7:

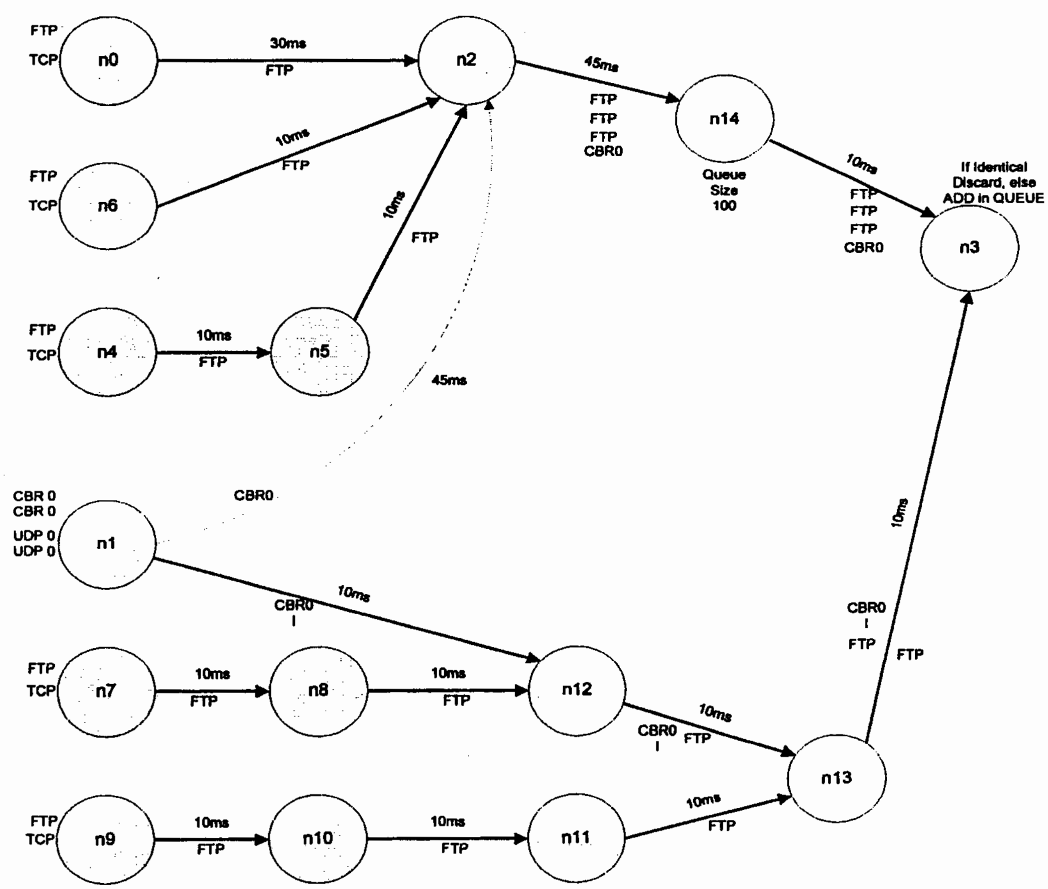


Fig 3.7 Network Architecture Design Example1

3.3 Modeling Algorithms

The application is divided into two phases and both the phases are further divided in the analysis of two different algorithms to find the better algorithm for the estimation of packet loss and delay.

3.3.1 Analysis of Bernoulli Algorithm

The basic classes to capture and store the sending time and receiving time are made very similar; there are no obvious differences on the surface. The general thought is to keep all basic classes modular, to the extreme in fact, not considering where the base class is to fit in. That is thought better handled using introduction, and composition of functionality through the weaving process.

As an example, the base classes' mudp and mudpsink are both created from the tcl base class `TclClass`. The agents in both the classes are made in a similar way. They are also tightly coupled to elements packet type, packet size and sequence number of the packet. This needs to be closed together because all the time capturing needs to be the same in both the files for specific period.

The next thing that needs to be done for the estimation of packet loss by the Bernoulli model is to capture the total number of packets that are received. The Bernoulli model needs to get the probability of the next packet lost for getting the estimation of packet losses in the future.

With Bernoulli Model, it is quite possible to get the estimation of packet loss very close to the actual loss of packet in the case of very bursty network. If the network is an average kind of network and the packet loss burst is not very lengthy then the Bernoulli packet loss estimation under-estimate the packet loss.

3.3.2 Design of Bernoulli Algorithm

The estimation of packet loss using Bernoulli model can be calculated by defining a procedure having different required variables. This implementation was based on sending the packets form source to destination for some specific period of time and then calculating total number of packets sent. Then, we start the calculation of actual number of lost packets within the specific stream and the specific packet type for the specified source and the destination. Furthermore, we need to calculate the exact probability of packet loss using Bernoulli model. The result of the probability leads us to the estimation of the packet loss using Bernoulli model of packet loss. Now we need to calculate the length of the burst for specific packet lost. The result of the packet loss estimation will be received by the calculation of Bernoulli probability and the total packets that are lost. Figure 3.8 shows the sequence of model flow for the estimation of packet loss.

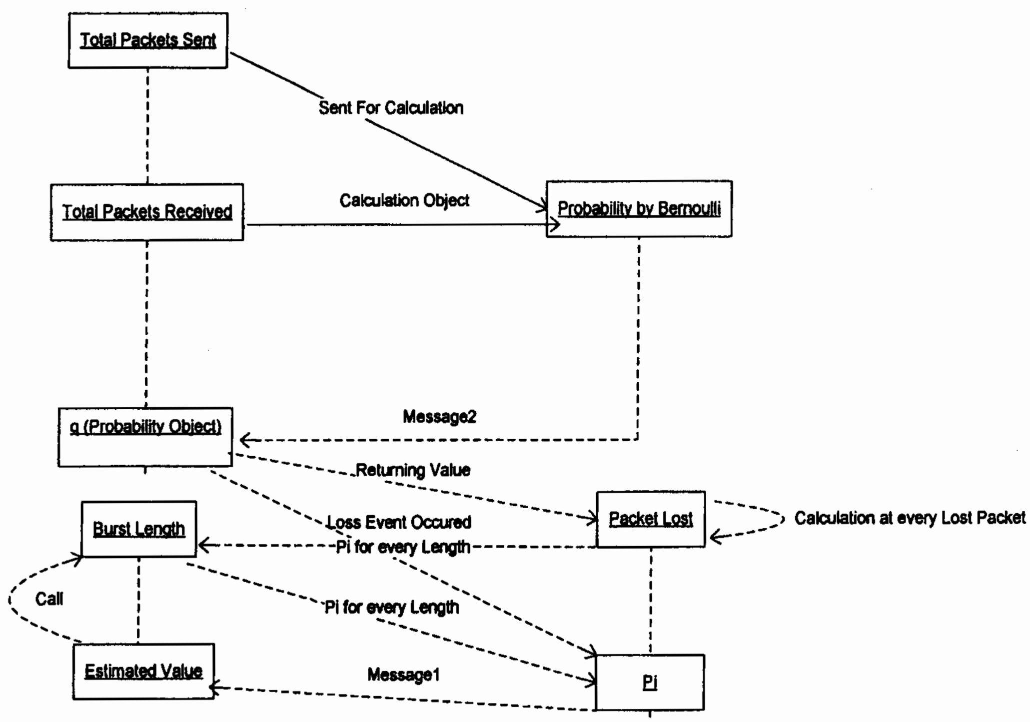


Fig 3.8 Sequence Diagram for Bernoulli Model

3.3.3 Analysis of Extended Gilbert Algorithm

The extended Gilbert model is normally the better model for the estimation of packet loss. The advantage of the extended Gilbert model over any other model is that it estimates the packet loss not on the basis of all the packets sent, or lost but it always estimates the packet loss on the bases of new probability for the upcoming packet.

Therefore every new coming packet has equal probability on the basis of past record. This gives extended Gilbert algorithm an edge over any other model.

For the calculation of packet loss using extended Gilbert model, we need to store the total packets sent and the total packets that are received on the destination. The loss of packet burst having length equal or less then three will be estimated using Gilbert model, the rest of the packet loss bursts will be treated using the extended Gilbert model of packet loss and delay.

3.3.4 Design of Extended Gilbert Algorithm

The estimation of packet loss using the extended Gilbert is calculated very differently as compare to the packet loss of any other packet loss model.

```
//*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*///
//    EXTENDED GILBERT MODEL IMPLEMENTATION          //
//*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*///
      upper = 0.0;
      lower = 0.0;
      for (j = 3; j < 21; j++)
      {
            upper += pktLost[j] * (j-1);
            lower += pktLost[j] * j;
      }
      q = 1 - ( (1.0 * upper) / (1.0 * lower) );
      for (j = 3; j < 21; j++)
      {
            gTotal = 0;
            for (int k = j; k < 21; k++)
            {
                  gTotal += pktLost[k];
            }
            pow = 1.0;
            for (int k = j; k > 1; k--)
                  pow *=  (1 - q);
            GPk[j] = (q * 1.0) * pow;
            gEstLost[j] = GPk[j] * gTotal;
      }
//*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*///
//    EXTENDED GILBERT MODEL IMPLEMENTATION END //
//*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*///
```

Fig 3.9 Extended Gilbert Model Code

In the figure 3.9, the calculation of packet loss starts with the calculation of the length of every burst. We have to calculate and find the number of times the single packet are lost. It needs to be store because the packet loss estimation will also be done for every burst length. After all the burst length and their frequency will be calculated, we have to calculate the value of pie. The next step is to find the probability of packet loss using the extended Gilbert model. The probability using extended Gilbert model for every burst length can be calculated by the value of already found probability multiply by the times of power for which the probability needs to be calculated. For example: If we need to find the estimation of packet loss for the packet loss burst of length three that means the power of the earlier found probability is three, which means the result is multiplied by itself for three times.

The specific estimation of packet loss now can be calculated by the resulted probability for packet loss. This is the estimated packet loss for the specific burst length of lost packets. The sequence diagram of Extended Gilbert model for the estimation of packet loss is shown in figure 3.10.

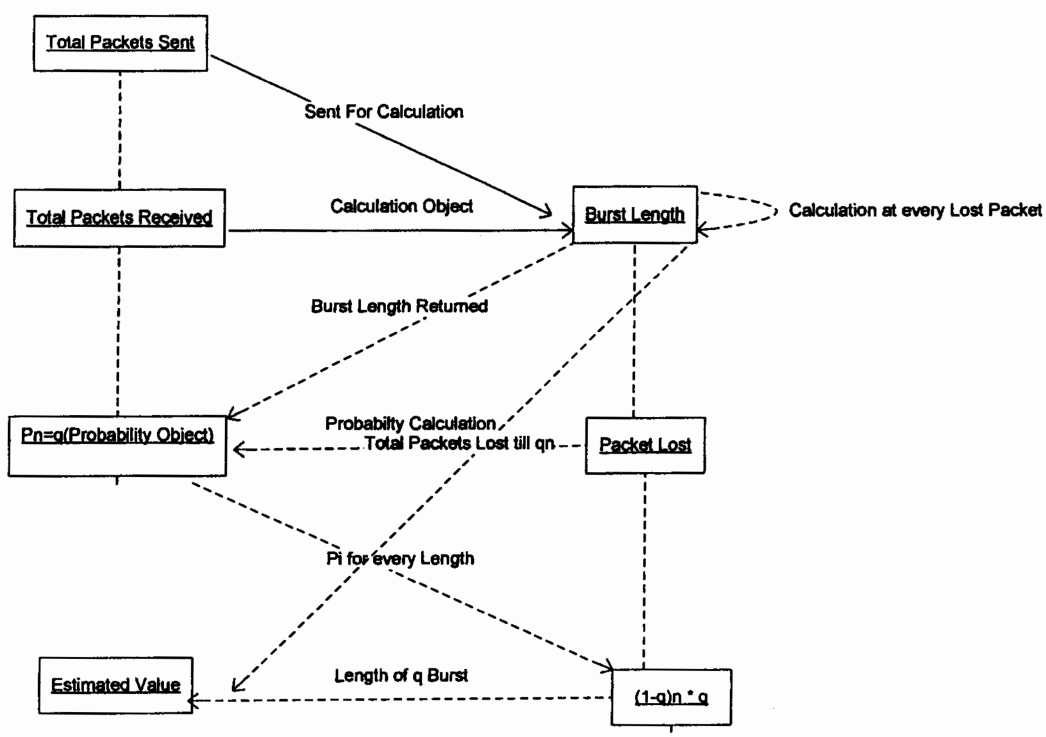


Fig 3.10 Sequence Diagram for Extended Gilbert Model

Chapter 4

Implementation

4.0 Implementation

The modeling of packet loss and delay using multi-path diversity was implemented on simulation using Network Simulator (ns-2.27) which provides following benefits simulating wireless simulations

4.1 Ns-2 and Wireless Simulations

Ns-2 provides a framework for simulation of wired and wireless networks, including some facility for emulation. The ns-2 simulator is written in C++ with a Tcl shell front-end that uses oTcl (object-oriented Tcl) libraries Scenarios are run by feeding an oTcl script to the ns-2 executable. The output can be read directly or post-processed by an interactive graphics viewer called NAM. NAM does not allow changing parameters on the fly; it is for post-viewing of a simulation dump (a .nam file).

As of this writing the graphics viewer NAM is not advertised to work with wireless simulations, but there is apparently work-arounds1. Generally ns-2 has a different architecture for wireless and wired node simulation. This report will only examine wireless and wired simulations, the code can be adapted for mixed wired/wireless environments. A community of users and designers has grown around the ns-2 software. The website has pointers to the latest code and documentation, and additional modules submitted by the developer community. There are several mailing lists for users and developers, and these are conveniently archived on the site. The design section will delineate the specifications for the Bernoulli and Extended Gilbert algorithms, and describe how design decisions were made. The implementation section walks through the functionality of the code. The conclusion should tell what this module looks like, what it can be used for, and the scope of its utility.

4.2 Path Routing in NS-2

The implementation of the path routing in the NS-2 provides both static and dynamic routing. If the route is selected as the static route with in the specific network topology then it means that the packets are going to flow through this specific route and they will not change their path even if certain link is down within the traveling path. In this scenario the packet loss may be huge because if there will be any congestion or link failure the packets cannot change their path. But the advantage of the static path is that they don't need to use any dynamic algorithm to reach the destination.

