FLOW CONTROL OF REAL-TIME UNICAST MULTIMEDIA APPLICATIONS IN BEST-EFFORT NETWORKS

A Dissertation

by

ANINDA BHATTACHARYA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2007

Major Subject: Mechanical Engineering

FLOW CONTROL OF REAL-TIME UNICAST MULTIMEDIA APPLICATIONS IN BEST-EFFORT NETWORKS

A Dissertation

by

ANINDA BHATTACHARYA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Alexander G. Parlos
Committee Members,	Suhada Jayasuriya
	Won-Jong Kim
	Xi Zhang
Head of Department,	Dennis O' Neal

August 2007

Major Subject: Mechanical Engineering

ABSTRACT

Flow Control of Real-Time Unicast Multimedia
Applications in Best-Effort Networks. (August 2007)

Aninda Bhattacharya, B.E., S.V. National Institute of Technology Surat India;
M.S., Texas A&M University
Chair of Advisory Committee: Dr. Alexander G. Parlos

One of the fastest growing segments of Internet applications are real-time multimedia applications, like Voice over Internet Protocol (*VoIP*). Real-time multimedia applications use the User Datagram Protocol (*UDP*) as the transport protocol because of the inherent conservative nature of the congestion avoidance schemes of Transmission Control Protocol (*TCP*). The effects of uncontrolled flows on the Internet have not yet been felt because UDP traffic frequently constitutes only $\sim 20\%$ of the total Internet traffic. It is pertinent that real-time multimedia applications become better citizens of the Internet, while at the same time deliver acceptable Quality of Service (*QoS*).

Traditionally, packet losses and the increase in the end-to-end delay experienced by some of the packets characterizes congestion in the network. These two signals have been used to develop most known flow control schemes. The current research considers the flow accumulation in the network as the signal for use in flow control. The most significant contribution of the current research is to propose novel end-to-end flow control schemes for unicast real-time multimedia flows transmitting over best-effort networks. These control schemes are based on predictive control of the accumulation signal. The end-to-end control schemes available in the literature are based on reactive control that do not take into account the feedback delay existing between the sender and the receiver nor the forward delay in the flow dynamics.

The performance of the proposed control schemes has been evaluated using the ns-2 simulation environment. The research concludes that active control of hard real-time flows delivers the same or somewhat better QoS as High Bit Rate (*HBR*, no control), but with a lower average bit rate. Consequently, it helps reduce bandwidth use of controlled real-time flows by anywhere between 31.43% to 43.96%. Proposed reactive control schemes deliver good QoS. However, they do not scale up as well as the predictive control schemes. Proposed predictive control schemes are effective in delivering good quality QoS while using up less bandwidth than even the reactive control schemes. They scale up well as more real-time multimedia flows start employing them.

To Amar Nath and Reena Bhattacharya

ACKNOWLEDGMENTS

I am indebted to Dr. Alexander Parlos for helping me understand what it means to be a researcher. Without his ideas this work would not have been possible.

For the past one and half years, I have been supported by the teaching and research funds of two professors here at Texas A&M University - Dr. Percy Galimbertti of Rural Community Health Institute (RCHI, TAMHSC) and Dr. Ana E. Goulart of Department of Engineering Technology and Industrial Distribution (ETID). After losing my funding as a Research Assistant in December 2005, I had almost given up hope of ever finishing my PhD. I would not have been able to finish my dissertation work without their financial and moral support during this time of crisis in my life.

My dissertation committee members - Dr. Suhada Jayasuriya, Dr. Won-Jong Kim, and Dr. Xi Zhang have always helped me out whenever I needed their advice. I am indebted to them for their advice. Dr. Jayasuriya and Dr. Kim taught me SISO and MIMO control theory. Dr. Zhang taught me the fundamentals on "Internet Protocols and Modeling" and ns-2 in his graduate level course ELEN $689-606$. This course played a very important role in helping me complete my PhD dissertation research.

I am grateful to all of my friends, especially, Parasuram Harihara, Bharati Hegde, Dan Ye, Lin Wang, Benjamin Chu, and Priyadarshini Kurup, for nourishing me with lots of fun and stimulating discussions during good as well as bad times here at TAMU. Association for India's Development - TAMU chapter (AID-TAMU), provided me a useful outlet to appreciate India in a better way. Prakash Krishnan, Ravi Karedla, Puneet Singla, Vinod Srinivasan, Rangakrishnan Srinivasan, Shivkumar Jolad, Nitin Ved, Bharath Thandri, Satish Natti, Ganesh Venkataraman, Kapil Bhattad, Dipa Brahmabhatt, and Chinmay Mishra, provided me with enough intellectual input re-

garding the politics and economics of India during the last three years. Without their companionship, life would have been really boring and frustrating here in College Station.

Last, but not the least, I would like to express my heartfelt appreciation for the faculty members of the Department of Mechanical Engineering and the Department of Electrical Engineering at Texas A&M University. I would not have been inspired to seek new boundaries of knowledge without their helping hand.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Multimedia Flows Over the Internet

The Internet has become the most important communication tool in today's world. As more and more people have logged onto the Internet, the push for providing and sharing different kinds of information through this medium has increased. The Internet has been instrumental in trying to level the playing field for everybody in this world by reducing what economists term as the "information asymmetry".

David D. Clark summarizes the goals in the design of the Internet in his paper [1], as follows:

1. Internet communication must continue despite loss of networks or gateways.

2. The Internet must support multiple types of communications service.

3. The Internet architecture must accommodate a variety of networks.

4. The Internet architecture must permit distributed management of its resources.

5. The Internet architecture must be cost effective.

6. The Internet architecture must permit host attachment with a low level of effort.

7. The resources used in the Internet architecture must be accountable.

Ten years ago, the Internet was used predominantly for reading, sharing, and delivering documents using a web browser or some other application like the E-mail or File Transfer Protocol (*FTP*) clients. The primary application that needed to be

The journal model is *IEEE Transactions on Automatic Control.*

supported in the earlier days of the Internet was the reliable data delivery and transfer between two connected systems. Initially, the designers developed Transmission Control Protocol (*TCP*) in such a way that it could be general enough to support any needed type of service. The nature of the information available on the Internet is undergoing a shift towards multimedia. Audio-visual media plays an important role in the dissemination of information. A reliable data delivery protocol like TCP is not good enough to meet the requirements of multimedia applications over the Internet. Delivery of multimedia content using the Internet has thrown up new challenges for the research community.

Multimedia applications can be divided into two classes - real-time and non real-time. Non real-time applications include some types of streaming applications. Streaming applications deliver audio or video streams from a server to a client. Commercial products like Real Audio, Windows Media Player are examples of streaming applications. These applications do not need any human interaction.

Real-time multimedia applications can be subdivided further into two categories - interactive and non-interactive. Applications like Internet Protocol Television (*IPTV*) and Video On-Demand are examples of non-interactive real-time multimedia applications. All types of conferencing applications fall under the category of interactive real-time multimedia applications. Examples of popular conferencing applications are Skype, Google Talk, Windows NetMeeting, and Yahoo Chat. Conferencing multimedia applications need human interaction. A brief overview of these two categories of the multimedia applications will be provided in the next two sections of this chapter.

1. Non Real-Time Multimedia Flows: Streaming Applications

Multimedia streaming refers to transmission of stored video or audio over the Internet. There are two modes for transmission of stored multimedia content over the Internet - download mode and the streaming mode. In the download mode, a user downloads the entire multimedia file and then plays it back. This is good enough for end-users who are willing to wait for the entire content to be delivered on their machine before opening the file to see or hear the content. Multimedia files are generally big in size. They take some time to be downloaded. The downloading time depends on the bandwidth of end-user connectivity to the network. IP networks have no problem in dealing with this kind of content delivery. In fact, a reliable data transfer protocol like TCP is used for delivering the multimedia content to the end-user.

Stream mode is different from the download mode. In the streaming mode, the multimedia content need not be downloaded in full. The content is played out while parts of the content are being received and decoded. The downloading and viewing happen simultaneously. Multimedia streaming has bandwidth, delay and loss requirements. The design of best-effort networks without any provisions for Quality of Service (QoS) guarantees is not good enough to support this kind of content delivery.

The current Internet does not offer any QoS guarantees to streaming video. In addition, for multicast applications, it is difficult to efficiently support video while providing service flexibility to meet a wide range of QoS requirements from users. Thus, designing mechanisms and protocols for Internet streaming video poses many challenges. Many strategies have been developed by researchers to make sure that certain basic QoS guarantees can be provided to multimedia applications by best-effort IP networks.

Wu et. al. [2] describe a comprehensive framework for video streaming applications and summarize the research work in each category of the framework. They identify six areas that they believe to be the core constituents of a video streaming architecture. Although Wu et. al. concentrate on video streaming in their paper, some of the survey presented in their paper is equally applicable in the context of audio and other multimedia applications. These six areas are:

1. Video/Audio compression: Raw audio and video must be sampled and compressed before transmission over the network. Many different compression schemes have been developed to achieve this objective.

2. Application layer QoS control: Different application layer QoS control strategies have been devised in order to serve end-users under varying best-effort network conditions. These application-layer QoS techniques are either used for congestion avoidance or error control. Congestion avoidance can be defined as the efforts made by an application to prevent or respond to overload conditions. In effect, this means reducing the end-to-end delay and loss of the multimedia packets traversing through best-effort networks.

   The objective of error control schemes is to devise strategies to recover from dropped packets and late packet arrivals to improve presentation during the transmission of real-time audio and video data streamed over unreliable IP networks. Error control mechanisms include forward error correction (FEC), retransmission, error-resilient encoding, and error concealment. Perkins, Hodson, and Hardman [3] provide a nice overview of packet loss recovery techniques for streaming audio. Wah, Su, and Lin [4] wrote a survey paper with the classification and details of various popular error-concealment schemes.

3. Continuous media distribution services: Best-effort networks do not have any inherent provisions to provide QoS support for real-time multimedia streaming

flows. Different frameworks like network filtering, multicasting, and caching have been developed in order to modify best-effort networks to provide some kind of QoS support. These frameworks have been developed and implemented in overlay networks built on top of the existing network. An overlay network can be defined as "a virtual network of nodes and logical links that is built on top of an existing network with the purpose to implement a network service that is not available in the existing network." [5].

4. Streaming servers: Streaming servers are the software programs responsible for delivering multimedia content streams to end-users. Functionally, they can be compared to web servers. However, streaming servers deliver the content at the exact data rate associated with the compressed audio and video streams. Streaming servers have to deliver media components like audio and video in a synchronous fashion. A lot of research work has been done in developing operating system support for streaming servers. Process, resource, and file management strategies for an operating system need to be different for supporting a streaming server as compared to a normal file or web server. Some of the challenging issues in designing storage systems for streaming servers involve - (a) increasing throughput with data striping, (b) increasing capacity of storage of large multimedia files using concepts like tertiary and hierarchical storage, and, (c) devising fault tolerance techniques to prevent breakdown in streaming services in case of drive failures.

5. Media synchronization mechanisms: Media synchronization refers to maintaining the temporal relationships within one data stream and between various media streams. There are three levels of synchronization - intra-stream, inter-stream, and inter-object.

The basic unit of multimedia content, like the video frame, has to maintain strict time deadlines. If it does not, the presentation of the content from the point of view of human sensory perceptions gets distorted. Therefore, intra-stream synchronization between the fundamental units is important for the final presentation of the content to end-user.

If an end-user is watching a streaming movie, the entire process involves two different streams - audio and video. If the audio stream is not synchronized with the video stream, it will lead to an unpleasant user experience. This explains the importance of inter-stream synchronization between various streams in the context of multimedia. This is a higher level synchronization as compared to intra-stream synchronization.

The highest layer of a multimedia document is the object layer, which integrates streams and time-independent data such as text and still images. Synchronization at this layer is referred to as inter-object synchronization.

6. Protocols for streaming media: Numerous innovative protocols have been designed for streaming media delivery from a server to a client. Real-time Transport Protocol (*RTP*) has been designed on top of the User Datagram Protocol (*UDP*) for delivering multimedia content over the Internet. RTP Control Protocol (*RTCP*) transfers information between a client and a server regarding the state of the network. This information can be used selectively by the application to provide congestion avoidance and flow control. The Real Time Streaming Protocol (*RTSP*) is a protocol for use in streaming media systems which allows a client to remotely control a streaming media server, issuing VCR-like commands such as “play” and “pause”, and allowing time-based access to files on a server. RTSP is a session control protocol. The combination of the protocols

in the application, session, and transport layer, provide a protocol framework to deliver multimedia content over the Internet.

The current research work is not oriented towards exploring problems related to non real-time multimedia applications. The streaming applications do not need to worry too much about human interaction like conferencing applications. This provides a lot of flexibility to accommodate the variable delays suffered by the packets when traversing through best-effort networks. Let us consider a streaming application that delivers its contents using a combination of the protocols like RTP, RTCP, and UDP over best-effort networks. A simple solution to solve the QoS guarantee problems will involve a good adaptive buffering strategy on the receiver side coupled with an effective congestion avoidance strategy based on RTCP information provided by the receiver to the sender. There is an initial delay before playback starts at the receiver's end. This can take care of the flow control and congestion avoidance problems arising out of the multimedia content traversing over networks without any QoS guarantees. The sender can vary the bit-rate of the information being pushed into the network using different strategies like hierarchical or layered coding of the multimedia content. Figure 1 depicts this simple solution. The source of this figure is the textbook by Stallings [6].

2. Real-Time Multimedia Flows: Conferencing Applications

Audio and video conferencing applications are real-time multimedia applications. However, conferencing applications that use best-effort networks to deliver information have features that make congestion avoidance and flow control more challenging as compared to streaming applications. Most of these challenges arise out of limits placed by human perception factors. For example, if two people are conversing with

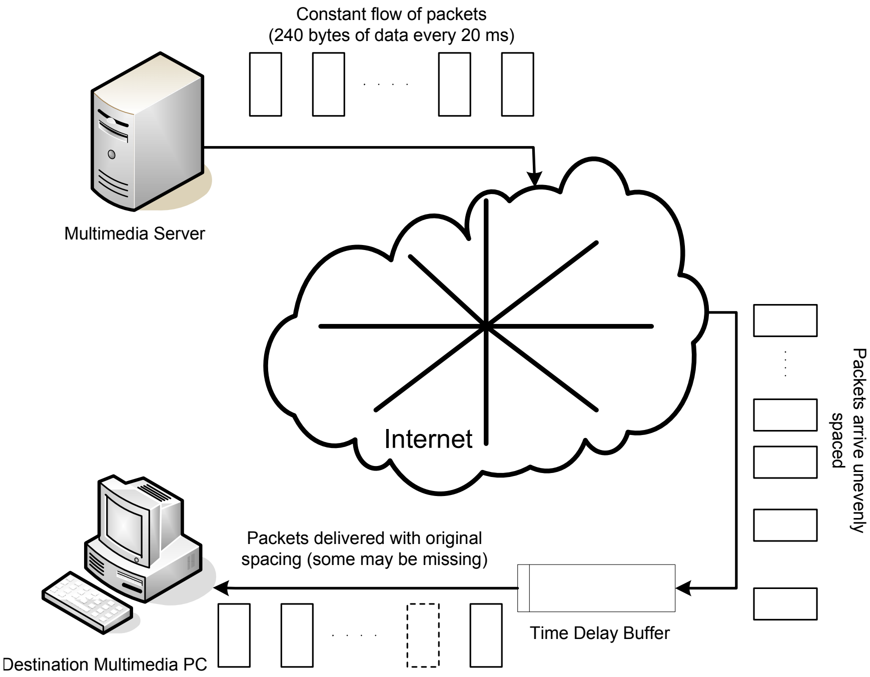

Constant flow of packets
(240 bytes of data every 20 ms)
Multimedia Server
Internet
Packets arrive unevenly spaced
Packets delivered with original spacing (some may be missing)
Time Delay Buffer
Destination Multimedia PC

Fig. 1. A simple solution for addressing QoS issues of real-time streaming application over best-effort networks [6].

.

each other using phones that send information through packet switched best-effort networks, the maximum one-way delay that the packets can tolerate is 200 ms [7]. This upper value of end-to-end delay has been shown to be commercially acceptable. When the end-to-end delay reaches 800 ms, adverse psychological factors impede normal telephonic conversation. A flexible end-to-end delay range between 200 ms to 800 ms is conditionally acceptable for a short portion of the conversation. However, occurrences of delays in the above-mentioned range should be rare and far apart. Yensen, Goubran, and Lambadaris [8] report that long end-to-end delays are unsuitable for interactive conversations since they impair normal conversation between participants and may intensify the effect of acoustic or network echoes.

a. Voice Over Internet Protocol

Transmitting voice signal over IP networks using a set of facilities and protocols for setting up and managing the sessions can be defined as Voice-over-IP (*VoIP*). Internet Telephony is one of the typical applications of VoIP. VoIP implementation can be categorized into two types of architectures [9]:

- based on H.323 framework, and
- based on Session Initiation Protocol (*SIP*) framework.

These two architectures provide far more services than just setting up voice calls. H.323 is a set of protocols for voice, video, and data conferencing over packet-based network. SIP is an application-layer control signaling protocol for creating, modifying, and terminating sessions with one or more participants. The basic architectures of these two implementations are the same. They consist of three main logical components: terminal, signaling server and gateway. They differ in the specific definitions of voice coding, transport protocols, control signaling, gateway control, and call management. A good tutorial about both the frameworks and their comparison has been provided in the article by Glassman, Kellerer, and Müller [10]. SIP based systems do not work well in network environments guarded by firewalls.

One of the latest and most exciting entrants into the world of VoIP is Skype. Most machines connected to the Internet are protected by several layers of firewall. These range from personal firewalls installed on Personal Computers (*PCs*) to firewalls used by the Internet Service Providers (*ISPs*). Most end-users also block unutilized ports in their machines because of security issues. This prevents VoIP applications from using unutilized and unregistered ports to transfer packets.

Skype takes care of the voice information transport problems through the network arising because of blocked ports due to firewall. The voice payload of Skype is sent using UDP in the transport layer. It uses proprietary protocols and techniques that allow encoded audio packets to go through any firewall by using the http protocol over TLS/SSL (*https*) port. Skype even works with port 80 meant for http, but it does not use the SIP standard. Therefore, it can not work with other existing Internet Telephony systems.

Skype allows users to communicate using peer-to-peer (*P2P*) networking. If a user wants to communicate using Skype, he/she has to start by logging into Skype. The Skype central server authenticates the user and logs him/her in. After the process of authentication is over, the user initiates a voice call. Some intermediate nodes termed as the *Supernodes* help the end-user to locate the machine of the other user that he/she wants to call. Location of the other end-user with the help of the *Supernodes* is followed by establishment of connection between the two users. Unlike other Internet telephony applications, voice call connection between the two end-users is not routed through a central server. This improves performance by not loading the central server. The quality of the call is not depended on the load of the central server but on the available bandwidth of the set of links that the voice packets traverse to reach one end from the other end. Figure 2 shows the functionality of P2P Internet telephony application Skype. The source of this figure is the book on Skype by Gough [11].

Skype has serious issues regarding security. If the application decides to mark a machine as one of the *Supernodes* in the Skype P2P network, the user of the machine has no control over the third-party traffic that is routed through his/her machine. Skype encrypts data using 256-bit AES encryption. The symmetric AES keys are negotiated between the server and the client using 1536 and 2048-bit RSA. Packet filters and proxy servers that examine the content of the data in the network are ineffective to prevent Skype traffic from going out or coming in. Gough [11] writes

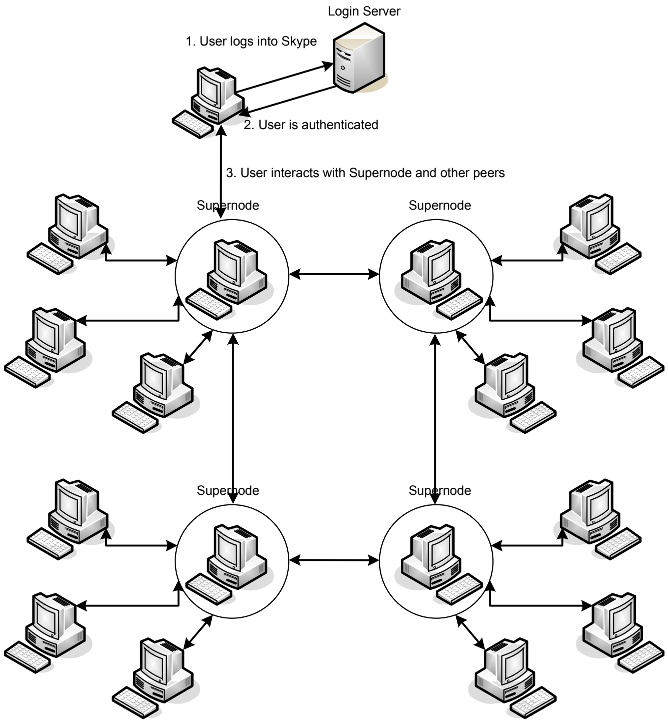


Fig. 2. Skype P2P Internet telephony application [11].

that "there is no straightforward way to block Skype" in a corporate network.

Skype uses the codec - iLBC. This codec is a multi-mode variable bit-rate codec that varies send rate between 3 kbps to 16 kbps depending on the quality of the link between nodes at the beginning of a session. iLBC also employs error concealment scheme to recover from lost packets during a session. However, Skype does not employ either flow control schemes or congestion avoidance schemes. This means that Skype doesn't react to congestion.

b. Video Conferencing over the Internet

The issues facing the networking community related to transporting real-time video along with audio for video conferencing applications over best-effort networks are similar to the issues faced in the case of VoIP applications. Transporting real-time interactive video over best-effort networks requires higher capacity than that of VoIP applications. In order to efficiently utilize the bandwidth available in the network, the video pictures are compressed. MPEG encoding is one of the most popular compression techniques [12]. Other encoding techniques are Motion JPEG [13] and H.261 [14].

Baldi and Ofek [15] show that MPEG constant bit rate (CBR) video encoding is not advisable for high-quality videoconferencing applications. In videoconferencing applications, a picture (video frame) is captured, compressed, and sent over the network once there is enough data ready to be sent in a packet. This raises two important questions about conferencing applications:

1. When will the compressed video data units be ready to be sent? Data unit generation during compression is difficult to predict.

2. How many data units will be generated after the compression of each picture?

Both of these questions need to be answered in order to reserve communication resources inside the network. A lot of research has been done to solve the problem of predicting traffic generated by multimedia sources, especially video [16], [17], [18], [19], [20], [21], [22], and [23].

In terms of implementation, current video conferencing architectures are same as that of audio conferencing architectures. Like audio conferencing applications, video conferencing applications use the SIP or H.323 set of protocols to transmit video from one end to another. Applications like Skype and Windows NetMeeting have provisions for sending video feed of the speakers along with the audio. The basic difference of the video conferencing applications lies in the encoding strategies, bandwidth requirements, and the task of synchronization of the voice with the pictures. Besides these issues, the problems faced by the video conferencing applications over best-effort networks without any QoS guarantees are similar to the problems faced by the audio conferencing Applications.

The current research will discuss flow control of unicast multimedia flows from the point of view of VoIP applications rather than video applications. The algorithms developed for flow control of unicast VoIP flows through best-effort networks can be applied for video applications too. However, studies related to this aspect have not been performed. Audio codecs are less complex when compared to the video codecs. The emphasis of the current work is not to develop new audio or video codecs but to develop flow control algorithms. The relatively simple nature of audio codecs as compared to video codecs helps to keep the focus on the control algorithm development. Typical traffic generated by audio codecs are used in this study to demonstrate the efficacy of the developed control algorithms. The proposed control schemes can be incorporated in the next generation of audio or video codecs during the implementation stage. Henceforth, whenever the word multimedia is mentioned in the dissertation, it will be in the context of audio applications. All definitions of technical terms will be defined and explained from the point of view of multimedia audio flows.

B. Quality of Service

The design of best-effort networks does not guarantee any kind of QoS to multimedia flows traversing the networks. Multimedia flows originating from conferencing applications depend on several factors for providing a good end-user experience. These factors are:

1. Loss of packets: Packets lost or dropped in best-effort networks while traveling from the sender to the receiver.

2. End-to-end packet delay: End-to-end packet delay can be defined as the time difference between the time instant at which the contents of a packet is played out at the receiver and the time instant at which the packet is generated at the sender.

3. Delay jitter: The end-to-end delay change between two consecutive packets in a stream [24].

The above mentioned factors that characterize the QoS of multimedia flows have to be described more rigorously in terms of mathematics. In order to describe the three factors that affect QoS, the following notation is used. These notations have been partly adopted from the paper by Ramjee, Kurose, and Towsley [25]: s_i is the time at which packet i is generated at the sending host, a_i is the time at which packet i is received at the receiving host, and p_i is the time at which packet i is played out at the receiving host. Let D_{prop} be the propagation delay experienced by a packet while traveling from the sender to the receiver through the network. Propagation delay is assumed to be constant for every packet. D_i^t is the transmission delay of the packet i. Let v_i be the queuing delay experienced by the packet i as it is sent from the source to the destination host. b_i is the amount of time spent by the packet i in the buffer of the receiver awaiting its scheduled playout time. The arrival time at the receiver host for packet i can be written down as

$$a_i = s_i + D_{prop} + D_i^t + v_i. \tag{1.1}$$

The playout time of packet i is the sum of the time taken

$$p_i = a_i + b_i. \tag{1.2}$$

Therefore, d_i, the end-to-end packet delay of packet i can be defined by the following equation:

$$\begin{aligned} d_i &= p_i - s_i, \\ \Rightarrow d_i &= D_{prop} + D_i^t + v_i + b_i, \end{aligned} \tag{1.3}$$

The total delay, n_i, introduced by the network can be written as

$$n_i = D_{prop} + D_i^t + v_i. \tag{1.4}$$

Delay jitter d_i^j can be expressed as

$$d_i^j = |d_i - d_{i-1}|. \tag{1.5}$$

Packets lost are not counted in the jitter measurement.

As mentioned earlier, one-way end-to-end delay of voice packets traversing the Internet should not exceed 200 ms. The upper bound on the jitter tolerance is determined by the maximum size of the playout buffer at the receiver end. The packet loss tolerance is not only dependent on how effective the error recovery techniques are but also on the encoder used. Markopoulou, Tobagi, and Karam [26] state that the encoding scheme employed by the audio application has great influence on the maximum allowed packet loss in the network to sustain acceptable quality.

The Mean Opinion Score (*MOS*) is a scale that provides a quantitative estimate of the perceptual voice quality. MOS tests for voice are specified by ITU Telecommunication Standardization Sector (*ITU-T*) [27]. The MOS scale varies from 1 to 5, with 1 representing very bad quality audio and 5 representing excellent quality audio without any impairments due to encoding and decoding. Markopoulou, Tobagi, and Karam [26] are interested in the "results pertaining to speech clipping, whereby speech clips of fixed durations are uniformly distributed across time, with loss rates ranging from 0% to 20%." They summarize that for audio applications employing packet loss concealment techniques and speech clipping duration of 10-ms loss, the MOS drops by roughly $1 - 1.5$ units for every 10% of packet loss. For audio applications that do not employ loss concealment techniques, the MOS drops much faster, by roughly one unit for every 1% of packet loss. Larger loss durations used for speech clipping result in increased degradation. By conventional standards of the Plain Old Telephone System ($POTS$) a "toll quality" telephone service has a MOS level of 4.0 or above. MOS and its implications in measuring encoded/decoded voice quality will be discussed later.

Best-effort networks do not provide any guarantee on the bounds of the loss, end-to-end delay, and delay jitter of the audio packets traversing through the network. Therefore, the applications must tolerate some degradation in QoS. However, many times the performance of best-effort effort networks is not good enough for real-time applications.

There are three ways to provide a better framework for better QoS guarantees to real-time multimedia applications, like audio-conferencing:

1. Development of new frameworks in network architecture to provide higher QoS assurances to applications. The network should be able to provide information before the establishment of a multimedia session about the QoS that it can support in the current state. Moreover, it should be able to deliver the performance that it promises while negotiating with the application at the beginning of the session during the session.

2. Improve the end-to-end flow control strategies of the real-time multimedia conferencing applications.

The next two sections of this chapter will discuss these two strategies in detail. Later on, the motivation for the current research will be provided.

1. New Frameworks in Network Architecture to Provide QoS

Many different kind of frameworks have been developed to provide a range of QoS. These can be divided into two broad categories [28]:

- fine-grained approaches, which provide QoS to individual applications or flows,
- coarse-grained approaches, which provide QoS to large classes of data or aggregated traffic.

"Integrated services", a QoS architecture developed by the Internet Engineering Task Force (*IETF*), falls in the first category. "Integrated Services" is associated with the Resource Reservation Protocol (*RSVP*). "Differentiated services" belongs to the second category.

a. Integrated Services

The Integrated Services working group developed specifications for a number of *service classes* designed to meet the needs of some of the application types. This group also defined how RSVP could be used to make reservations using these service classes. A brief description of various service classes are:

- Guaranteed service: Intolerant multimedia applications need all the packets to arrive before their respective deadline. In this service, the application negotiates with the network for an upper bound on the network delay. The network provides this guarantee and applications set their playback point so that no packet arrives late.

- Controlled load: Tolerant, adaptive multimedia applications perform quite well on networks that are lightly loaded. The aim of this service is to emulate the conditions of a lightly loaded network within a congested network. This is achieved by simultaneously using admission control to limit the total amount of controlled load traffic on a link and use queuing mechanism like Weighted Fair Queuing (*WFQ*) to isolate the controlled load traffic from other traffic.

Flowspec is the set of information that the application provides to the network in order to negotiate the level of services. There are two separable parts of the flowspec - the part that describes the traffic characteristics of the flow *TSpec* and the part that describes the service requested from the network *RSpec.* RSpec is very service specific and relatively easy to describe. If the application needs a controlled load service with no additional parameters, RSpec can describe it easily to the network.

When some new flow wants to receive a particular level of service, admission control looks at the TSpec and RSpec of the flow and tries to decide if the desired service can be provided to that amount of traffic without causing any previously admitted flow to receive worse service than it had requested. If it can provide the service, the flow is admitted. If it can not provide the service, the flow is rejected admission. Issues of policy and resource availability are addressed when admission control decisions are made.

A lot of information needs to be provided to the network if we want the network to provide QoS support for real-time multimedia applications. While there have been a number of setup protocols proposed for the Internet, Resource Reservation Protocol (*RSVP*) is the most popular choice. RSVP aims to support both multicast flows as well as unicast flows.

Let us take the example of a sender trying to establish a connection with a receiver using RSVP. The receiver needs to know what traffic the sender is likely to send by reading the sender's TSpec so that it can make appropriate reservation. The receiver also has to know the path that the packets from the sender take in order to reach the receiver. This helps in establishing reservation of resources on every router on the path. These two actions are achieved by the sender sending a message to the receiver that contains the required TSpec. This message is called the PATH message. This message is read by every router in the path during its traversal from the sender to the receiver. After receiving the PATH message, the receiver responds by sending back a RESV message. This message contains the sender's TSpec and an RSpec describing the requirements of the receiver. Each router on the path looks at the reservation request and tries to allocate the necessary resources to satisfy it. If the reservation can be made, the RESV request is passed on to the next router. If not, an error message is returned to the receiver who made the request. As long as the receiver wants to retain the reservation, it sends the same RESV message once every 30 seconds. Figure 3 shows the process of making reservations of resources on the network using the Intserv framework.