The dynamic routing in the NS-2 gives support in the lesser number of packet loss and even reduces the packet delay in reaching the destination. We have used the combination of dynamic routing in the multi-path diversity along with the static routing to set the packets to move from the specific source to the specified destination because the multi-path diversity phenomenon needs dynamic routing at the top to generate duplicate packets from certain source to the destination using different streams.

4.3 Modeling of Packet Loss

Modeling of the packet loss has been implemented using two better algorithms that are Bernoulli model and Extended Gilbert Model. These are implemented on the single path and better one was planned to be used for the multi-path diversity. But the results from the single path using very complex network gave us a chance to use both the models for the multi-path diversity. The results from the single path for the modeling of packet loss using the Bernoulli model and Extended Gilbert Model proves that both the algorithms are equally beneficial for the modeling the packet loss in different cases. If the network is too bursty and the number of lengthy packets loss is high then the number of single packet loss then Bernoulli estimates it better because it always over estimates the packet loss for double packet loss and above that, it also under estimate the single packet loss.

In most of the networks, the single packet losses are the most loss packets and then the double packets and so on. Specifically, in the case of multi-path diversity phenomenon, the number of single packet loss is very high, therefore Extended Gilbert model estimates it better.

4.4 System Requirements

This project was developed under Linux Fedora Core 2. The ns-2 version was "ns-allinone-2.27", which is a single tar ball with all the requisite packages that easily installs with one command. The link to the package of C++ based coding is done with the link creation between C++ module and ns-2.

4.5 Design of the Packet Loss Modeling for ns-2

The network design is based on static routing. The sources of the nodes generate either TCP packets or UDP packets to the destination. The TCP packets generating node is having FTP at the application layer, whereas the CBR is running at the application layer of the UDP generating node. The delay among the nodes is same as that of the actual voice and data networks. The bandwidth capacity of the links is either 1MB or 2MB on the basis of requirement. The capacity of the links is capable enough to handle massive voice and data traffic.

The queues used in the links are SFQ, the Stochastic Fair Queue. The advantage of SFQ over DropTail Queuing is that it does fair queuing. Therefore, the dropping of TCP and UDP packets will be same. The size of the packets of UDP is similar to the original voice packets. The sample network design and packet structure and flow are shown in figure 4.1.

Firstly, we need the implementation of packet loss modeling using Bernoulli and Extended Gilbert Model in single path environment. The implementation is based on sending the packets form source to destination for specific period of time and calculating total number of voice packets that were sent over the network in single stream. The next step is to calculate the sending and receiving time of every packet. This information is used to calculate total delay each packet incur to arrive at the destination. The calculated delay is the single side delay from source to the destination. The next target was the calculation of actual number of lost packets along with the calculation of the exact probability of packet loss using Bernoulli model. The result of the Bernoulli probability leads to the estimation of the packet loss using Bernoulli model. Repeating the above steps results in the calculation of probability by using Extended Gilbert model and the estimation of packet loss using Extended Gilbert Model.

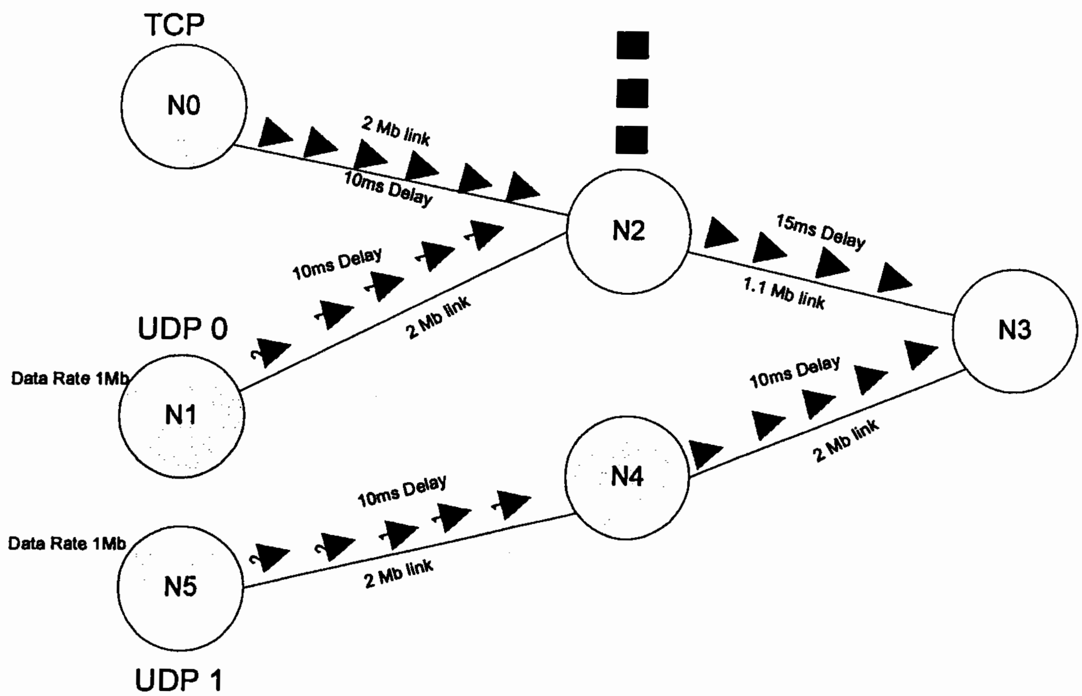


Fig 4.1 Network Diagram of Single-path

The results clearly shows that Bernoulli model over estimates the single packet loss and under estimates the loss of lengthy bursts of packets, which means that if the length of packet loss is greater then equals to two packets then it normally under estimates the packet loss. That creates terrible probability calculation of packet loss and leads to the incorrect calculation in estimating packet loss using Extended Gilbert Model.

The Extended Gilbert model on the other hand, estimated the single loss very close to the actual calculated loss. The results showed us that the extended Gilbert model estimates the packet loss very close to actual loss when there is single packet loss and almost calculated equally right when there are lengthy bursts of packet losses.

So using the results of above mentioned scenario, we took the Extended Gilbert model and implement it in multi-path diversity mechanism. The Multi-path Diversity is basically based on sending the same packets from two different streams.

The advantage of Multi-path is that if there is a loss of packets from one stream then there is fair chance of receiving those packets from the other stream. Therefore, there will be less probability of losing packets in burst and the length of burst of lost packets will be very small as well. The probability of estimating the packet loss to the actual loss will also be well estimated. As these algorithms finds the estimation very accurate when the length of loss packets is diminutive.

So we made two streams of voice traffic (Figure 4.2) having same packets coming out of them. The source and the destination were same but the path of the streams was different. In this regard the major advantage was that there was positive chance that the packet will reach the destination through at least one of the paths. Although the disadvantage of multi-path is that sometimes it needs some extra time and bandwidth to discard and receive the packets respectively but still it is better to have little delay and cost of more bandwidth then losing the packets or resending the packets from the start.

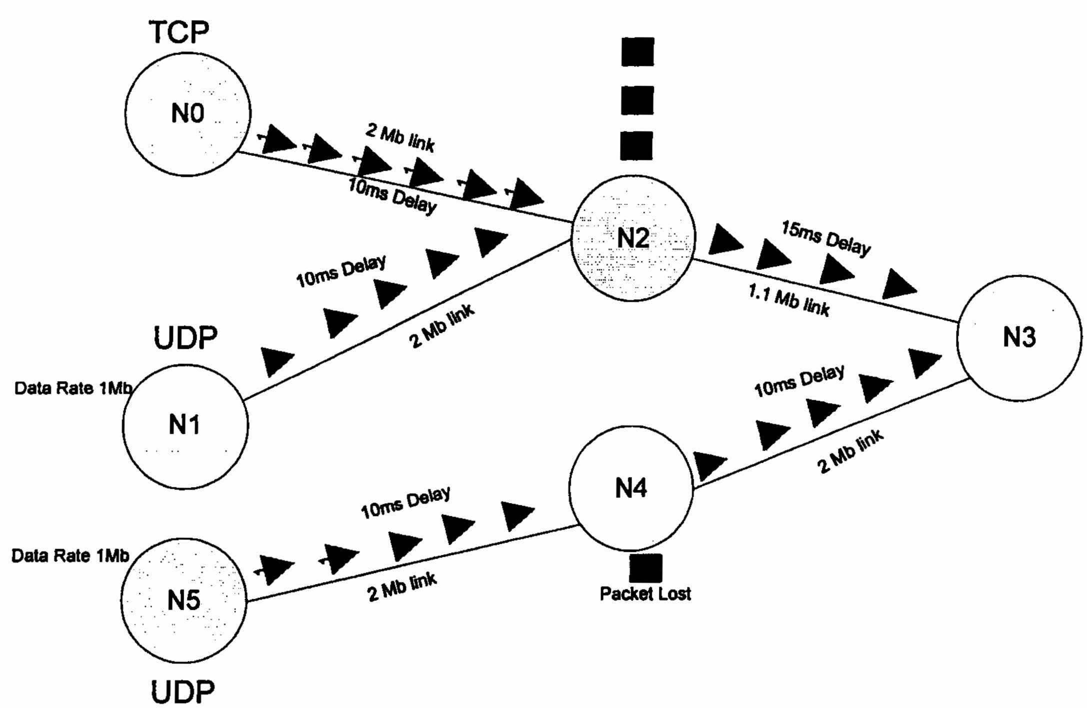


Fig. 4.2 Network Diagram of Multi-Path Diversity

4.6 Overview of the code

The code consists of C++ code which was converted into objects files that need to be used in the tool command language. That can be simulated in the Ns-2.

The combination of C++ files consists of headers and .CC files as NS-2 supports .CC files not the .C or .CPP files. The main headers files are listed below with some of its description

Some of header files (.h) are listed below

- agent.h
- stdio.h
- udp.h
- rtp.h
- mudp.h
- address.h
- ip.h

Some of .cc files are listed below

- mudpsink.cc
- mudp.cc
- packet.cc
- ip.cc

4.6.1 agent.h

This header file (Shown in figure 4.3) is responsible for implementing the source type at every node. It selects the source type at application layer that could be FTP for TCP and CBR for UDP. It also includes dependency graph which are used during the implementation of resulting graphs.

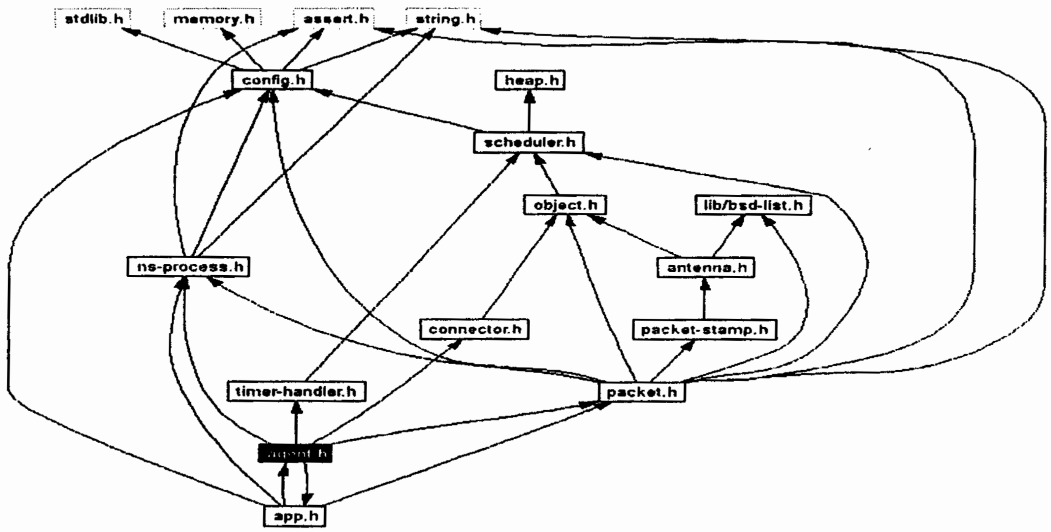


Fig. 4.3 agent.h

4.6.2 udp.h

This header file (Shown in figure 4.4) is responsible for implementing the udp packet at transport layer. This protocol stores and keeps track of all udp packets from source to destination.

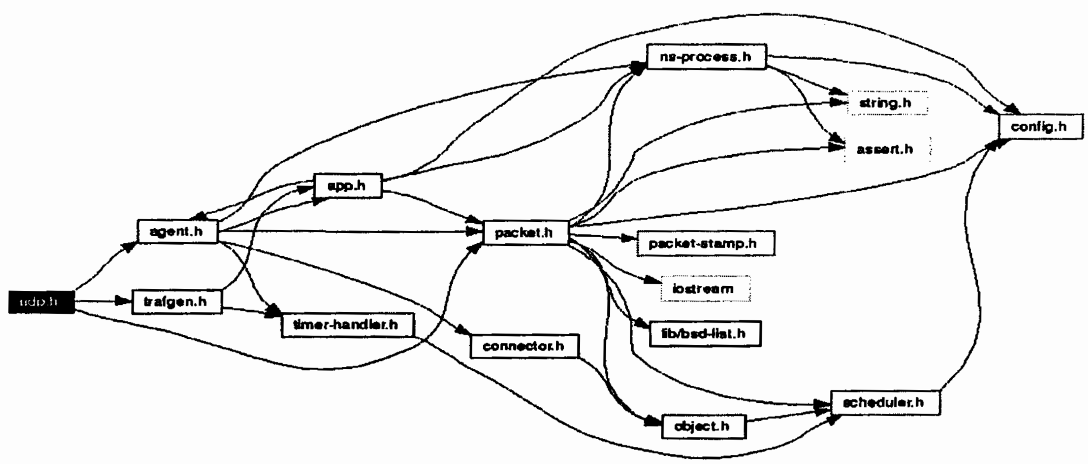


Fig. 4.4 udp.h

4.6.3 rtp.h

This header file (Shown in figure 4.5) is responsible for implementing RTP which is inside UDP. It is use for the voice communication over IP.

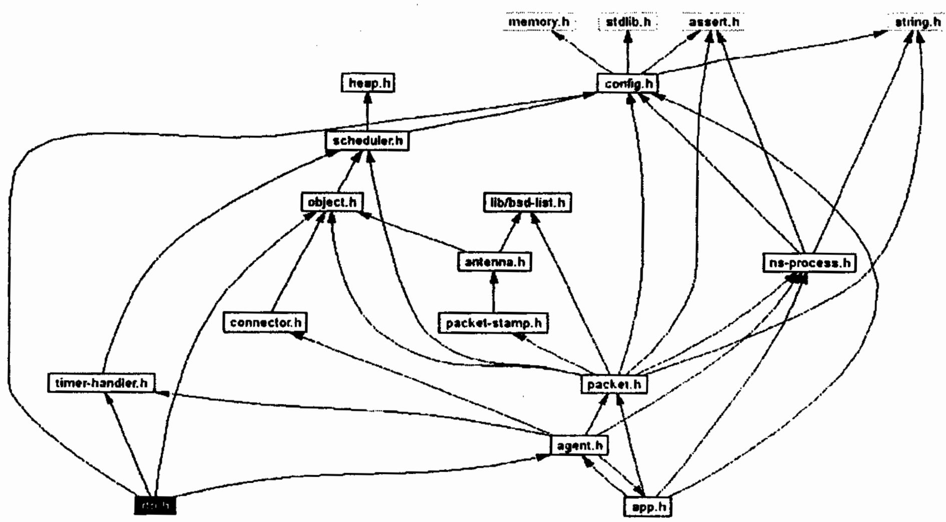


Fig. 4.5 rtp.h

4.6.4 mudp.h

This header file is responsible for implementing the modeling of packet loss and stores all the header files required for the implementation.

4.6.5 address.h

This header file is responsible for storing the IP addresses of the nodes or the devices.

4.6.6 ip.h

This header file is responsible for implementing the ip for the routing of the packets, in which the routing protocols are defined.

4.7 Tcl Code

The tcl implementation of modeling of packet loss and delay is given as follows.

4.7.1 Single Path

The single path implementation for the estimation of packet loss and delay for getting better model is comprised of two phases. The first phase is the introduction of

Bernoulli model for the modeling of packet loss to judge that whether Bernoulli model estimates the packet loss close to the actual loss of the packets. The second phase is the implementation of the above mentioned procedure using the extended Gilbert model.

The network implementation is based on different network scenarios to get the better model in each case. So the resulted model should be efficient in the estimation of packet loss of average based network having lesser packet loss and the network should be equally efficient in the estimation of packet loss in bursty networks.

We first check the total number of packets that are sent to the destination. When the last packets are received to the destination, the modeling algorithm starts estimating the packet loss and the loss probability using the Bernoulli model. The formula and the patch of specific implementation of this model for the packet loss is illustrated below in figure 4.6 :

```
if (id[i] == 4475)
{
        int n = id[i]+1;  // TOTAL PKTS RCVD
        // PROBABILITY
        P = totalLost / ((n * 1.0) - totalLost);
        // CALCULATE q
        for (j = 1; j < 21; j++)
        {
                upper += pktLost[j] * (j-1);
                lower += pktLost[j] * j;
        }
        q = 1 - ( (1.0 * upper) / (1.0 * lower) );
        // CALCULATE Pi
        Pi = (1.0 * P) / ( (P * 1.0) + (q * 1.0) );
        for (j = 1; j < 21; j++)
        {
                for (int k = j; k > 1; k--)
                        pow *= Pi;
                BPk[j] = (1 - Pi) * pow;
                bEstLost[j] = BPk[j] * totalLost;
                pow = 1.0;
        }
```

Fig. 4.6 Code for Bernoulli for Packet Loss

The second phase starts with the implementation of Gilbert model and the extended Gilbert model (As shown in Figure 4.7)for the probability of packet loss and the estimation of total number of single, double and other consecutive packet losses.

```
for (j = 1; j < 3; j++)
```

```
{
      gTotal += pktLost[j];
}
upper = 0.0;
lower = 0.0;
for (j = 1; j < 3; j++)
{
      upper += pktLost[j] * (j-1);
      lower += pktLost[j] * j;
}
q = 1 - ( (1.0 * upper) / (1.0 * lower) );
for (j = 1; j < 3; j++)
{
      pow = 1.0;
      for (int k = j; k > 1; k--)
            pow *=  (1 - q);
      GPk[j] = (q * 1.0) * pow;
      gEstLost[j] = GPk[j] * gTotal;
}

upper = 0.0;
lower = 0.0;
for (j = 3; j < 21; j++)
{
      upper += pktLost[j] * (j-1);
      lower += pktLost[j] * j;
}
q = 1 - ( (1.0 * upper) / (1.0 * lower) );
for (j = 3; j < 21; j++)
{
      gTotal = 0;
      for (int k = j; k < 21; k++)
      {
            gTotal += pktLost[k];
      }
      pow = 1.0;
      for (int k = j; k > 1; k--)
            pow *=  (1 - q);
      GPk[j] = (q * 1.0) * pow;
      gEstLost[j] = GPk[j] * gTotal;
}
```

Fig. 4.7 Code of Packet Loss using Gilbert Model

4.7.2 Multi-Path

The major change in the multi-path diversity phenomenon as compare to the single path architecture is the network simulation of the packets. The network simulation and its implementation start with the code "Node set multiPath_ 1". This specific line is use to send the duplicate packets from two different paths. Now when we are going to send the duplicate packets, this line of code use different streams (if available) to send the packets to the destination. The implementation of sending the duplicate packets from diversified paths is dependent on the fid of the packet. If the fid of the packet is same then the sent packets from the source are treated as similar packets, hence two packets are always sent from the source to the specified destination over the network having similar information.

The important implementation starts when the packet reaches the destination. Therefore we are implemented the code which can detects the arrival of duplicate packets at the destination and if it happens the destination node has to do an extra functionality in discarding the second received duplicate packet from the other stream. If the single packet is received and the other was lost then the packet is placed in the buffer and played out before the play-out time is expired. The destination node also have to keep an eye on the packet which was lost from one stream and received to the destination through other stream with late arrival but before its play out time expiration to place in the playing buffer and play it at its time, hence reducing the packet loss of the packets.

4.7.3 packet.cc

In this specific file (Shown in Figure 4.8), the size of the packet's header is set which will be added in the actual data size of the packet. The static offset of common header size and the static offset of flag header are set in this code file.

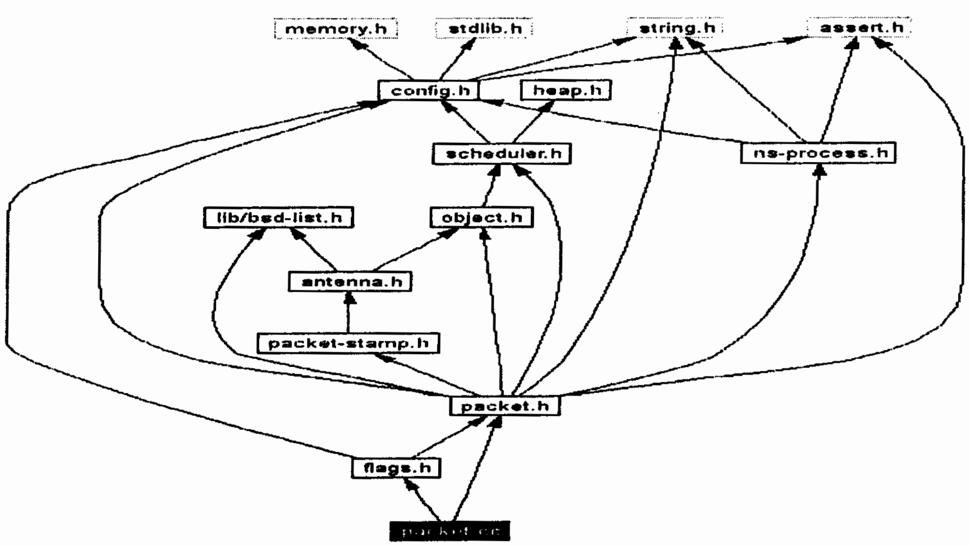


Fig. 4.8 Packet.cc

4.7.4 ip.cc

In this code file, the multicast and time to live of the packet is set. In this source file whose flow of communication is mentioned in figure 4.9, the type of the packet that whether it will joint some multicast or not is also set. This class file takes the relevant data from the ip header.

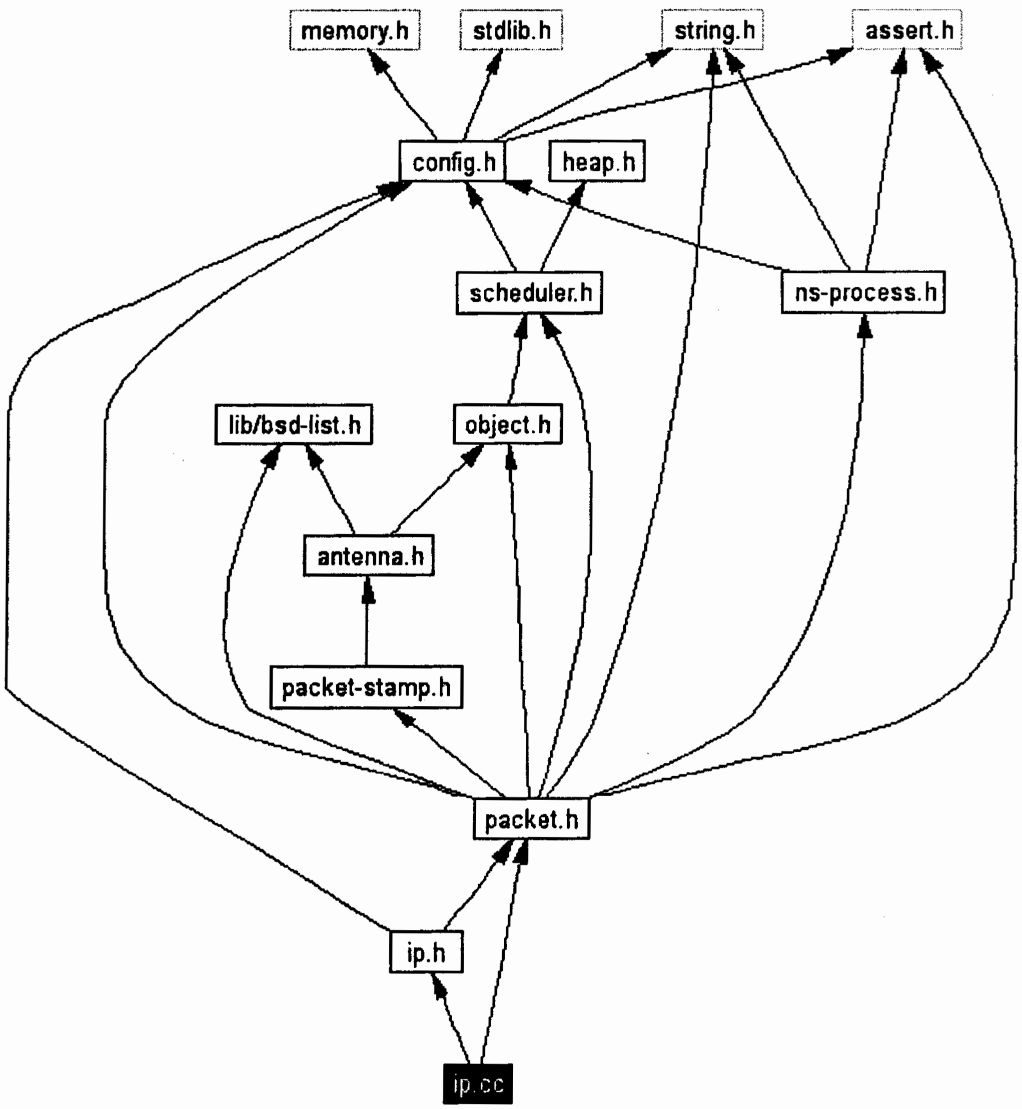


Fig. 4.9 ip.cc

Chapter 5

Testing

5.0 Testing

Testing phase is the most phase in the of the software development cycle, it sorts out the bugs in the software and the negatives of the software that is why the software's being developed are tested taking double as time required by implementation. There are different techniques with which software can be tested but the major are described as under.

5.1 Black Box Testing

Test type: *Black Box Testing* **Developed by :** Asad **Description:** In black box testing, the user interface is exercised over a full range of inputs and the corresponding outputs are observed for correctness. **Environment:** ns-2.27 running under Linux Fedora core 2
Steps Performed for Test 1. Specified the links of the nodes 2. Specified the data rate of the network 3. The tcl file was compiled which successfully generated the trace file 4. The user opens the trace file. 5. The result can be viewed by plotting the graph out of values obtained after simulation.
Result : ☑ Pass ☐ Fail

5.2 White Box Testing

Test type: *White Box Testing* **Developed by :** Asad

Description: This testing technique do structure analysis, branch coverage, functional coverage, boundary conditions checking, and input generation are used and run against the static code.

Environment: ns-2.27 running under Fedora core2

Steps Performed for Test

We specifies the number of node for a large network with a very small bandwidth and checked the results it was very congested

Result :

☑ Pass ☐ Fail

5.3 Unit Testing

Test type: *Unit Testing* **Developed by** : Asad

Description: In unit testing, different modules of the developed system are tested independently. The purpose is to determine that each module is functioning properly and to locate errors in the modules.

Environment: ns-2.27 running under Fedora core2

Steps Performed for Test

We have done the unit test of each individual unit. An example test of module function parsing is given below,

- As modeling of packet loss is a combination of different modeling algorithms we implemented tested the individual protocols and got the correct results.

Result :

☑ Pass ☐ Fail

Chapter 6

Results & Conclusion

6.0 Results

The major advantage of Extended Gilbert model over Bernoulli model results is that the probability distribution of every packet depends upon only the **n** consecutive loss packets, whereas in the Bernoulli model of probability distribution the future probability loss of the packet is dependent on all **n** packets. Therefore, the probability in Extended Gilbert model is well calculated to the actual loss of the packets as compare to the Extended Gilbert model [7].