After informing the network about the service guarantees needed by the multimedia flow and reserving the resources of the network, the next task is to ensure that the network entities have the mechanism to provide the guarantees. Every network entity between the sender and the receiver needs to perform two tasks - associate each packet with the appropriate reservation so that it can be handled correctly (*classifying packets*) and manage the packets in the queues so that they receive the service that has been requested (*packet scheduling*). Classification of the packets is done by examining the following fields in the IP header: source address, destination address, protocol number, source port, and the destination port. Based on the information

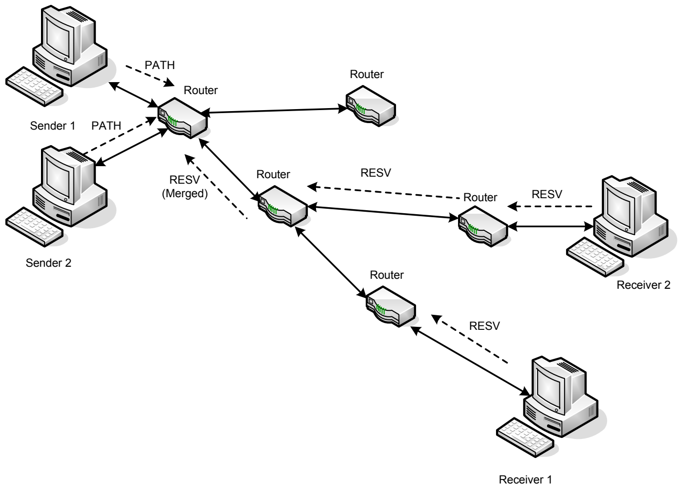


Fig. 3. Making reservations using the Intserv framework.

gathered from the five fields the packets can be categorized into a suitable class of flow. Packet scheduling is implemented by using algorithms like Weighted Fair Queuing (*WFQ*) to manage the queues in the routers that support the Intserv framework.

In spite of a functional design to solve the QoS guarantee problem over best-effort networks Intserv failed to be deployed in large scale by the Internet Service Providers (*ISPs*) for one reason - scalability. The state of the flows passing through the routers need to be stored in memory and refreshed periodically for the Intserv framework to succeed. As the number of flows passing through a router increases, the memory requirement to store the state of the flows becomes a bottleneck. Each router deploying the Intserv need to classify, police, and queue each flow passing through it. Admission control decisions need to be made every time a flow requests reservation. Some other approaches that do not need the "per-flow" state have been developed. These approaches fall under the "Differentiated services" discussed in the next subsection.

b. Differentiated Services

The Differentiated services framework divides the flows into few categories and allocates resources for each of these categories. Some proposed approaches to Differentiated Services divide traffic into two classes. The IETF decided to take the old Type of Service (*TOS*) byte from the IP header and redefine it. Six bits of this byte have been allocated for the Differentiated Services Code Points (*DSCP*). Each DSCP is a 6 bit value that identifies a particular "per-hop behavior" (*PHB*) to be applied to a packet. PHB is a term that is associated with the individual routers in the network rather than end-to-end services. The PHB information conveyed through the DSCP by the packets provide direction to the routers in the network that implement the Differentiated Services framework on how to treat the individual packets. The routers do not have to maintain the state of each flow. They just take action depending on the PHB desired by the packets.

Some very basic types of PHB are:

Expedited Forwarding (*EF*) : Packets marked for EF should be forwarded by the router with minimal delay or loss. There are many ways in which EF can be implemented. One is to give the EF packets priority over all other packets. Another is to perform a WFQ between the EF packets and the other packets, with the weight of EF set sufficiently high that all EF packets can be delivered quickly.

Assured Forwarding (*AF*) : This PHB is based on "RED with In and Out" (*RIO*) or "Weighted RED". Both of these approaches are enhancements to the basic RED algorithm [29]. The end-user (corporate/individual) pays the ISP to provide QoS guarantees to the flows originating from the domain specified by the user. The ISP provides the guarantee by agreeing to provide QoS for a certain fixed rate of the traffic from the end-user. The edge router employed by the ISP can determine whether the traffic sent by the customer is within the traffic profile that the customer is paying for. If the traffic exceeds the profile, the packets are marked as “out”. Otherwise, they are marked as ”in”. This task is generally performed at the edge of the networks.

RIO employs two separate queues on the router for the packets marked “in” and “out”, respectively. These queues employ Random Early Detection (RED) as the queue management scheme. The drop probability curves of the two queues are different. The “out” curve has a lower $MinThreshold$ than the “in” curve. This means that under low levels of congestion, packets marked “out” will be discarded first by the RED algorithm. If the congestion becomes more serious, a higher percentage of “out” packets are dropped. If the average queue length exceeds Min_{in} then RED starts to drop the “in” packets as well.

The combination of the profiling strategy at the edges and RIO in all the other routers of the service provider’s network should provide the end-user with high assurance that packets within his profile are delivered according to the promised guarantees.

Weighted Fair Queuing Scheduler : A third way to provide “Differentiated Services” is to use the DSCP value to determine which queue to put a packet into in a weighted fair queuing scheduler. One code of DSCP can point to a “best-effort” queue. Another code can point to select the “premium” queue. Depending on the weights chosen, the “premium” queue packets will be serviced more frequently than the “best-effort” queue. The success of this strategy depends on the careful setting of the weights of different queues. This approach can be extended to allow more than two classes represented by different code points.

As mentioned in this subsection earlier, the effectiveness of “Differentiated Services” depends on the right selection of parameters. Moreover, the success of “Differentiated Services” to provide QoS guarantees between machines located in different domains depends on all the domains that fall in the path of travel of the packets from the source to the receiver, adopting the specified framework. “Differentiated Services” requires investment from network managers. Many private networks have adopted the strategies laid down by the “Differentiated Services” framework to provide QoS guarantees for real-time multimedia flows within their domain. Carpenter and Nichols [30] provide a more detailed explanation of the current status of the implementation and deployment of “Differentiated Services” over best-effort networks in order to provide QoS guarantees.

2. End-to-End Flow Control Strategies to Improve QoS

In spite of the development of frameworks like Intserv and “Differentiated Services” to enable best-effort networks to provide QoS guarantees, end-to-end flow control remains important. Most of the multimedia applications, real-time and non real-time, use the User Datagram Protocol (*UDP*) in the network layer to transport information over the networks from one end to the other. Unlike TCP, UDP is a simple protocol that does not have any control mechanism to avoid congestion. It also does not have any mechanism for flow control. If a multimedia application generates information at a certain rate of X kbps and uses UDP to transport this information over the Internet, UDP maintains the rate of sending information into the network irrespective of its state. If congestion occurs, UDP flows do not help in relieving the congestion. In fact, UDP flows might exacerbate congestion.

Floyd and Fall [31] state the potentially negative impacts of increasing deployment of non-congestion-controlled best-effort traffic on the Internet. They contend that the negative impacts of deploying non-congestion-controlled traffic on best-effort networks range from extreme unfairness against competing TCP traffic to the potential for congestion collapse. The end-to-end congestion control mechanisms of TCP have been a critical factor in the robustness of the Internet till now. TCP ensures fairness in sharing bandwidth between competing flows. It also enables reliability by incorporating features like retransmission.

Kim, Won, and Hong [32] performed a detailed study to characterize the Internet traffic from the perspective of flows. The experiments for the study were conducted during three weeks of February and March 2004. In order to collect IP traffic trace data, they deployed a real-time traffic monitoring and analysis system for high-speed network links on the Internet junction at the campus of Pohang University of Science and Technology, Korea (*POSTECH*). Some of their experimental observations related to the composition of the Internet traffic and relevant to the current research work are:

- Over 90% of the total Internet traffic to and from POSTECH in terms of packets, as well as bytes, is TCP traffic.
- TCP is used by the majority of current Internet applications.
- The number of total UDP flows is about two times greater than the number of total TCP flows.

Fomenkov, Keys, Moore, and Claffy [33] analyzed more than 4000 trace samples collected at a number of academic, research, and commercial sites during years 1998-2003. They state that, typically, TCP traffic is between 60% and 90% of the total load. UDP traffic is between 10% and 40%. All other protocols combined produce less than 5 percent. The aggregated average composition of traffic by protocols at all the sites whose data is available to Fomenkov et. al., except one, is shown in the Table I. Fomenkov et. al. also observe that the protocol mix of the Internet traffic has remained stable over time despite the increase in the number of UDP applications in recent years.

Table I. Composition of traffic by protocols for different metrics [33].

Metric	TCP(%)	UDP (%)	Other (%)
Bytes	83 ± 11	16 ± 11	1 ± 1
Packets	75 ± 12	22 ± 11	3 ± 2
Flows	56 ± 15	33 ± 10	11 ± 7

Multimedia applications have become popular on the Internet. Their increasing popularity is going to change the composition of the current Internet traffic. Floyd and Fall [31] define congestion collapse in the network as a state “when an increase in the network load results in a decrease in the useful work done by the network.” Congestion collapse due to undelivered packets happens when the bandwidth of the network is used up in delivering packets that are dropped on the way to their destination. The increasing rate of deployment of applications with no feedback (e.g. multimedia) can lead to congestion collapse situations in the future. Therefore, there is a need for development of effective end-to-end control mechanisms in order to support multimedia flow packets in best-effort networks.

Some reactive congestion avoidance and flow control strategies based on RTP/RTCP deployed on top of UDP have been developed to make the real-time multimedia flows adapt better to congestion in a best-effort network. However, these solutions have remained confined to the academic world. None of the existing audio/video conferencing applications in the real world use congestion avoidance and flow control. Lot of recent work in this area has also focused on developing control strategies for multimedia applications that can be grouped under TCP Friendly Rate Control (*TFRC*). More discussion about the existing research related to development of congestion avoidance and flow control mechanisms for real-time multimedia flows is presented in chapter II of this dissertation.

Krasic, Li, and Walpole [34] make the case that TCP is a viable and attractive choice for quality-adaptive video streaming. They discuss the main challenges for video applications using TCP, which are due to TCP retransmissions and congestion avoidance and how they are not sufficient to disqualify TCP from transporting multimedia streams. Audio/video conferencing applications are far more strict in their network latency requirements as compared to streaming applications. Therefore, the arguments put forth by Krasic, Li, and Walpole do not diminish the need for development of flow control mechanisms for real-time multimedia flows.

Transport protocols like TCP can not be used for delivering multimedia flows. This is because TCP, in spite of developments like fast retransmit and fast acknowledgement, is too conservative to maintain the deadlines for transmitting real-time multimedia flow packets through best-effort networks. TCP uses reactive control by additively increasing or multiplicatively decreasing the congestion window. TCP uses a timeout mechanism at the sender's side to decide whether a packet has been lost on the Internet because of congestion in the path. A timeout of the clock is an indication of congestion in the network. If the acknowledgement of the packet does not arrive from the receiver within the time reserved for timeout at the sender side, TCP cuts down its congestion window size by half. Current versions of TCP like TCP-Reno use *duplicate ACKs* to determine congestion along with timeout of the clock. The time period of the timer clock is maintained by calculating the Round Trip Time (RTT) from the acknowledgement of the packets sent by the receiver to the sender.

Martin, Nilsson, and Rhee [35] address some of the issues related with the delay based congestion avoidance control schemes employed by newer versions of TCP like TCP-Vegas. The authors define several performance metrics that quantify the level of correlation between packet loss and RTT in the network. Based on measurement analysis they contend that although there is useful congestion information contained within RTT samples, the level of correlation between an increase in RTT and packet loss is not strong enough to allow a TCP source to reliably improve throughput. Delay based congestion avoidance schemes are able to reduce the packet loss rate experienced by a connection but these algorithms react unnecessarily to RTT variations that are not associated with packet loss. The result is degraded throughput.

C. Research Objectives and Proposed Approach

There is a need to deploy some kind of end-to-end flow control scheme for real-time multimedia applications. As seen earlier, employing TCP for these kind of applications is not an option. The need for end-to-end flow control is going to increase in the future with the popularity of using the Internet for delivering audio-visual content. None of the popular real-time audio or video conferencing applications employ adaptive flow control. Many existing applications probe the network for bandwidth availability and decide the rate of sending information to the network at the beginning of the session. They maintain this rate throughout the entire session. This is not an optimal solution as multimedia applications perform best when they are adaptive to the network conditions.

Adaptive end-to-end flow control schemes also help real-time multimedia applications be better citizens of the Internet. Assuming that the installed bandwidth capacity of the backbone links of the Internet fail to keep pace with the demand of consumers and the composition of the Internet traffic undergoes changes on account of more contribution by UDP flows, the issues regarding congestion of the network caused by real-time multimedia flows will become more important in the future.

Existing control schemes devised for multimedia real-time streams employ suitable variations of reactive control strategies. The event of congestion has already taken place in the network and the application is merely reacting to that event in order to help alleviate the congestion. Reactive control schemes are very effective for systems that have instantaneous feedback and no time delays in their dynamics. Most electro-mechanical systems employ different kinds of reactive control strategies. However, network dynamics have delays and measurement feedback is not instantaneous. There is a variable time delay associated with the feedback from the receiver to the sender. This complicates the case when reactive control is applied in these kind of systems.

Instead of reactive control, it might be better to employ predictive control strategies in order to provide end-to-end flow control. Predictive control refers to a class of control algorithms that compute a sequence of manipulated variable adjustments in order to optimize the future behavior of a system. The algorithms make explicit use of a model of the process to be controlled. Such approaches have proved to be very effective for systems with long delays. Developing a predictive control strategy for a single flow of a real-time multimedia application that anticipates congestion in the network before it actually takes place and devising a control signal to manipulate the future state of the flow is the main objective of the current research. The main class of multimedia applications utilized to demonstrate the findings is VoIP.

Additionally, this research will investigate the following:

- Develop some feasible reactive control strategies and compare their performance with the performance of predictive control strategies.

- Demonstrate the effects of employing end-to-end flow control schemes in enabling real-time multimedia flows to become more friendly with respect to other flows in the network.

To summarize, the objective of the research is to improve the QoS delivered to the receiver end in audio conferencing applications when they communicate through best-effort networks, while trying to remain friendly to the other competing flows. This will be done by the development of novel adaptive end-to-end flow control strategies.

Figure 4 provides a pictorial representation of the end-to-end flow control framework for VoIP applications that will be investigated in the current research. From the point of view of control systems terminology, flow transport over the best-effort network is the system that needs to be modeled and controlled. The flow bit-rate generated by the encoder at the sender side is the input signal. The entire cross-flow traffic is the disturbance to the system. Flow accumulation in the network is the feedback signal supplied by the receiver to the sender. More details about the accumulation signal as the output of the system will be discussed in later parts of this dissertation. The control strategy is based on accumulation feedback. The control decisions taken by the control algorithm is implemented by the encoder (actuator). The encoder implements these decisions by changing its coding mode to modify the bit-rate it generates in a suitable manner.

D. Research Contributions

The contributions of the current research work are as follows:

1. Using accumulation signal for development of flow control schemes for hard

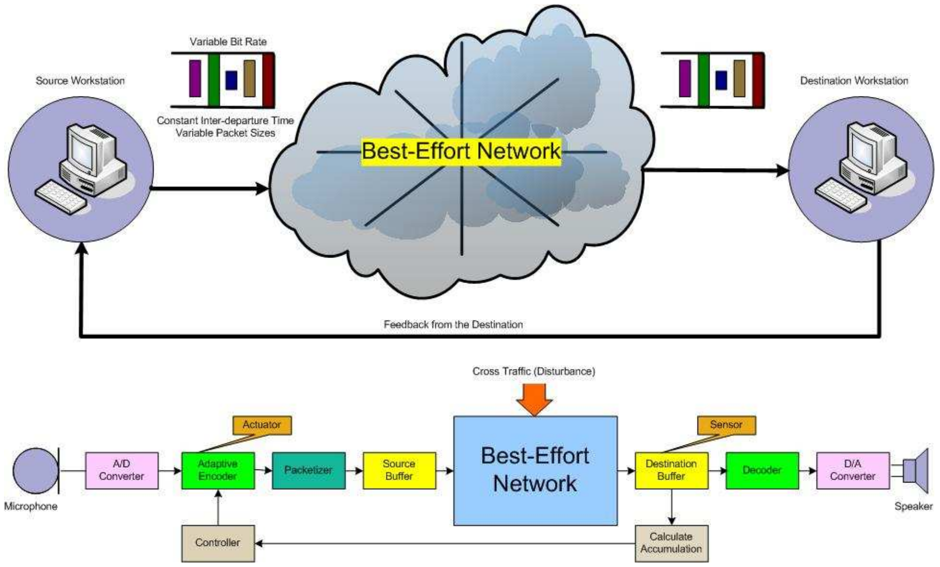


Fig. 4. Framework for end-to-end flow control for VoIP applications.

real-time multimedia applications

2. Development of two simple reactive flow control schemes - Linear Control Law (LCL) and Nonlinear Control Law (NCL).

3. Exploiting the process control framework to develop predictive control schemes for interactive real-time multimedia applications transmitting over best-effort networks

4. Determining the feasibility of control schemes to provide good QoS while utilizing less bandwidth in varied network conditions of different scales

E. Organization of the Dissertation

The dissertation has been divided into eight chapters. Each chapter deals with one aspect of the research. The current chapter introduces the problem to the readers and provides motivations behind the research. Chapter II provides a brief review of the existing literature on end-to-end flow control of the real-time multimedia flows over the Internet.

A brief overview of the technical details about the voice codecs used in VoIP applications and the algorithms employed by them is covered in chapter III. Determining the quality of encoded-decoded voice is an important topic for the current research. Chapter IV discusses the concepts of determining quality of voice in the context of current research. The quality metrics discussed in chapter IV help to depict the results of the research in later chapters.

Chapter V explains the failure of the search for a suitable real world test bed in order to demonstrate the efficacy of the designed adaptive flow control strategies for real time multimedia applications over best-effort networks. In lieu of the real world test bed, simulated network topologies using the network simulator ns-2 are used to perform the experiments. Chapter V also discusses the structure of two topologies designed for achieving the objectives of the current research.

Chapter VI describes the modeling of single flows in best-effort networks. This chapter begins with a discussion about the limitations of reactive control strategies. It lists different types of signals (packet loss rate, end-to-end-delay, accumulation) used for modeling best-effort networks and their relative merits. Accumulation signal is utilized for modeling the flows. Theoretical derivations of the state-space equations to model conservative real-time multimedia flows is presented. This is followed by the description of the empirical linear models of multimedia flows and their relative merits. A brief summary of the various failed linear as well as nonlinear modeling strategies is also provided.

Chapter VII describes the theoretical foundations for the design of flow control algorithms. Model Predictive Control (MPC) formulation to flow control for real-time multimedia flows over best-effort networks is presented in this chapter. This chapter also investigates the performance of the different control schemes and determines how effective they are in controlling single flows while trying to provide a good end-to-end QoS. All the experiments are performed in a simulated environment using the framework provided by ns-2.

Chapter VIII summarizes the research and provides final conclusions. This is followed by a brief discussion on the limitations of the current work. The investigation into every aspect of the flow control problem in real-time multimedia applications over best effort networks is not complete. The experiments raise many questions about the theoretical as well as practical aspects of flow control schemes presented in the current work. These question need to be answered in the future. Therefore, chapter VIII also recommends areas of the current research that need to be further investigated in the future.

CHAPTER II

LITERATURE REVIEW

It is important to review the existing literature related to adaptive control strategies for real-time multimedia flows over the Internet. Review of these papers helps to understand various aspects of the problem under consideration. The summary of the existing literature related to end-to-end flow control of multimedia flows is divided into two broad categories - audio and video applications. These two categories include literature review of both real-time streaming as well as conferencing applications.

The current research work is focused on real-time audio applications that have very stringent requirements on the loss rates and end-to-end delay of the packets while traversing from the source to destination using the Internet. However, multimedia applications need not only include audio, but other applications like video, online gaming etc. Many ideas that have been developed for one set of applications can also be used for other sets of applications. Therefore, research work related to all kinds of multimedia applications is reviewed in this chapter.

A. Audio Applications

One of the earliest important works regarding adaptive flow control has been done by Ramjee, Kurose, Towsley, and Schulzrinne [25]. They investigate the performance of four different algorithms for adaptively adjusting the playout delay of audio packets in an interactive packet-audio terminal application, in the face of varying network delays. Pinto and Christensen [36] extend the work proposed in [25].

Sanneck, Stenger, Ben Younes, and Girod [37], propose a new error concealment technique for audio transmission. In another paper Liang, Färber, and Girod [38], propose a new receiver-based playout scheduling scheme. This playout scheme estimates the network delay from past statistics and adaptively adjusts the playout time of the voice packets. The adjustment is not only performed in between talkspurts, but also within the talkspurts in a highly dynamic way. Proper reconstruction of continuous output speech is achieved by scaling individual voice packets using a time-scale modification technique which modifies the rate of playout while preserving voice pitch.

Ranganathan and Kilmartin [39] use novel neural networks and fuzzy systems as estimators of network delay characteristics. The estimated delay is used for developing adaptive playout algorithms for flow control. The performance of their proposed scheme is analyzed in comparison with a number of traditional techniques for both inter and intra-talkspurt adaptation paradigms. The design of a novel fuzzy trend analyzer system (*FTAS*) for network delay trend analysis and its usage in intra-talkspurt playout delay adaptation are presented in the paper.

Sisalem and Schulzrinne [40] present a new scheme called the *Loss-Delay based Adjustment Algorithm* (LDA) for adapting the transmission rate of multimedia applications to the congestion level of the network. The LDA algorithm is designed to reduce losses and improve utilization in a TCP-friendly way that avoids starving competing TCP connections.

DeLeon and Sreenan, in a conference paper [41], invoke the idea of using accurate prediction of network delay to adjust the buffer delay. They present an algorithm based on a simple *Normalized Least-Mean-Square* (NLMS) adaptive predictor.

The paper by Casetti, DeMartin, and Meo [42], discusses most of the issues of interest to the current research. In practice, the authors implement a very simple empirical control algorithm to control the flow of audio on the network. This algorithm is similar to the often-used AIMD algorithm.

Sreenan, Chen, Agrawal, and Narendran [43] propose an algorithm that records historical information and uses it to make short-term predictions about network delay with the aim of not reacting too quickly to short-lived delay variations. This allows an application-controlled tradeoff of packet lateness against buffering delay. The algorithm is suitable for applications which demand low delay but can tolerate or conceal a small amount of late packets.

In the paper by Wang et. al. [44], a new rate-based flow control mechanism, based on control theory, is proposed with the feedback of *Buffer Occupancy* (BO). The mechanism applies feedback control to keep BO running to a given level away from buffer overflow and underflow. Beritteli, Ruggeri, and Schembra [45] state that although several TCP friendly algorithms have been introduced to support realtime applications on the Internet, the only target in optimizing them is to achieve fairness with TCP flows in the network. No attention has been paid to the QoS perceived by their users. Their paper analyzes this problem of transmitting VoIP when voice sources use one of the most promising TCP-friendly algorithms, RAP and TFRC. They also propose a modification of both RAP and TFRC in order to take care the QoS in real-time multimedia flows.

The most recent survey on transport adaptation techniques has been provided in the article written by Homayounfar [46]. This article deals with the voice encoders of mobile networks. An *Adaptive Multi Rate* (AMR) encoder has been considered for VoIP applications. For VoIP, the purpose of rate adaptation is to reduce network congestion, and this can be thought of as an indirect form of implementing TCP-friendliness. The key question is what kind of metrics could be available for adaptive VoIP. In [47], for example, a method based on time stamps is proposed that is both simple and sensible. Recognizing that AMR voice frames arrive in packets with a combination of RTP/UDP/IP headers and that each RTP packet header actually includes a timestamp, the interarrival delay of AMR frames can be readily computed.

This delay is then used to detect network congestion: large delays (greater than 140 ms) indicate that congestion is present and that the lowest AMR rate of 4.75 kb/s ought to be used. Shorter delays imply otherwise. By quantizing the delay into eight segments, the appropriate rate of AMR is selected.

The work that seems very close to the current research work is presented in an article published by Abreu-Sernandez and Garcia-Mateo [48]. They propose a voice coder with three different bit rates adapted for the transmission of VoIP packets. The encoder has a rate control device that analyzes the traffic congestion of the network and orders the voice coder to switch among five operation modes if necessary. These modes include mitigation techniques of packet losses. The ideas proposed in this paper are similar to the ideas that will be presented in the current research. Like their work, the current research also proposes a six-mode adaptive voice encoder that changes the bit-rate of the voice encoding at the source in order to respond to congestion in the network. However, the details of the adaptive flow control scheme as proposed by Abreu-Sernandez and Garcia-Mateo are different from the schemes proposed in the current work. The authors of this paper mention dynamic adaptation of send-rate of the encoder without any specific details. The framework to vary the send-rate does not use predictive control. The results shown in the paper do not prove the efficacy of the codec conclusively.

Bai and Ito [49] have written a good survey paper on different approaches to provide QoS control for video and audio communication in conventional and active networks. The three most popular approaches to provide QoS of multimedia are by using integrated services, differentiated services, and active networking. The authors examine these three approaches from the perspective of their relative strengths and weaknesses. Akyildiz, Altunbasak, Fekri, and Sivakumar [50] provide an overview of AdaptNet, an adaptive protocol suite for next-generation wireless data networks.

AdaptNet consists of protocol solutions at different layers of the protocol stack addressing several problems, including rate adaptation, congestion control, mobility support, and coding.

Some more recent work regarding adaptive end-to-end flow control schemes for real-time audio applications have been done by Roychoudhury and Al-Shaer [51]. This paper presents an adaptive rate control framework that performs proactive rate control based on packet loss prediction and on-line audio quality assessment for real-time packet audio. The proposed framework determines the optimal combination of various audio codecs for transmission, utilizing a thorough analysis of audio quality based on individual codec characteristics, as opposed to ad-hoc codec redundancy used by other approaches.

Roychoudhury and Al-Shaer track the increase patterns or trends of the delay as an indication of congestion causing packet loss. The measured loss condition in the network is utilized to solve an optimization problem. The objective function of the optimization problem is the audio quality to be maximized, and is expressed in terms of the sum of the product of codec percentage and the codec quality score under current loss condition. The constraints of the optimization problem are:

1. Total bandwidth consumption by the codecs, expressed as the sum of the products of bit-rate generated and percentage of each codec. Total bandwidth should not exceed the measured available bandwidth.

2. Total codec delay, expressed as the sum of products of encode/decode delay and percentage of each codec. Total codec delay should not exceed the difference of the maximum allowable Mouth-to-Ear delay of 400ms and the link one way delay.

3. Quality of the voice signal encoded by the combination of codecs cannot exceed

the maximum quality value 4.3 (the MOS score of G.711 under no loss), and should be greater than or equal to 3.5 (lower bound of acceptable voice quality).

This work does not show any results of the effect of adoption of the proposed adaptive control scheme with respect to loss rates in network. Therefore, it is not possible to comment about the efficacy of the control scheme in best-effort network scenarios involving not only packet losses but also packet loss bursts.

B. Video Applications

Bolot and Turletti, [52] and [53], describe a feedback control mechanism for a variable bit rate video in which the parameters of a video coder, and hence the output rate of the coder, are adjusted in response to changing conditions in the Internet. In these papers, the researchers consider more specifically video coders based on the H.261 standard. The control algorithm focuses on three parameters - the refresh rate, the quantizer, and the movement detection threshold.

Multimedia data have specific temporal presentation requirements. In video conferencing applications voice and images of participants must be delivered and presented synchronously. These requirements can be achieved by scheduling or managing system resources. Gibbon and Little [54], present a technique called *Limited A Priori* (LAP) scheduling to manage the delivery channel from source to destination for digital multimedia data.

Jacobs and Eleftheriadis [55], propose an architecture that can support video services in best-effort networks, i.e. networks which have no QoS. The proposed technique involves an explicit attempt at avoiding network congestion. The same authors present a new technique for streaming real time video on the Internet in the paper [56]. This technique is based on dynamic rate shaping and TCP congestion avoidance.

The paper by Chung, Kim, and Kuo [57], examines the problem of real-time video streaming over the Internet by introducing an *adaptive Least-Mean-Squares* (LMS) bandwidth controller to adjust the amount of video data uploaded to the network so that the packet loss can be minimized in face of network congestion.

Rejaie, Estrin, and Handley [58] present a mechanism for using layered video in the context of unicast flow control. The quality adaptation mechanism adds and drops layers of the video stream to perform long-term coarse-grain adaptation. It uses a TCP-friendly congestion avoidance mechanism to react to congestion on very short timescales.

Song, Kim, and Kuo [59], define a GOP as a group of predictive frames without I-frame. The bit rate constraint is satisfied in each GOP. The proposed rate control uses this new GOP as a basic rate control unit. Furthermore, they propose an effective encoding frame rate control algorithm with low-latency under time-varying *Constant Bit Rate* (CBR) channel based on the R-D models. It can be more robust to the channel bandwidth fluctuation. The proposed frame rate control algorithm adopts a sliding window approach that does not impose additional encoding time-delay. Research papers [60] and [61], extend the work done in [59].

In paper [62], the authors, Wu et.al., tackle the problem of designing an efficient video delivery system that can maximize the perceptual quality while achieving high resource utilization. The main contributions of this paper are: a feedback control algorithm based on *Real Time Transport Protocol* (RTP) and *Real Time Control Protocol* (RTCP), an adaptive source-encoding algorithm for MPEG-4 video which is able to adjust the output rate of MPEG-4 to desired rate, and an efficient and robust packetization algorithm for MPEG video bit streams at the sync layer for Internet transport.

The paper by Wu, Hou, Li, Zhu, Zhang, and Chao [63] states that *Rate-distortion* (R-D) optimized mode selection is a fundamental problem for video communication over packet-switched networks. The classical R-D optimized mode selection only considers quantization distortion at the source. Such an approach is unable to achieve global optimality under the error-prone environment since it does not consider the packetization behavior at the source, the transport path characteristics, and receiver behavior. The researchers present an end-to-end approach to generalize the classical theory of R-D optimized mode selection for point-to-point video communication. A notion of global distortion is proposed by taking into consideration both the path characteristics (i.e., packet loss) and the receiver behavior (i.e., the error concealment scheme), in addition to the source behavior (i.e., quantization distortion and packetization). A set of accurate global distortion metrics for any packetization scheme is also proposed in this paper.

Feamster [64] talks about adaptive delivery of real-time streaming video. The thesis focuses on recovering from errors using mechanisms that do not rely on alteration of the encoded bitstream or processing of the decoded image at the receiver. The research is specifically interested in recovering I-frame data to limit error propagation. Real-time streaming applications that require low latency cannot afford to wait for the complete retransmission of all lost packets. The thesis focusses on using a combination of selective retransmission (enabled by backwards-compatible extensions to *Real-time Transport Protocol* (RTP), called SR-RTP) and postprocessing error concealment at the receiver to recover from packet losses that occur in more important portions of the compressed bitstream.

Adaptation to bandwidth changes are achieved by performing congestion avoidance - when packet loss is detected, the sender slows its transmission rate accordingly. Feamster uses binomial congestion avoidance for obtaining a flow that is TCP-friendly.

A scheme to help the server implement *quality adaptation* is also proposed in the thesis.