6.1 Packet Loss measurements

The implementation of packet loss modeling in single path and multi-path using Bernoulli and Extended Gilbert model ensures that the quality of service can be extendable and the voice quality over the packet network can get better and it can be estimate able better. The measurement has also shown that if the voice packets are sent over the network from two different streams as redundant packets, they have very positive chance to reach the destination. The simulation has proved with the results shown in Table 6.1 that if certain packet is lost under the multi-path diversity phenomenon, it mostly reaches the destination through other voice stream, utilizing the basic advantage of multi-path diversity.

Table 6.1 Single Path Network Structure

Trace	Data Rate	Packet Size	Total nodes	Total Links	Total Links UP	Delay	Band width	Packets Sent	Packets Received	Packets Lost
1	1mb	1K	15	15	14	30ms/10ms	2mb	2237	1646	591
2	1mb	1K	15	15	15	30ms/10ms	2mb	2237	1605	632
3	1mb	1K	15	15	15	30ms/10ms	1.5mb	2237	1572	665
4	1mb	1K	15	15	15	30ms/10ms	1.1mb	2237	1690	547
5	2mb	1K	15	15	14	30ms/10ms	2mb	4475	3041	1434

All the measurements from the traces are having either 1mb or 2mb data rate with packet size 1k. The results from the traces differ on the basis of bandwidth within the links. The numbers of packets that are sent to the destination are also dependent upon the number of links up for the packets and the bandwidth available at each node.

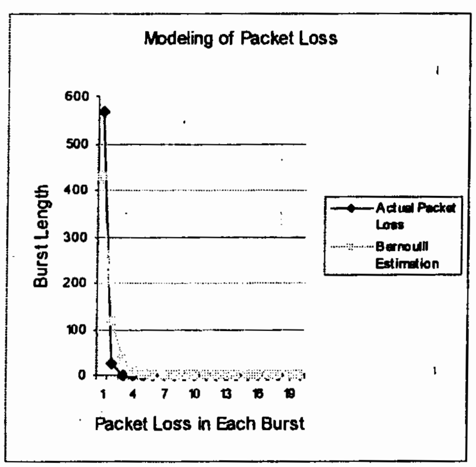
Table 6.2 Multi-Path Network Structure

Trace	Data Rate	Packet Size	Total nodes	Total Links	Total Links UP	Delay	Band width	Packets Sent	Packets Received	Packets Lost
1	1mb	1K	15	15	14	30ms/10ms	2mb	2237	2115	122
2	1mb	1K	15	15	15	30ms/10ms	2mb	2237	2041	196
3	1mb	1K	15	15	15	30ms/10ms	1.5mb	2237	1973	264
4	1mb	1K	15	15	15	30ms/10ms	1.1mb	2237	1865	372
5	2mb	1K	15	15	15	30ms/10ms	2mb	4475	3158	1317
6	2mb	1K	15	15	14	30ms/10ms	2mb	4475	3211	1264

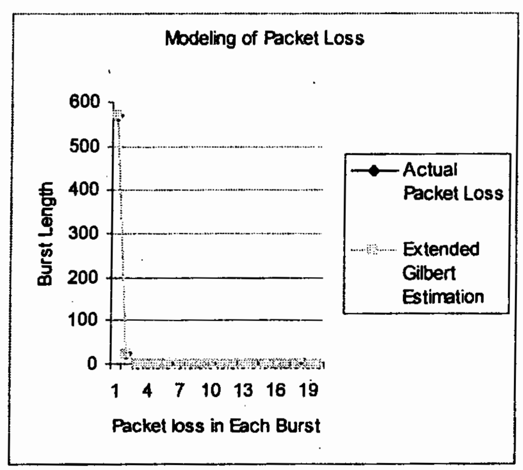
All the measurements from the traces are having either 1mb or 2mb data rate with packet size 1k, which is also shown in Table 6.2. The results from the traces differ on the basis of bandwidth within the links. The numbers of packets that are sent to the destination are also dependent upon the number of links up for the packets and the bandwidth available at each node.

6.2 Determining the Packet Loss Model

The under mentioned graphs prove that if the total packets lost in each burst are having length of one or two packets then Extended Gilbert Model accurately estimates the packet loss to the actual loss of the packets. The Extended Gilbert model estimates the packet loss every time with a probability of collective consecutive lost packets. Therefore it estimates very close to the actual loss of the packets.



Graph 6.1 Bernoulli Loss Model in Single Path (Trace 1)



Graph 6.2 Extended Gilbert Loss Model in Single Path (Trace 1)

On the other hand, the Bernoulli loss model over estimates the number of packet loss in every burst, therefore gives results far away from the actual loss of packets. We have also seen after the results that if the network is bursty network and having packet loss of lengthy bursts more then the short length bursts then Bernoulli calculates the packet loss better the extended Gilbert model. The only reason gain is that the Bernoulli model over estimates the loss of the packets hence luckily gives better results then the extended Gilbert model. The results regarding the comparison are also mentioned in Graph 6.1 and Graph 6.2.

6.3 Comparison of Single and Multi-path

It is seen with the results that multi-path give better results as compare to the single path phenomenon. The multi-path diversity provides flexibility to the packets to reach the destination in time. The packet loss in s single path are always more then the multi-path. Also the length of packet loss burst is short in case of multi-path diversity as compare to the single path phenomenon. Another advantage of multi-path diversity over the single path is that there is small number of lengthy burst during the flow of packets from the source to the destination.

Multi-path diversity provides an extra edge to the packets for the recovery. The efficiency of the forward error correction can be better with the combined use of multi-path. Because in forward error correction the recovery of the lost packet is highly dependent of the next coming packet and if the next packet is lost then the recovery of two continuous lost packets are depending on the next arriving packets. Therefore if the network is very bursty and the packets are lost in lengthy bursts then the recovery of the packets is very difficult. The multi-path diversity provides mechanism to the forward error correction for having very short length of the bursts. So if the packet is lost form one stream, it reaches the destination through the second stream. Hence the forward error correction receives the packet and finds the way to recover the earlier lost packet within the network.

The results have proved the advantage of modeling the packet loss with multi-path diversity phenomenon over the single path. But there is one major disadvantage of multi-path over the single path but it is negligible. The destination side has to work more to check if the coming packet is already received or not, if it is received then the destination has to discard the packet. If the coming packet is the recovery of already lost packet, then destination has to reassemble it before forwarding to the listener. Still it is better then losing the entire packets. The recovery mentioned in the above sentence is not by using the forward error correction, but it is the recovery of lost packets by receiving it from the other stream by utilizing the feature of multi-path diversity phenomenon.

Table 6.3 Comparison of Loss models using Trace 1

Length of Loss Burst	Actual Packet Loss (Single Path)	Estimated by Bernoulli Model (Single Path)	Estimated by Extended Gilbert Model (Single Path)	Actual Packet Loss (Multi-Path)	Estimated by Bernoulli Model (Multi- Path)	Estimated by Extended Gilbert Model (Multi-Path)
0	1646	1646	1646	2115	1996	2115
1	567	430.3	567.9	121	215.64	121
2	24	116.9	22.1	1	23.4	0.9
3	0	31.8	0	0	2.5	0
4	0	8.6	0	0	0.03	0
5	0	2.3	0	0	0.003	0
6	0	0.6	0	0	0	0
7	0	0.17	0	0	0	0

The mentioned comparison table in Table 6.3 shows that the multi-path diversity provides better quality of service, as it allows more packets to reach the destination. The simulation proves that using multi-path phenomenon provides packets more options to reach the destination, hence less number of packet loss. If we look at the table 6.2, the frequency of double packet loss really reduced by the introduction of multi-path diversity phenomenon. The table also verifies that the estimation of packet loss by the extended Gilbert model is better then the Bernoulli model on most of the instances. For further analysis and clarification for the use of Bernoulli model and extended Gilbert model, the following table results are illustrated.

Table 6.4 Comparison of Loss models using Trace 3

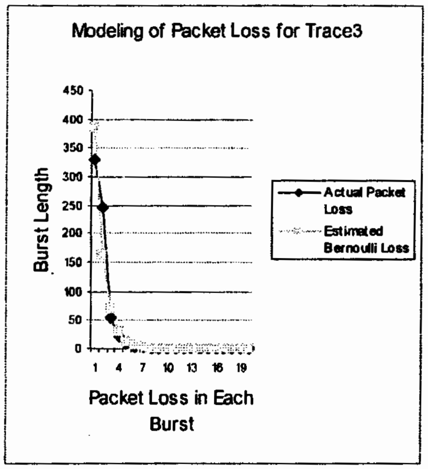
Length of Loss Burst	Actual Packet Loss (Single Path)	Estimated by Bernoulli Model (Single Path)	Estimated by Extended Gilbert Model (Single Path)	Actual Packet Loss (Multi-Path)	Estimated by Bernoulli Model (Multi- Path)	Estimated by Extended Gilbert Model (Multi-Path)
0	1572	1572	1572	1973	1973	1973
1	330	385	404	249	223.7	249.8
2	247	162	121	15	26.6	13.4
3	54	68	12	0	3.1	0
4	20	28.6	3	0	0.3	0
5	10	12	1	0	0	0
6	2	5	0.21	0	0	0
7	2	2	0.07	0	0	0

In the above mentioned tables i.e. Table 6.3 and Table 6.4, the length of the lost burst describes the actual length of the lost packets. If the length of the lost burst is zero then it describes the total number of packets that are received at the destination.

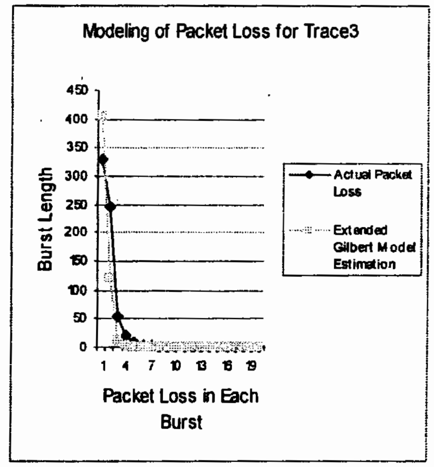
6.4 Comparison of Modeling Algorithms

We graded the modeling algorithms according to the performance measures. We have sent the packet from source to the destination and calculate the accuracy of the models for both single and multi-path diversity phenomenon.

The mentioned tables and graphs give very interesting results regarding the loss models. The results of the research prove that Bernoulli model also provides effective results as Extended Gilbert do, but for some specific Scenarios. The analysis of Table 6.2 and Table 6.3 explains that if the actual burst of packet loss is very lengthy, each burst includes massive frequency of packet losses and comparatively limited number of single packet loss, then Bernoulli Loss model can give better results as compare to Extended Gilbert Model, because it always over estimates the loss of packets, hence close down the estimation of packet loss to the actual packet loss. So we have used both the models for the estimation of packet loss in multi-path diversity as well.



Graph 6.3 Bernoulli Loss Model in Single path (Trace 3)



Graph 6.4 Extended Gilbert Loss Model in Single path (Trace 3)

6.5 CONCLUSION

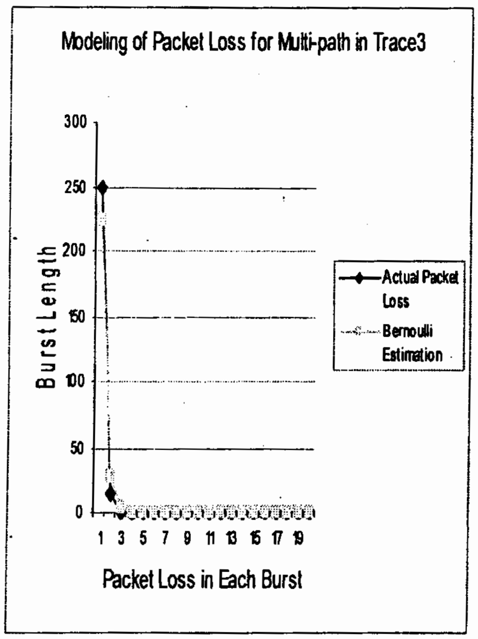
The above mentioned results have also shown that the implementation of multi-path diversity has given better quality of service because we have very limited number of actual packet loss. Therefore, the estimation of packet loss using Extended Gilbert model is very near to the real packet loss. The extended Gilbert model is better option for estimating the packet loss specifically for multi-path, as the burst length in multi-path diversity not exceeding then two consecutive packet losses; also extended Gilbert Model estimates the packet loss better then Bernoulli in most of the scenarios.

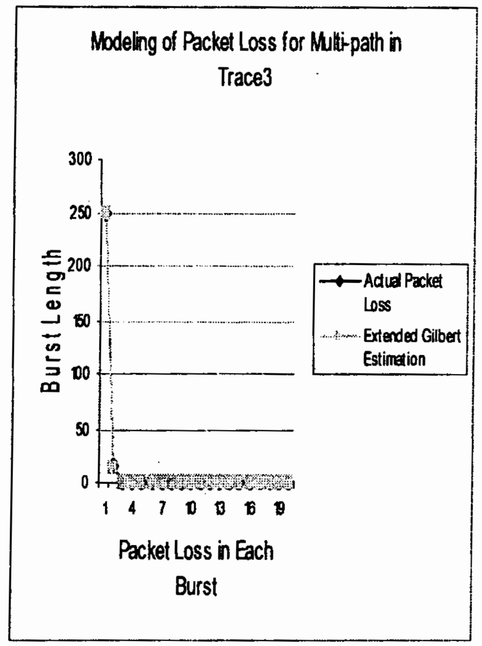
It is also concluded that it is better to use both packet loss models for estimating the packet loss and delay while implementing them in single path and multi path. But if the network is very congested and there is huge probability of losing the lengthy burst of packets in high frequency then Bernoulli model estimates the packet loss more accurately as compare to Extended Gilbert model, just because Bernoulli model always over estimates the packet loss. Conversely, if the network is less congested and the frequency of packet loss in each burst is limited then Extended

Gilbert model provides better results. As Multi-path has less number of packet losses, therefore Extended Gilbert always estimates the packet loss accurately.

6.6 Output Graphs on Modeling of Packet Loss

The following graphs show the performance of both the models for the single and multi-path on different scenarios, the scenarios include the increase and decrease in bandwidth, the increase in the packet size, variation in data rate etc.

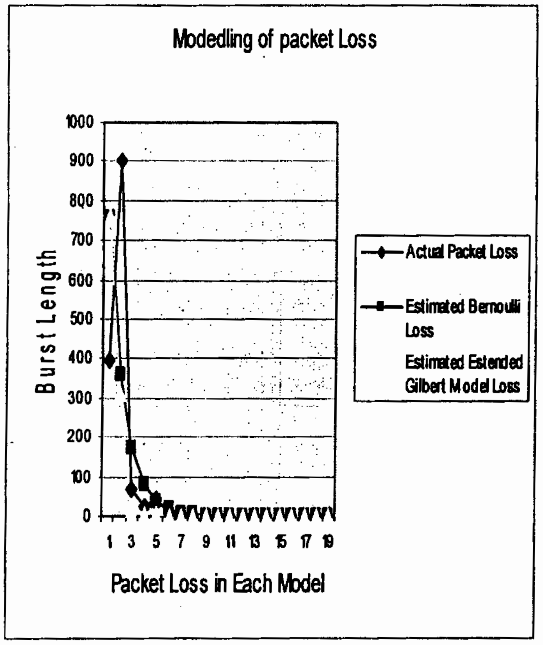




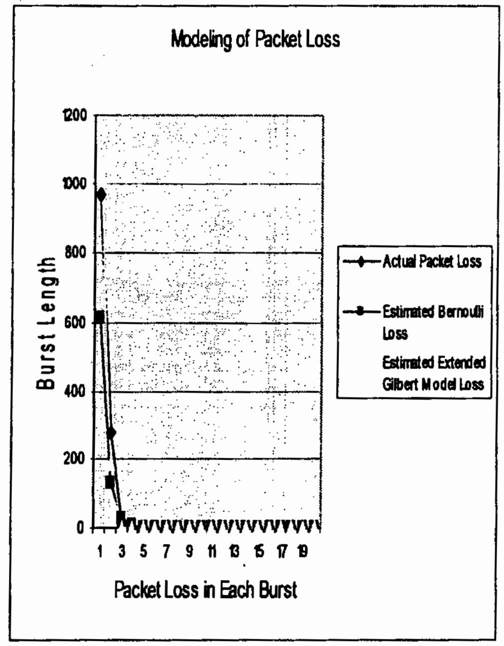
Graph 6.5 Bernoulli Loss Model in Multi-path (Trace 3)

Graph 6.6 Extended Gilbert Loss Model in Multi-path (Trace 3)

The Graphs i.e. Graph 6.5 and Graph 6.6 of Trace3 and Graph 6.7 and Graph 6.8 of Trace6 for the single and multi-path diversity demonstrate that the results of the expected number of Packet Loss by the Extended Gilbert model are very close to the actual loss of the packets only if the number of single packet loss is high. If there is huge number of packet loss with burst length more then one creates under estimation by the Extended Gilbert model. Thus the trace3 graphs prove that multi-path confers better results.



Graph 6.7 Comparison of Loss Models in Single path (Trace 6)



Graph 6.8 Comparison of Loss Models in Multi-path (Trace 6)

Reference & Bibliography

1. http://en.wikipedia.org/wiki/Voice_over_IP.

2. http://www.unet.univie.ac.at/aix/aixbman/commadmn/tcp_protocols.htm.

3. http://en.wikipedia.org/wiki/Packet_loss.

4. http://www.protocols.com/papers/voip2.htm.

5. http://www.cisco.com/en/US/tech/tk869/tk769/technologies_white_paper09186a00801b1a1e.shtml.

6. www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax_c/vclgloss.htm.

7. Wenyu Jiang, Henning Schulzrinne {Wenyu, Schulzrinne} @cs.colombia.edu. Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality Department of Computer Science, Columbia University. May 2000. book title = "Proc. NOSSDAV", url = "citeseer.ist.psu.edu/jiang00modeling.html"

8. www.voiptroubleshooter.com/indepth/burstloss.html.

9. Yi J.Liang, Eckehard G. Steinbach, and Bernd Girod, Real-time Voice Communication over the internet Using Packet Path Diversity, Information Systems Laboratory, Stanford University, Stanford. booktitle = "{ACM} Multimedia", pages = "431--440", year = "2001", url = "citeseer.ist.psu.edu/liang01realtime.html"

10. Philippe Golle and Nagendra Modadugu, Authentication Streamed data in the Presence of Random Packet Loss, Stanford. ISOC Network and Distributed System Security Symposium (2001), pp 13--22.", year = "2001", url = "citeseer.ist.psu.edu/golle01authenticating.html"

11. J. Ignacio Alvarez-Hamelin and Pierre Fraigniaud, Reducing Packet-Loss by Taking Long-Range Dependencies into Account. Laboratoire de Recherche en Informatique, B^at 490 Universit'e Paris Sud, 91405 Orsay CEDEX, France, {ihameli,pierre}@lri.fr

12. Hiroyuki Ohsaki, Masayuki Murata and Hideo Miyahar, Modeling End-to-End Packet Delay Dynamics of the Internet using System Identification. International Teletra#c Congress 17, Sept. 2001.", year = "2001"

13. Yi J.Liang and G.Steinbac, Real-Time Voice communication over the Internet using Packet Path Diversity. Information Systems Laboratory, Stanford University, Stanford. booktitle = "{ACM} Multimedia", pages = "431--440", year = "2001", url = "citeseer.ist.psu.edu/liang01realtime.html".

14. Jagadeesh Balam and Jerry D. Gibson, Path Diversity and Multiple Descriptions with Rate Dependent Packet Losses. Information Theory and Applications Workshop, University of California, San Diego, La Jolla, CA, February 6-10, 2006

15. http://en.wikipedia.org/wiki/fec.

16. Matulya Bansal and Avideh Zakhor, Path Diversity Based Techniques for Resilient Overlay Multimedia Multicast. Packet Video Workshop, Irvine, CA, Dec. 2004.

17. Thinh Nguyen and Avideh Zakhor, Path Diversity with Forward Error Correction (PDF) System for Packet Switched Networks. Proceedings of IEEE INFOCOM, 2003.", year = "2003"

Appendix A

Abbreviations

Appendix A

It includes the abbreviation used for the different terminologies, which are stated as following.

FEC	Forward Error Correction
VoIP	Voice over Internet Protocol
MDC	Multiple Description Coding
QOS	Quality of Service
SISO	Single Input Single Output
ARX	Auto-Regressive eXogenous
SDC	Single Description Coding
RON	Resilient Overlay Network
OLSR	Optimized Link State Routing Protocol
DAG	Directed Acyclic Graphs
ROM	Robust Overlay Multicast
SPS	Simple Parent Selection
PDD	Post Dial Delay
ALOC	Average Length of Call
ASR	Average Success Rate
DSR	Dynamic Source Routing protocol

UDP	User Datagram Protocol
TCP	Transport Control Protocol
SFQ	Stochastic fair Queue
DDR	Distributed Dynamic Routing Algorithm
GSR	Global State routing protocol
NS-2	Network Simulator 2
NAM	Network Animator

Appendix B

NS Code

Appendix B

```
#Create a simulator object

set ns [new Simulator]

############### Color Scheme for the TCP and UDP sources ###########

$ns color 1 Blue

$ns color 2 Red

$ns color 3 Green

$ns color 4 brown

$ns color 5 yellow

############### Open the NAM trace file ######################

set nf [open out.nam w]

$ns namtrace-all $nf

############### Open the Trace file #########################

set tf [open out.tr w]

$ns trace-all $tf

############### Define a 'finish' procedure ####################

proc finish {} {

    global ns nf tf

    $ns flush-trace

    #Close the NAM trace file
```

```
    close $nf

    #Close the Trace file

    close $tf

    #Execute NAM on the trace file

    exec nam out.nam &

    exit 0

}

############## Create nodes ##################################

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

set n5 [$ns node]

set n6 [$ns node]

set n7 [$ns node]

set n8 [$ns node]

set n9 [$ns node]

set n10 [$ns node]

set n11 [$ns node]
```

```
set n12 [$ns node]

set n13 [$ns node]

############### Create links between the nodes #########################

$ns duplex-link $n0 $n2 2Mb 30ms SFQ

$ns duplex-link $n1 $n2 2Mb 30ms SFQ

$ns duplex-link $n2 $n3 2Mb 45ms SFQ

$ns duplex-link $n4 $n5 2Mb 10ms SFQ

$ns duplex-link $n5 $n2 2Mb 10ms SFQ

$ns duplex-link $n5 $n2 2Mb 10ms SFQ

$ns duplex-link $n6 $n2 2Mb 10ms SFQ

$ns duplex-link $n7 $n8 2Mb 10ms SFQ

$ns duplex-link $n9 $n10 2Mb 10ms SFQ

$ns duplex-link $n10 $n11 2Mb 10ms SFQ

$ns duplex-link $n1 $n12 2Mb 10ms SFQ

$ns duplex-link $n12 $n13 2Mb 10ms SFQ

$ns duplex-link $n13 $n3 2Mb 10ms SFQ

$ns duplex-link $n8 $n12 2Mb 10ms SFQ

$ns duplex-link $n11 $n13 2Mb 10ms SFQ

############## Set Queue Size of link (n2-n3) to 10 ###################

$ns queue-limit $n2 $n3 100
```

```
############## Give node position (for NAM) #########################

$ns duplex-link-op $n0 $n2 orient right-down

$ns duplex-link-op $n1 $n2 orient right-up

$ns duplex-link-op $n2 $n3 orient right

$ns duplex-link-op $n5 $n2 orient right

$ns duplex-link-op $n6 $n2 orient right

$ns duplex-link-op $n7 $n8 orient right

$ns duplex-link-op $n9 $n10 orient right

$ns duplex-link-op $n10 $n11 orient right

$ns duplex-link-op $n1 $n12 orient right

$ns duplex-link-op $n12 $n13 orient right

$ns duplex-link-op $n13 $n3 orient right

$ns duplex-link-op $n8 $n12 orient right

$ns duplex-link-op $n11 $n13 orient right

############### Monitor the queue for link (n2-n3). (for NAM) ############

$ns duplex-link-op $n2 $n3 queuePos 0.5

############### Setup a TCP connection for n0-n3 #########################

set tcp [new Agent/TCP]

$tcp set class_ 2

$ns attach-agent $n0 $tcp
```

```
set sink [new Agent/TCPSink]

$ns attach-agent $n3 $sink

$ns connect $tcp $sink

$tcp set fid_ 1

############ Setup a FTP over TCP connection ##########################

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type_ FTP

############### Setup a TCP connection for n4-n3 #########################

set tcp1 [new Agent/TCP]

$tcp1 set class_ 4

$ns attach-agent $n4 $tcp1

set sink1 [new Agent/TCPSink]

$ns attach-agent $n3 $sink1

$ns connect $tcp1 $sink1

$tcp1 set fid_ 3

############ Setup a FTP over TCP connection ##########################

set ftp1 [new Application/FTP]

$ftp1 attach-agent $tcp1

$ftp1 set type_ FTP
```

```
############### Setup a TCP connection for n6-n3 #########################

set tcp2 [new Agent/TCP]

$tcp2 set class_ 2

$ns attach-agent $n6 $tcp2

set sink2 [new Agent/TCPSink]

$ns attach-agent $n3 $sink2

$ns connect $tcp2 $sink2

$tcp2 set fid_ 4

############ Setup a FTP over TCP connection ##########################

set ftp2 [new Application/FTP]

$ftp2 attach-agent $tcp2

$ftp2 set type_ FTP

############### Setup a TCP connection for n7-n3 #########################

set tcp3 [new Agent/TCP]

$tcp3 set class_ 2

$ns attach-agent $n7 $tcp3

set sink3 [new Agent/TCPSink]

$ns attach-agent $n3 $sink3

$ns connect $tcp3 $sink3

$tcp3 set fid_ 5
```

```
############ Setup a FTP over TCP connection ##########################

set ftp3 [new Application/FTP]

$ftp3 attach-agent $tcp3

$ftp3 set type_ FTP

############### Setup a TCP connection for n9-n3 #########################

set tcp4 [new Agent/TCP]

$tcp4 set class_ 2

$ns attach-agent $n9 $tcp4

set sink4 [new Agent/TCPSink]

$ns attach-agent $n3 $sink4

$ns connect $tcp4 $sink4

$tcp4 set fid_ 6

############ Setup a FTP over TCP connection ##########################

set ftp4 [new Application/FTP]

$ftp4 attach-agent $tcp4

$ftp4 set type_ FTP

#############Setup a UDP connection for n1-n3 ###########################

set udp [new Agent/mUDP]

$udp set_filename sd

$ns attach-agent $n1 $udp
```

```
set null [new Agent/mUdpSink]

$null set_filename rd

$ns attach-agent $n3 $null

$ns connect $udp $null

$udp set fid_ 2

############### Setup a CBR over UDP connection for n1-n3 ##############

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type_ CBR

$cbr set packet_size_ 1000

$cbr set rate_ 2mb

$cbr set random_ false

############### Schedule events for the CBR, FTP and FTP1 agents ######

$ns at 0.1 "$ftp start"

$ns at 0.1 "$ftp1 start"

$ns at 0.1 "$ftp2 start"

$ns at 0.1 "$ftp3 start"

$ns at 0.1 "$ftp4 start"

$ns at 1.1 "$cbr start"

$ns at 19.0 "$cbr stop"
```

```
$ns at 18.0 "$ftp stop"

$ns at 18.0 "$ftp1 stop"

$ns at 18.0 "$ftp2 stop"

$ns at 18.0 "$ftp3 stop"

$ns at 18.0 "$ftp4 stop"

#######Call the finish procedure after 12 seconds of simulation time #######

$ns at 20.0 "finish"

############# Print CBR packet size and interval #######################

puts "CBR packet size = [$cbr set packet_size_]"

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

#################### Run the simulation ########################

$ns run
```