Wai [65] proposes a *Sender-Adaptive Rate Control for Layered Video Multicast* (SARC) in his MS thesis. This provides a rate control mechanism for the layered video multicast over the internet. SARC determines the number of video layers that can be accommodated by the receiver at the start of the video session. This is done by determining the real-time status of the network bandwidth.

Lu, Morando, and El Zarki, [66], define a measurement metric that tries to define the quality of a picture from a perceptual point of view. The paper has implementation details about the end-to-end real-time application involving a TCP based feedback channel that transmits the perceptual video quality parameters from receiver to sender. The application, based on the feedback information of perceptual quality degradation, can dynamically adjust the output bit rate of the video encoder that is delivered to the network.

C. Chapter Summary

This chapter provides a comprehensive summary of the previous research work into the field of end-to-end adaptive flow control schemes for both video and audio applications. No categorization of the literature surveyed has been done on the basis of streaming and conferencing applications. However, work related to only real-time multimedia applications have been presented. In brief, the most relevant works for the current research work can be found in these papers: [42], [43], [46], [47], [48], [51], and [52]. Although these papers propose various flow control and congestion avoidance strategies yet they rely exclusively on reactive control framework. The paper by Roychoudhury and Al-Shaer [51] try to estimate the occurrence of loss based on delay calculations. The paper by Bolot and Turletti [52] proposes simple feedback of the network status to control QoS of videoconferencing application. without taking care of network delay dynamics and feedback delay. The estimation strategy is not predictive flow control. The other papers in this section have been included in the literature search for completeness.

CHAPTER III

VOICE ENCODERS AND DECODERS

This chapter provides a brief overview of the voice encoders. Voice encoders are responsible for the bit rate of the real-time audio applications. From the perspective of control system theory, voice encoder is the actuator of the system. Many of the concepts described in this chapter has been extracted from the book on speech coding algorithms by Wai C. Chu [67].

Wai defines speech coding as the procedure "to represent a digitized speech signal using as few bits as possible, maintaining at the same time a reasonable level of speech quality." Speech coding is performed using numerous steps or operations specified as an algorithm. Figure 5 shows the block diagram of a speech coding system. The continuous-time analog speech signal from a given source is digitized by a standard connection of filter (eliminates aliasing), sampler (discrete-time conversion), and analog-to-digital converter. The source of this figure is the textbook by Chu [67]. The output is a discrete time speech signal whose sample values are discretized. This signal is referred to as the digital speech. Source encoder attempts to reduce the input bit-rate of the digital speech. Low bit-rate encoded digital speech is the output of the source encoder. The encoded digital speech data is further processed by the channel encoder. The channel encoder provides error protection to the bit-stream before transmission to the communication channel.

The channel decoder processes the error-protected data to recover the encoded data. Later on, the source decoder generates the output digital speech signal with the original sampling rate. Digital to analog conversion combined with anti-aliasing filters is used to convert the digital signal to an analog signal at the decoder end. There are many schemes to design the encoder-decoder pair. Different schemes provide differing

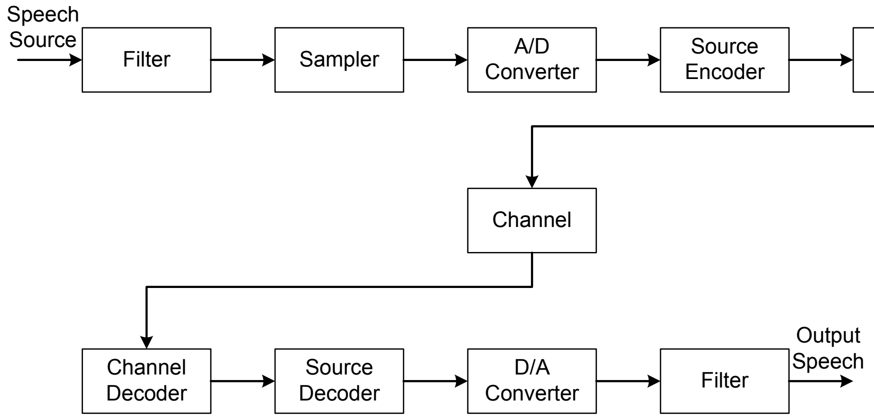


Fig. 5. Block diagram of a speech coding system [67].

speech quality and bit-rate. The complexity of these schemes also varies a lot. It is safe to presume that the complexity of the encoder-decoder pair increases with the increase in compression of the speech signal. Some of these techniques of speech coding will be discussed further in this chapter.

The main goal of speech coding is to either maximize the perceived quality at a fixed bit-rate or to decrease the bit-rate for a particular perceptual quality. Some the desirable properties in a speech encoder are:

- Low bit-rate: The lower the bit-rate generated by an encoder, the lesser is the bandwidth required for transmission.
- High speech quality: The encoded-decoded speech should have a quality acceptable to the end-users.
- Robustness across different speakers/languages
- Robustness to channel errors: Encoder-decoder pair should have algorithm that help them recover from the errors due to problems in the communication channel.

- Low memory size and computational complexity: This decides the cost of the encoder. The lower the cost the better it is.

- Low coding delay: In the process of speech coding, delay is introduced. The delay is the time shift between the input speech of the encoder with respect to the output speech of the decoder. An excessive delay creates problems with real-time two-way conversations.

A. Encoder-decoder Coding Delay

Let T_c be the time at which a speech sample arrives at the input. Let T_d be the time at which the encoded input speech sample arrives at the output of the decoder. The difference between the two time instants i.e. $T_c - T_d$ can be characterized as the coding delay. The process of measuring the coding delay is illustrated in figure 6. The source of this figure is the textbook by Chu [67].

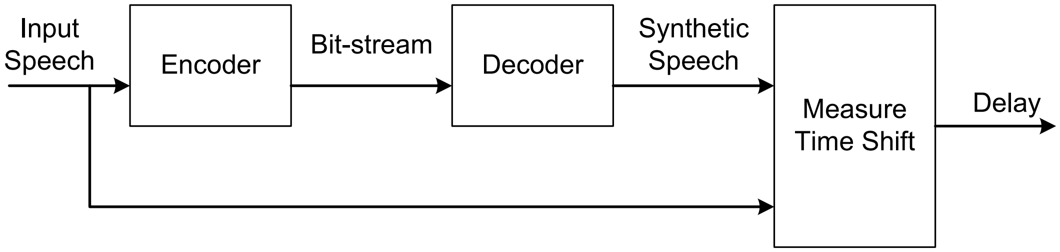


Fig. 6. Measurement of coding delay [67].

Coding delay is comprised of four components. One component is the *transmission delay.* This is the time taken by the encoded speech signal to reach the decoder through the communication channel. In context of the current research, this is the end-to-end delay experienced by the packets while traversing through best-effort networks. A mathematical formulation of the end-to-end delay has already been provided in chapter I. The other three main components of the coding delay are:

1. Encoder buffering delay: speech encoders require the collection of a certain number of samples before processing them. This collection of a certain number of speech samples is called a *frame*. There is lot of variation in the size of frames.

2. Encoder processing delay: The algorithm of the encoder requires certain processing time to encode the buffered data and construct a bit-stream. This delay is dependent on the complexity of the algorithm and the processing power of the hardware on which the algorithm is implemented. The processing delay must always be smaller than the buffering delay.

3. Decoder processing delay: This is the time required to decode in order to produce one frame of synthetic speech. Decoder processing delay is less as compared to the encoder processing delay.

When the term mouth-to-ear delay is used in the field of speech communication over packet switched networks, it refers to a delay comprising of the sum of the end-to-end packet delay, encoder buffering delay, encoder processing delay, and the decoder processing delay.

B. Classification of Speech Coders

1. Waveform Coders

Waveform coders preserve the original shape of the signal waveform. These coders produce high bit-rate coding. Their performance in terms of perceptual quality of the encoded-decoded speech drops sharply with reduced bit-rates. These coders work best at a bit-rate of 32 kbps and higher. Examples of this kind of coders are Pulse Code Modulation (*PCM*) and Adaptive Differential Pulse Code Modulation (*ADPCM*).

2. Parametric Coders

The principle of modeling speech using parametric models is used in parametric coders. During encoding, parameters of the model are estimated from the analysis of the input speech signal. The encoded bit-stream consists of the parameters. This type of encoder makes no attempt to preserve the original shape of the waveform. Perceptual quality of the decoded speech is related to the accuracy and sophistication of the underlying model. Most successful underlying models are based on linear prediction. This class of coders work well for low bit-rate. Examples of this kind of coders are Linear Prediction Coding (*LPC*) and Mixed Excitation Linear Prediction (*MELP*).

3. Hybrid Coders

The hybrid coders combine the strength of a waveform coder with that of a parametric coder. The parameters of the underlying model are calculated in the same way as they are calculated in the case of parametric coders. Some additional parameters of the model are also estimated in order to make the synthetic speech as close to the original waveform as possible with the closeness often measured by a perceptually weighted original signal. Examples of this type of encoders are Code Excited Linear Prediction (*CELP*) and its variants. The hybrid coders behave like waveform coders at high bit-rate and like a parametric encoder at low bit-rate.

4. Single Mode and Multi Mode Coders

Single mode coders are the category of coders that do not change their bit-rate irrespective of the speech type or network conditions. These encoders generate traffic at a constant rate leading to Constant Bit-Rate (*CBR*) flows. Multi mode coders adapt to time-varying network conditions or properties of the speech signal.

In an open-loop system, the modes of an encoder are selected by analyzing the properties of the input signal. In a closed-loop approach, encoded outcomes of each mode are taken into account in the final decision. The mode selection information is transmitted as part of the bit-stream. This is used by the decoder to select the proper mode. Most multi mode coders have variable rate. Each mode has a particular fixed value of bit-rate. Example of a multi mode coder is ETSI AMR ACELP. Some popular multi mode open source encoders are Speex [68] and Internet Low Bitrate Codec (*iLBC*) [69].

C. Codecs Used in VoIP Applications

The current research work is not oriented towards design of new codecs suitable for VoIP applications. The objective is to devise new adaptive flow control algorithms for real-time multimedia flows. The flow control algorithms devised in the current work can be implemented in the form of a new multi-rate codec that changes the bit-rate of the encoded voice signal based on the control algorithm. These algorithms can also be implemented separately that keeps varying the modes of the existing multi-rate codecs. In terms of control system theory, the codec is the actuator. The controller can be implemented separate from the actuator or can be built inside the actuator as a single unit.

Since the focus is on development of control strategies and not designing new codecs, it is not feasible to cover the entire spectrum of information regarding existing codec technologies and the algorithms behind them. Constraints of space and time has prevented a more detailed description of the codecs used in real-time audio applications. Chu's book [67] provides a good foundation for understanding the algorithms that are responsible for the codecs.

For the purpose of the simulations performed to devise new control strategies, it is pertinent to discuss some of the existing codecs in brief and summarize the bit-rates generated by them. This information is used in designing the parameters of the designed controllers that implement the control strategies.

1. G.711

G.711 is an ITU-T standard for audio coding. The standard was released for usage in 1972. G.711 represents logarithmic PCM samples for signals of voice frequencies, sampled at the rate of 8000 samples/second.

There are two main algorithms defined in the standard, the -law algorithm (used in North America and Japan) and A-law algorithm (used in Europe and the rest of the world). Both are logarithmic, but A-law was specifically designed to be simpler for a computer to process. The μ-law and A-Law algorithms encode 14-bit and 13-bit signed linear PCM samples (respectively) to logarithmic 8-bit samples. Thus, the G.711 encoder will create a 64 kbit/s bitstream.

2. G.723.1A

G.723.1A is the implementation of the ITU G.723.1 standard. Music or tones such as fax tones cannot be transported reliably with G.723.1. G.723.1 became an ITU-T standard in 1995. The coder operates on 30 ms frames of speech sampled at an 8 kHz rate, which together with a 7.5 ms look-ahead results in a total algorithmic delay of 37.5 ms. The codec has the features of Voice Activity Detection / Comfort Noise Generation (*VAD / CNG*). There are two bit rates at which G.723.1 can operate:

- 6.3 kbps (using 24 byte chunks and MPC-MLQ algorithm
- 5.3 kbps (using 20 byte chunks and ACELP algorithm

3. G.726

G.726 is an ADPCM speech codec that generates voice traffic at rates of 16, 24, 32, and 40 kbps. The bit size of a sample determine the four bit rates associated with G.726. The most commonly used mode of this codec is 32 kbit/s. This mode is half the rate of G.711. This increases the usable network capacity by 50 percent. G.721 was introduced in 1984. G.723 was introduced in 1988. They were combined into G.726 in 1990.

4. G.728

ITU-T G.728 is low delay speech coder standard, for compressing toll quality speech (8000 samples/second). G.728 coders have low algorithmic delay. G.728 is a very robust speech coder, with very good speech quality, comparable to 32 kbit/s ADPCM.

G.728 codec is based on Low Delay-Code Excited Linear Prediction (*LD-CELP*). This codec has the conventional approach of analysis-by-synthesis for fixed codebook search. The low algorithmic delay of the coder is achieved using backward adaptation of predictors and gain. Although the algorithmic delay can be as low as 0.625 ms (5 samples), the frame size of the coder is 2.5 ms (20 samples).

5. G.729a

This codec was developed by a consortium of France Telecom, Mitsubishi Electric Corporation, Nippon Telegraph and Telephone Corporation (*NTT*), and Université de Sherbrooke. G.729a works at speech sampling rate of 8 kHz. G.729A is based on the principle of Complementary Symmetry - Algebraic Code Excited Linear Prediction (*CS-ACELP*). The coder works on a frame of 80 speech samples (10 ms). It has a look ahead delay of 40 samples (5 ms). The total algorithmic delay for the coder is 15 ms.

6. AMR

The Adaptive Multi-Rate (*AMR*) codec is the fourth speech compression algorithm standardized by the European Telecommunications Standards Institute (*ETSI*). AMR was adopted as the standard speech codec by 3GPP in October 1998 and is now widely used in Global System Mobile (*GSM*) communications. It uses link adaptation to select from one of eight different bit rates based on link conditions.

7. iLBC

iLBC (*internet Low Bit-rate Codec*) is a popular open source codec. The codec is designed for narrow band speech and results in a payload bit rate of 13.33 kbit/s with an encoding frame length of 30 ms and 15.20 kbps with an encoding length of 20 ms. The iLBC codec enables graceful speech quality degradation in the case of lost frames, which occurs in connection with lost or delayed IP packets. Some features of this codec are:

- The codec generates two different bit-rates - 13.33 kbps and 15.2 kbps
- The basic quality of the codec is higher then G.729A

- iLBC has high robustness to packet losses
- The computational complexity of the codec is in the range of G.729A

8. Speex

Speex is an open source, patent-free, multi mode audio compression format designed for speech. Speex is well-adapted to Internet applications and provides useful features that are not present in most other codecs. Speex is based on CELP and is designed to compress speech at bit-rates ranging from 2 to 44 kbps. Some of Speex's features include:

- Narrow-band (8 kHz), wide-band (16 kHz), and ultra-wideband (32 kHz) compression in the same bitstream
- Intensity stereo encoding
- Packet loss concealment
- Variable bit-rate operation (VBR)
- Voice Activity Detection (VAD)
- Discontinuous Transmission (DTX)
- In-progress fixed-point port
- Acoustic echo canceler

Many properties of the Speex encoder has been extensively used in the current research work. The quality of speech produced by Speex in the presence of network delays and packet losses has been analyzed. E-model parameters [70] for the narrow band mode of the Speex have been evealuated to determine the perceptual quality of the speech coded at different quality levels. More details about E-model and tests to determine the perceptual quality of speech without conducting a subjective MOS [27] test is provided in the next chapter. The details of the encoder shown in Table II are meant for Speex narrow band encoder operating with Constant Bit-Rate (*CBR*) without employing any VAD, packet loss concealment, and DTX.

D. Bit-rate Generated by the Codecs

The bit-rate generated by a VoIP application consists of two components. The first component is the bit-rate generated by the codecs. This part has been discussed quite comprehensively in this chapter. The second part is the overhead generated by the headers that are attached to the packets that transport encoded voice data. Packet-switched networks like IP networks need these headers to perform flow control, reconstruct the voice signal at the destination, and route the packets to the destination. The effect of the headers in the final bit-rate become more pronounced as encoded audio data per packet transmitted decrease.

In order to explain this with a case study, G.711 - the simplest codec is selected. G.711 generates 160 byte packets every 20 ms. This turns out to be bit-rate equal to 64 kbps. Each of this encoded data packet is generated at the application layer. This payload is wrapped in successive layers of information in the subsequent layers of the OSI model.

The first header added is RTP. The size of RTP header is 12 bytes. RTP allows the samples to be reconstructed in the correct order and provides a mechanism for measuring delay and jitter.

The second header added is the UDP header. UDP header has a size of 8 bytes. UDP routes the data to the correct destination port.

The network layer adds the third header in the form of the IP header. An IP header has 20 bytes. This header is responsible for delivering the data to the destination host.

If we sum up the size of the RTP, UDP, and IP headers that get added to the packet of 160 bytes carrying one frame of the encoded voice, we get a total of 40 bytes. This implies that for the delivery of 160 bytes of encoded voice over best-effort IP networks, an overhead of 40 bytes is required. The headers added to the packet till the network layer form $\frac{40*100}{(160+40)} = 20$ percent of the bit-rate generated.

In order to travel through the IP network, the IP packet is wrapped in another layer by the physical transmission medium. Most VoIP transmissions start their journey over the Ethernet. Parts of the core transmission network is also likely to be Ethernet. The Ethernet header starts with an 8 byte preamble. A header of 14 bytes follows the preamble. The payload including all the previous headers is attached after the 14 byte header. The payload is followed by a 4 byte CRC. Tow subsequent Ethernet packets must be separated by a 12 byte gap. This mean that the physical transmission medium adds another 38 bytes to the 200 byte packet making the actual packet size traversing through the net equal to 238 bytes. Although the encoder generates 64 kbps bit-rate, the actual bit-rate traversing through the network is 95.2 kbps. 32.77 percent of the total bandwidth needed by the VoIP flow is used up in delivering the headers attached to the encoded data packet by the various layers. Figure 7 depicts the overhead due to other OSI layers on an encoded data packet traversing through a best-effort IP network. The source of this figure is the white paper of the Newport Networks [71].

Table II provides a list of technical details for different codecs mentioned earlier in this chapter. These details have been summarized from the following sources: [68], [69], [71], and [72]. The column “Net Bit-rate” provides information about the final

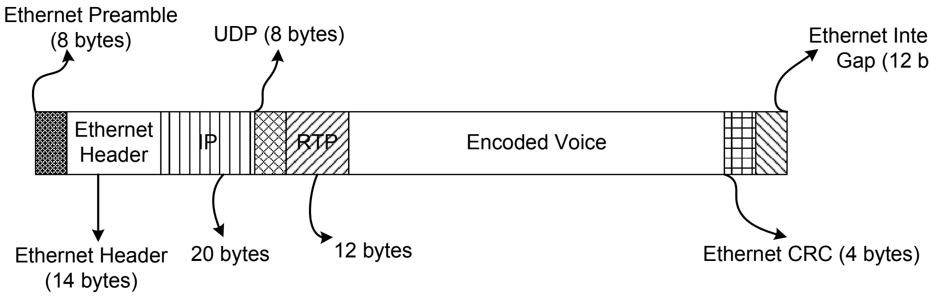

Ethernet Preamble (8 bytes)
UDP (8 bytes)
Ethernet Inter-Frame Gap (12 bytes)
Ethernet Header
IP
RTP
Encoded Voice
Ethernet Header (14 bytes)
20 bytes
12 bytes
Ethernet CRC (4 bytes)

Fig. 7. Overhead due to other OSI layers on an encoded voice data packet [71].

bit-rate inclusive of all the headers that traverses through an ethernet based best-effort IP network if a certain codec is used. The range of the numbers regarding the final bit-rate inclusive of all the headers provided in this table will be used to design the controllers later on in this work.

E. Chapter Summary

This chapter provides details about the speech codecs used commonly in VoIP applications. The first section provides an introduction to different kind of delays involved in a voice codec. Mouth-to-ear delay of an audio signal is the sum of end-to-end packet delay as defined in chapter I and the delay incurred due to coding and decoding of a frame in the codec. The second section of this chapter discusses various voice coding techniques in brief and classifies the codecs. This is followed by the section that lists various codecs used in VoIP applications. The last section of this chapter provides an insight into the actual bit-rate generated by the speech codecs and the traffic overhead generated by the headers attached by the protocols at various OSI layers to transport the encoded voice data packets to the receiver through a best-effort IP network.

Table II. Details of various popular codecs [68], [69], [71], and [72].

Codec	Algorithm	Sample	Lookahead	Frame Sz	Bit-rate	Frame/Pkt	Net Bit-rate
		Period	Delay	(Bytes)	(kbps)		((kbps)
G.711	PCM	20 ms	0 ms	160	64	1	95.2
G.723.1A	ACELP	30 ms	7.5 ms	20	5.3	1	26.1
G.723.1A	MP-MLQ	30 ms	7.5 ms	24	6.4	1	27.2
G.726	ADPCM	20 ms	0 ms	80	32	1	63.2
G.728	LD-CELP	2.5 ms	0 ms	5	16	4	78.4
G.729a	CS-CELP	10 ms	5 ms	10	8	2	39.2
AMR	ACELP	20 ms		12	4.75	1	36.0
AMR	ACELP	20 ms		19	7.4	1	38.8
AMR	ACELP	20 ms		31	12.2	1	43.6
AMR-WB/G.722.2	ACELP	20 ms		17	6.6	1	38.0
Speex (NB, CBR)	CELP	20 ms	10 ms	27-144	10.8-57.6	1	42-88.8
iLBC	LPC	20 ms	5 ms	38	15.2	1	46.4
iLBC	LPC	30 ms	10 ms	50	13.33	1	34.1

CHAPTER IV

VOICE QUALITY MEASUREMENT TESTS

As mentioned in chapter I, the main objective of the research is to improve the quality of the voice at the receiver end in audio conferencing applications when they communicate through best-effort IP networks while trying to remain friendly to the other competing flows. Quality of voice and metrics to determine the quality of voice is an integral part of the current research. This chapter provides a review of the various aspects of the voice quality measurement tests.

Performance of a voice codec is determined by the quality of the synthetic voice generated by it. The presentation of the results of the current research work depend on using an objective metric of voice quality. This metric should be able to show the change in the quality of voice due to application of the designed adaptive flow control algorithms to the multimedia flow generated by a real-time audio conferencing application over best-effort IP networks. This chapter is a brief review of the existing subjective and the objective metrics of the voice quality as prescribed by the various standards' committees all around the world.

It is not an easy task to determine the perceptual quality of voice encoded and decoded by the codec. Perceptual quality of voice has many dimensions. Some of them are:

- Intelligibilty: This dimension signifies the ability to understand the underlying message of the synthetic voice generated from the decoder.
- Naturalness and pleasantness: These dimensions refer to the presence of distortions, artifacts, and noise in the synthetic voice.
- Speaker recognizability

These dimensions are very difficult to be translated into a quantitative measure of the voice quality. Voice quality is affected by many factors - background noise, channel errors, power level of the input signal, language of speech, sex and age of the speaker, and type of input signal.

Background noise includes sound from sources like activity in the streets, transient noises, interfering speakers, and music. Loss of packets while traversing through a best-effort IP network can be characterized as a channel error. The quality of the codec is not uniform with respect to the variations in input power level. The output from the decoder should perform well for input signals with extremely low and high power levels of the input signal. Different languages of this world have different sounds (phonemes). Some of these phonemes might not be encoded properly by the codecs. Codec performance is definitely language dependent. Quality of the encoded and decoded voice is depended on the age and the sex of the speaker too. Sometimes, codecs that work well with human voice might not work well with the non-voice signal. In general, it is not expected for the voice coder to be able to synthesize music.

A. Subjective Tests

Subjective tests require different voice signals encoded and decoded by a specific codec to be played out to a set of listeners under different conditions. The listeners are asked to rate the quality of the voice signal they heard. The ratings are collected and averaged to calculate the final score indicating the quality of encoding performed by the codec. Reliability of the subjective tests depend on the number of listeners participating in the test. Most of the tests require at least 16 listeners.

1. Absolute Category Rating (ACR)

In this test the listeners provide a single rating for the voice sample. The listeners have five ratings to chose from: excellent (5), good (4), fair (3), poor (2), and bad (1). The average rating of all the listeners is known as the Mean Opinion Score (*MOS*). MOS is a scale that has been recommended by ITU to measure the quality of received audio voice signal after compression or transmission.

2. Degradation Category Rating (DCR)

This test is a modification over ACR. In this test the listeners are presented with an original voice signal and made to compare the synthetic encoded-decoded voice signal with it. The listeners give a rating based on whether they can perceive the difference between the two signals. Their choice of ratings are - not perceived (5), perceived but not annoying (4), slightly annoying (3), annoying (2), and very annoying (1). When the ratings of all the listeners are averaged, the mean score is called the Degradation Mean Opinion Score (*DMOS*).

3. Comparison Category Rating (CCR)

This is a modification of the DCR test. In this subjective test, the order in which the voice signals are presented to the listeners for comparative quality rating is changed. Unlike the DCR test where the original voice signal is presented first, followed by the synthetic voice, the order of the signals in this test can be random. The choice of ratings for the listeners participating in the test are: much better (3), better (2), slightly better (1), about the same (0), slightly worse (-1), worse (-2), and much worse (-3).

B. Perceptual Evaluation of Speech Quality

The subjective tests for voice quality for the codecs under different operating conditions are very costly. They can not be performed easily. These tests also consume a lot of time. Therefore, the research community started focusing their effort on voice quality tests directed towards perceptually based objective measures.

Perceptual Evaluation of Speech Quality (*PESQ*) is an enhanced perceptual quality measurement for voice quality in telecommunications. PESQ is an objective measurement tool that predicts the results of subjective listening tests on telephony systems. PESQ has been developed to be applicable to end-to-end voice quality testing under real network conditions, like VoIP, POTS, ISDN, GSM etc.

PESQ has been officially approved as new ITU-T recommendation P.862 in February 2001. It is considered to be a successor of Perceptual Speech Quality Measure (*PSQM*) as recommended in the document ITU-T P.861. PESQ has been developed by KPN Research, Netherlands and British Telecommunications (BT), by combining the two advanced voice quality measures PSQM+ and PAMS. PESQ is able to predict subjective quality with good correlation in a very wide range of conditions, that includes coding distortions, errors, noise, filtering, delay and variable delay

In PESQ test, the original and degraded signals are mapped onto an internal representation using a perceptual model. The difference in this representation is used by a cognitive model to predict the perceived voice quality of the degraded signal. This perceived listening quality is expressed in terms of MOS.

C. E-Model

The subjective tests like ACR and objective tests like PESQ provide effective frameworks to assess voice quality. However, these frameworks fail to provide reliable results in the scenario of best-effort packet-switched IP networks. These frameworks have not been designed to take care of the effects arising out of variable time-dependent end-to-end delay, packet losses, and delay jitter. A suitable framework for estimating call quality from the measured network performance such as delay and loss characteristics of a path is the ITU-T E-Model [70]. E-Model is used by the VoIP industry to measure the the quality of the voice signals.

E-Model defines an R-factor that combines different aspects of voice quality impairments. Equation 4.1 provides the mathematical definition of the R-factor.

$$R = R_0 - I_s - I_e - I_d + A \tag{4.1}$$

In the equation 4.1, R_0 refers to the effects of various noises, I_s represents the effects of the impairments that occur simultaneously with the original signal, I_e is the effect of impairments caused by packet losses in the network, and I_d shows the effect of impairments because of the delay suffered by the packets. A compensates for the above impairments under various user conditions.

I_d and I_e are the two parameters among all the others that are important for the VoIP systems. Equation 4.1 can be modified after substituting the default values of the other parameters [73]. Equation 4.2 is the modified and final equation to determine the R-factor that determines voice quality in a VoIP application using best-effort networks to transmit information.

$$R = 94.5 - I_e - I_d \tag{4.2}$$

The R-factor ranges from 0 to 100 and can be mapped to MOS using a nonlinear mapping [70].

$$MOS = 1 + 0.035R + 7 * 10^{-6}R(R - 60)(100 - R) \tag{4.3}$$

Figure 8 shows the results of the equation 4.3 graphically. Equations 4.2 and 4.3

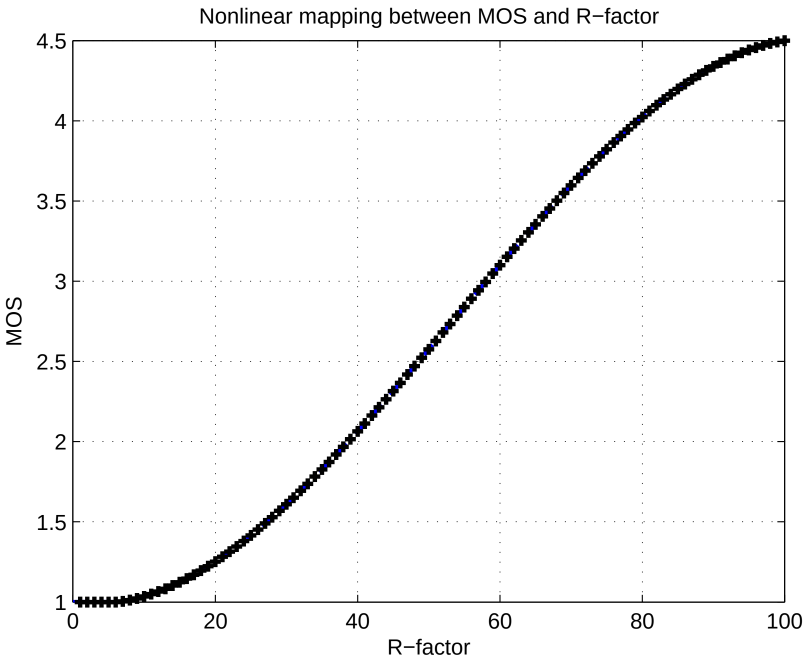


Fig. 8. Nonlinear mapping between MOS and R-factor.

provide the facility of measuring the quality of voice in terms of MOS in a VoIP application based on the measurement of delay and packet loss in the network. The two variables that needs to be calculated in order to calculate the R-factor and eventually the MOS are I_e and I_d.

I_d is dependent on the mouth-to-ear delay, d. Mouth-to-ear delay, as defined earlier, is the sum of the end-to-end delay of the packet containing the encoded voice data and the delay due to coding and decoding of the signal. End-to-end delay comprises of the playout delay due to buffering at the receiver. Equation 4.4 shows how the mouth-to-ear delay affects I_d.

$$I_d = 0.024d + 0.11(d - 177.3)\mathbf{I}(d - 177.3) \tag{4.4}$$

$\mathbf{I}(x)$ is an indicator function that implies

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

As seen from equation 4.4, I_d does not depend on the type of codec used for encoding voice. It depends on the mouth-to-ear delay. However, calculating the impact of packet losses on quality of voice within the E-Model framework is not easy. I_e depends on parameters that are determined by the properties of the encoder. The relation between I_e and overall packet loss rate e can be expressed by the following equation:

$$I_e = \gamma_1 + \gamma_2 ln(1 + \gamma_3 e) \tag{4.5}$$

γ_1 is a constant that determines voice quality impairment caused by encoding, γ_2 and γ_3 describe the impact of loss on perceived voice quality for a given codec. e includes both network losses and playout buffer losses.