Multi-Path Code

```
#Create a simulator object

set ns [new Simulator]

############### Color Scheme for the TCP and UDP sources ###########

$ns color 1 Blue

$ns color 2 Red

$ns color 3 Green

$ns color 4 brown
```

```
$ns color 5 yellow

############### Opeln the NAM trace file ######################

set nf [open out.nam w]

$ns namtrace-all $nf

############### Open the Trace file #########################

set tf [open out.tr w]

$ns trace-all $tf

############### Define a 'finish' procedure ####################

proc finish {} {

    global ns nf tf

    $ns flush-trace

    #Close the NAM trace file

    close $nf

    #Close the Trace file

    close $tf

    #Execute NAM on the trace file

    exec nam out.nam &

    exit 0

}

############## Create nodes #################################
```

```
Node set multiPath_ 1

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

set n5 [$ns node]

set n6 [$ns node]

set n7 [$ns node]

set n8 [$ns node]

set n9 [$ns node]

set n10 [$ns node]

set n11 [$ns node]

set n12 [$ns node]

set n13 [$ns node]

set n14 [$ns node]

############### Create links between the nodes ########################

$ns duplex-link $n0 $n2 2Mb 30ms SFQ

$ns duplex-link $n1 $n2 2Mb 30ms SFQ

$ns duplex-link $n2 $n14 2Mb 45ms SFQ
```

```
$ns duplex-link $n4 $n5 2Mb 10ms SFQ

$ns duplex-link $n5 $n2 2Mb 10ms SFQ

$ns duplex-link $n5 $n2 2Mb 10ms SFQ

$ns duplex-link $n6 $n2 2Mb 10ms SFQ

$ns duplex-link $n7 $n8 2Mb 10ms SFQ

$ns duplex-link $n9 $n10 2Mb 10ms SFQ

$ns duplex-link $n10 $n11 2Mb 10ms SFQ

$ns duplex-link $n1 $n12 2Mb 10ms SFQ

$ns duplex-link $n12 $n13 2Mb 10ms SFQ

$ns duplex-link $n13 $n3 2Mb 10ms SFQ

$ns duplex-link $n8 $n12 2Mb 10ms SFQ

$ns duplex-link $n11 $n13 2Mb 10ms SFQ

$ns duplex-link $n14 $n3 2Mb 10ms SFQ

############## Set Queue Size of link (n2-n3) to 10 ####################

$ns queue-limit $n2 $n14 100

############## Give node position (for NAM) #########################

$ns duplex-link-op $n0 $n2 orient right-down

$ns duplex-link-op $n1 $n2 orient right-up

$ns duplex-link-op $n2 $n14 orient right

$ns duplex-link-op $n5 $n2 orient right
```

```
$ns duplex-link-op $n6 $n2 orient right

$ns duplex-link-op $n7 $n8 orient right

$ns duplex-link-op $n9 $n10 orient right

$ns duplex-link-op $n10 $n11 orient right

$ns duplex-link-op $n1 $n12 orient right

$ns duplex-link-op $n12 $n13 orient right

$ns duplex-link-op $n13 $n3 orient right

$ns duplex-link-op $n8 $n12 orient right

$ns duplex-link-op $n11 $n13 orient right

$ns duplex-link-op $n14 $n3 orient right

############### Monitor the queue for link (n2-n3). (for NAM) ############

$ns duplex-link-op $n2 $n14 queuePos 0.5

############### Setup a TCP connection for n0-n3 #########################

set tcp [new Agent/TCP]

$tcp set class_ 2

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $n3 $sink

$ns connect $tcp $sink

$tcp set fid_ 1
```

```
############ Setup a FTP over TCP connection ##########################

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type_ FTP

############### Setup a TCP connection for n4-n3 #########################

set tcp1 [new Agent/TCP]

$tcp1 set class_ 4

$ns attach-agent $n4 $tcp1

set sink1 [new Agent/TCPSink]

$ns attach-agent $n3 $sink1

$ns connect $tcp1 $sink1

$tcp1 set fid_ 3

############ Setup a FTP over TCP connection ##########################

set ftp1 [new Application/FTP]

$ftp1 attach-agent $tcp1

$ftp1 set type_ FTP

############### Setup a TCP connection for n6-n3 #########################

set tcp2 [new Agent/TCP]

$tcp2 set class_ 2

$ns attach-agent $n6 $tcp2
```

```
set sink2 [new Agent/TCPSink]

$ns attach-agent $n3 $sink2

$ns connect $tcp2 $sink2

$tcp2 set fid_ 4

############ Setup a FTP over TCP connection ##########################

set ftp2 [new Application/FTP]

$ftp2 attach-agent $tcp2

$ftp2 set type_ FTP

############## Setup a TCP connection for n7-n3 ##########################

set tcp3 [new Agent/TCP]

$tcp3 set class_ 2

$ns attach-agent $n7 $tcp3

set sink3 [new Agent/TCPSink]

$ns attach-agent $n3 $sink3

$ns connect $tcp3 $sink3

$tcp3 set fid_ 5

############ Setup a FTP over TCP connection ##########################

set ftp3 [new Application/FTP]

$ftp3 attach-agent $tcp3

$ftp3 set type_ FTP
```

```
############### Setup a TCP connection for n9-n3 #########################

set tcp4 [new Agent/TCP]

$tcp4 set class_ 2

$ns attach-agent $n9 $tcp4

set sink4 [new Agent/TCPSink]

$ns attach-agent $n3 $sink4

$ns connect $tcp4 $sink4

$tcp4 set fid_ 6

############ Setup a FTP over TCP connection ##########################

set ftp4 [new Application/FTP]

$ftp4 attach-agent $tcp4

$ftp4 set type_ FTP

#############Setup a UDP connection for n1-n3 ###########################

set udp [new Agent/mUDP]

$udp set_filename sd

$ns attach-agent $n1 $udp

set null [new Agent/mUdpSink]

$null set_filename rd

$ns attach-agent $n3 $null

$ns connect $udp $null
```

```
$udp set fid_ 2

############### Setup a CBR over UDP connection for n1-n3 ##############

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type_ CBR

$cbr set packet_size_ 1000

$cbr set rate_ 1mb

$cbr set random_ false

#############Setup a UDP connection for n1-n4-n3 ##########################

set udp1 [new Agent/mUDP]

$udp1 set_filename sd

$ns attach-agent $n1 $udp1

$null set_filename rd

$ns attach-agent $n3 $null

$ns connect $udp1 $null

$udp1 set fid_ 2

############### Setup a CBR over UDP connection for n1-n4-n3 ##############

set cbr1 [new Application/Traffic/CBR]

$cbr1 attach-agent $udp1

$cbr1 set type_ CBR
```

```
$cbr1 set packet_size_ 1000

$cbr1 set rate_ 1mb

$cbr1 set random_ false

$ns rtproto DV

############### Schedule events for the CBR, FTP and FTP1 agents ######

$ns at 0.1 "$ftp start"

$ns at 0.1 "$ftp1 start"

$ns at 0.1 "$ftp2 start"

$ns at 0.1 "$ftp3 start"

$ns at 0.1 "$ftp4 start"

$ns at 1.1 "$cbr start"

$ns at 1.1 "$cbr1 start"

$ns at 19.0 "$cbr stop"

$ns at 19.0 "$cbr1 stop"

$ns at 18.0 "$ftp stop"

$ns at 18.0 "$ftp1 stop"

$ns at 18.0 "$ftp2 stop"

$ns at 18.0 "$ftp3 stop"

$ns at 18.0 "$ftp4 stop"

#######Call the finish procedure after 12 seconds of simulation time #######
```

```
$ns at 20.0 "finish"

#################### Run the simulation ########################

$ns run
```

Published Paper

Journal of Theoretical and Applied Information Technology

JATIT

Modeling of Packet Loss and Delay using Multi-Path Diversity

[1]Muhammad Asad Khan, [1]Yasir Irshad Abassi, [2] Imran Baig, [2] S.Tauseef-ur-Rehman
[1]Department of Computer Science, Faculty of Applied Sciences, International Islamic University, Islamabad, Pakistan
[2]Chairman, Department of Telecommunication, Federal Urdu University of Arts, science & Technology Islamabad, Pakistan
asad.khan1@hotmail.com, gallian@hotmail.com, imran_baig_mirza@yahoo.com, stauseef@fuuastisb.edu.pk

Abstract

The quality of voice in IP based networks is highly reliant on Packet loss and delay. Packet loss is the failure of packets to reach the destination [1,2]. The multi-path diversity is a phenomenon to send identical packets from diversified paths. Therefore, if certain packet is lost from one stream, still its identical packet has fair chance to arrive at the destination [3]. We first discuss the modeling of packet loss and delay using Bernoulli and Extended Gilbert Model [4]. We then evaluate the results of both models. We have developed a modified loss model for modeling of packet loss using multi-path diversity. The present research aims to increase the efficiency of Forward Error Correction (FEC) by the modeling of packet loss and delay in the real time communication. The set of examined results from our research shows that multi-path diversity bestows positive effect on the voice quality of VoIP network. The results also validate our model as comparison with actual data results in similar trend.

Key Words: Packet Loss, VoIP, Forward Error Correction, Multi-path Diversity

1 Introduction

Voice over Internet Protocol (VoIP) is the conversion of analog and digital communication into data transmission of digital packets over internet or intranet [5]. Basically, it is the toggling from circuit switching to packet switching. Voice over IP (VoIP) is susceptible to network behaviors, referred to as delay and packet loss, which can degrade the voice application to the point of being unacceptable to the average user [1,2,5].

The loss of the packets mostly occurs due to the congestion or the link failure between the nodes. The clogging with in the nodes mostly occurs due to the limited bandwidth and heavy load of voice and data traffic, which also falls out as packet loss [1,2,4]. The loss or dropping of packets results in highly noticeable performance issues and affects all other network applications [1,2].

The loss of data packets can be recovered either by resending the packets to the destination or by using some recovery algorithm. The data

packets are time independent, so they are easy to recover. On the other hand, voice packets are
based on the time limitations. As a result, if voice packets are lost or delayed, they are hard to recover for the reason that if they cross the play out time then they are futile. Therefore, it creates either the short duration calls or very appalling voice quality [5].

1.1 Multi-path Diversity

In Multi Path diversity, the copies of identical packets are sent over a network to achieve the advantage of uncorrelated packet loss and delay [3]. The advantage of multi-path diversity over single path is that there is less probability of losing the packet. If certain packet is lost from one path, still it has positive probability of reaching the destination from the second path. Another advantage of multi-path diversity is that the extent of the packet loss burst is relatively diminutive. The major shortcoming of multi-path diversity is that it needs superfluous bandwidth on both source and the destination links. In addition the destination node needs further processing to discard the already received packets. Therefore, it creates some extra delay in the arrival of voice packet to the destination, but it is better then losing the packets [3].

1.2 Loss Modeling

The analytical modeling of the packets loss and delay for the assessment of the packet loss needs to be done to hit upon the better model for estimating packet loss [4,6].

1.2.1 The Bernoulli Model

The Bernoulli loss model is based on a geometric distribution. It is the most widely used model and based on simple independent losses. It is very basic modeling algorithm for estimating packet loss. Therefore, often use in modeling of packet loss for IP voice and multicast systems [4,6]. The probability of packet loss in the Bernoulli model is represented by **p**. If there are large number of packets **n** to be transmitted over a network then the expected number of lost packets is **n*p**. Bernoulli loss model is a two-state process. The one state (State 0) symbolizes a packet loss, and the other state (State 1) stands for a packet reaching the destination. The mean loss probability **p** represents the probability that the current packet is lost given that the last packet was also lost. **q** is the probability that the current packet is arrived provided that the previous packet was also arrived [6].

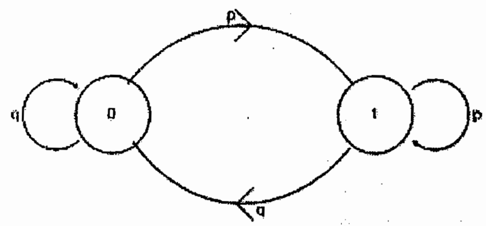


Figure 1.0: The Bernoulli Model

The Bernoulli model gives an estimation of loss probability by calculating the total number of packets that were lost and then dividing the result by the total transmitted packets. In Bernoulli model, each packet transmitted on a network has fixed and independent loss probability with constant loss rate of the link [4,6]. Networks in which time interval between the packets is very short, the loss of packets can not be estimated properly by a Bernoulli model [4,6].

1.2.2 The Extended Gilbert Model

The Bernoulli model maintains the record of all past **n** number of losses to calculate the probability of losing the next packet, where as Extended Gilbert model consider only last **n** number of consecutive loss of packets to calculate the probability of the next packet to be lost. Therefore the probability calculation and loss estimation are in the vicinity of the actual loss of packets. The extended Gilbert model needs **n+1** states to remember **n** events [4].

The Extended Gilbert Algorithm is used with a structure that maintains a counter l, which is the number of consecutive packet loss. But it is reset whenever the packet is received [4]. The extended Gilbert model is the extension of two-state Gilbert model which calculates the burst state with almost the same transition probability as of Bernoulli algorithm by considering burst state as 1 and then according to state transition probabilities P01 and P11 is the probability for the burst length of packet loss.

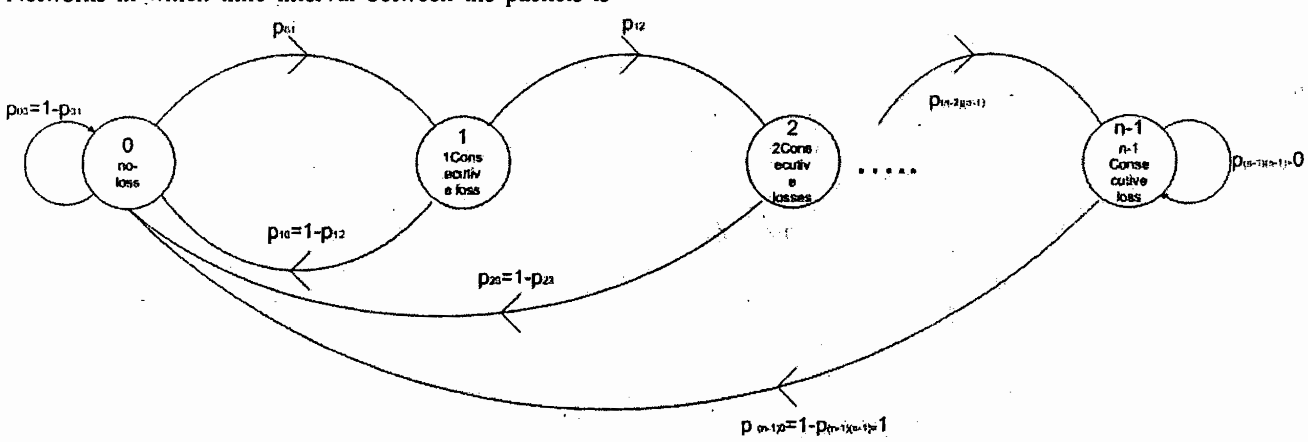


Figure 1.2: The Extended Gilbert Model

2 Literature Review

The work has been done on the algorithms of recovering packets and modeling of voice and data packets loss. The basic modeling algorithm of packet loss and delay are Bernoulli model, Markov model, Gilbert Model and Extended Gilbert Model.

Wenyu Jiang effort was based on the efficiency and perceptual quality of FEC. They started with the modeling of packet loss and delay. Then they proposed the joint use of Extended Gilbert model and inter-loss distance (ILD). Then they implemented the internet traces to validate the necessity and effectiveness of these models. The next phase was the assessment of the consequences of this reliance on VoIP. But their research lack in finding out that how burstiness relays to quality of voice [4].

Yi J. Liang, Eckehard G. Steinbach, and Bernd Girod contributed in the reconstruction of the voice transmission over internet. They worked to improve the tradeoff among delay, late loss rate, and speech quality using multi-stream transmission of real-time voice over the Internet, where multiple redundant descriptions of the voice stream are sent over independent network paths. Scheduling the play out of the received voice packets is based a Lagrangian cost function to trade delay versus loss. They observed significant reductions in mean end-to-end latency and loss rates as well as improved speech quality when compared to FEC protected single-path transmission at the same data rate. In addition to our Internet measurements, they analyzed the performance of the proposed multi-path voice communication scheme using the *ns* network simulator [3].

Jean Yves Le implemented a joint play out buffer and Forward Error Correction (FEC) adjustment scheme for Internet Telephony that incorporates the impact of end-to-end delay on the perceived audio quality. Their results publicized that it offers a better quality than the adjustment schemes for play out buffer and FEC. This is important because of a threshold effect when the end-to-end delay of interactive audio is around 150 ms. They represented the perceived audio quality as a function of both the end-to-end delay and the distortion of the voice signal. They validated their approach by simulation that their scheme allows a source to increase its utility by avoiding increasing the play out delay when it is not really necessary. But the problem in that development is the static size of buffer, which again creates the loss of the packets and the delay by the bursting of traffic on certain conditions. The problem remains the same, as when the packets cross the size of the buffer, they start to loss or creates the delay [7].

In the earlier research, different loss models were designed to calculate the packet loss and delay for the single path. The emphasis was on estimating the packet loss and delay using diverse loss models. But no one has compared the loss models to unearth the better among them and subsequently implement the resulted better model using multi-path diversity phenomenon to observe any advantage of multi-path diversity for the modeling of packet loss. Mostly the packet loss and delay are due to the congestion or the link failure. Therefore, it requires a technique to recover the lost data. Forward Error Correction (FEC) is use to recover the packets. The snag in FEC for the single path is that it cannot recover lengthy burst of packet loss, as a result all the interlinked packets are also lost, because the information of each packet is added in its subsequent packet. So the sender needs to send the packets again, hence creating more delay.

We have proposed a solution to have better quality of service using the multiple paths. We will compare the extended Gilbert model with Bernoulli model for the single path. The better model will be implemented using the multi-path diversity phenomenon for the modeling of the packet loss and delay. The whole emphasis is on the efficiency of Forward error correction by the modeling of packet loss and delay in the real time communication and to make it more useable to recover the lost packets. By the implementation, we will validate any change in the efficiency of forward Error Correction (FEC) by the introduction of multi path diversity.

3 Methodology

3.1 Estimating Packet Loss:

The implementation for estimating packet loss and delay has been done using network Simulator *ns*. The source of the nodes generates either tcp packets or udp packets to the destination. The tcp packets generating nodes are boasting ftp at the application layer, whereas the cbr is running at the application layer of the udp generating nodes. The capacity of the links is capable enough to handle massive voice and data traffic.

The queues used in the links are SFQ, the stochastic fair queue. The advantage of SFQ over DropTail Queuing is that it does fair queuing. Therefore, the dropping of tcp and udp packets will be same [8].

The first step is the implementation of packet loss modeling using Bernoulli and Extended Gilbert Model in single path environment. The implementation is based on sending the packets form source to destination for specific period of time and calculating total number of voice packets that were sent over the network in single stream. The outcome of the Bernoulli probability and Extended Gilbert probability leads to the estimation of the packet loss using Bernoulli model and Extended Gilbert model respectively. The results will be compared with the actual loss of the packets to obtain the better loss model.

3.2 Bernoulli Loss Model for Single Path

The Implementation was based on the following mentioned steps:

- Design a complex network which comprises of both data and voice traffic from different nodes to the destination nodes.
- Assign starting and ending time to the simulation to extract relevant data during specific time.
- Dispense starting and ending time to all the nodes which are generating the packets.
- Start sending the packets.
- Calculate the sent time, received time and the total delay that packets required to reach the destination.
- Calculate the total voice packets sent by specific source to destination.
- Calculate the actual loss of voice packets for the above mentioned link.

- Calculate the actual length of bursts of packet loss.
- Start calculating the mean Probability using under mentioned method of Bernoulli algorithm.

$$\prod = \left(\sum_{i=1}^{n-1} Li\right) / L0$$

Where
- Li is the loss bursts numbers with length i.
- The value of i=1, 2, 3.........n-1. i=1 means that single packet loss whereas n-1 is the longest burst of packet loss.
- L0 is the total number of packets sent from source to the destination [4].

- Now as we have the Mean Probability using Bernoulli model, we need to approximate the Probability of losing the packets for the different lengths of bursts using following probability formula.

$$\prod k = \prod^{k-1} * (1 - \prod)$$

Where
- $\prod k$ is the actual probability for estimating the packet loss.
- k is the length of burst of packet loss. e.g. If k=1 that means $\prod 1$ is calculating the probability of single packet loss [1].

- The result of probability of packet loss will lead us to the estimation of packet loss by using following algorithm.

$$L(est) = \text{Total packet loss} * \prod k$$

Where
- $\prod k$ is the probability taken from the above step.
- L(est) is the estimated packet loss using Bernoulli model of packet loss.

3.3 Extended Gilbert Loss Model for Single Path:

The Implementation was based on the following mentioned steps:

- The steps till the calculation of actual length of burst are same as that of Bernoulli Model.
- Start calculating the mean Probability using under mentioned method of Extended Gilbert Algorithm

$$\mu = 1 - \left(\sum_{j=1}^{n-1} Lj * (j-1)\right) / \left(\sum_{j=1}^{n-1} L(j-1) * (j-1)\right)$$

Where
- Lj is the loss bursts numbers with length j.
- The value of j=1, 2, 3.........n-1. j=1 means that single packet loss whereas n-1 is the longest burst of packet loss.
- Now as we have the Mean Probability using Extended Gilbert model, we need to approximate the Probability of losing the packets for the different lengths of bursts using following probability formula.

$$\mu k = (1 - \mu(k-1)(k)) * \mu^{k-1}$$

Where
- μk is the actual probability for estimating the packet loss.
- k is the length of burst of packet loss. e.g. If k=1 that means $\mu 1$ is calculating the probability of single packet loss [4].

- The result of probability of packet loss will lead us to the estimation of packet loss by using following algorithm of extended Gilbert Model.

$$L(est)k = \sum_{j=1}^{k-1} Lj * \mu k$$

Where
- μk is the probability taken from the above step.
- L(est)k is the estimated packet loss using Extended Gilbert model of packet loss.
- k is the length of packet loss for which estimation has been calculated.

3.4 Extended Gilbert Model for Multi-path

The major change in implementing the multi-path diversity over single path is that we designed a network in which there were two streams from the source to the destination for the specific voice communicating link. The duplicate streams of voice packets from source to destination start sending the packets at the same time. All the packets in these streams are duplicate packets of each other. The sending time and the packet fid of duplicate packets are same. Therefore, if there will be no packet loss in the two streams that means the destination node receives every packet twice. So there is a need of discarding the packets which are already arrived from the other stream. When any packet is arrived to the destination, its unique id is compared with all earlier received packets. If the id of the packet is matched with any of the earlier received packet's id that means it is already received and there is no need to receive that packet again [4]. Therefore, we made a mechanism to drop any packet which was earlier received. The end result by implementing the multi-path diversity was that there were very limited packet losses and even the length of packet loss bursts was not lengthy [3,4].

Rest of the implementation of the packet loss was similar to that of single path implementation.

- Design a complex network which comprises of both TCP and UDP packets that means that the network should comprise of data and voice traffic from different nodes to the destination nodes.
- Assign starting and ending time in the simulation to extract relevant data for the analysis.
- Assign starting and ending time to all the nodes which are generating the packets.
- Start sending the packets.
- Calculate the sent time, received time and the total delay that packet required to reach the destination.
- Calculate the total voice packets sent by some specific source to destination.
- Calculate the actual loss of voice packets for the above mentioned link.
- Calculate the actual length of bursts of packet loss.
- Start calculating the mean Probability using under mentioned method of Extended Gilbert Algorithm.

$$\mu = 1 - \left(\sum_{j=1}^{n-1} L_j * (j-1)\right) / \left(\sum_{j=1}^{n-1} L(j-1) * (j-1)\right)$$

Where

- Lj is the loss bursts numbers with length j.
- The value of j=1, 2, 3..........n-1. j=1 means that single packet loss whereas n-1 is the longest burst of packet loss.
- Now as we have the Mean Probability using Extended Gilbert model, we need to approximate the Probability of losing the packets for the different lengths of bursts using following probability formula.

$$\mu k = (1 - \mu(k-1)(k)) * \mu^{k-1}$$

Where

- μk is the actual probability for estimating the packet loss.
- k is the length of burst of packet loss. e.g. If k=1 that means μ1 is calculating the probability of single packet loss.

- The result of probability of packet loss will lead us to the estimation of packet loss by using following algorithm of extended Gilbert Model.

$$L(est)k = \sum_{j=1}^{k-1} L_j * \mu k$$

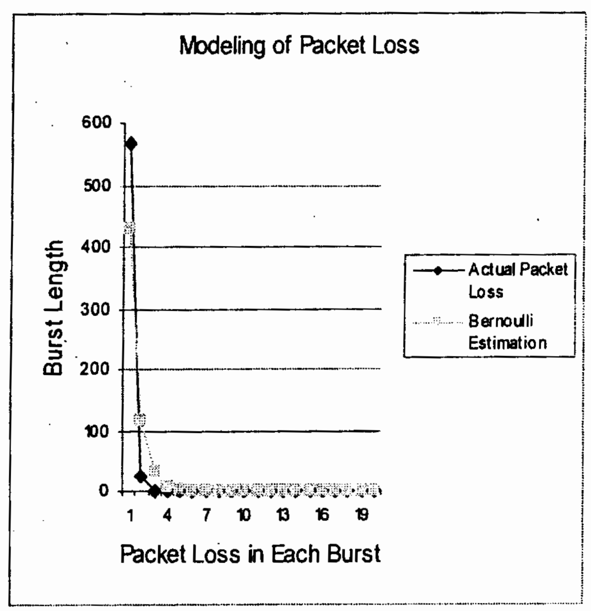
Where

- μk is the probability taken from the above step.
- L(est)k is the estimated packet loss using Extended Gilbert model of packet loss.
- k is the length of packet loss for which estimation has been calculated [4].

Another advantage of modeling of packets using multi-path diversity is that the Forward error correction works better if the bursts of packet loss are not very lengthy. Multi-path packet forwarding gives the advantage to forward error correction to even recover very few lost packets [3,4].

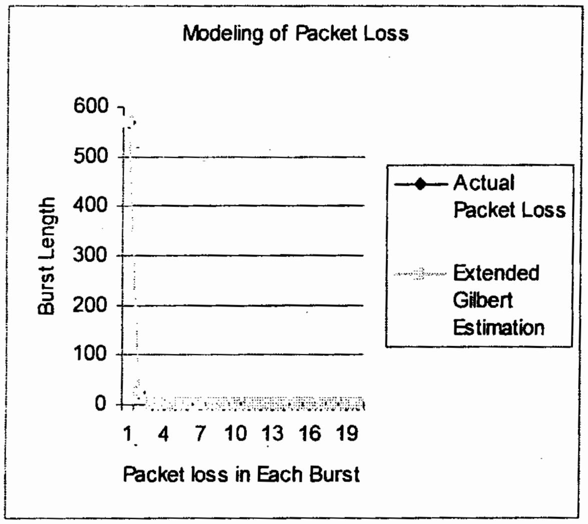
4 Results and Conclusion

The major advantage of Extended Gilbert model over Bernoulli model results is that the probability distribution of every packet depends upon only the **n** consecutive loss packets, whereas in the Bernoulli model of probability distribution the future probability loss of the packet is dependent on all **n** packets. Therefore, the probability in Extended Gilbert model is well calculated to the actual loss of the packets as compare to the Extended Gilbert model [4,6].



Graph 4.0 Bernoulli Loss Model in Single Path (Trace 1)

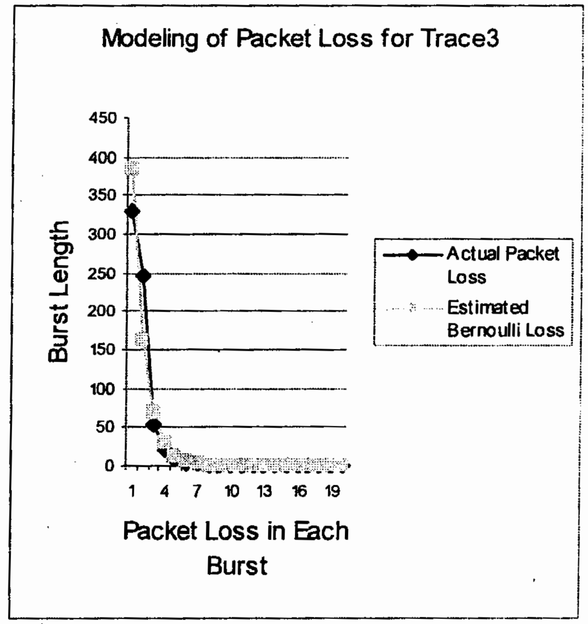
The implementation of packet loss modeling in single path and multi-path using Bernoulli and Extended Gilbert model ensures that the quality of service can be extendable and the voice quality over the packet network can get better and it can be estimate able better.