γ_1, γ_2, and γ_3 are determined by using voice quality testing results under different loss conditions. To read more about E-Model and its applications in determining quality of voice in VoIP applications, the papers by Cole and Rosenbluth [73], Ding and Goubran [74], Tao et. al. [75], and Sun and Ifeachor [76] need to be referred to. The above mentioned papers also provide the experimentally determined values of the three parameters - γ_1, γ_2, and γ_3, for various existing encoders. Ye [77] provides the values of γ_1, γ_2, and γ_3 for Speex at different bit-rates operating without any error-concealment algorithms and in CBR mode in his research work. Ye also describes the experimental procedure of obtaining the parameter values for Speex in his dissertation. Table III summarizes the results for several codecs as mentioned in [73] and [75]. Table IV provides the γ values for Speex encoder without any error-concealment, encoded at CBR as determined by Ye [77]. Speex is a multi mode codec.

In Table IV, the column "Quality" refers to the Speex encoding that is controlled by a quality parameter that ranges from 0 to 10. In CBR operation, the quality parameter is an integer. Figure 9 shows the variation of the parameter I_e with respect to the percentage of packet lost in the network. Eight different quality levels of the Speex encoder is shown in the diagram. γ values of the three levels in Speex overlap each other resulting into three pairs of overlapped I_e curves.

Table III. The values of γ_1, γ_2 and γ_3 for various codecs [73] and [75].

Codec	PLC	γ_1	γ_2	γ_3
G.711	[78]	0	30	15
G.723.1.B-5.3	silence	19	71.38	6
G.723.1.B-6.3	silence	15	90.00	5
G.729	silence	10	47.82	18
G.723.1.A+VAD-6.3	none	15	30.50	17
G.729A+VAD	none	11	30.00	16

D. Chapter Summary

Chapter IV provides details about various tests available to determine the quality of voice in VoIP applications. MOS is the metric that subjective tests like ACR use to quantify quality. However, subjective tests are not only difficult but also costly to perform. Objective tests that provide the final results in terms of MOS like PESQ provide a viable alternative to the subjective tests. E-Model is a framework standardized by ITU-T that determines quality of the voice in VoIP applications communicating over best-effort networks. The R-rating determined by the E-Model

Table IV. The values of γ_1, γ_2 and γ_3 for Speex [77].

Quality	γ_1	γ_2	γ_3
3	31.01	36.99	10.29
4	31.01	36.99	10.29
5	23.17	29.36	20.03
6	23.17	29.36	20.03
7	16.19	24.91	36.17
8	16.19	24.91	36.17
9	9.78	22.81	58.76
10	6.89	21.99	72.75

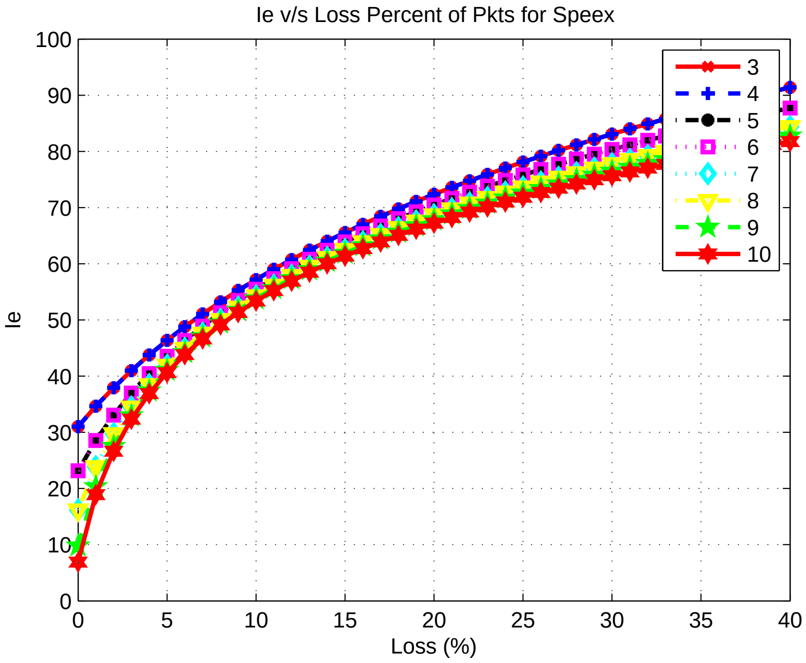


Fig. 9. Plot of I_e versus packet loss percentage for Speex at different voice quality levels.

on the basis of packet losses and the delay of the packets can be easily mapped to MOS scale using a nonlinear relationship. E-Model is widely used in the VoIP sector.

CHAPTER V

EXPERIMENTAL AND SIMULATION SETUPS

The first step in solving the problem of flow control in real-time multimedia applications is to collect data in order to model a single flow as a system. Development of suitable linear and nonlinear models using system identification techniques precedes the development of control strategies for the system.

Initially, one of the objectives of the current research was to implement the developed real-time multimedia flow control algorithms in a VoIP application operating on the Internet. This would have helped in demonstrating the efficacy of the flow control algorithms in a real world scenario. A real-world test bed is needed to perform experiments in order to collect data for modeling the flows. PlanetLab - an overlay network maintained by a consortium of corporate and academic entities all around the world, provided the most logical solution in the search for a test bed to conduct the experiments for the current research. However, the data from the experiments conducted on the PlanetLab overlay network failed to fulfill the expectations of a real world test bed. These experiments failed because of the inability of PlanetLab overlay network test bed to provide the right level of network congestion. More details about the experiments conducted on PlanetLab and the results from the experiments will be discussed later in this chapter.

After the failure of the PlanetLab network to provide the right level of congestion conditions to perform real world experiments, a simulation environment provided by ns-2 - an open source simulator, is selected to demonstrate the ideas of adaptive flow control of real-time multimedia flows traversing through best-effort IP networks. In the absence of a real world test bed to implement the adaptive flow control algorithms, the simulation scenarios developed in ns-2 environment provide the framework for the current research.

A. PlanetLab

PlanetLab [79] is an open, globally distributed platform for developing, deploying and accessing planetary-scale network services. PlanetLab nodes support both short-term experiments and long-running network services. PlanetLab creates a unique environment to conduct experiments at the Internet scale. The network services deployed on PlanetLab experience all of the behaviors of the real Internet where the only thing predictable is unpredictability (latency, bandwidth, paths taken). PlanetLab provides a diverse perspective on the Internet in terms of connection properties, network presence, and geographical location. Figure 10 shows the active nodes of PlanetLab network.

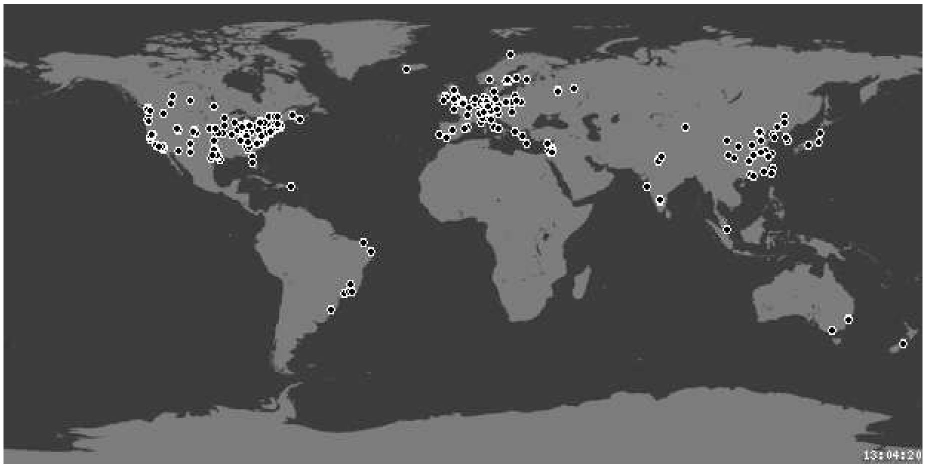


Fig. 10. PlanetLab: current distribution of 723 nodes over 354 sites.

B. Experiments Conducted Using PlanetLab

One of the important premises of the current research is that in a congested network, a flow occupying lesser bandwidth will be able to provide a better end-to-end QoS as compared to a flow that demands more bandwidth. High quality multimedia flows require larger bandwidth as compared to low quality flows. During congestion, the probability of the loss of packets belonging to a high bandwidth flow is higher than that of packets belonging to a low bandwidth flow. The loss of packets of the higher bandwidth flow deteriorates the quality of the flow to make it inferior to that of the quality achieved by the lower bandwidth flow during congestion. The delays and the losses suffered by a UDP flow in a congested network can be reduced by reducing the send-rate of the flow at minimal expense to voice quality. In order to verify this statement, experiments are conducted on the overlay network PlanetLab. Real-world delay traces and packet losses are collected using tools on servers of PlanetLab.

The bit-rate of a multimedia flow has two degrees of freedom. It can be manipulated by either varying the inter-departure times between the packets or by varying the sizes of the packets comprising the flow. From the point of view of control systems as well as development and implementation of multi-mode variable bit-rate encoding algorithms, it is far easier to deal with flows that have varying packet sizes as compared to varying inter-departure times between the packets.

The packets of a flow sent during a connection have dual roles. The first role is related to the transmission of the information from one end to the other. This facilitates the interaction between the two end users. The second role is related to providing the feedback signal about the the state of the network through which the packets pass in order to deliver information.

Since packets also provide the information about the state of the network, varying inter-departure time between the packets means that the information about the network is sampled at different frequencies. This complicates the formulation of control strategies for the system. While traversing through the network, the packets undergo time-variant queuing and processing delays leading to variation in the inter-arrival times and even losses. This implies that depending on the state of the network, the sampling rate of the state of the network also varies considerably. These time-varying time delays introduce a lot of complexities in the subsequent mathematical analysis. If it is decided to vary the inter-departure time of the packets then the control algorithms become too complicated. It is prudent that the send-rate of the flow to the network be varied by varying the size of the packets generated.

In order to select end nodes that have the potential to show the quality problems faced by users in the real world, the Digital Subscriber Line (*DSL*) nodes available on the PlanetLab overlay network are chosen. The DSL nodes help in the collection of realistic end-to-end traffic. The servers under the domain name gti-dsl.nodes.planet-lab.org are chosen as one end of the network connection. The two nodes - planetlab1.gti-dsl.nodes.planet-lab.org and planetlab2.gti-dsl.nodes.planet-lab .org, share a 500Kbps symmetric DSL line. The other end of the network comprises of the servers belonging to the domain name nbgisp.com. The servers - planetlab1.nbgisp.com etc. are also connected to the network using DSL lines.

The experiments between the nodes selected as the source and the destination are performed by sending one UDP flow emulating a real-time multimedia flow from one end to another in a controlled manner. The control applied to the experimental UDP flow is open loop and is not based on any feedback. Analysis of the delays and the losses suffered by the packets in the experimental flow provides information about the state of the network comprising of the links on the path of journey of packets from sender to receiver. The duration of this experimental flow lasts for half an hour for one run of the experiment. The experiment is repeated for the entire day from 0900 hrs to 2300 hrs every day for a week in order to capture the short-term and long-term variation of cross-flow dynamics on the Internet.

The open loop control of the experimental UDP flow emulating real-time multimedia is applied by varying the bit-rate of the flow periodically. The bit-rate of the flow is switched alternately between a low bit rate and a high bit rate. The flow during low bit rate comprises of 32 byte packets sent across the network with the inter-departure time period of 120 ms. This is equivalent to 7.33 kbps after taking into account the size of the UDP, IP, and the ethernet headers. The flow during high bit rate comprises of 626 byte packets sent across the network with the inter-departure time period of 120 ms. This is equivalent to 46.93 kbps. It is important to note that the high bit rate part of the flow is almost 6.4 times the low bit rate.

The switching period between the high bit rate and the low bit rate is also varied in order to gauge the sensitivity of the system to changes in the flow bit rate. It might happen that the system (the single flow) has a very slow rate of response. This means that a faster rate of change of the bit rate may not produce any noticeable change in the response as measured in terms of the number of packets lost or delayed. The experiments on the PlanetLab are designed to measure the sensitivity of the network too. Figure 11 shows the variation of the send rates and the time period of switching between the low bit rate and the high bit rate while conducting experiments between the two DSL servers located within the United States.

In a span of five days of a week, 107 experiments were conducted by sending data from 5 planetlab1.gti-dsl.nodes.planet-lab.org to planetlab1.nbgisp.com. Out of these 107 experiments, data obtained from 3 experiments were rejected due to problems in recording the data files on both the servers. Final data comprises of 104 data sets that provide a glimpse of the network path between the two servers mentioned above.

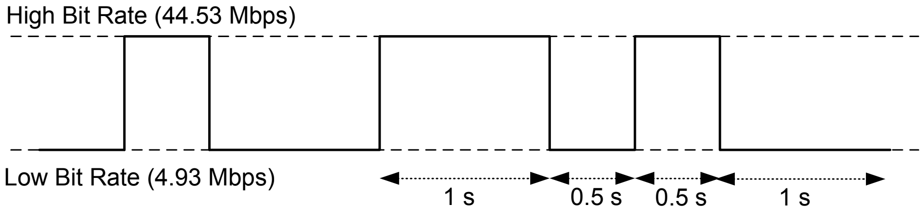


Fig. 11. Send rate variation of experiments conducted between planetlab1.gti-dsl.nodes.planet-lab.org and planetlab1.nbgisp.com.

While analyzing the collected data, it is assumed that any packet that has a forward end-to-end delay of more than 200 ms is assumed to be lost. These type of losses can be categorized as Delay Threshold Losses (DTL). Each data set is a time-series containing the information of the delays and losses suffered by the packets while going through the network.

Table V provides information about the loss rate suffered by the packets in the forward path of the 104 data sets collected by performing experiments between the two DSL nodes over a week. It can be observed that in spite of the fact that the high bit rate (HBR) is 6.4 times that of the low bit rate (LBR), percentage of packets lost remains the same irrespective of the send rate. This means that the variation of the send bit-rate has no effect on the number of losses that the packets encounter while traveling through the network. The switch rates of 0.5 s and 1.0 s fail to provide any insight into the sensitivity of the system.

Some of the descriptive loss statistics as tabulated in Table V might hide something that can be unearthed only after closer inspection. In the send rate signal used to push information into the network, a high bit-rate segment is always followed by a low bit-rate segment of similar duration. If the larger size packets are more prone to loss than the smaller size packets, it is pertinent to examine them in the consecutive segments in the whole series. In order to mitigate the effects of momentary change in the nature of the network to prevent bias, an ensemble average of the packet loss

Table V. Loss rate statistics derived from experiments conducted between planetlab1.gti-dsl.nodes.planet-lab.org and planetlab1.nbgisp.com.

S No	Switch Rate	Bit Rate	Pkts. Sent	Pkts. Lost	Loss(%)
1	0.5s	LBR	278720	41909	15.04
2	0.5s	HBR	278720	43130	15.47
3	1.0s	LBR	501696	74518	14.85
4	1.0s	HBR	501691	78075	15.56

comprising of 104 data sets is taken. The set of equations are:

$$L_{LBR,0.5}(i) = \sum_{k=1}^{104} l^k_{LBR,0.5}(i), \tag{5.1}$$

$$L_{HBR,0.5}(i) = \sum_{k=1}^{104} l^k_{HBR,0.5}(i), \tag{5.2}$$

$$L_{LBR,1.0}(i) = \sum_{k=1}^{104} l^k_{LBR,1.0}(i), \tag{5.3}$$

$$L_{HBR,1.0}(i) = \sum_{k=1}^{104} l^k_{HBR,1.0}(i), \tag{5.4}$$

where $L_{[L/H]BR,y}(i)$ represents the total number of lost Lower/Higher bit rate packets in the segment where the switching time period is y secs, and where $l^k_{[L/H]BR,y}(i)$ is the number of lost Lower/Higher bit rate packets in the segment of data set k where the switching time period is y secs. There are four different modes of the UDP flow that is being sent to the Internet. The two parameters that make up these modes are bit-rate and the switching period. $i\epsilon\{1, \ldots, N\}$ such that N is the number of times change of mode takes place within the UDP flow during the course of a single experiment of half an hour. If each data set from one run of the experiment contains data of M secs approximately, one cycle of four mode switches consumes $0.5 + 0.5 + 1.0 + 1.0 = 3.0$ secs. Thus, we have $\lceil M/3 \rceil = N$ cycles.

Using the four time-series extracted from the whole set of 104 data sets, we generate two more time-series. The new time-series represent whether number of packet losses during high bit rate is higher for a segment as compared to the number of packet losses during the low bit rate.

$$S_{0.5}(i) \quad = \quad sign(L_{HBR,0.5}(i) - L_{LBR,0.5}(i)), \tag{5.5}$$

$$S_{1.0}(i) \quad = \quad sign(L_{HBR,1.0}(i) - L_{LBR,1.0}(i)), \tag{5.6}$$

where $S_x(i)$ represents the signum function of the difference between the packets lost during high bit rate and low bit rate when the switching rate between them is x secs. Equations 5.5 and 5.6 describe the generation of the two time-series mentioned earlier in the paragraph. The metric that helps to evaluate whether packets with larger size are more prone to being lost than packets with smaller size is to calculate number of times $S_x(i)$ is positive, negative, or 0. In mathematical terms these metrics - $M^1_{0.5}$, $M^0_{0.5}$, $M^{-1}_{0.5}$, $M^1_{1.0}$, $M^0_{1.0}$, and $M^{-1}_{1.0}$, can be defined by the following equations:

$$M^1_{0.5} \quad = \quad \sum_{i=1}^{N} \mathbf{I}(S_{0.5}(i)), \tag{5.7}$$

$$M^0_{0.5} \quad = \quad \sum_{i=1}^{N} \mathbf{I}(S_{0.5}(i) + 1) - M^1_{0.5}, \tag{5.8}$$

$$M^{-1}_{0.5} \quad = \quad N - (M^1_{0.5} + M^0_{0.5}), \tag{5.9}$$

$$M_{1.0}^{1} = \sum_{i=1}^{N} \mathbf{I}(S_{1.0}(i)), \tag{5.10}$$

$$M_{1.0}^{0} = \sum_{i=1}^{N} \mathbf{I}(S_{1.0}(i)+1) - M_{1.0}^{1}, \tag{5.11}$$

$$M_{1.0}^{-1} = N - (M_{1.0}^{1} + M_{1.0}^{0}), \tag{5.12}$$

where $\mathbf{I}(x)$ is an indicator function that implies

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

Table VI shows that the number of times larger size packets are lost as compared to smaller size packets is really not much different from the number of times smaller size packets are lost as compared to larger size packets. This shows that bit rate variation in the send rate does not really help in reducing packet losses in between the two servers planetlab1.gti-dsl.nodes.planet-lab.org and planetlab1.nbgisp.com on the PlanetLab overlay network.

Table VI. Loss of HBR and LBR packets while traversing the network.

	Switch Rate	Metric	Frequency
1	0.5s	$M_{0.5}^{1}$	263
2	0.5s	$M_{0.5}^{-1}$	273
3	0.5s	$M_{0.5}^{0}$	0
4	1.0s	$M_{1.0}^{1}$	282
5	1.0s	$M_{1.0}^{-1}$	254
6	1.0s	$M_{1.0}^{0}$	0

1. Search for Other Suitable Servers on PlanetLab

During the time when the experiments were conducted on PlanetLab, only five other DSL servers were available to the end users. These were - planetlab[1/2/3].ucb-dsl.nodes.planet-lab.org, uw1.accretive-dsl.nodes.planet-lab.org, and planetlab4-dsl.cs .cornell.edu. However, none of these servers were reliable enough to perform a week long continuous experiment.

Non-DSL servers within United States did not exhibit any suitable characteristics of a moderately congested network. The aim of the current research is to investigate how end-to-end dynamic send rate variation improves the QoS of multimedia real-time flows. In order to investigate the problem and demonstrate the efficacy of the end-to-end flow control scheme, the network under investigation should be moderately congested with packet loss rates between 5% to 10%. Most of the PlanetLab nodes located within United States do not exhibit any kinds of congestion. Packet loss rates between these nodes is consistently less than 3%.

The search for servers to extract suitable network traffic profiles for the current research finally led to the selection of overseas nodes. PlanetLab has hundreds of servers, many of them located outside United States. In order to make a preliminary list of the prospective servers that might be eligible for further investigation, short-term ping experiments of 600 sec. time span were conducted from the server planetlab1.pbs.org to various servers overseas. Table VII provides a brief glimpse into the results of the ping experiments

Most of the Western European servers exhibited similar network congestion conditions as the mainland United States servers. The Eastern European servers were too unreliable to be used for conducting long sustained experiment for more than a week. Some of the links dropped almost half the packets that were sent to it from

Table VII. Ping experiment results with overseas servers.

Server	Location	Min. RTT (ms)	Mean RTT (ms)	Max. RTT (ms)	Loss (%)
x2.cs.vmk.unn.ru	Russia	145.97	3278.96	27599.13	8
planetlab3.singapore.equinix.planet-lab.org	Singapore	241.56	243.96	255.96	0
planetlab2.pop-rs.rnp.br	Brazil	360.27	396.72	416.71	2
planetlab1.lsd.ufcg.edu.br	Brazil	385.60	419.54	490.61	9
planetlab2.informatik.uni-kl.de	Germany	106.00	108.29	119.11	0
planetlab1.mini.pw.edu.pl	Poland	113.83	116.85	164.88	48
planetlab1.cs.unibo.it	Italy	142.11	145.14	190.88	0
planet5.cs.huji.ac.il	Israel	177.56	180.79	195.41	0
planetlab1.iis.sinica.edu.tw	Taiwan	201.97	211.11	228.76	0
planetlab-01.ipv6.lip6.fr	France	114.29	118.13	132.47	0
fudan1.6planetlab.edu.cn	China	278.94	283.23	295.27	0
planetlab2.iitb.ac.in	India	387.84	410.31	440.57	1
csplanetlab1.kaist.ac.kr	S.Korea	223.14	228.16	270.96	0
planet-lab-1.csse.monash.edu.au	Australia	226.11	230.49	281.93	0

the PBS server planetlab1.pbs.org.

Two paths exhibited some promise initially because the *Round Trip Time* (RTT) measured from planetlab1.pbs.org seemed to be ideal for the experiments ($\sim$ 250 ms). These were the paths to the end nodes planet-lab-1.csse.monash.edu and fudan1.6planetlab.edu.cn. Week long experiments between planetlab1.pbs.org and fudan1.6planetlab.edu.cn or planet-lab-1.csse.monash.edu.au yielded traffic profiles that exhibited very few losses ($< 5\%$) over a period of one week. However, these experiments did manage to provide evidence in support of the hypothesis that higher bit-rate flows undergo more losses in a network as compared to lower bit-rate flows.

Figures 12 and 13 provide snapshots of the losses encountered in the forward path by the flows traveling to Australia and China from mainland United States. Although the total number of losses in each experiment in both cases is low, yet it can be noticed that more high bit-rate packets are lost as compared to low bit-rate packets. However, this difference in the losses between high and low bit-rate packets is marginal and not significant.

2. Failure of the PlanetLab Experiments

One of the main reasons for the failure to locate a suitably congested node is the current state of the Internet. In the late 1990s, telecom firms embarked upon fiber-laying binge. Networking gear makers also developed methods to dramatically increase the amount of data that could be transmitted on a single strand of fiber. Using optical-networking technology, carriers can divide a single strand of fiber into 96 or more separate channels, each handling about 10 Gbps of traffic. The bandwidth available to the users grew exponentially from the late 1990s to early 2001. However, only a small fraction of the installed bandwidth is currently used leading to “bandwidth glut”. This excess bandwidth has been responsible in reducing the packet losses in

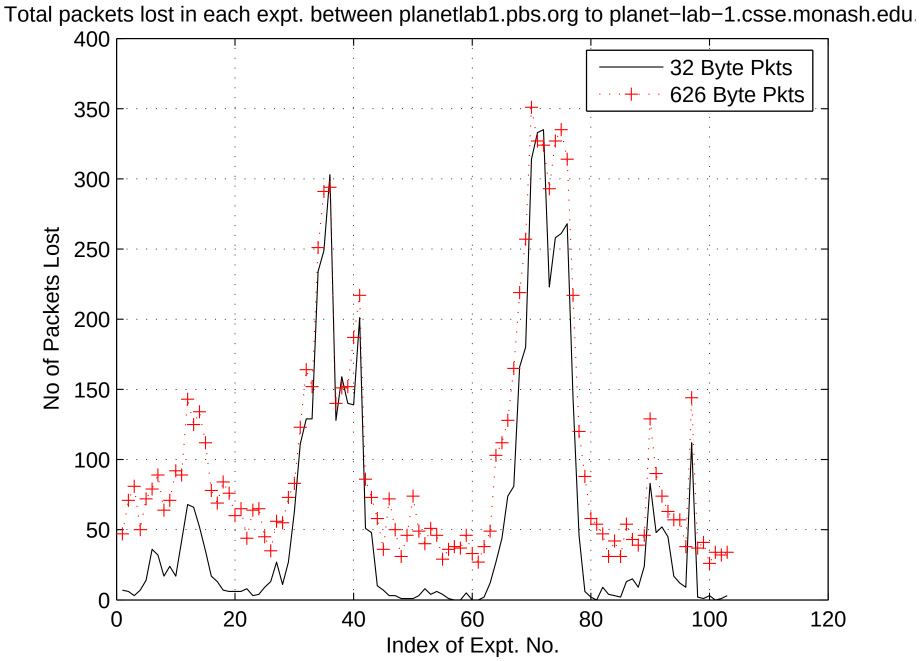


Fig. 12. Packets lost in experiments conducted between planetlab1.pbs.org and planet-lab-1.csse.monash.edu.au.

the links leading to subsequent increase of the quality of end-to-end paths between the servers.

Many of the experiments conducted on the PlanetLab network also exhibited high packet losses when the bandwidth needed by the experimental flows exceeded 10 Mbps. These packet losses were not because of the congestion in the network. Unfortunately, the high loss rates were most likely because PlanetLab enforces a 10 Mbps allocation of bandwidth for all the flows per slice. PlanetLab setup also requires all flows to implement congestion avoidance.

PlanetLab does not provide a controlled environment for our experiments. Repeatability of the experiments is very difficult to achieve. Most PlanetLab nodes are also heavily loaded. Precise probing at a certain rate and sensitive timing measurements over the duration of several seconds are quite difficult to perform. This problem

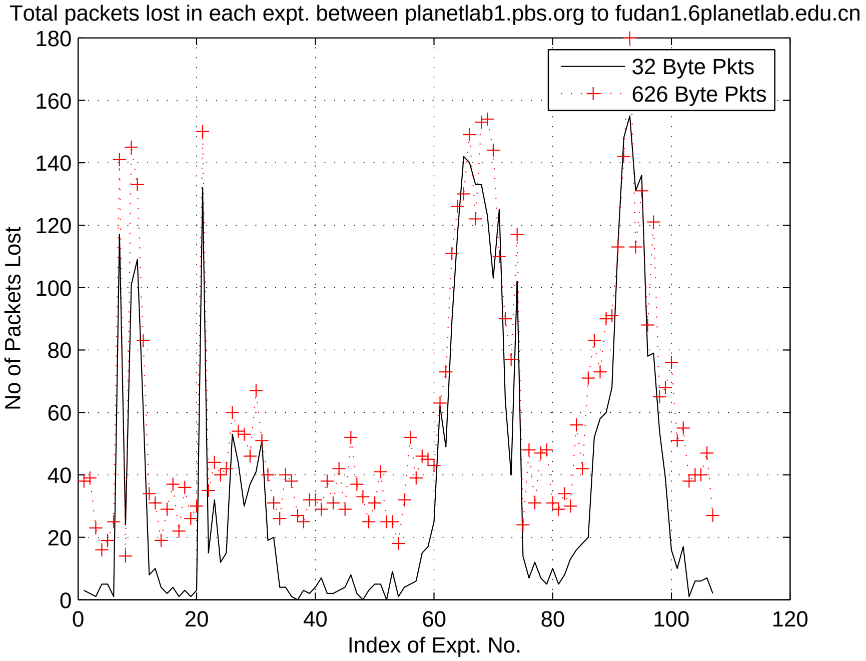


Fig. 13. Packets lost in experiments conducted between planetlab1.pbs.org and fudan1.6planetlab.edu.cn.

has been experienced by many researchers who have tried to perform network characterization measurements on the PlanetLab. PlanetLab provides infrastructure for testing experimental network protocols. It helps in testing distributed applications in real world scenarios. However, it does not provide a suitable test bed for controlled experiments.

Some researchers have proposed that together with the existing model of virtual slicing in PlanetLab, some PlanetLab nodes can be reserved for a “single user” for short time periods. This option would really help a lot of network measurement studies. However, these nodes should not allow the running of long-term services. The “exclusive access” nodes can be a small subset of the PlanetLab network. At the time of writing this dissertation, the facility of “exclusive access” on PlanetLab has not materialized yet.

3. Need for Simulations

After the failure to get a suitable real world test bed environment from the PlanetLab overlay network, the need for a controlled environment that can provide a framework for performing repeatable and controlled experiments is satisfied by the use of network simulators. The two most popular network simulators in the networking research community are ns-2 and Opnet. For the purpose of the current research, ns-2 is selected as the simulator of choice. This is because ns-2 is open source and widely accepted in the networking community. The application level flow control algorithms for improving the QoS of VoIP applications for this research have been implemented as separate modules in the ns-2 framework and integrated, later on. More about this will be discussed in later chapters.

ns-2 is an object-oriented, discrete event driven network simulator developed at UC Berkeley. It is written in C++ and OTcl. ns-2 is primarily useful for simulating local and wide area networks. ns-2 also provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.

C. Simulation Strategy

Creating a network topology that is sufficiently saturated to serve the purpose of the current research is easy in a simulator framework. The user has control over all aspects of the topology. The user can decide on the level of congestion by varying the parameters that determine the final topology suitable for demonstrating the efficacy of the ideas. Some important aspects that were taken care of while designing the simulation topologies are:

- The percentage of UDP flows in the network as compared to the TCP flows should reflect what is currently prevalent on the Internet.
- The queue capacity of the routers that carry the main traffic should not be too large so that there is no drop of packets. The capacities should not be too small so that they get saturated by a small number of flows transferring packets at moderate levels.
- The propagation delay between the source and the destination should be large enough to reflect real world conditions. The propagation delay across mainland United States vary from 60 ms to 90 ms.

For the purpose of the experiments two different network topologies were generated. The first topology was used to design and validate the flow control strategies. The second topology was used for studying the scalability of the flow control strategies. The next two subsections describe the two topologies used for performing the experiments in detail.

1. Simulation Network Topology for Validating Designed Control Strategies

The first topology for validating the designed control strategies is depicted in figure 14. This topology has two links that resemble the backbone links found in networks like Internet2. The backbone links are formed by the three routers named Router 0 (R0), Router 1 (R1), and Router 2 (R2).

Some of the initial experiments to collect delay and loss signals in order to simulate the Internet for the current research were performed between the two servers - niml.tamu.edu situated at the Texas A&M University campus and hal.bu.edu located at the Boston University campus in the east coast of the United States. During these experiments and the subsequent experiments on the PlanetLab overlay network, it was observed that the cost-to-coast propagation delay between servers located at different academic institutions in the United States varies between 60 ms to 90 ms. The link between R0 and R1 (R0-R1) has a propagation delay of 30.87 ms. The link between R1 and R2 (R1-R2) also has the same propagation delay. Therefore, the total propagation delay in the backbone links adds up to 61.74 ms. The propagation delay in the backbone links is never changed during the experiments.

The path to hal.bu.edu had 16 hops. Using the measurement tool, Pathrate [80], the minimum bandwidth capacity of the bottleneck link in the path to hal.bu.edu was determined to be 73.26 Mbps. The link between routers R1 and R2 (R1-R2) has been fixed to 73.26 Mbps.

The two backbone links in the current topology, R0-R1 and R1-R2, serve unique roles. The purpose of the link R0-R1 is to cause losses in the flows. Meanwhile, the R1-R2 link causes variation in the end-to-end delay of the flows under observation. The term “flows under observation” refers to UDP flows that we are trying to control.

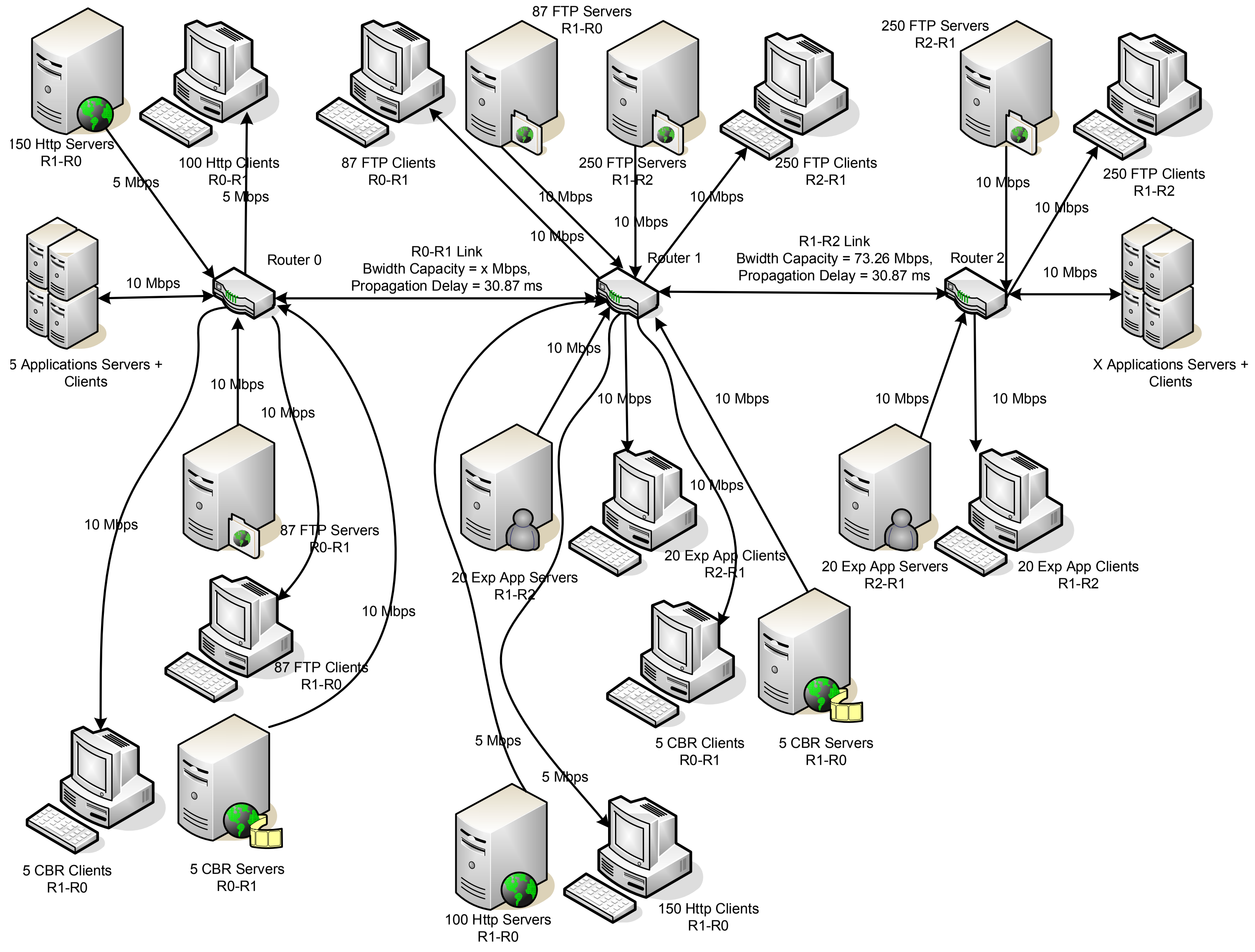


Fig. 14. Topology created in the ns-2 simulation environment to design and study the adaptive end-to-end multimedia flow control schemes.

By changing the bandwidth of the link R0-R1 in a suitable manner, the loss rates of the flows under observation are varied. The bandwidth capacity of the link R0-R1 varies between 28.92 Mbps and 50.92 Mbps. Subsequently, the average loss rate of the five UDP flows under observation vary from 15% to 3%.

Besides the bandwidth capacity and the propagation delay of the queues, the queue sizes of the links R0-R1 and R1-R2 are important. All the queues in the current topology have First In First Out (*FIFO*) buffer management scheme. In ns-2, the output queue of a node is implemented as part of a link. The duplex link R0-R1 has two queues with a capacity of 460800 bytes. The next backbone duplex link R1-R2 has two queues of capacity equal to 1024000 bytes.

If we consider left to right as the forward direction of the flow of traffic in figure 14, the forward cross-flow traffic at the link R0-R1 comprises of 150 HTTP (TCP) flows, 5 CBR (UDP) flows, and 87 FTP (TCP) flows. Also, 100 HTTP (TCP), 5 CBR (UDP) flows, and 87 FTP (TCP) flows make up the cross-flow traffic in the backward direction in the same link. The HTTP nodes are connected to Router 0 and Router 1 using 5 Mbps capacity links. The rest of the nodes are connected using 10 Mbps capacity links.

There is no cross-traffic from HTTP flows in the link R1-R2. The cross-traffic of the link R1-R2 is composed of 250 FTP (TCP) and 20 Exponential (UDP) flows in the forward as well as the backward direction. All the application nodes are connected to the routers R1 and R2 using 10 Mbps links. Table VIII summarizes the information about cross-traffic in the topology in terms of the flows. The cross-traffic flows constitute 98.98 % of the total flows in the current simulation scenario. The last column of the table provides the information about the contribution of a type of flow as a percentage of total flows in the topology.

The remaining flows in the topology (1.02%) is comprised of the nodes that run multimedia applications using UDP in the transport layer. These applications use the designed flow control schemes. Henceforth, these flows will be referred to as "**controlled UDP**" flows in the dissertation. The controlled UDP flows are the flows to be observed and measured to test the efficacy of he designed controlled schemes. The 10 nodes that provide controlled flows are connected to Router 0 (R0) and Router 2 (R2) using 10 Mbps bandwidth capacity links. The controlled UDP flows traverse both the backbone links.

Table VIII. Details of the composition of cross-traffic in terms of flows for the network topology used to design and validate the control schemes.

Flow Type	R0-R1 (F)	R0-R1 (B)	R1-R2 (F)	R1-R2 (B)	%
HTTP	150	100	0	0	25.41
FTP	87	87	250	250	68.50
CBR	5	5	0	0	1.02
EXP	0	0	20	20	4.06
Total					98.98

Describing the topology in terms of the number of flows fails to provide a realistic summary of the contribution of the flows to the cross-traffic. For example - ten 64 kbps flows generate twice as much traffic as ten 32 kbps flows. However, in terms of flows, their contribution is the same to the network traffic. Therefore, it is pertinent to describe the composition of the traffic in the network during simulations in terms of the percentage of traffic bytes contributed by each type of flow. Table IX provides the composition of traffic in terms of bytes during simulations performed using the current topology. Although the loss rate of the controlled UDP flows vary from 3%

to 15% because of the variation of the bandwidth of the link R0-R1, the relative contribution of each type of traffic to the simulation does not undergo much changes. The variations are within a bound of ±1%.

Table IX. Details of the composition of cross-traffic in terms of bytes for the network topology used to design and validate the control schemes.

Type	% of Total TCP	% of Total UDP	% of Total Traffic
TCP (HTTP)	∼ 15.69	N.A.	∼ 12.64
TCP (FTP)	∼ 84.31	N.A.	∼ 67.92
Total % (TCP/Total Traffic)			∼ 80.56
Controlled UDP	N.A.	∼ 2.98	∼ 0.58
UDP (Exp)	N.A.	∼ 84.30	∼ 16.39
UDP (CBR)	N.A.	∼ 12.72	∼ 2.47
Total % (UDP/Total Traffic)			∼ 19.44

2. Simulation Network Topology for Studying Scalability of the Control Strategies

The second topology is designed to study the scalability performance of the adaptive flow control algorithms and it is based on the previous topology. There are two objectives of the scalability studies:

- Test the performance of the flow control algorithms as the share of controlled UDP flows to the total UDP traffic in the network goes up. The ratio of the UDP flows and the TCP flows to the total traffic generated during this set of simulations remains constant. This test is designed to find out whether increasing the percentage of controlled UDP flows as compared to the uncontrolled

UDP flows while keeping the contribution of the total UDP flows to the total traffic in the network constant affects the performance of the TCP flows.

- Test the performance of the flow control algorithms as the share of the UDP flows (both the controlled as well as the uncontrolled UDP flows) to the total traffic in the network goes up. This scalability test is aimed at investigating the advantages of using flow controlled real-time multimedia applications as the share of UDP traffic in the overall traffic over the Internet goes up in the future.

These experiments, using the designed network topology, will help to demonstrate the advantages of using real-time multimedia applications deploying adaptive flow control algorithms over the Internet.

The difference between the current topology and the topology designed earlier is with respect to the location of the nodes generating the exponential (UDP) traffic. In the earlier topology used to validate the efficacy of the control algorithms, the nodes running exponential applications on top of UDP were sending traffic between the link R1-R2 in both directions. In the current topology, the traffic from these nodes traverse the same path as that of the packets from the controlled UDP flows. This means that the nodes having exponential UDP applications are connected either to Router 0 (R0) or to Router 2 (R2). Figure 15 shows the topology generated using ns-2 to study concept scalability. The changed position of the nodes running Exponential application using UDP have been shown using shaded areas in the figure 15.

Link R0-R1 has to tolerate extra traffic because of the shifting of the locations of the nodes running exponential applications. Simultaneously, the link also has to maintain the average loss rate of the controlled UDP flows around 3% to 4%. Therefore, the sizes of the queues present in the duplex link R0-R1 are increased to 768000 bytes from 460800 bytes. Unlike the previous topology, the bandwidth of

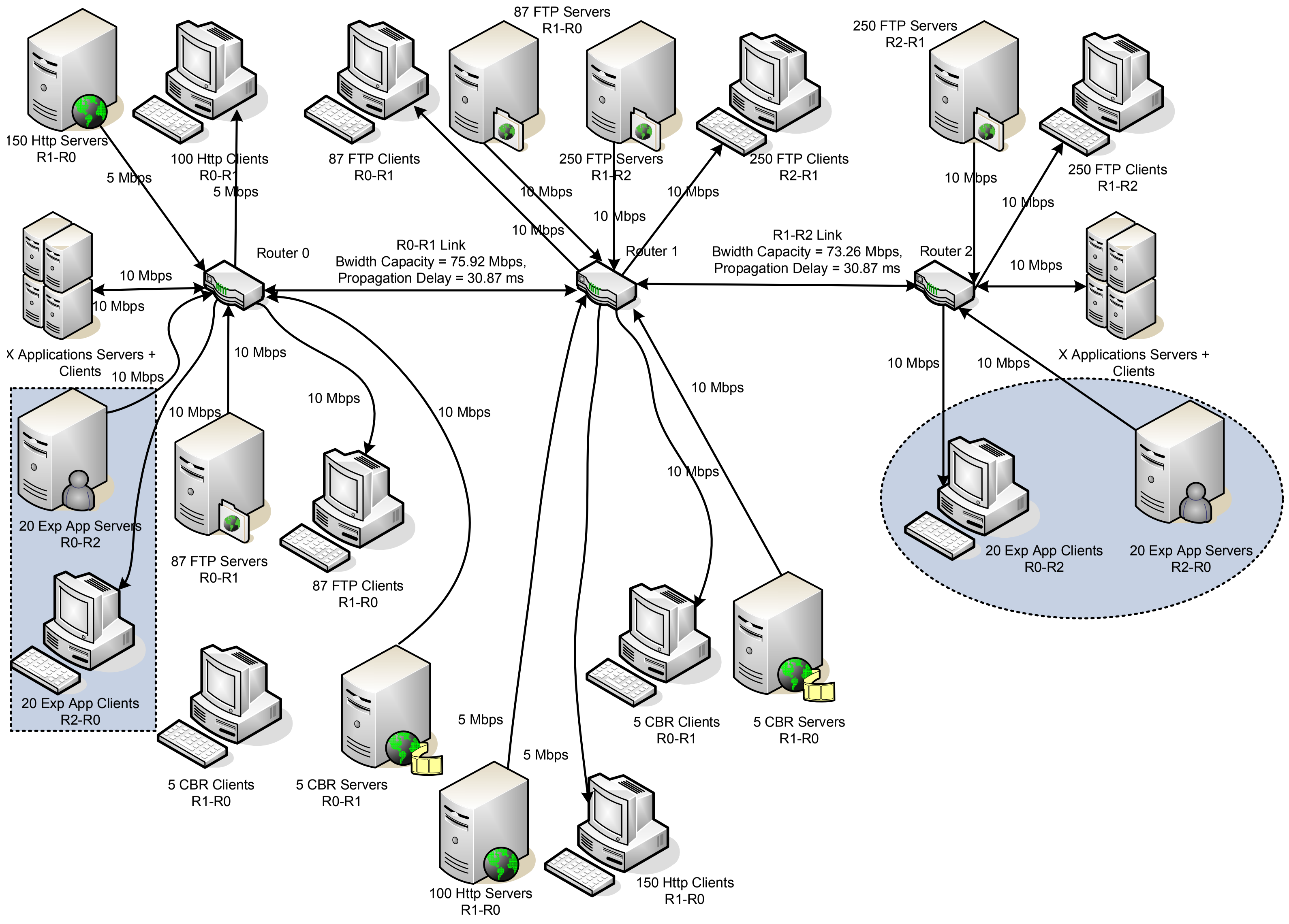


Fig. 15. Topology created in the ns-2 simulation environment to study scalability of the adaptive end-to-end multimedia flow control schemes.

the R0-R1 link is fixed at 75.92 Mbps. This is because this set of experiments do not require the loss rates of the controlled UDP flows to be varied by changing the bandwidth or the queuing parameters of the network.

The topology for studying scalability is modified in many ways during the experiments as per the requirements. In order to study the effect of increasing the ratio of controlled UDP flows with respect to the uncontrolled UDP flows, the ratio of total UDP to total TCP traffic is kept constant. The number of nodes using flow controlled multimedia UDP applications, connected to the routers R0 and R2, is steadily increased. The bit rates generated by the nodes running exponential applications using UDP, connected to the same routers, is suitably decreased. For studying the second objective of the scalability studies, the number of nodes using flow controlled multimedia UDP applications is steadily increased without making any further changes in the network topology. This leads to an increase in the percentage of UDP traffic contribution in the total traffic during the simulations. Figures 16.(a), 16.(b), and 16.(c) show the objectives of the scalability experiments.

Figure 16.(a) represents the original topology. The bandwidth capacity in the network is consumed by five different categories of flows. The two TCP categories (HTTP and FTP) have the largest share. This is followed by the shares of the three types of UDP flows. Figure 16.(b) shows the change in percentage of the bandwidth used among the UDP flows. The percentage of bandwidth consumed by the controlled UDP flows goes up as compared to the other two categories of the UDP flows. The proportion of the total UDP contribution to the total traffic does not change. Finally, figure 16.(c) depicts the second objective of the scalability tests visually. The proportion of the UDP traffic to the total traffic goes up.

Let $\mathbb{T}_1$ be the total traffic in the network measured in bytes. $\mathbb{T}_1$ is the sum of UDP traffic $\mathbb{U}_1$ and TCP traffic $\mathbb{P}_1$ in the network. Both of them are also measured

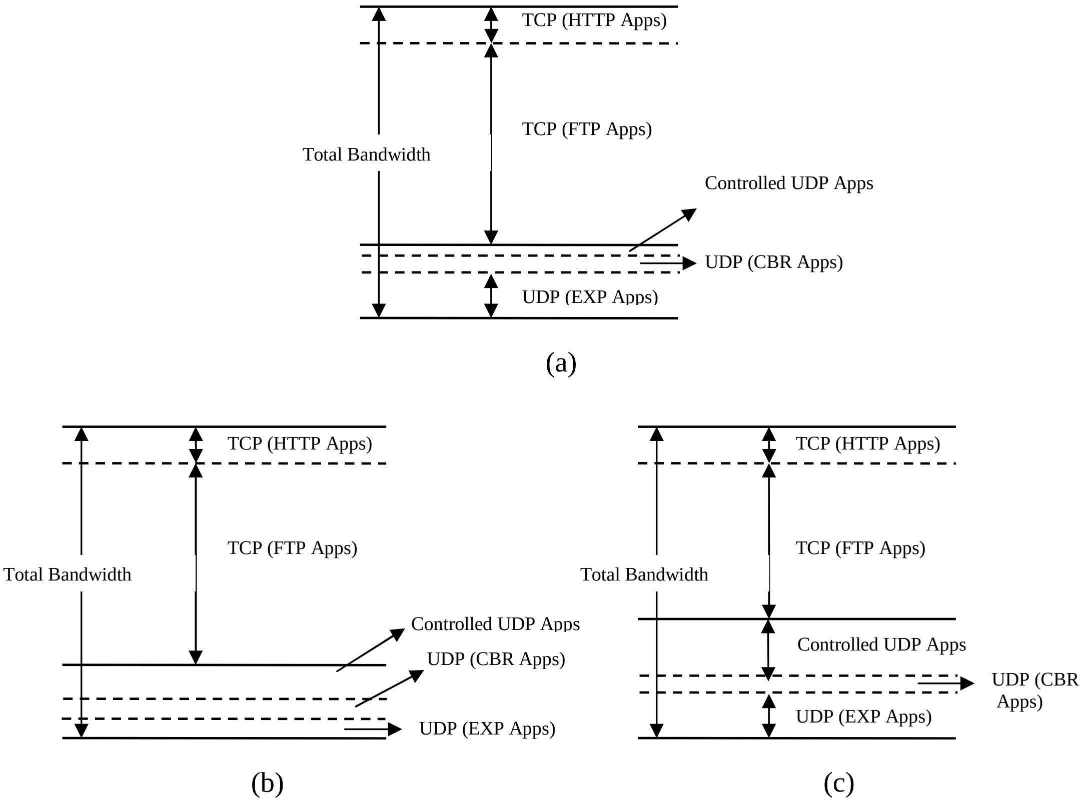


Fig. 16. Objectives of the scalability experiments.

in bytes. Therefore,

$$\frac{\mathbb{P}_1}{\mathbb{T}_1} + \frac{\mathbb{U}_1}{\mathbb{T}_1} = 1 \tag{5.13}$$

$\mathbb{U}_1$ comprises of controlled UDP traffic U_{c_1} and uncontrolled UDP traffic U_{u_1}. This implies that

$$\frac{U_{c_1}}{\mathbb{U}_1} + \frac{U_{u_1}}{\mathbb{U}_1} = 1 \tag{5.14}$$

In the first case of scalability experiments, overall contribution of the UDP flows in the total traffic is kept constant. The traffic composition is changed by increasing the share of the controlled UDP flows and decreasing the share of the uncontrolled UDP flows by the same magnitude.

$$\begin{aligned} \frac{\mathbb{P}_2}{\mathbb{T}_2} + \frac{\mathbb{U}_2}{\mathbb{T}_2} &= 1, \qquad (5.15) \\ \frac{U_{c2}}{\mathbb{U}_2} + \frac{U_{u2}}{\mathbb{U}_2} &= 1, \qquad (5.16) \end{aligned}$$

such that $\frac{U_{c2}}{\mathbb{U}_2} > \frac{U_{c1}}{\mathbb{U}_1}$ and $\frac{U_{u_2}}{\mathbb{U}_2} < \frac{U_{u_1}}{\mathbb{U}_1}$. Moreover, $\frac{\mathbb{U}_2}{\mathbb{T}_2} = \frac{\mathbb{U}_1}{\mathbb{T}_1}$. This means that $\frac{\mathbb{P}_2}{\mathbb{T}_2} = \frac{\mathbb{P}_1}{\mathbb{T}_1}$.

In the second case of scalability experiments, overall contribution of the UDP flows in the total traffic is increased. This increase is achieved by increasing the share of the controlled UDP flows without changing any other aspect of the traffic as compared to the original case.

$$\begin{aligned} \frac{\mathbb{P}_3}{\mathbb{T}_3} + \frac{\mathbb{U}_3}{\mathbb{T}_3} &= 1, \qquad (5.17) \\ \frac{U_{c3}}{\mathbb{U}_3} + \frac{U_{u3}}{\mathbb{U}_3} &= 1, \qquad (5.18) \end{aligned}$$

such that $\frac{U_{c3}}{\mathbb{U}_3} > \frac{U_{c1}}{\mathbb{U}_1}$, $\frac{U_{u3}}{\mathbb{U}_3} < \frac{U_{u1}}{\mathbb{U}_1}$, and $\frac{\mathbb{U}_3}{\mathbb{T}_3} > \frac{\mathbb{U}_1}{\mathbb{T}_1}$. This implies that $\frac{\mathbb{P}_3}{\mathbb{T}_3} < \frac{\mathbb{P}_1}{\mathbb{T}_1}$.

Table X provides information about the composition of traffic in the network topology with five controlled UDP sources to study scalability of the control algorithms measured in terms of the bytes contributed by each type of flow. This traffic composition undergoes changes as the topology is modified for achieving the objectives of the scalability tests.

Table X. Details of the composition of cross-traffic in terms of bytes for the network topology used to study scalability of the control schemes.

Type	% of Total TCP	% of Total UDP	% of Total Traffic
TCP (HTTP)	$\sim$ 15.19	N.A.	$\sim$ 12.35
TCP (FTP)	$\sim$ 84.81	N.A.	$\sim$ 68.95
Total % (TCP/Total Traffic)			$\sim$ 81.30
Controlled UDP	N.A.	$\sim$ 3.0	$\sim$ 0.56
UDP (Exp)	N.A.	$\sim$ 84.15	$\sim$ 15.73
UDP (CBR)	N.A.	$\sim$ 12.85	$\sim$ 2.41
Total % (UDP/Total Traffic)			$\sim$ 19.45

D. Chapter Summary

This chapter describes the experimental test beds and the simulation scenarios designed for the current research. Initially, attempt is made to demonstrate the feasibility of the adaptive flow control algorithms in real time multimedia applications by performing experiments in real world scenarios. The PlanetLab overlay network is chosen as the test bed for the experiments. However, PlanetLab failed to serve as the suitable test bed primarily because of the lack of availability of suitable paths between pairs of nodes that show packet loss rates between 3% to 10%. After the failure in the search for a suitable real world test bed, a simulated network environment is selected for development and testing of the flow control algorithms. The framework provided by the ns-2 network simulator has been used to create topologies to develop flow control algorithms and study their scalability. These topologies have been discussed in detail in this chapter.

CHAPTER VI

MODELING OF SINGLE FLOWS IN VoIP APPLICATIONS

The main objective of the current research is to design innovative end-to-end flow control strategies for real-time multimedia applications that transfer information over best-effort IP networks. This necessitates a brief discussion on the existing flow control strategies that have been developed for real-time multimedia applications.

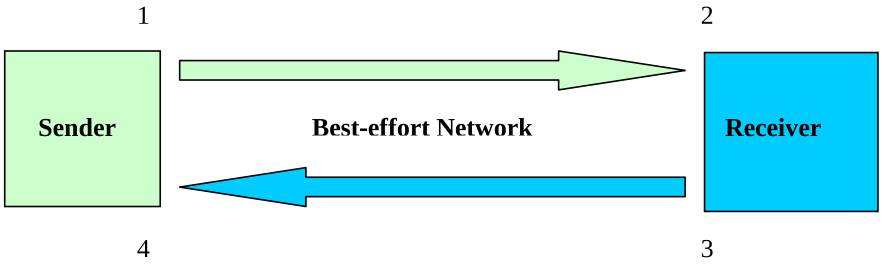
Many of the existing flow control strategies have been listed either in chapter I while introducing the problem or in chapter II while surveying the literature related to the current research. Almost all the real-time multimedia control strategies reviewed till now are based on "reactive" control strategies. Presence of network dynamic delay and time varying delayed feedback is part of the inherent nature of the end-to-end flow control problem over best-effort networks.

Figure 17 depicts the sequence of events that takes place when a packet containing information is sent into the network. A packet i is sent into the network at send time s_i. It arrives at the receiver at the arrival time a_i. The receiver calculates the state(s) of the network. The state(s) of the network can be one or a combination of or all of the signal(s) used for modeling the network mathematically. The state(s) of the network is sent back to the sender at the departure time f_i. The feedback of the state(s), as determined by the passage of the packet i in the forward path, arrives at the sender at the receive time r_i.

The forward and backward path delays can be written as

$$\begin{aligned} T_i^f &= a_i - s_i, & (6.1) \\ T_i^b &= r_i - f_i. & (6.2) \end{aligned}$$

The total Round Trip Time (RTT), T_i^R, of the packet is the sum of the forward and



1. Sender sends packets containing information into the network. s_i is the time at which packet i is generated at the sender.
2. The receiver receives packet i at the arrival time a_i. Forward delay for the packet i to get transferred from the sender to receiver can be calculated as $T_i^f = a_i - s_i$. State of the network is calculated by the receiver.
3. The feedback about the state of the network is sent back to the sender at feedback departure time $f_i > a_i > s_i$.
4. The feedback is received by the sender at the receive time $r_i > f_i > a_i > s_i$. The backward delay for the feedback to reach the source is $T_i^b = r_i - f_i$.
5. The round trip time of the packet i to reach receiver, generate a feedback, send it to the sender, and make the sender receive the feedback is $T_i^R = T_i^f + T_i^b$.

Fig. 17. Sequence of events for a packet i sent to the receiver from the sender.

the backward delays. Mathematically, this can be expressed as

$$T_i^R = T_i^f + T_i^b \tag{6.3}$$

$$\Rightarrow T_i^R = (r_i + a_i) - (f_i - s_i). \tag{6.4}$$

For the sake of clarity, it is important to note that the forward delay of the packet i is not the same as end-to-end delay of the packet. The end-to-end delay of a packet includes the amount of time the packet spends in the playout buffer b_i along with the propagation delay D_{prop}, transmission delay D_i^t, and the queuing delay v_i. Mouth-to-ear delay of a talkspurt includes the coding and the decoding delays associated with the specific codec in use along with the end-to-end delay.

The important point from the above definitions is that the feedback of the network takes approximately T_i^b time to reach the sender. It is assumed that the amount of time taken by the receiver to calculate the state(s) of the network, generate the feedback information, and send it out through the network in a packet is negligible. Moreover, the time T_i^b varies depending on the level of congestion in the network. The control action of reducing or increasing the bit-rate based on the state of the network measured T_i^b seconds ago might not be the most effective way to control the flow.

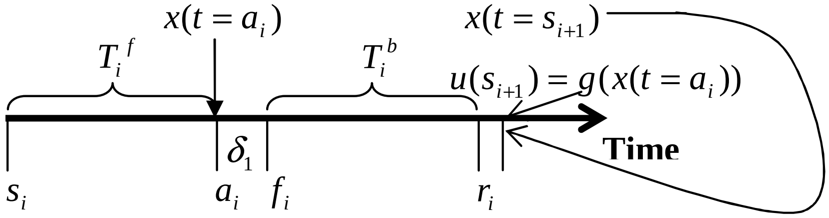


Fig. 18. Timeline of the events during a reactive flow control strategy.

Figure 18 shows the order in which the events take place during a reactive flow control strategy. The state of the network, x, is measured at time $t = a_i$. The feedback of the state of the network is received by the source at the time r_i. The control input is calculated and implemented at the time $t = s_{i+1}$. If the processing time of the control input calculation is neglected, it can be assumed that the control signal $u(t = s_{i+1})$ is based on the measurements of the state(s) of the network at time $t = a_i$. Since the time period T_i^b is not negligible, it can be safely assumed that $x(t = a_i) \neq x(t = s_{i+1})$. This exposes the critical flaw of reactive control strategies in the network environment.

If the future state(s) of the network are known then these state(s) can be utilized to compute a far more effective control signal. This is the reason that a predictive control strategy seems to be more suitable for real-time flow control of multimedia applications. Figure 19 shows the sequence of events in a predictive control strategy. In this figure, the control signal $u(t = s_{i+1})$ is based on the prediction of the network state(s) $\hat{x}(t = a_{i+1})$ at time $t = a_{i+1}$ in the future. If the prediction accuracy of the network state(s) is good enough, the calculated control signal based on predicted state(s) is a better candidate for flow control with respect to the control signal calculated based on the state(s) of the network measured T_i^b seconds ago. This concept, in brief, is the main proposal of the current research. The designed predictive flow control algorithm will be implemented at the application layer of the OSI model.

A. Modeling End-to-End Single Flow Dynamics in Best-Effort Networks

Modeling the end-to-end dynamics of a single flow in best-effort networks is a difficult task. Traditionally, queuing theory has been used extensively to analyze communication networks. However, queuing theory fails to model the dynamics in packet

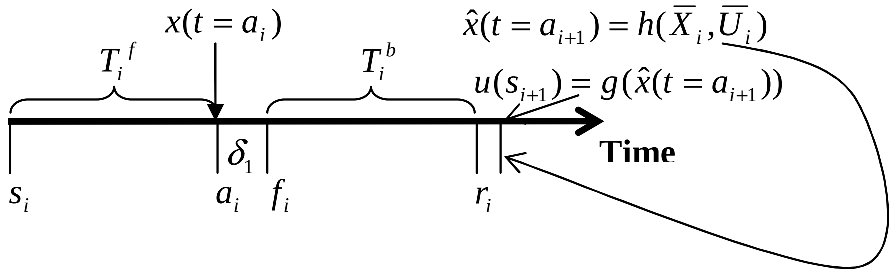

$x(t = a_i)$
T_i^f
T_i^b
$\hat{x}(t = a_{i+1}) = h(\overline{X}_i, \overline{U}_i)$
$u(s_{i+1}) = g(\hat{x}(t = a_{i+1}))$
Time
δ_1
s_i
a_i
f_i
r_i

Fig. 19. Timeline of the events during a predictive flow control strategy.

switched networks. Moreover, traffic on the Internet is self-similar and long-range dependent in nature. Kihong and Willinger [81] state that scale-invariant burstiness is a persistent trait existing across a range of network environments.

System identification is a technique of inferring models of dynamical systems based on observed data from the system. The current research work used linear and nonlinear system identification techniques to develop models of the traffic traversing through the best-effort networks. These models were used for designing control laws to control the end-to-end flow for real-time multimedia applications.