Graph 4.1 Extended Gilbert Loss Model in Single Path (Trace 1)

The above mentioned graphs prove that if the total packets lost in each burst (except burst length of one and two) are short then Extended Gilbert Model accurately estimates the packet loss to the actual loss of the packets. On the other hand, the Bernoulli loss model over estimates the number of packet loss in every burst, therefore gives results far away from the actual loss of packets.

The following tables present results regarding the simulation for both single path and multi-path phenomenon and the estimation of packet loss using the two loss models.



Graph 4.2 Bernoulli Loss Model in Single path (Trace 3)

Trace	Data Rate	Packet Size	Total nodes	Total Links	Total Links	Delay	Band width	Packets	Packets	Packets Lost
1	1mb	1K	15	15	14	30ms/10	2mb	2237	1646	591
2	1mb	1K	15	15	15	30ms/10	2mb	2237	1605	632
3	1mb	1K	15	15	15	30ms/10	1.5mb	2237	1572	665
4	1mb	1K	15	15	15	30ms/10	1.1mb	2237	1690	547
5	2mb	1K	15	15	15	30ms/10	2mb	4475	XXX	XXX
6	2mb	1K	15	15	14	30ms/10	2mb	4475	3041	1434

Table 4.0 Single Path Network Structure

Trace	Data Rate	Packet Size	Total nodes	Total Links	Total Links UP	Delay	Band width	Packets Sent	Packets Received	Packets Lost

1	1mb	1K	15	15	14	30ms/10ms	2mb	2237	2115	122
2	1mb	1K	15	15	15	30ms/10ms	2mb	2237	2041	196
3	1mb	1K	15	15	15	30ms/10ms	1.5mb	2237	1973	264
4	1mb	1K	15	15	15	30ms/10ms	1.1mb	2237	1865	372
5	2mb	1K	15	15	15	30ms/10ms	2mb	4475	3158	1317
6	2mb	1K	15	15	14	30ms/10ms	2mb	4475	3211	1264

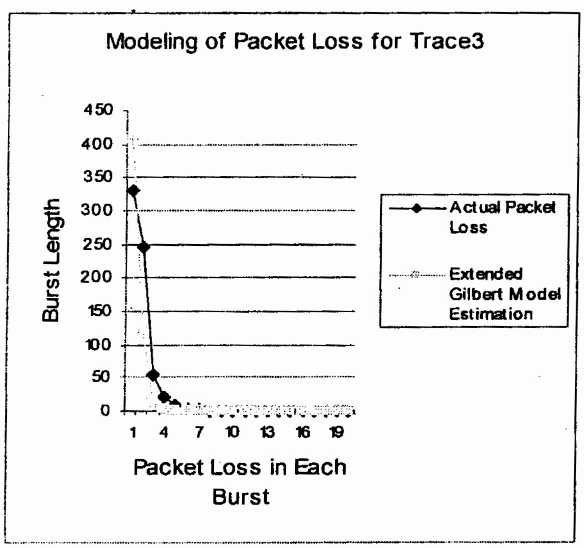
Table 4.1 Multi-Path Network Structure

Table 4.2 Comparison of Loss models using Trace 1

Table 4.3 Comparison of Loss models using Trace 3

Length of Loss Burst	Actual Packet Loss (Single Path)	Estimated by Bernoulli Model (Single Path)	Estimated by Extended Gilbert Model (Single Path)	Actual Packet Loss (Multi-Path)	Estimated by Bernoulli Model (Multi-Path)	Estimated by Extended Gilbert Model (Multi-Path)
0	1572	1572	1572	1973	1973	1973
1	330	385	404	249	223.7	249.8
2	247	162	121	15	26.6	13.4
3	54	68	12	0	3.1	0
4	20	28.6	3	0	0.3	0
5	10	12	1	0	0	0
6	2	5	0.21	0	0	0
7	2	2	0.07	0	0	0

Length of Loss Burst	Actual Packet Loss (Single Path)	Estimated by Bernoulli Model (Single Path)	Estimated by Extended Gilbert Model (Single Path)	Actual Packet Loss (Multi-Path)	Estimated by Bernoulli Model (Multi-Path)	Estimated by Extended Gilbert Model (Multi-Path)
0	1646	1646	1646	2115	1996	2115
1	567	430.3	567.9	121	215.64	121
2	24	116.9	22.1	1	23.4	0.9
3	0	31.8	0	0	2.5	0
4	0	8.6	0	0	0.03	0
5	0	2.3	0	0	0.003	0
6	0	0.6	0	0	0	0
7	0	0.17	0	0	0	0



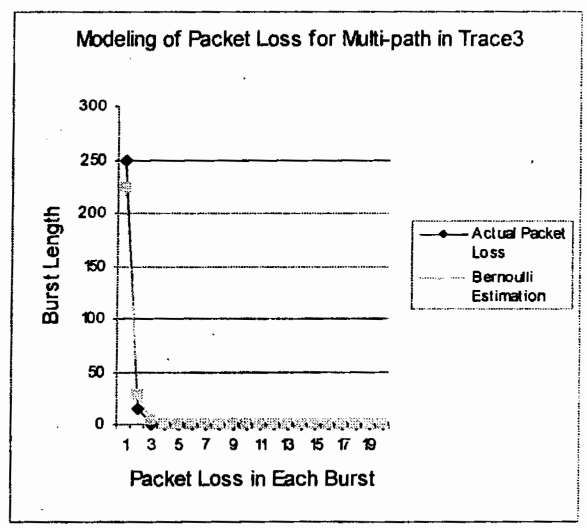
Graph 4.3 Extended Gilbert Loss Model in Single path (Trace 3)

The mentioned tables and graphs give very interesting results regarding the loss models. The results of the research prove that Bernoulli model also provides effective results as Extended Gilbert do, but for some

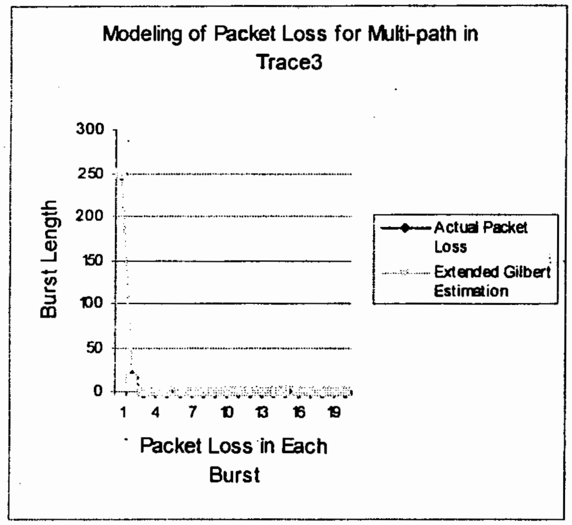
The above mentioned comparison table shows that the multi-path diversity provides better quality of service, as it allows more packets to reach the destination. The simulation proves that using multi-path phenomenon provides packets more options to reach the destination, hence less number of packet loss.

In the above mentioned tables, the length of the lost burst describes the actual length of the lost packets. If the length of the lost burst is zero then it describes the total number of packets that are received at the destination.

specific Scenarios. The analysis of Table 4.2 and Table 4.3 explains that if the actual burst of packet loss is very lengthy, each burst includes massive frequency of packet losses and comparatively limited number of single packet loss, then Bernoulli Loss model can give better results as compare to Extended Gilbert Model, because it always over estimates the loss of packets, hence close down the estimation of packet loss to the actual packet loss. So we have used both the models for the estimation of packet loss in multi-path diversity as well.

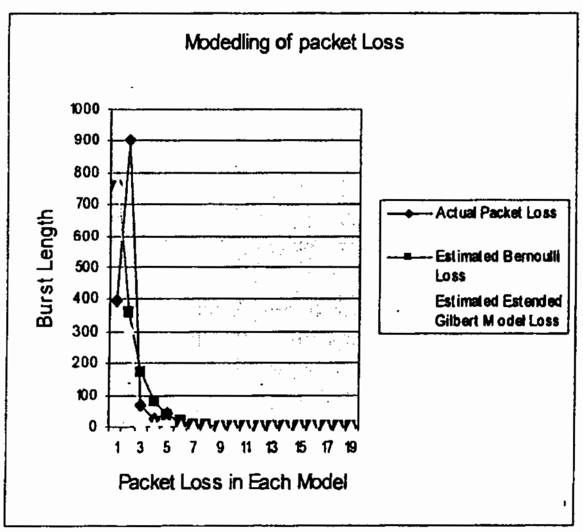


Graph 4.4 Bernoulli Loss Model in Multi-path (Trace 3)

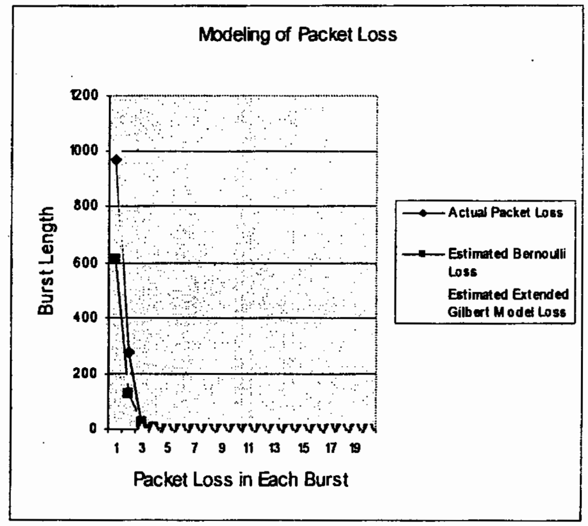


Graph 4.5 Extended Gilbert Loss Model in Multi-path (Trace 3)

The Graphs of Trace3 for the single and multi-path diversity demonstrate that the results of the expected number of Packet Loss by the Extended Gilbert model are very close to the actual loss of the packets only if the number of single packet loss is high. If there is huge number of packet loss with burst length more then one creates under estimation by the Extended Gilbert model. Thus the trace3 graphs prove that multi-path confers better results.



Graph 4.6 Comparison of Loss Models in Single path (Trace 6)



Graph 4.7 Comparison of Loss Models in Multi-path (Trace 6)

The above mentioned results have also shown that the implementation of multi-path diversity has given better quality of service because we have very limited number of actual packet loss. Therefore, the estimation of packet loss using Extended Gilbert model is very near to the real packet loss. The extended Gilbert model is better option for estimating the packet loss specifically for multi-path, as the burst length in multi-path diversity not exceeding then two consecutive packet losses; also extended Gilbert Model estimates the packet loss better then Bernoulli in most of the scenarios.

It is better to use both packet loss models for estimating the packet loss and delay while implementing them in single path and multi path. But if the network is very congested and there is huge probability of losing the lengthy burst of packets in high frequency then Bernoulli model estimates the packet loss more accurately as compare to Extended Gilbert model, juts because Bernoulli model always over estimates the packet loss. Conversely, if the network is less congested and the frequency of packet loss in each burst is limited then Extended Gilbert model provides better results. As Multi-path has less number of packet loss, therefore Extended Gilbert always estimates the packet loss accurately.

5 Future Work

The research is based on the modeling of packet loss using single path and multi-path phenomenon. We plan to research that what will be the advantage of using FEC in Multi-path diversity and its effect of the integration on bandwidth and performance of network in terms of packet loss. We have also plan to integrate both the loss models for the estimation of packet loss and see their performance.

6 References

1. *Packet Loss*, http://en.wikipedia.org/ wiki/Packet_loss

2. Yoshiaki Hori, *udp packet loss, Effect of TCP Traffic*, http://www.isoc .org /inet97/ proceedings/F3/F3_1.HTM,

3. Yi J.Liang, Eckehard G. Steinbach, and Bernd Girod, *Real-time Voice Communication over the internet Using Packet Path Diversity*, Information Systems Laboratory, Stanford University, Stanford.

4. Wenyu Jiang, Henning Schulzrinne {Wenyu,Schulzrinne} @cs.colombia.edu. *Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality* Department of Computer Science, Columbia University., May 2000.

5. VoIP vs. IP telephony http://vertical.com/voip/voip-vs-ip-telephony.html.

6. Understanding the Bernoulli Model, http://www.commsdesign.com/design_corner/showArticle.jhtml?articleID=18901597.

7. *Jean Yves Boudes*, Adaptive Joint playout Buffer and FEC Adjustment for Internet Telephony, 2001.

8. Manual NS.

9. D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. The case for resilient overlay networks. In Proceedings of the 8th Annual Workshop on Hot Topics in Operating Systems (HotOS-VIII), May 2001. Online at: http://nms.lcs.mit.edu/projects/ron/.

10. J. G. Apostolopoulos. Reliable video communication over lost packet networks using multiple state encoding and path diversity. In Proceedings Visual Communication and Image Processing, pages 392–409, Jan. 2001.

11. M. Arlitt and T. Jin. Workload characterization study of the 1998 World Cup web site. IEEE Network, 14(3):30–7, May 2000.

12. J.-C. Bolot. End-to-end packet delay and loss behavior in the Internet. Computer Communication Review, ACM SIGCOMM '93, 23(4):289–298, Sept. 1993.

13. J.-C. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive FEC-based error control for Internet telephony. In Proceedings of IEEE INFOCOM '99, volume 3, pages 1453–1460, Mar. 1999.

14. V. Hardman, A. Sasse, and A. Watson. Reliable audio for use over the Internet. In Proceedings of INET '95, pages 171–178, June 1995.

15. R. V. Hogg and E. A. Tanis. Probability and statistical inference. Macmillan Publishing Company, 4th edition, 1993.

16. ITU-T Recommendation G.726. 40, 32, 24, 16 kbit/s Adaptive differential pulse code modulation (ADPCM), Dec. 1990.

17. ITU-T Recommandation P.862. Perceptual evaluation of speech quality (PESQ), an objective method for End-to-end speech quality assessment of narrow-band telephone networks and speech codec, Feb. 2001.