Ljung [82] describes a system as “an object in which variables of different kinds interact and produce observable signals.” Among all the observable signals, some signals are of interest to the observer. These signals are called the outputs. The signals that affect the system under observation can be categorized into two categories - input and disturbance. The signals affecting the system under observation but under the control of the observer can be termed as input signals. The other signals affecting the system but not under the control of the observer are termed as the disturbances.

In the current scenario end-to-end single real-time multimedia flows is considered to be the system that needs to be modeled. The cross traffic in the intermediate nodes of the path, not under the control of the applications at both the ends, is considered to be the disturbance. The rate at which information is generated by the application is considered to be the input signal to the system.

A model of a dynamic system is constructed out of the observed data. Output signal(s) is(are) the most important aspect of the development of a model. The last critical aspect for the development of a model for the best-effort networks is the choice of the output signal that reflects the state(s) of the system under observation. Among a host of signals that can be considered as the output signal of the system, three signals, packet loss rate, forward delay, and accumulation, hold the most promise. Each of these signals will be discussed in the next three subsections.

1. Packet Loss Signal

The packet loss signal of a real-time multimedia application comprises of not only the packets that are dropped in the queues of the intermediate nodes while traveling through the network, but also the packets that arrive at the destination after the expiration of the deadline of their arrival. A more suitable name for this signal would be Comprehensive Loss Rate (CLR).

In a real-time multimedia flow each packet sent by the sender has a deadline before which it needs to be played out by the receiver. If the packets arrive after their respective deadline decided by the synchronizer, they are considered to be of no use. Therefore, a packet that fails to meet the delay threshold is also considered to be lost.

The loss rate signal is correlated with the existence of congestion in the network. An increase in the magnitude of the loss rate signal signifies congestion in the network. However, this assertion is only valid for wired networks where the Bit Error Rate (BER) is low. In the case of wireless networks, a high loss rate might not be an indication of congestion in the network.

It is very difficult to judge whether a packet has been lost while traversing through the network in the presence of flow reversal. Flow reversal is used to describe the phenomenon that takes place when the sequence at which packets arrive at the receiver does not increase monotonically. Flow reversal happens when the packets take different paths while traversing through the network to reach the same destination. Figure 20 shows the phenomenon of flow reversal. The packet sent at time s_i arrives at the destination a_i. The packet sent at time s_{i+1} arrives at the destination at time $a_{i+1} < a_i$. The forward delays of packets i and $i+1$ are T_i^f and T_{i+1}^f, respectively. Also, T_d is the inter-departure time between the packets at the sender. T_d can also be viewed as the sampling interval of the system. It can be observed from the figure

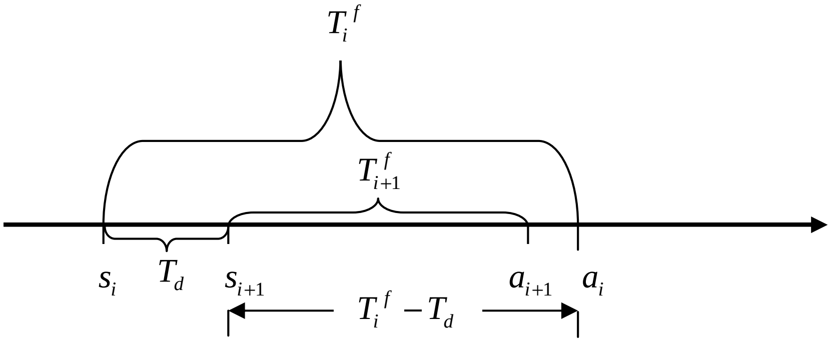


Fig. 20. Timeline of the events showing flow reversal of the packets.

20 that the flow reversal happens if

$$\frac{T^f_{i+1} - T^f_i}{T_d} < -1, \tag{6.5}$$

$$\Rightarrow T^f_{i+1} - T^f_i < -T_d, \tag{6.6}$$

$$\Rightarrow T^f_{i+1} < T^f_i - T_d.$$

If we model flow reversal using a differential equation in the continuous time domain, it can be described as:

$$\frac{dT^f}{dT_d} < -1, \tag{6.7}$$

where T^f is the forward delay associated with the flow. The subscript i has been dropped to signify that T^f is associated with a continuous flow rather than discrete packets.

During this research, it is assumed that flows do not undergo any significant flow reversal. This is a valid assumption as real world experiments have shown that the Internet shows very few instances of flow reversal. This simplifies the calculation of the loss signal in real-time. In the absence of flow reversal, if a packet with a sequence number j reaches the receiver after the packet with a sequence number i such that $j - i > 1$, all the packets with sequence numbers in the range of $[i + 1, j - 1]$ are assumed to be lost. This type of loss of packets is categorized as *pure loss.*

The packets that fail to meet their playout time deadline need to be accounted for in the loss signal too. Any packet i that satisfies equation

$$T^f_i > T_{Th}, \tag{6.8}$$

where T_{Th} is a threshold time that is determined by the needs of the real-time multimedia application, is considered to be lost. This type of loss is categorized as *delay* *threshold loss.* The larger the T_{Th}, the lower is the interactivity between the two end-users.

Comprehensive loss of packets L_{CL} is defined as

$$L_{CL} = L_T + L_P \tag{6.9}$$

where L_T and L_P are delay threshold loss and pure loss of the packets of a multimedia flow, respectively. Comprehensive Loss Rate (CLR) of packets in a flow can be defined as the ratio of the number of comprehensive loss of the packets and the total number of packets sent N_T from sender to receiver in a time interval $[t_1, t_2)$

$$CLR = \frac{L_{CL}}{N_T} \times 100. \tag{6.10}$$

It is expressed in percentage.

CLR was tried as an output signal in order to model the end-to-end dynamics of a single flow. However, the linear and nonlinear models developed using bit-rate of the flow as the input and the loss signal as the output were not very accurate. Figure 21 provides an example of the CLR of flow #1 in a simulation that uses the topology generated to validate designed control strategies. The simulation run time is 120 s and it has five CBR flows, using UDP, in the forward direction. The delay threshold, T_{Th}, used for calculating the CLR is 230 ms. This is the approximate upper bound of the end-to-end delay for interactive real-time multimedia flows exchanging information over best-effort networks.

2. Packet Delay Signal

Packet delay signal seems to hold more promise for modeling a single flow as compared to the CLR signal described in the previous subsection. Packet delay signal seems to be more smoother and content rich than the loss signal. However, there is one major

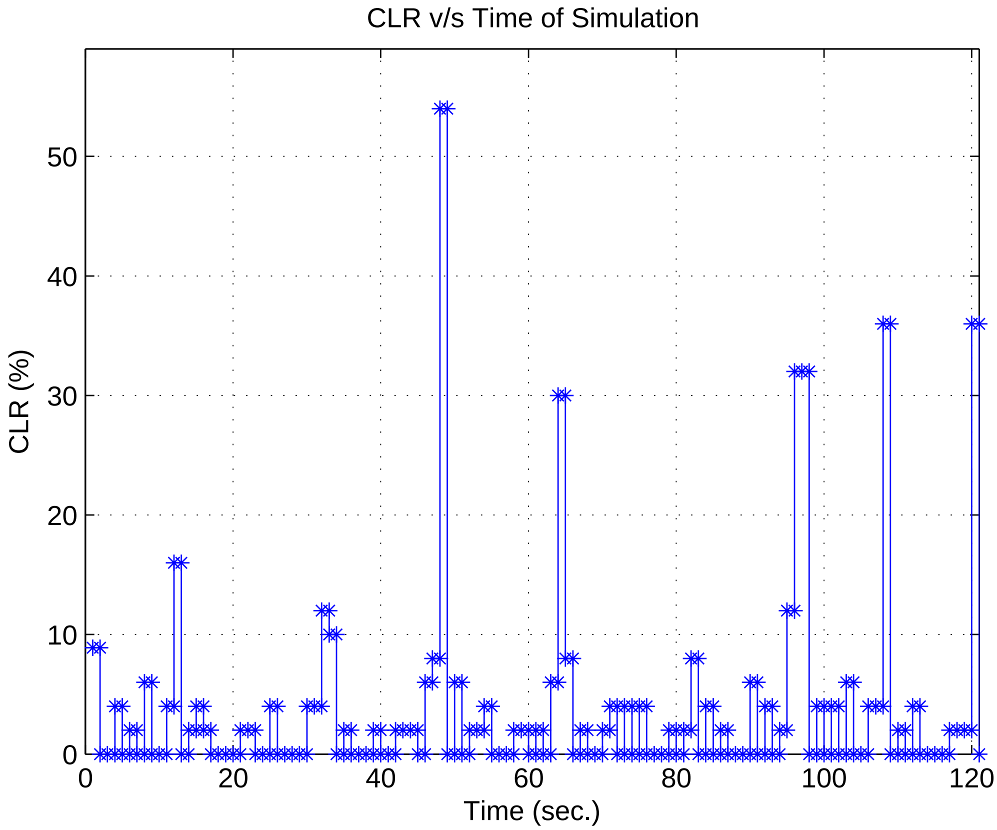


Fig. 21. CLR for flow #1 in a simulation using the topology to validate designed control strategies. The simulation run time is 120 s. It has five CBR flows sending traffic at the rate of 96.02 kbps.

problem in measuring this signal in real-time. The problem is - how to deal with the packet losses while measuring packet delays over the network?

The packet delay signal exists only if the packets reach the receiver at the other end. If the packets are lost during their travel, the delay signal can not be defined or measured. Researchers have used various techniques to determine the value of the packet delay signal in the presence of packet losses. The simplest way to deal with this issue is to ignore the packet delay values whenever the packet losses occur. However, this approach leads to lot of mathematical complications while dealing with the signal during system identification. The most rational approach to this issue is to allocate an artificial delay value to the signal in the case of packet losses. This artificial packet delay value determines the shape of the signal in the time-domain.

If it is decided to allocate an artificial delay value to the signal in the case of packet losses, the issue at hand is what artificial value should be allocated to the packet delay signal to substitute packet losses. The answer to the above question determines the difficulty with which the flow can be modeled based on the bit-rate of the flow as the input signal and packet delay as the output signal. Figure 22 shows the end-to-end delay experienced by the packets of flow #1 in a simulation that uses the topology generated to validate designed control strategies. The simulation run time is 120 s and it has five CBR flows, using UDP, in the forward direction. The lost packets of flow #1 in the network have been ignored while generating the signal shown in figure 22.

3. Accumulation Signal

The most suitable contender for the role of the output signal in order to model best-effort IP network is the accumulation signal. Accumulation signal is the difference between the number of bytes sent into the network by the source and the number

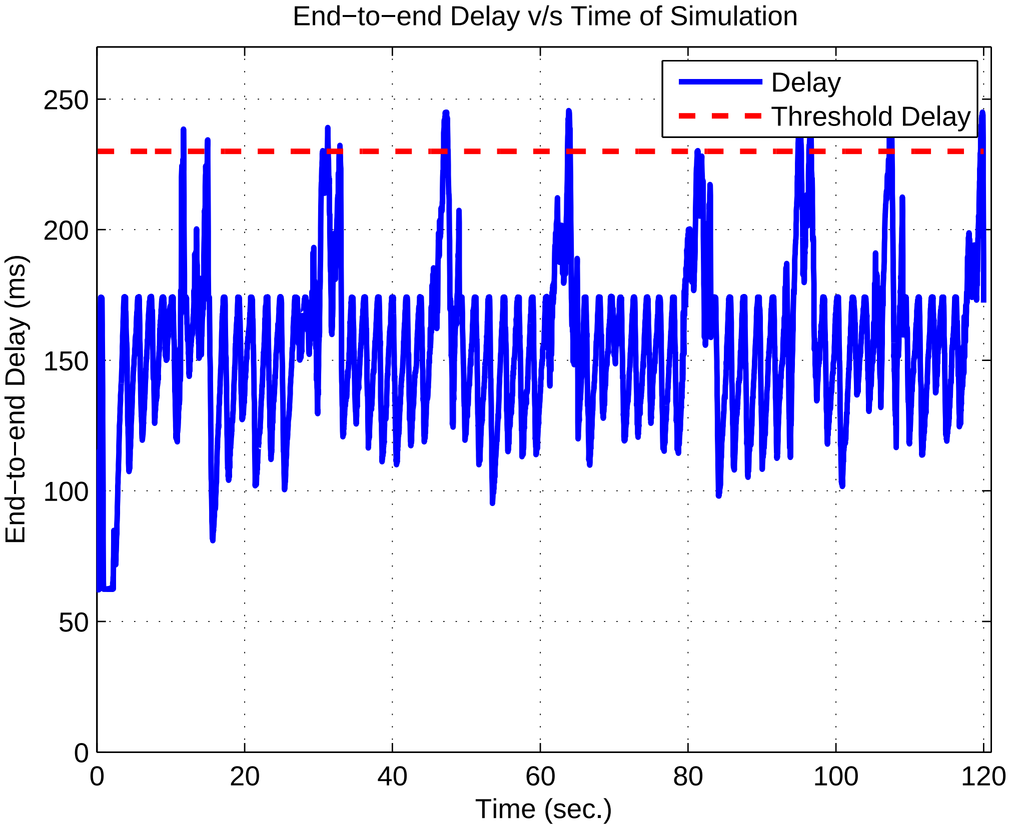


Fig. 22. End-to-end delay of packets for flow #1 in a simulation using the topology to validate designed control strategies. The simulation run time is 120 s and has five CBR flows sending traffic at the rate of 96.02 kbps.

of bytes received by the destination at a time t. Both the cross-traffic flow as well as the input bit-rate from the source affects the accumulation signal. Accumulation signal reflects both packet losses and the effects of delays in the network. Moreover, accumulation signal is a continuous-time signal. It has a definite value at each instant of time irrespective of packet losses.

Figure 23 shows the comparison of the accumulation signal w.r.t. the delay and comprehensive loss signals of a single UDP flow in a simulation of 120 s. It can be noticed that the delay signal has the same trend as the accumulation signal. Whenever delay goes up, accumulation increases. This shows a positive correlation between delay and accumulation. The packet losses increase when the accumulation goes up. They also happen intermittently in areas where the accumulation is at a normal level.

4. Linear State-Space Models and End-to-End Multimedia Flows

In the current research, a single flow in a network is modeled using fluid flow rather than packet flow. This means that differential equations in continuous time domain are used to describe the flow dynamics. Data traffic is modeled as a fluid. This can be done based on certain assumptions. These assumptions are:

1. The packet sizes of single flows have to be small enough to be considered as continuous fluid.

2. The end-to-end flow should be persistent. This means that the source of the traffic should have inexhaustible supply of packets to send into the network at specified time intervals.

Traffic in the Asynchronous Transfer Mode (ATM) networks can be easily modeled using fluid flows because the size of the packets, known as cells in ATM networks,

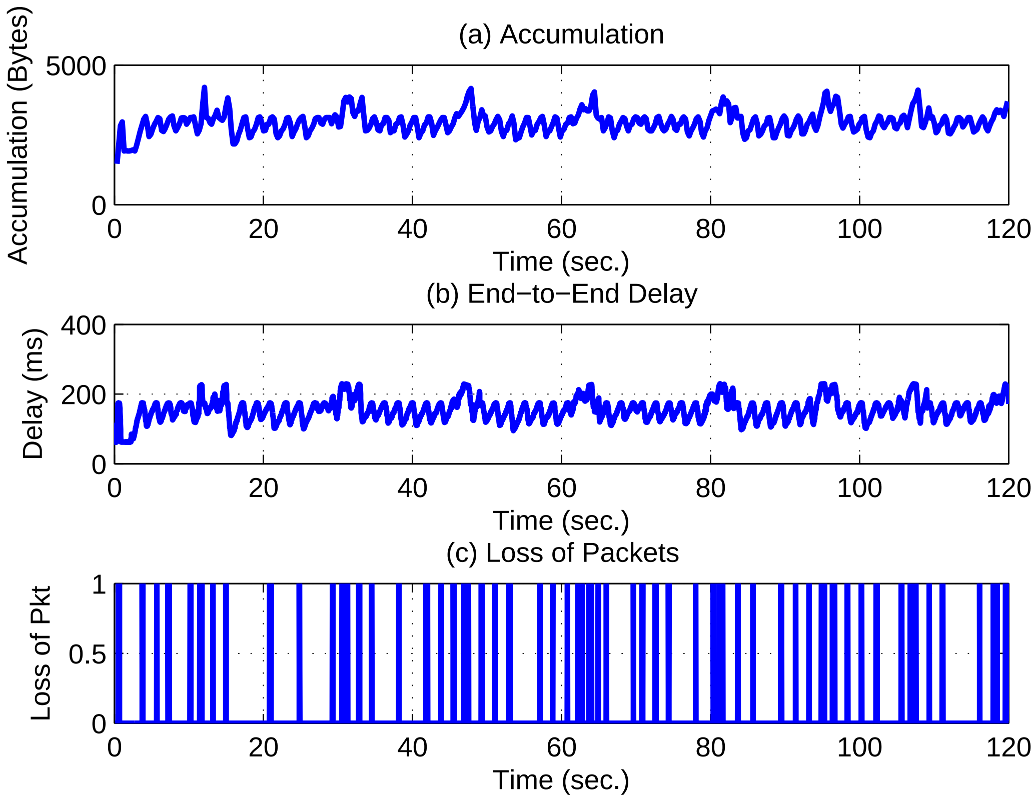


Fig. 23. Comparison of the accumulation signal w.r.t. the delay and comprehensive loss signals of a single flow.

is 53 bytes. This is very small. However, the packet sizes in the IP networks can be larger than even 1000 bytes. The assumptions of the fluid flow modeling can be justified in case of IP networks if the size of the packets in a single flow is compared to the total traffic flowing through the network. The ratio of the packets of single flow to the total traffic in an instant is very small. Therefore, the first assumption is still valid.

There are several different ways to describe a system of linear differential equations. The so-called state-space representation is given by the following equations in the continuous time domain:

$$\begin{aligned} \frac{d\bar{x}(t)}{dt} &= A\bar{x}(t) + B\bar{u}(t), \\ \bar{y}(t) &= C\bar{x}(t) + D\bar{u}(t), \end{aligned} \tag{6.11}$$

$\bar{x}(t)$ is a vector of l states that describe the system, A is an $l \times l$ square matrix, $\bar{u}(t)$ is a vector of m input signals to the system, B is an $l \times m$ matrix describing the effect of the input signals on the system, $\bar{y}(t)$ represents the output signal vector of size n, C is an $n \times l$ matrix showing the relationship between the measured output signals and the states, and D is a $n \times m$ matrix representing the feed-forward term of the system. There are l first-order differential equations. State space representation can be used to describe systems with Single Input Single Output (*SISO*) as well as for systems with Multiple Inputs and Multiple Outputs (*MIMO*). The state-space representation of the system shown by the set of equations 6.11, can be represented by the following equations in the discrete time domain:

$$\begin{aligned} \bar{x}(k+1) &= F\bar{x}(k) + G\bar{u}(k) \\ \bar{y}(k) &= H\bar{x}(k) + J\bar{u}(k) \end{aligned} \tag{6.12}$$

The matrices F, and G, H, and J have the same dimensions and serve the same roles as those mentioned for the matrices A, B, C, and D in the continuous time systems.

In the next few pages of this chapter, the differential and difference state-space equations to model a single flow as seen from end-to-end applications are derived. A SISO model with bit-rate as the input signal and accumulation as the output signal is developed. Although this model is derived from the conservation principles, yet it is not used for the development of different control laws. The main purpose for this model is to provide physical insights into the structure of the empirical models designed later on. A more information-rich model developed using black-box system identification techniques is used for the design of control laws later on.

While deriving the flow equations to characterize single flows in communication networks as observed from the end-to-end multimedia applications, it is important to describe the different definitions of time and relate one definition to the other. In the earlier chapters, forward and backward delay experienced by the packets while traversing from the receiver to the sender are denoted by the notations T_i^f and T_i^b, respectively, where i represents the sequence number of the packet. Now, a different notation is introduced to denote delay in the continuous time domain in order to derive the equations describing accumulation signal. Let $\tau(t)$ be the time delay present in the network for the flow traversing from one end to the other under observation at time t.

The multimedia communication network with a sender and a receiver at the two ends has two definitions of time. In order to maintain consistency with the notations already used in this research, the time associated with the event of sending packets of a flow into the network from the sender is called the send time and is represented by s. Similarly, the time associated with the event of arrival at the destination is called the arrival time and can be represented by a. The subscript i has been dropped from the notations to signify that the notations are now associated with flows in continuous domain rather than packets in the discrete domain. The two definitions of time are related to each other by the following equation:

$$a = s + \tau(s) \tag{6.13}$$

where $\tau(s)$ is the delay suffered by the flow while traversing through the network. The delay is a function of the send time s and is time varying.

If the state equations are defined with respect to the send time s, the resultant system incorporates the effect of flow reversal. As mentioned earlier, we had assumed that flow reversal will be ignored in the current work. Therefore, the state equations of the system are written down with respect to the arrival time a at the receiver in the continuous time domain. However, when the output or the measurement equations of the system are defined, they are defined with respect to the send time s.

5. Deriving the State Equations of a Conservative Single Flow System

Let $A(a)$ be the measure of the accumulation signal at arrival time a and $u(a)$ be the rate at which information is sent into the multimedia flow under observation. Let $z(a)$ be the arrival rate of the information at the destination. It is important to note that all these observations are being made at the receiver end of a conservative system. A network flow is considered to be conservative when there are no packet losses within the flow as it traverses from the source to the destination. The continuous time state space equations are derived first. Later on, these equations are discretized to obtain discrete time state space equations.

Applying the law of conservation of mass, the rate of change of the accumulation of the signal from a single source can be described by the following expression with respect to the arrival time a.

$$\frac{dA(a)}{da} = u\left(a - \tau(s)\right) - z(a), \tag{6.14}$$

where $\tau(s)$ is the variable delay suffered by the packets. By multiplying both sides of equation 6.14 with da and integrating between the time starting from t_1 to t_2 such that $t_2 > t_1$, we get

$$\int_{t_1}^{t_2} dA(a) = \int_{t_1}^{t_2} u\left(a - \tau(s)\right) da - \int_{t_1}^{t_2} z(a)da, \tag{6.15}$$

For the sake of simplicity, let us assume that the delay $\tau(s)$ remains constant. Therefore, $\tau(s) = \tau$ and equation 6.15 can be simplified as

$$\begin{aligned} \int_{t_1}^{t_2} dA(a) &= \int_{t_1}^{t_2} u\left(a - \tau\right) da - \int_{t_1}^{t_2} z(a)da, \\ A(t_2) - A(t_1) &= \int_{t_1}^{t_2} u\left(a - \tau\right) da - \int_{t_1}^{t_2} z(a)da. \end{aligned} \tag{6.16}$$

Let $x = a - \tau$. This implies $dx = da$ and the transformed equation 6.16 can be written as

$$\begin{aligned} A(t_2) &= A(t_1) + \int_{t_1-\tau}^{t_2-\tau} u(x)dx - \int_{t_1}^{t_2} z(a)da, \\ A(t_2) &= A(t_1) + \int_{t_1-\tau}^{t_1} u(x)dx + \int_{t_1}^{t_2-\tau} u(x)dx \\ &\quad - \int_{t_1}^{t_2} z(a)da. \end{aligned} \tag{6.17}$$

Let $t_1 = kT$ and $t_2 = kT + T$, where T is the sampling period of the system and $k \in \mathbb{N}$. Therefore, the above equation can be written as

$$\begin{aligned} A(kT+T) &= A(kT) + \int_{kT-\tau}^{kT} u(x)dx + \int_{kT}^{kT+T-\tau} u(x)dx \\ &\quad - \int_{kT}^{kT+T} z(a)da \end{aligned} \tag{6.18}$$

We have to express equation 6.18 for three different cases - each pertaining to the value of the constant time delay τ, which can have three different values i.e. $\tau < T$, $\tau = T$, and $\tau > T$.

It is assumed that the input signal passes through a Zero Order Hold *(ZOH)* before being applied to the system. This implies that the input rate $u(t)$ does not change between time instants kT and $kT + T$.

a. Case I: $\tau < T$

Equation 6.18 can be simplified as

$$\begin{aligned} A(kT+T) &= A(kT) + u(kT-T)\int_{kT-\tau}^{kT} dx + u(kT)\int_{kT}^{kT+T-\tau} dx \\ &\quad - \int_{kT}^{kT+T} z(a)da, \\ \Rightarrow A(kT+T) &= A(kT) + u(kT-T)\tau + u(kT)(T-\tau) - z(kT)T, \\ \therefore A((k+1)T) &= A(kT) + u((k-1)T)\tau + u(kT)(T-\tau) - z(kT)T. \quad (6.19) \end{aligned}$$

The final form of equation 6.19 can be written as

$$A(k+1) = A(k) + u(k-1)\tau + u(k)(T-\tau) - z(k)T. \quad (6.20)$$

b. Case II: $\tau = T$

When $\tau = T$, equation 6.18 takes the form

$$\begin{aligned} A(kT+T) &= A(kT) + u(kT-T)\int_{kT-T}^{kT} dx - z(kT)T, \\ \Rightarrow A(kT+T) &= A(kT) + u(kT-T)T - z(kT)T, \\ \therefore A((k+1)T) &= A(kT) + u((k-1)T)T - z(kT)T. \quad (6.21) \end{aligned}$$

The final form of equation 6.21 can be written as

$$A(k+1) \;=\; A(k) + u(k-1)T - z(k)T. \tag{6.22}$$

c. Case III: $\tau > T$

The final equation depends on how large the value of τ is. Without a bound on the value of τ, it is impossible to determine the exact expression for $A(k+1)$. In order to complete the set of equations, we assume that $T < \tau < 2T$ and derive the final form for this case. In this special case, we can represent the time delay τ as $T + \delta$ with range of δ being defined as $0 < \delta < T$. Equation 6.18 takes the form

$$\begin{aligned}
A(kT+T) &= A(kT) + \int_{kT-T-\delta}^{kT-T} u(x)dx + \int_{kT-T}^{kT} u(x)dx \\
&\quad + \int_{kT}^{kT-\delta} u(x)dx - \int_{kT}^{kT+T} z(a)da, \\
\Rightarrow A(kT+T) &= A(kT) + u((k-2)T)\delta + u((k-1)T)T \\
&\quad - \int_{kT-\delta}^{kT} u(x)dx - z(k)T, \\
\Rightarrow A(kT+T) &= A(kT) + u((k-2)T)\delta + u((k-1)T)T \\
&\quad - u((k-1)T)\delta - z(k)T. \qquad (6.23)
\end{aligned}$$

The final form equation 6.23 can also be written as

$$A(k+1) \;=\; A(k) + u(k-2)\delta + u(k-1)(T-\delta) - z(k)T. \tag{6.24}$$

d. Generalization of State Equations With Constant Network Time Delay, τ

If we take a closer look at equations 6.20 and 6.24, we can arrive at a generalized equation for single flow accumulation in a best-effort network for delay τ that lies between the values $(N-1)T$ and NT, such that $N \in \mathbb{N}$. The delay can be expressed as

$$\tau = (N-1)T + \delta, \tag{6.25}$$

where $0 < \delta < 1$. The accumulation at time instant $(k+1)T$ can be expressed in the following form:

$$A(k+1) \;=\; A(k) + u(k-N)\delta + u(k-N+1)(T-\delta) - z(k)T. \tag{6.26}$$

A simple exercise is performed to check the generalized state equation 6.26. Let the network delay τ be equal to the sampling period T. From equation 6.25, it can be seen that δ reduces to 0 and $N = 2$. The accumulation at time instant $(k+1)T$ can be expressed as:

$$A(k+1) \;=\; A(k) + u(k-1)T - z(k)T.$$

This expression is the same as that of equation 6.22.

6. Deriving the Output/Measurement Equations of a Conservative Single Flow System

The measurement of the accumulation signal (output) is performed by the sender after receiving the feedback from the receiver. Therefore, all the continuous time equations regarding measurement of accumulation (output), $y(s)$, are written with respect to send time s. Accumulation measured at time s is the delayed feedback provided by the receiver at the other end. This implies

$$y(s) \;=\; A(s-\tau(s)). \tag{6.27}$$

For the sake of simplicity, let us assume that the delay $\tau(s)$ remains constant.

This implies $\tau(s) = \tau$. Equation 6.27 can be written as

$$y(kT) \;=\; A(kT - \tau). \tag{6.28}$$

Like the previous section, we have to express 6.28 for the three cases of τ i.e. $\tau < T$, $\tau = T$, and $\tau > T$.

a. Case I: $\tau < T$

Equation 6.28 takes the form

$$\begin{aligned} y(kT) &= A((k-1)T) + \Delta A(\tau), \\ \Rightarrow y(kT) &= A((k-1)T) + (u((k-1)T) - z((k-1)T)))\tau. \end{aligned} \tag{6.29}$$

The final form of measurement equation in the case of $\tau < T$ can be written as

$$y(k) \;=\; A(k-1) + (u(k-1) - z(k-1))\tau. \tag{6.30}$$

b. Case II: $\tau = T$

Equation 6.28 takes the form

$$\begin{aligned} y(kT) &= A((k-1)T) \\ \Rightarrow y(k) &= A(k-1). \end{aligned} \tag{6.31}$$

This is the final form of measurement equation when $\tau = T$.

c. Case III: $\tau > T$

The final measurement equation, when $\tau > T$, depends on the upper bound of τ. In order to complete the set of equations, we assume that $T < \tau < 2T$ and derive the final form for this case. In this special case, we can the represent the time delay τ as

$T+\delta$ with range of δ being defined as $0<\delta<T$. Equation 6.28 takes the form

$$\begin{aligned} y(kT) &= A((k-2)T)+\Delta A(\tau), \\ \Rightarrow y(kT) &= A((k-2)T)+(u((k-2)T)-z((k-2)T))\delta. \end{aligned} \tag{6.32}$$

Therefore, the final form of measurement equation in the case of $T<\tau<2T$ can be written as

$$y(k) = A(k-2)+(u(k-2)-z(k-2))\delta. \tag{6.33}$$

d. Generalization of Measurement/Output Equations With Constant Flow Time Delay

Let τ be expressed as $\tau=(N-1)T+\delta$ such that $0<\delta<1$. Referring to equations 6.30 and 6.33, the generalized equation of measured accumulation $y(kT)$ in the network with the flow delay τ between time instant $(N-1)T$ and NT, such that $N\in\mathbb{N}$ is:

$$y(k) = A(k-N)+(u(k-N)-z(k-N))\delta. \tag{6.34}$$

7. Final State-Space Equations for a Conservative Flow in a Best Effort Network

If the constant delay, τ, for a conservative flow falls between time instant $(N-1)T$ and NT, such that $N\in\mathbb{N}$, then τ can be written as

$$\tau=(N-1)T+\delta, \tag{6.35}$$

with the range of δ being $0<\delta<T$, then the state-space linear equations describing multimedia flow in terms of accumulation signal A(k) as the output and the bit-rate signal u(k) as the input can be written as

$$A(k+1) = A(k)+u(k-N)\delta+u(k-N+1)(T-\delta)-z(k)T, \tag{6.36}$$

$$y(k) \quad = \quad A(k-N) + (u(k-N) - z(k-N))\delta. \tag{6.37}$$

B. Empirical Modeling of Single Flow Dynamics in Best Effort Networks

Instead of using the state equations derived in the previous subsection, the current research will use empirical models based on data collected from the experiments performed in simulation framework. These empirical models are more accurate as they capture the dynamics of single flow better. However, there is an important reason for deriving the state equations earlier - determining the structure of the empirical models. As explained later in this section, selection of a right model from a set of candidate models is a very difficult task. An insight into the basic model developed from conservation of flow principles helps to make the task a bit easier.

System identification is based on experimentation. Input and output signals from the system are recorded and subjected to data analysis in order to infer a model. This approach is also known as data-driven modeling. The construction of a model from data involves three basic entities:

1. The data: The input-output data are sometimes recorded during a specifically designed identification experiment, where the user may determine which signals to measure and when to measure them and may also choose the input signals. The data should have maximum information. In other cases, the user may decide to use the data obtained during the normal course of operation of the system.

2. A set of candidate models: A set of candidate models is obtained by specifying within which collection of models are tested in order to select the most suitable one. This is the most important and most difficult choice of the system identification procedure. In many cases standard linear models may be employed,

without reference to the physical background. Many times linear models fail to provide satisfying results. Neural networks can be used to develop nonlinear models. Model set, whose parameters are basically viewed as vehicles for adjusting the fit to the data and do not reflect physical considerations in the system is called a black-box.

3. A rule by which candidate models can be assessed using the data: Determining the best model in the set, guided by data is the next step. This is the identification method. The assessment of model quality is based on how models perform when they attempt to reproduce the measured data. Here we try to reject the model based on old and new data. Failing to do so does not mean that the model is good. In fact, all attempts are made to invalidate a model.

The models derived after performing the above steps need to be tested to determine whether they are valid for their purpose. The tests done to validate a model are known as model validation. They involve various procedures to assess how the derived model relates to observed data, to prior knowledge, and to its intended use. An excellent text for the study of the system identification techniques has been written by Ljung [82].

1. Search for Model Structures

Many different approaches were tried out for modeling a single flow in a best-effort network using the bit-rate and the accumulation signal. Initially, the research focused on using nonlinear models. Feed-forward Multi Layer Perceptron ($FMLP$) and Recurrent Multi Layer Perceptron ($RMLP$) neural networks were tried out. Some promising models were developed using the Radial Basis Functions (RBF). RBFs are very effective in capturing the nonlinear behavior of systems. However, when the single step ahead prediction results from these nonlinear models were compared with the prediction provided by the linear models like Auto-Regressive eXogenous (ARX), it was found that the improvement in the prediction did not justify the additional complexity of the nonlinear predictors.

Moreover, the Model Predictive Control (MPC) framework, described later on in the current chapter, is used to develop the predictive control strategies for adaptive flow control of the real-time multimedia flows over best-effort networks. The MPC framework is well established for linear models like ARX and ARMAX in the literature. If the system is represented by a nonlinear models like FMLP and RBF, a lot more theoretical work in the development of predictive control strategies must be done. Furthermore, the resulting controller is expected to be too complex for easy implementation. This task is beyond the scope of the current work because of the constraints of space as well as time. Therefore, the mathematical models developed to represent single flows in best-effort networks are linear in nature.

2. Linear Models

The current research utilizes parametric linear system identification methods to model the accumulation signal. These models describe the system with finite number of parameters. The parameters are determined by optimizing some objective function by methods such as linear regression. Linear models have the advantages that they are computationally easy to develop, and the associated controller design is much simpler. Figure 24 shows the general linear model structure. Other forms of linear models can be derived from this general model.

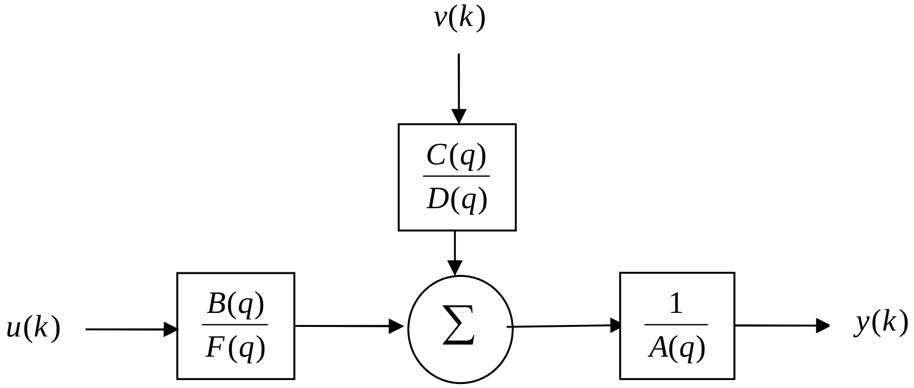


Fig. 24. General linear model structure.

a. Auto-Regressive Exogenous Model

ARX is the simplest and the most used model structure in linear system identification. The general SISO ARX model can be expressed by the following linear difference equation:

$$y(k) = a_1 y(k-1) + \ldots + a_{n_y} y(k-n_y) + b_1 u(t-n_k) + \ldots + b_{n_u} u(k-n_u-n_k+1) + \nu(k), \tag{6.38}$$

where $u(k)$ and $y(k)$ are the input and the output of the SISO ARX model, $\nu(k)$ is the noise disturbance, n_y and n_u are the number of past outputs and the number of past inputs used in the model, and n_k is the pure time delay (i.e. the dead time) in the system. The coefficients a_1, …, a_{n_y} and b_1, …, b_{n_u}, are known as the model parameters. The noise disturbance $\nu(k)$ is usually assumed to be white noise. Figure 25 depicts the structure of ARX models.

On the basis of the model defined by the ARX structure, the Single Step Pre-

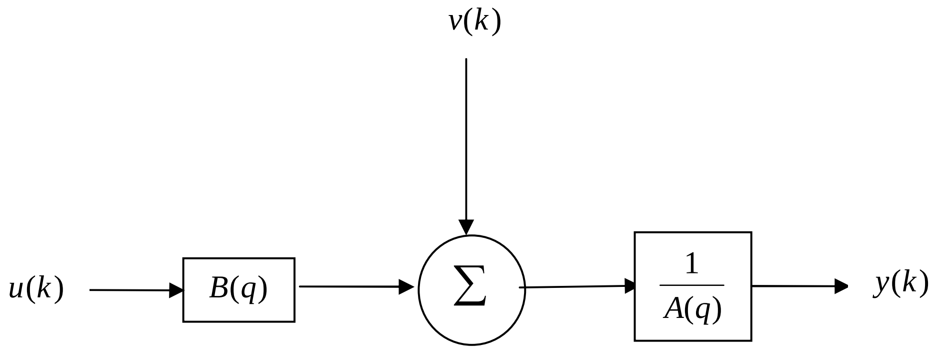


Fig. 25. ARX model structure.

diction (SSP) of the system output can be obtained by following expression

$$\widehat{\mathbf{y}}(k|k-1,\theta) = \phi^T(k)\theta, \tag{6.39}$$

where, $\phi(k) = [y(k-1), \ldots, y(k-n_y), u(k-n_k), \ldots, u(k-n_u-n_k+1)]^T$ and $\theta = [a_1, \ldots, a_{n_y}, b_1, \ldots, b_{n_u}]$.

Equation 6.39 is in the form of a linear regression. The model parameter vector θ is the regression vector. It is estimated using the least-squares technique that minimizes the mean-square error of the prediction error as compared to the actual data.

The Auto-Regressive (AR) model is the special case of the ARX model where only past values of the output are used for modeling the system. The AR model is also known as time-series model.

b. Auto-Regressive Moving Average Exogenous Model

A more general input-output model is given by the Auto-Regressive Moving Average Exogenous Model (*ARMAX*). The AR in the ARMAX model refers to the autoregressive part. The MA is the moving average and X corresponds to the extra input called the exogenous variable. The ARMA model formulates the disturbance term as a moving average of a white noise process. The SISO ARMAX model can be represented by the following equation:

$$\begin{aligned} y(k) &= a_1 y(k-1) + \ldots + a_{n_y} y(k-n_y) + b_1 u(t-n_k) + \ldots + \\ & \quad b_{n_u} u(k-n_u-n_k+1) + c_1 e(k-1) + \ldots + c_{n_e} e(k-n_e), \end{aligned} \tag{6.40}$$

where, $e(k)$ is the prediction error and n_e is the number of past noise terms used in the model. The other variables are the same as in the ARX model.

On the basis of the model defined by the ARMAX structure, the Single Step Prediction (*SSP*) of the system output can be obtained by following expression

$$\widehat{\mathbf{y}}(k|k-1,\theta) = \phi^T(k)\theta \tag{6.41}$$

where, $\phi(k) = [y(k-1), \ldots, y(k-n_y), u(k-n_k), \ldots, u(k-n_u-n_k+1), e(k-1,\theta), \ldots, e(k-n_e,\theta)]^T$ and $\theta = [a_1, \ldots, a_{n_y}, b_1, \ldots, b_{n_u}, c_1, \ldots, c_{n_e}]$.

Equation 6.41 is in the form of a pseudo-linear regression and hence iterative. The least squares method can be used to solve for θ. The Auto-Regressive Moving Average (ARMA) model is the special case of the ARMAX model where no input or exogenous variable is used while modeling the system. The system identification toolbox of Matlab is used for estimating the parameters of the linear models.

3. ARX Modeling of a Single Flow in a Best Effort Network

Ohsaki, Morita, and Murata [83] used system identification techniques to model the Round Trip Time (RTT) dynamics of a flow over the Internet. They treat the network measured from one end to the other by the source and the destination as a black box. The authors model the RTT variation as a Single Input Single Output (*SISO*) Auto-Regressive eXogenous (ARX) model with the inter-departure time of the packets as the input signal. Internet Control Message Protocol (*ICMP)* Echo messages are chosen as the probe packets to perform the measurements.

In order to model a single flow in a best-effort IP network using data driven linear predictors, it is very important to collect the input and output data in the right manner. The input data used for the development of the model should be able to excite all the important dynamics of the system that needs to be modeled for the resulting model to be effective.

a. Selection of the Sampling Interval for Different Signals

During the experiments using simulation topologies designed in ns-2, the control signal i.e. the bit-rate, needs to be calculated and applied every T_c seconds. The calculation of the control signal is dependent on the accumulation of the data in the system. The accumulation signal needs to be sampled at every T_a seconds in order to calculate the control signal every T_c seconds. It is important to select these parameters carefully because the accuracy of the linear mathematical model will be based on the selection of these parameters. The value of these time periods provide information about how fast the dynamics of the system responds to changes in the input. If a system is really slow to respond to changes, i.e. time constant is large, then there is no point in applying control at a rate faster than the time constant. In fact, this effect can destabilize the system.

A set of open loop control experiments are designed in order to determine the time period for the application of the control signal. Figure 26 shows the way the experiments are performed to determine T_c.

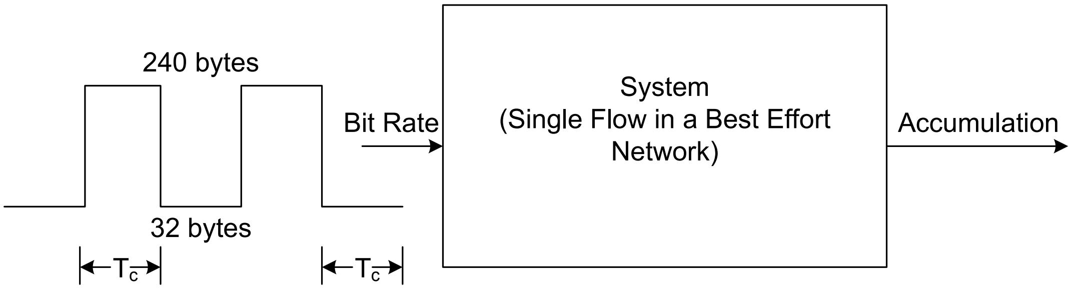


Fig. 26. Experiment designed to determine T_c, time period of the application control signal for flow control.

An input signal of alternating bit-rates, sending packets every 20 ms is selected. The high bit-rate section of the input signal generated traffic at the rate of 96 kbps (240 byte packets). The low bit-rate section of the signal generated traffic at the rate of 12.8 kbps (32 byte packets). This range of traffic, between the low bit rate of 12.8 kbps and 96 kbps, is selected based on the typical levels of traffic generated by the different codecs. Moreover, the frame size of the most encoders used in VoIP applications is around 20 ms. The time period of switching between the low bit rate and the high bit rate segments in the input signal, T_c, was varied from 60 ms to 3000 ms in a set of nine experiments performed using the network with mean CLR of three percent. Each simulation experiment is run for a simulation time of 120 s.

For determining which time period of switching the bit-rate is most effective in controlling the flow, the CLR of the controlled UDP packets for flows of interest are calculated. The switching time period, T_c, that makes the maximum impact to reduce the CLR is selected as the time period of control signal to the system. The results from the nine experiments can be seen in the figure 27. As can be seen from the figure, on an average, a time period of either 240 ms or 480 ms makes the maximum impact in reducing the packet loss for the controlled multimedia UDP flows. Therefore, T_c is chosen as 240 ms and in all the experiments that follow.

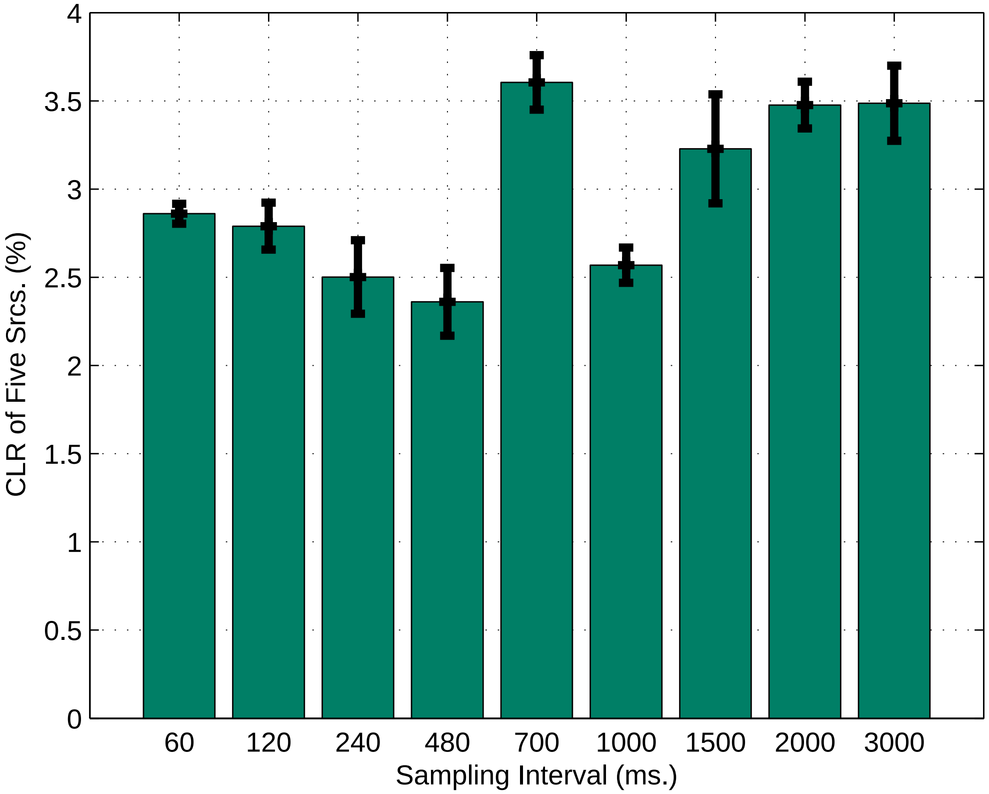


Fig. 27. Comprehensive packet loss percentages of the five flows in the forward direction with the bit-rate control switching time period T_c of 60 ms, 120 ms, 240 ms, 480 ms, 700 ms, 1000 ms, 1500 ms, 2000 ms, and 3000 ms.

Although the accumulation signal is measured every 20 ms, yet an averaged (smoothed) value of the signal over 240 ms is used for prediction purposes. This implies that T_a has also been selected to be 240 ms for the current research. Although, the actual sampling frequency of the accumulation signal is 50 Hz, yet the sampling frequency of the smoothed accumulation signal used for the purpose of prediction is 4.167 Hz. The smoothed accumulation signal is calculated from the sampled accumulation using the following equation:

$$A_m(k) = \frac{1}{\lceil \frac{T_a}{20} \rceil} \sum_{i=1}^{i=\lceil \frac{T_a}{20} \rceil} A_i, \tag{6.42}$$

where A_i is the ith sample of $A(t)$ such that $i \in \mathbb{N}$ and the range of i spans $[1, \lceil \frac{T_a}{20} \rceil]$. For the current research, the maximum value of index i is $\frac{240}{20} = 12$.

The sampling frequency of control signal is equal to the sampling frequency of the accumulation signal. This frequency is set to 4.167 Hz.

b. Input Signals Used in Data Collection

As per the survey of the codecs used in VoIP applications, the lowest and the highest bit-rate generated by the commercially available codecs is 36 kbps (90 bytes every 20 ms) and 96 kbps (240 byte packets every 20 ms), respectively. The bit-rates mentioned include the header sizes along with the actual traffic generated by the codecs. The control strategy for adaptive flow control involves changing the bit-rate of the real-time multimedia flow based on the feedback of the accumulation signal in the network. The bit-rates of a real-time multimedia application that has already started sending traffic after establishing a session with the receiver end can be changed using an adaptive encoder.

The current research work does not delve into the actual design of the adaptive codec. It tries to lay down the specifications of the final codec design in terms of the bit-rates that the codec can generate and the end-to-end quality of the voice associated with each of the bit-rate levels. The design of a codec that manages to meet the design specifications laid down in the current work is a topic for further research.

The desired adaptive encoder for the current research can switch between N different bit-rates in order to control flow over best-effort network while trying to keep the end-to-end QoS measured in terms of Mean Opinion Score (MOS) constant. The desired adaptive encoder has the following bit rates:

1. 36 kbps: 90 byte packets (including all headers) sent every 20 ms.
2. 48 kbps: 120 byte packets (including all headers) sent every 20 ms.
3. 60 kbps: 150 byte packets (including all headers) sent every 20 ms.
4. 72 kbps: 180 byte packets (including all headers) sent every 20 ms.
5. 84 kbps: 210 byte packets (including all headers) sent every 20 ms.
6. 96 kbps: 240 byte packets (including all headers) sent every 20 ms.

The number of unique bit-rates is varied from 2 to 12 levels. It is observed that if N is fixed to less than six levels then the designed control laws fail to improve upon or maintain the end-to-end voice QoS. The control law implementation becomes too course to make any difference in the system. Changing the number of levels beyond 6 does not make any difference to the eventual outcome.

Two experiments were conducted in order to collect data for the development of a linear predictor that models the flow. Both experiments utilize the network topology with 5 controlled UDP sources that undergo a mean CLR of 3% while sending traffic from one end to the other. Figure 28 shows the the input bit-rates during the two experiments as measured by the controlled UDP node #1 for flow #1.

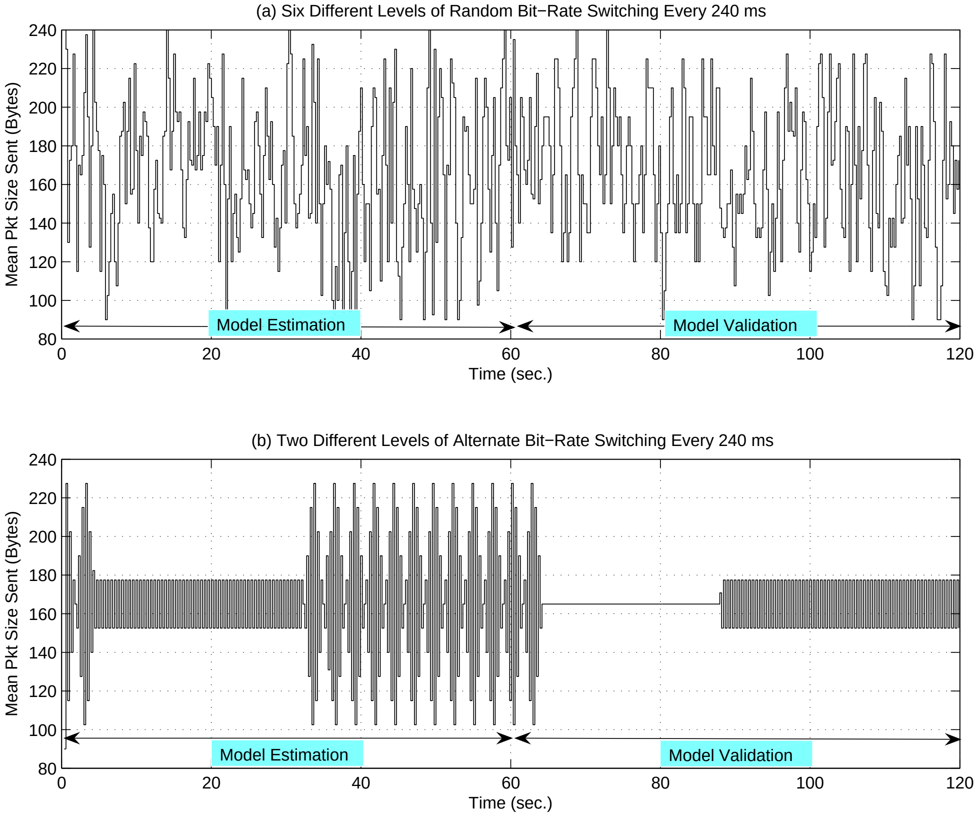


Fig. 28. Mean bit-rate of flow #1; (a) random switching between six bit rates, (b) alternate switching between HBR and LBR.

The first experiment involved sending a traffic that switches randomly between six different levels, mentioned in the previous paragraph, every 240 ms and measuring the accumulation. The second experiment involved sending a traffic that switches between the low bit rate of 36 kbps and a high bit rate of 96 kbps alternately and measuring the accumulation. The corresponding mean accumulation signals, measured every 20 ms and averaged over a time period of 240 ms at the sender of flow #1 are shown in figure 29.

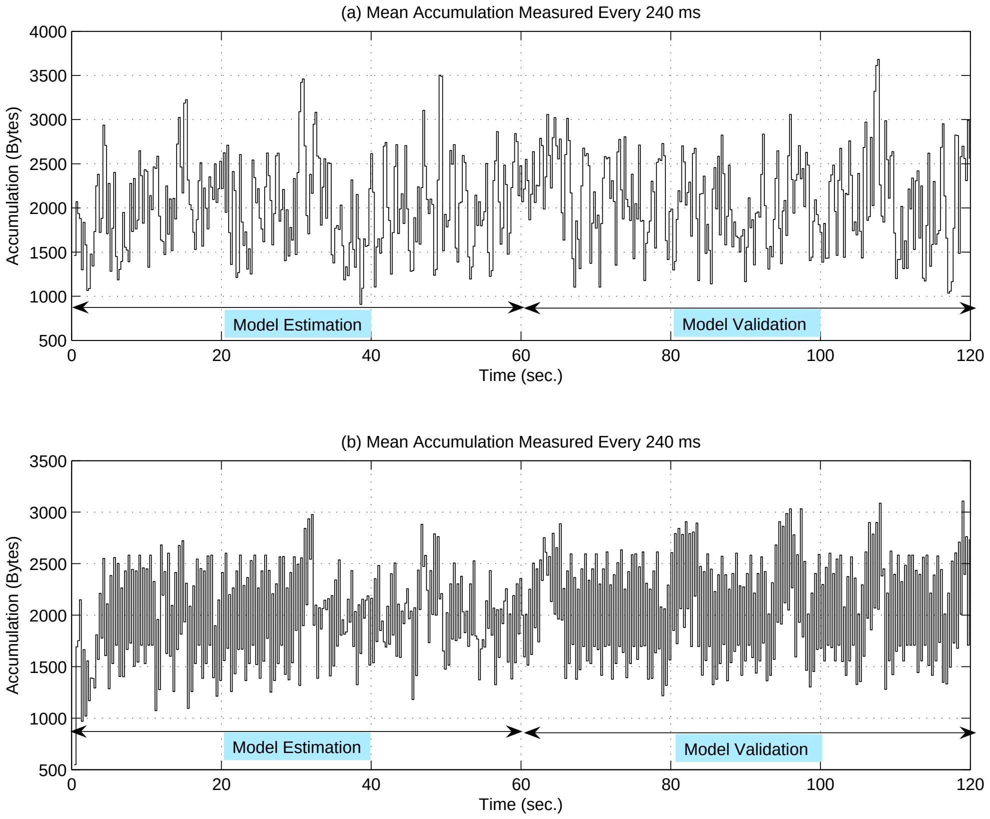


Fig. 29. Measured mean accumulation signal of flow #1; (a) random switching between six bit rates, (b) alternate switching between HBR and LBR.

The plots of the input bit-rate as shown in the figure 28(a) and (b) do not show the bit-rate exactly switching between levels as mentioned earlier. Especially, in the figure 28(b), the packet sizes lie between 90 bytes and 240 bytes instead of exactly at 90 bytes and 240 bytes. This is because the signals shown in the plots are the average packet sizes, measured every 240 ms. The figures do not show the individual packet sizes. The packet size averaging makes the signal look the way it is shown on the plot.

In order to estimate a model for the flow, half of the data, i.e. first 60 seconds from each experiment, are extracted and combined artificially to form the data set that is used for estimating the linear models. The leftover data, i.e. the last 60 seconds from each experiment, are combined in order to validate the designed models. The idea behind artificially creating the input and the output signals by using data from two separate simulation experiments is to try to model all the dynamics of the system. The first experiment provides an input signal to the system that excites the dynamics of the network when the the controller is operating with six different bit-rates. The second experiment tries to capture the dynamics of the network when the controller alternates between the lowest bit-rate and the highest bit-rate.

c. Variation of the Network Topology Used for Designing Control Laws to Obtain Different Loss Rates

In chapter V, two different network topologies are discussed. The first network topology is used for development of the control laws. Using this network topology with five controlled UDP flows in the forward as well as in the backward direction, networks with different loss rates are created. This is achieved by varying the bandwidth capacity of the R0-R1 link lying between Router 0 and Router 1. Table XI shows how the change in the bandwidth capacity of the R0-R1 link affects the average CLR of the packets of the five controlled UDP flows. By varying this parameter, six different network topologies with different loss percentages of the controlled UDP flows are obtained. The loss percentages range from 3% to 15%. These six network topologies are used to test the efficacy of the control laws and their robustness under different network loss rates.

Table XI. Bandwidth capacity of the R0-R1 link in the network topology #1 and resulting mean CLR of the 5 controlled UDP flows.

	BWidth Capacity (in Mbps) of R0-R1	CLR
1	50.92	3
2	44.92	5
3	40.92	7
4	36.92	9
5	35.92	11
6	28.92	15

4. Designed ARX Models

Linear predictors are simple and can be used easily in the MPC framework. Two ARX models are designed based on the input and output data obtained from the experiments performed on 3% and 9% CLR network topologies.

The structure of the ARX SSP designed using the data obtained from the 3% CLR network topology is $n_y = 15$, $n_u = 14$, and $n_k = 1$. This predictor is represented as ARX [15 14 1] in this document. The $A(q)$ and $B(q)$ polynomials along with the values of the coefficients of the model are

$$\begin{aligned} A(q) \quad = \quad & 1 + 0.0244q^{-1} - 0.5285q^{-2} + 0.1873q^{-3} + 0.07573 \\ & q^{-4} + 0.008386q^{-5} + 0.02054q^{-6} - 0.2202q^{-7} - \\ & 0.1836q^{-8} - 0.0229q^{-9} + 0.1109q^{-10} + 0.1383q^{-11} \\ & -0.09604q^{-12} + 0.1602q^{-13} - 0.1447q^{-14} + 0.04906 \\ & q^{-15}, \end{aligned} \tag{6.43}$$

$$\begin{aligned} B(q) \quad = \quad & 8.838q^{-1} - 2.487q^{-2} - 3.86q^{-3} + 4.355q^{-4} - \\ & 0.7367q^{-5} + 2.134q^{-6} - 1.112q^{-7} - 3.669q^{-8} \\ & -0.8616q^{-9} + 0.4826q^{-10} + 3.136q^{-11} - 0.07706q^{-12} \\ & +1.276q^{-13} - 0.3919q^{-14}, \end{aligned} \tag{6.44}$$

Similarly, the structure of the SSP designed using the data obtained from the 9% CLR network topology is ARX [14 15 1]. $A(q)$ and $B(q)$ polynomials along with the values of the coefficients of the model are

$$\begin{aligned} A(q) \quad = \quad & 1 - 0.1519q^{-1} - 0.5841q^{-2} + 0.2529q^{-3} + 0.04036q^{-4} \\ & -0.1455q^{-5} - 0.121q^{-6} - 0.189q^{-7} - 0.06726q^{-8} \\ & +0.07563q^{-9} + 0.2044q^{-10} + 0.08776q^{-11} - 0.1296q^{-12} \\ & +0.2397q^{-13} - 0.1537q^{-14}, \end{aligned} \tag{6.45}$$

$$\begin{aligned} B(q) &= 7.691q^{-1} - 3.891q^{-2} - 3.62q^{-3} + 5.121q^{-4} - 2.577q^{-5} \\ &\quad -0.1723q^{-6} - 2.662q^{-7} - 2.887q^{-8} + 0.5515q^{-9} \\ &\quad +2.167q^{-10} + 3.374q^{-11} - 0.8988q^{-12} + 2.346q^{-13} \\ &\quad +0.9643q^{-14} - 1.048q^{-15}. \end{aligned} \tag{6.46}$$

The notation $A(q)$ used for denoting the denominator of the filter modeling the system in the above equations should not be confused with the accumulation signal.

The SSP performance of the two models is measured in terms of four different performance metrics. These metrics are:

1. Mean Square Error (*MSE*): MSE is the ratio between the sum of the square of the prediction error and the sum of the square of the input data. MSE is defined by

$$MSE = \frac{\sum_{j=1}^{N_S} \left(A(j) - \hat{A}(j)\right)^2}{\sum_{j=1}^{N_S} A(j)^2} \times 100, \tag{6.47}$$

   where N_S is the total number of samples in the output data (accumulation), $A(j)$ is the jth sample of the actual accumulation, and $\widehat{A}(j)$ is the one step ahead prediction of the jth sample of the accumulation signal. MSE gives an idea of the overall quality of the prediction. The lower the MSE is the better is the performance of the predictor.

2. Maximum Absolute Error (*MAE*): MAE is the maximum error between the actual accumulation signal and the one step ahead prediction of the accumulation signal. It is defined as

$$ME = \max_{1 \leq j \leq N_S} |A(j) - \hat{A}(j)|. \tag{6.48}$$

   This definition of error tries to provide some insight into the worst case error.

The lower the MAE the better is the performance of the predictor.

3. Fit: The norm of a vector is a scalar that gives some measure of the magnitude of its elements. The fit of a vector is defined as

$$\|A\|_2 = \left(\sum_{j=1}^{N_S} |A(j)|^2 \right)^{1/2} \tag{6.49}$$

$$fit = \left(\frac{1 - \|(\widehat{\mathbf{A}} - \mathbf{A})\|_2}{\|\mathbf{A} - \overline{\mathbf{A}}\|_2} \right) * 100 \tag{6.50}$$

where $\widehat{\mathbf{A}}$ represents the vector of the SSP of the accumulation signal, $\mathbf{A}$ is the vector of the measured values of the accumulation signal, and $\overline{\mathbf{A}}$ is vector of the mean value of the measured accumulation signal from the simulation experiments. The higher the fit is the better is the performance of the predictor.

4. Probability of Normalized Error to be less than 10 percent (*PNE10*): During the design of the linear predictors, it was felt that the previous three metrics are not providing enough insight into the details of the prediction error. It happened many times that although the previous three metrics were pointing towards good prediction results, yet the predictions were not as good as expected. Therefore, a new metric to judge the quality of the prediction is devised. This metric shows the probability of the normalized absolute prediction error to be less than 10 percent. The higher PNE10 is the better the prediction.

$$PNE10 = P\left(\frac{|\widehat{A}(j) - A(j)|}{A(j)} \leq 0.1 \right) \tag{6.51}$$

such that $j \in \{1, \ldots, N_S\}$.

Table XII provides the SSP performance results of the two designed linear predictors. The table shows the performance of the two SSP predictors, measured in terms of four stated metrics, while predicting the accumulation data collected from four different network topologies.

Table XII. Performance of the one step ahead predictors.

Percent Loss	MSE (%)	MAE (Bytes)	Fit (%)	PNE10
ARX [15 14 1]				
3	2.44	2215.0	24.76	0.52
5	2.50	2342.5	24.75	0.50
9	2.84	2302.5	19.52	0.50
15	3.52	2282.5	9.66	0.47
ARX [14 15 1]				
3	2.34	2215.0	24.88	0.54
5	2.24	2342.5	27.02	0.55
9	2.36	2302.5	26.70	0.56
15	2.83	2282.5	21.39	0.57

Figure 30 shows the SSP of the mean accumulation signal of 3% CLR network using ARX [15 14 1] and the corresponding normalized prediction error associated with each sample. Figure 31 plots the probability of the normalized SSP error to be less than a certain x percent for ARX [15 14 1], while predicting the mean accumulation signal of the 3% CLR network.

Figure 32 shows the SSP of the mean accumulation signal of 9% CLR network using ARX [14 15 1] and the corresponding normalized prediction error associated with each sample. Figure 33 plots the probability of the normalized single step ahead prediction error to be less than a certain x percent for ARX [14 15 1], while predicting

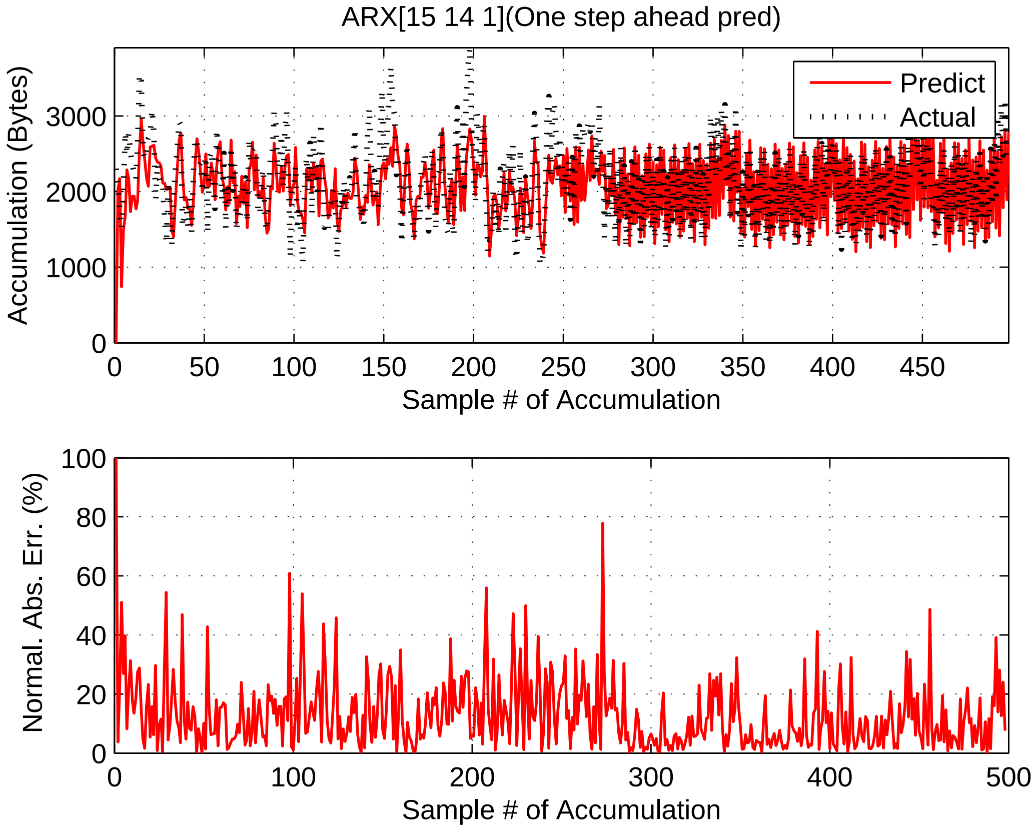


Fig. 30. Single step ahead prediction of the accumulation signal of the 3% CLR network using ARX [15 14 1] and the corresponding normalized prediction error.

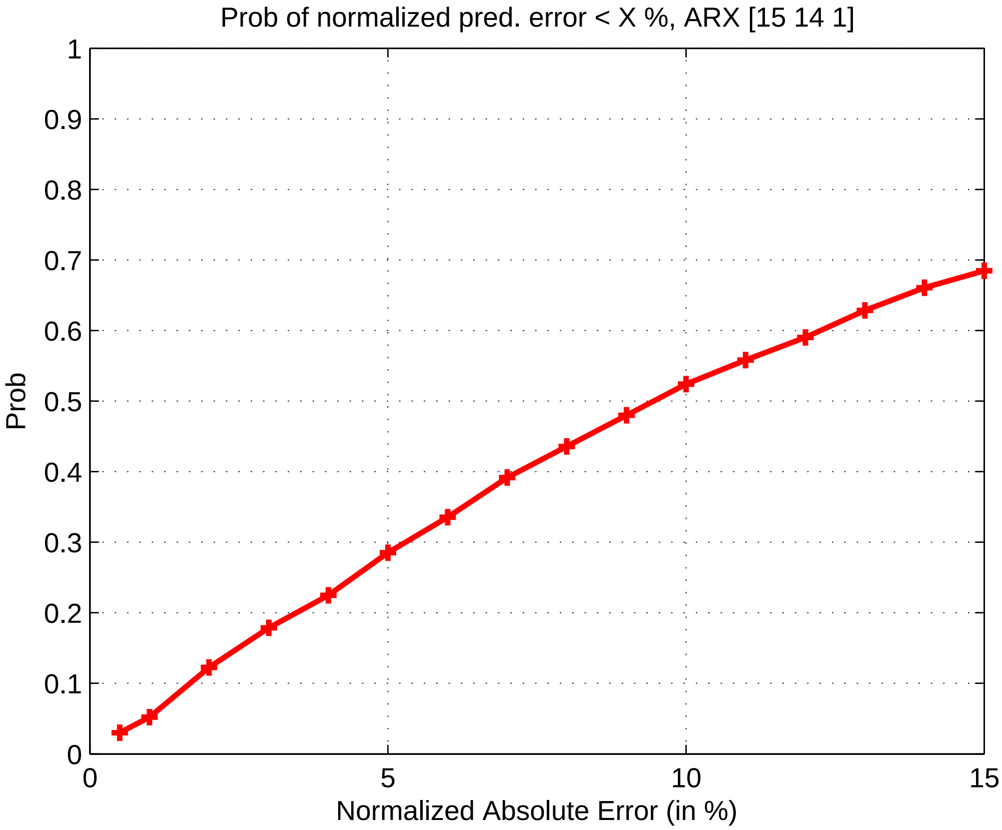


Fig. 31. Performance metric PNE10 for the predictor ARX [15 14 1] while predicting the accumulation signal from 3% CLR network topology.

the mean accumulation signal of 9% CLR network.

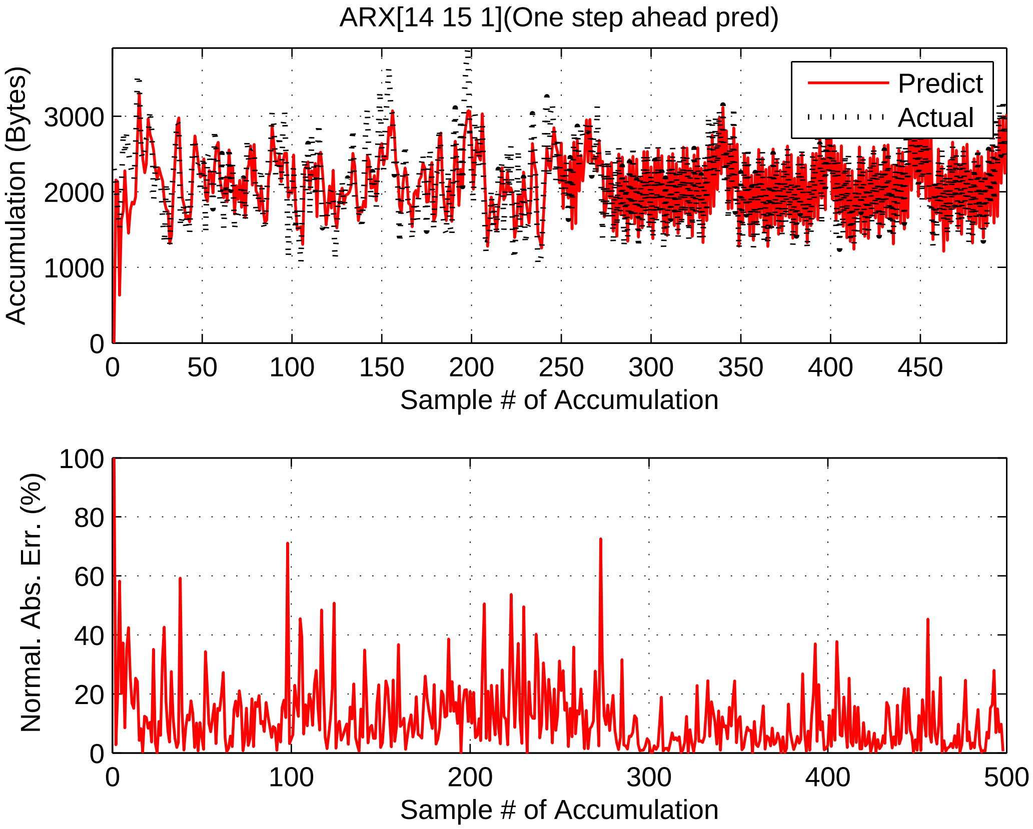


Fig. 32. Single step ahead prediction of the accumulation signal of 9 percent packet loss network using ARX [14 15 1] and the corresponding normalized prediction error.

Performance of the MPC based flow control schemes depend on the accuracy of predictors designed. If the predictors are bad, MPC schemes will turn out to be ineffective. Therefore, it is necessary to comment on the accuracy of the two ARX predictors.

The performance of both the SSPs seem to be good when judged in terms of the value of MSE. However, this does not provide the entire picture about the quality of the SSP prediction. If we look at the normalized error plots of both the predictors, the errors in the prediction of the accumulation signal for the entire duration seems

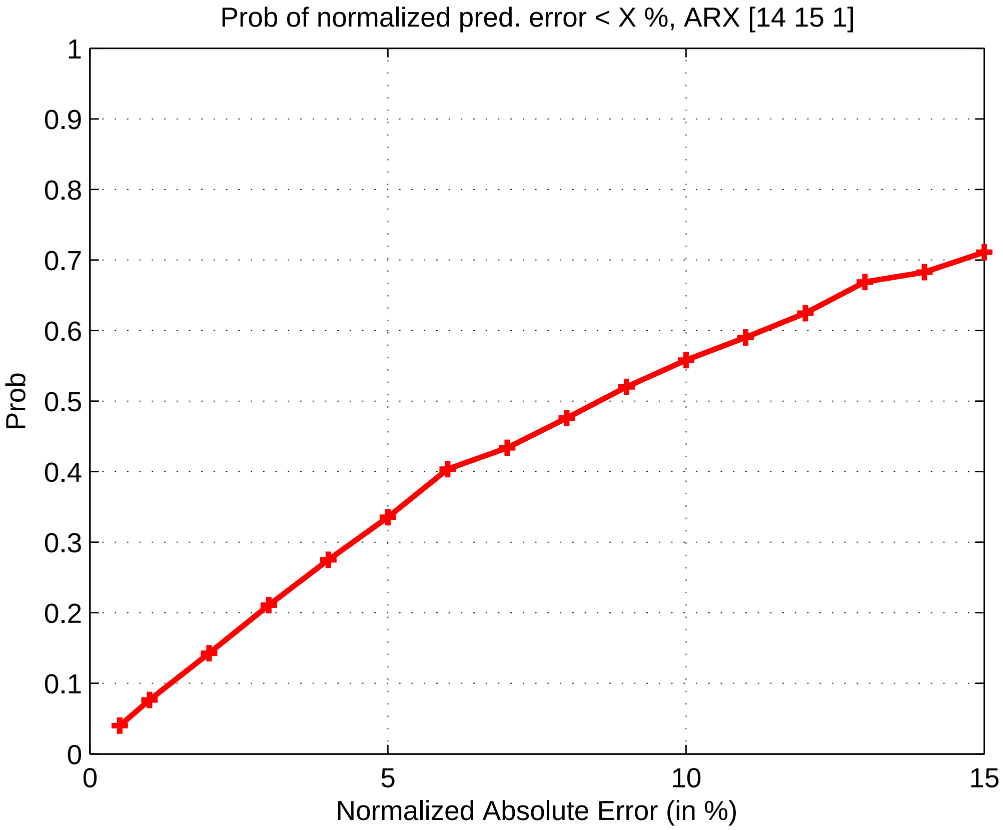


Fig. 33. Performance metric PNE10 for the predictor ARX [14 15 1] while predicting the accumulation signal from 3 percent packet loss network topology.

to be substantial at some instants. This is also indicated in the high values of MAEs for both the predictors in Table XII. As observed from both the figures 33 and 33, the probability that the normalized absolute error is less than 10% is less than 60%. In other words, the chance of SSP prediction errors of accumulation signal to be quite large (> 10%) is greater than 40%. The choice of the predictor does not matter. The nonlinear predictors do not provide better prediction results.

5. Simple Predictor

The two ARX predictors, described in the previous subsection, model the network topology better than any other nonlinear and linear models. These two ARX predictors will be used to develop predictive flow control frameworks for real-time multimedia applications explained in the next section of this chapter.

A very simple p step ahead predictor that can be used to design control schemes is defined by the following equation

$$\widehat{A}(k+p|k) = A(k), \tag{6.52}$$

where $p \in \mathbb{N}$. Equation 6.52 states that the value of the accumulation signal p step ahead in the future is equal to the current known value of the signal. Control strategies based on p step ahead simple predictor are essentially reactive control strategies.

Figure 34 shows the SSP, $\widehat{A}(k+1|k) = A(k)$, of the accumulation signal obtained from the 3% CLR network topology. The MSE of the prediction is 11.66%. This prediction is definitely poor as compared to the SSP obtained by using the two ARX models described in the previous section. Two control laws - one linear and the other nonlinear will be developed based on the simple predictor. The flow control results from these two control laws will be used to compare with the results from the control laws developed using the two ARX models.

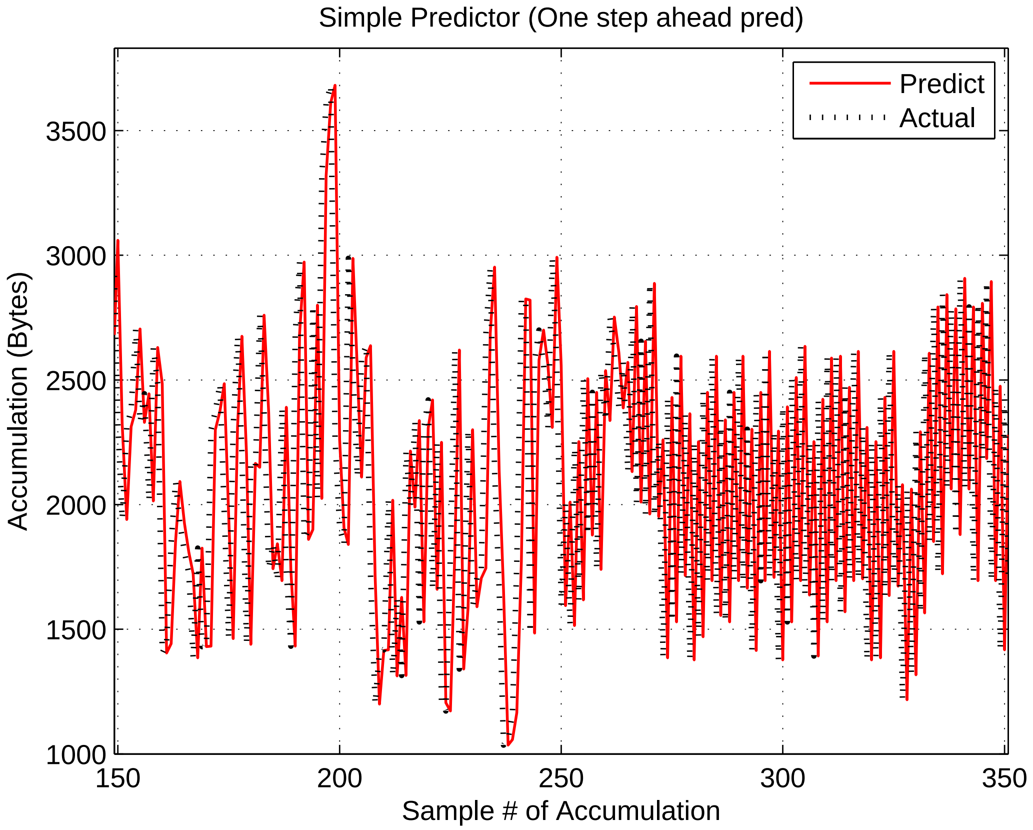


Fig. 34. Single step ahead prediction of the accumulation signal of 3% packet loss network using the simple predictor.

C. Chapter Summary

This chapter provided details about the main contribution of this research in the field of flow control of real-time multimedia flows. First part of the chapter discusses the modeling of the network using linear and nonlinear system identification techniques. Modeling of a system using system identification techniques involves the selection of the appropriate input and output signals to develop the mathematical model. The advantages of the accumulation signal over the packet loss and end-to-end delay signal in order to model best-effort IP networks is narrated in the next few subsections. The nonlinear system identification techniques did not provide any better model of the communication networks as compared to the linear models. Two linear models (ARX predictors) are chosen as the models to develop MPC control strategies because of their simplicity as compared to the nonlinear models.

CHAPTER VII

DEVELOPMENT AND VALIDATION OF THE PROPOSED FLOW CONTROL STRATEGIES

The first part of the chapter discusses the formulation of the flow control strategies. Before discussing the concepts of MPC and how to use this concept to formulate predictive control strategies, it is convenient to discuss two simple control laws based on the Simple Predictor (SP), the Linear Control Law (LCL) and the Nonlinear Control Law (NCL).

Figure 35 shows the schematic representation of a general control system. The control law utilizes the error signal calculated by subtracting the measured output $y(k)$ from the reference signal $r(k)$. The control law is the function of the error signal $e(k)$.

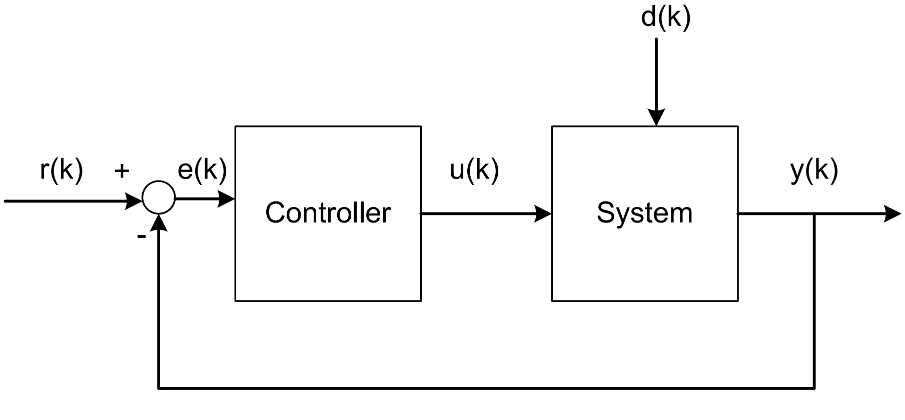


Fig. 35. Schematic of a system controlled with the help of a controller. The controller adjusts the input signal $u(k)$ based on the error signal $e(k) = r(k) - y(k)$.

In the current context, accumulation $A(k)$ is the output signal i.e. $y(k) = A(k)$, and the size of the packets in the real-time multimedia flows is the input signal $u(k)$.

The disturbance $d(k)$ comprises of the cross flow traffic in the best effort network that affects the QoS of the flow under observation. It is very difficult to model the disturbance $d(k)$ because of the distributed nature of the Internet.

If the control system has the structure as shown in the figure 35, then a reference accumulation $r(k)$ needs to be determined. This is the most difficult aspect of the problem at hand. If the reference accumulation selected is very small, then the controller will always act conservatively and the controlled real-time multimedia flow can never take advantage of the extra bandwidth available when the network is not congested. If the reference accumulation selected is very large, then the controller will never predict the congestion present in the network and take appropriate actions.

Assuming that the inter-departure time of the packets is fixed at 20 ms, the reference accumulation is dependent on the selection of the packet size of the multimedia flow. For example, if the maximum end-to-end delay that can be tolerated while transferring multimedia information over the best effort network is 230 ms and the bit-rate of the flow is 36 kbps, then the accumulation level signifying a normal network is $\frac{36000 \times 0.23}{8} = 1035$ bytes. The control action needs to be taken only if the accumulation level goes above or below 1035 bytes. If the bit-rate of the flow changes to 96 kbps, then the accumulation level signifying a normal network is $\frac{36000 \times 0.23}{8} = 2760$ bytes. This makes the mathematical formulation of the control strategy difficult for both design and implementation.

In order to overcome the difficulties of designing simple flow control strategies based on the error signal as prescribed by the conventional control system design framework, an alternative framework is used. This is shown in the figure 36.

$$u(k) \quad = \quad f(y(k)) \tag{7.1}$$

Equation 7.1 describes the new control law. Both the LCL as well as the NCL, described in the next two subsections, are based on this framework that does not have any reference signal and the control law is a function of the measurement of the output signal.

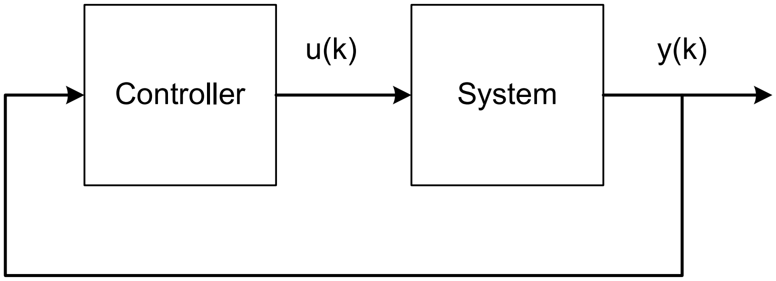


Fig. 36. Schematic of a single flow in best-effort network modeled as system controlled with the help of a controller employing either LCL or NCL. The controller adjusts the input signal $u(k)$ based on the output signal $y(k)$.

1. Linear Control Law with the Simple Predictor

The linear control law has been designed using the following logic: if the accumulation in the network goes up, then the bit-rate of the controlled real-time multimedia flow should go down and vice versa. The control law is not really linear in the right sense of the word. It incorporates saturation of the actuator, i.e. the inability of the codec to generate bit-rates beyond a certain level. The law is linear in a certain specific range of accumulation. In spite of this, the simple control law will be referred to as the "Linear Control Law" (LCL) in this research work.

As discussed earlier, the codec considered in this research has six different bit-rates, varying from 36 kbps to 96 kbps. The end-to-end delay of the VoIP application that has acceptable level of interactivity between the two users at the ends is considered to be 230 ms. The inter-departure time of the packets is 20 ms. In an ideal case, the maximum amount of accumulation, measured in bytes, that a flow can afford to have and preserve interactivity is

$$A_s = \frac{p_s \times 230}{20}, \tag{7.2}$$

where p_s is the size of the packets in the VoIP flow.

This means that the maximum accumulation that can be tolerated by the application is equivalent to the total number of packets in transportation when the end-to-end delay of the packets is equal to 230 ms. Equation 7.2 generates six different levels of accumulation for six different bit-rates, these are shown in Table XIII.

Table XIII. Equilibrium accumulation corresponding to each of the six bit rate levels.

	Packet Size (Bytes)	Bit-rate (kbps)	Accumulation (Bytes)
1	90	36	1035
2	120	48	1380
3	150	60	1725
4	180	72	2070
5	210	84	2415
6	240	96	2760

Let us assume that at time instant kT, the accumulation signal value $A\left((k-1)T\right)$ is known because of the feedback delay from the receiver to the sender. This measurement, $A\left((k-1)T\right)$, is utilized to predict the value of accumulation at time instant $(k+1)T$, using a two-step-ahead simple predictor. The two-step-ahead predictor is described by the following equation:

$$\widehat{A}(k+1|k-1) = A(k-1). \tag{7.3}$$

The LCL is described by

$$u(k+1) = \begin{cases} 90 & \text{, if } \widehat{A}(k+1|k-1) > 2760 \\ -0.087\widehat{A}(k+1|k-1) + 330 & \text{, if } 1035 \leq \widehat{A}(k+1|k-1) \leq 2760, \\ 240 & \text{, if } \widehat{A}(k+1|k-1) < 1035 \end{cases} \tag{7.4}$$

where $u(k+1)$ is the size of the packets that need to be sent every 20 ms.

The LCL and its implementation in the quantized form is shown together in figure 37. The quantized implementation of the LCL can be expressed as

$$u(k+1) = \begin{cases} 90 & \text{, if } \widehat{A}(k+1|k-1) \geq 2415 \\ 120 & \text{, if } 2070 \geq \widehat{A}(k+1|k-1) < 2415 \\ 150 & \text{, if } 1725 \geq \widehat{A}(k+1|k-1) < 2070 \\ 180 & \text{, if } 1380 \geq \widehat{A}(k+1|k-1) < 1725 \\ 210 & \text{, if } 1035 \geq \widehat{A}(k+1|k-1) < 1380 \\ 240 & \text{, if } \widehat{A}(k+1|k-1) < 1035 \end{cases} \tag{7.5}$$

The quantization is necessary as any realistic codec can not generate the bit-rates specified by equation 7.4.

2. Nonlinear Control Law with the Simple Predictor

A slight variation to LCL is used to design another simple flow control law. The LCL is designed on the basis of the concept that higher the accumulation signal, lower the bit-rate that needs to be sent into the network. If the network is congested the need for congestion avoidance is far more urgent. Therefore, the drop in packet sizes dur-

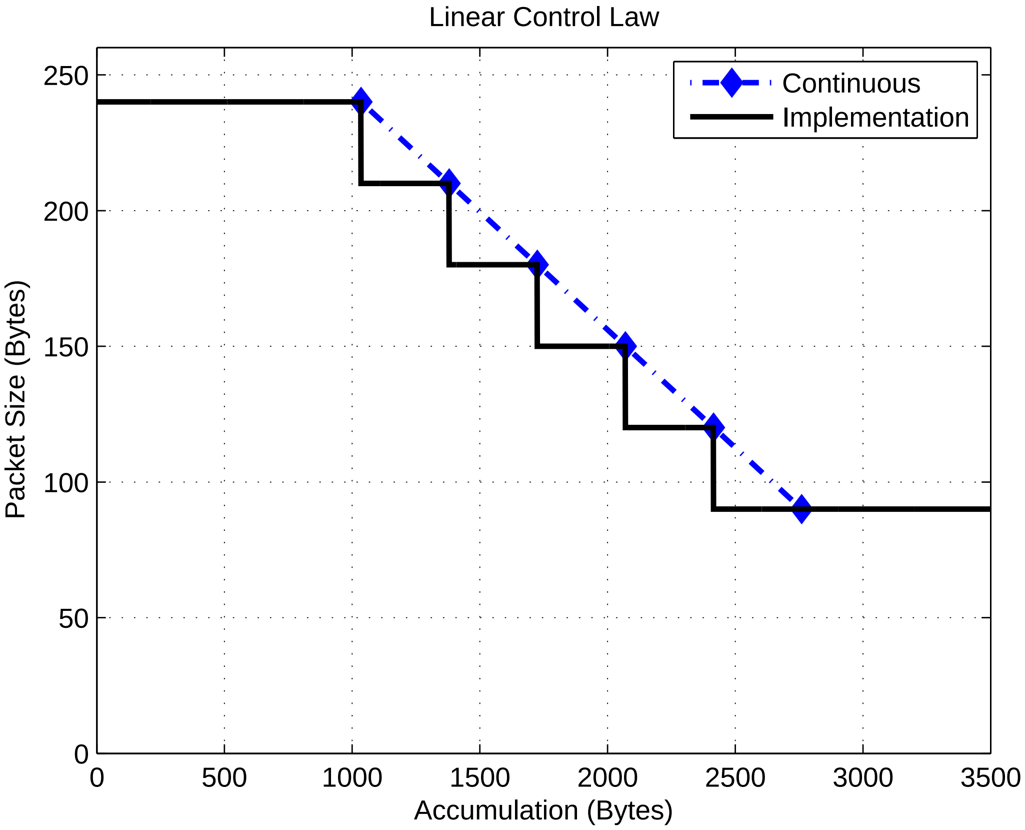


Fig. 37. Linear control law.

ing congestion should happen at a faster rate as compared to the increase in packet sizes when the network is not congested. This is inspired from the Additive Increase Multiplicative Decrease ($AIMD$) control employed by the TCP for congestion avoidance. TCP's AIMD mechanism of increasing the congestion window by roughly one segment per round-trip time in the absence of congestion, and halving the congestion window in response to a round-trip time with a congestion event ensures stability.

The LCL based on accumulation does not take care of the above mentioned concern. Therefore, a separate control law is designed. This control law operates nonlinearly in the specified range of accumulation. The quantized version of this law is employed for flow control. The nonlinear flow control law is described by the following equation:

$$u(k{+}1) = \begin{cases} 90 & \text{, if } \widehat{A}(k+1|k-1) > 2760 \\ \left(65700 - \frac{65700}{8688825}\widehat{A}(k+1|k-1)^2\right)^{\frac{1}{2}} & \text{, if } 1035 \leq \widehat{A}(k+1|k-1) \leq 2760. \\ 240 & \text{, if } \widehat{A}(k+1|k-1) < 1035 \end{cases} \tag{7.6}$$

The NCL and its implementation in the quantized form is shown together in figure 38.

The quantized implementation of the NCL can be described by equation

$$u(k+1) = \begin{cases} 90 & \text{, if } \widehat{A}(k+1|k-1) \geq 2415 \\ 144 & \text{, if } 2070 \geq \widehat{A}(k+1|k-1) < 2415 \\ 184 & \text{, if } 1725 \geq \widehat{A}(k+1|k-1) < 2070. \\ 208 & \text{, if } 1380 \geq \widehat{A}(k+1|k-1) < 1725 \\ 224 & \text{, if } 1035 \geq \widehat{A}(k+1|k-1) < 1380 \\ 240 & \text{, if } \widehat{A}(k+1|k-1) < 1035 \end{cases} \tag{7.7}$$

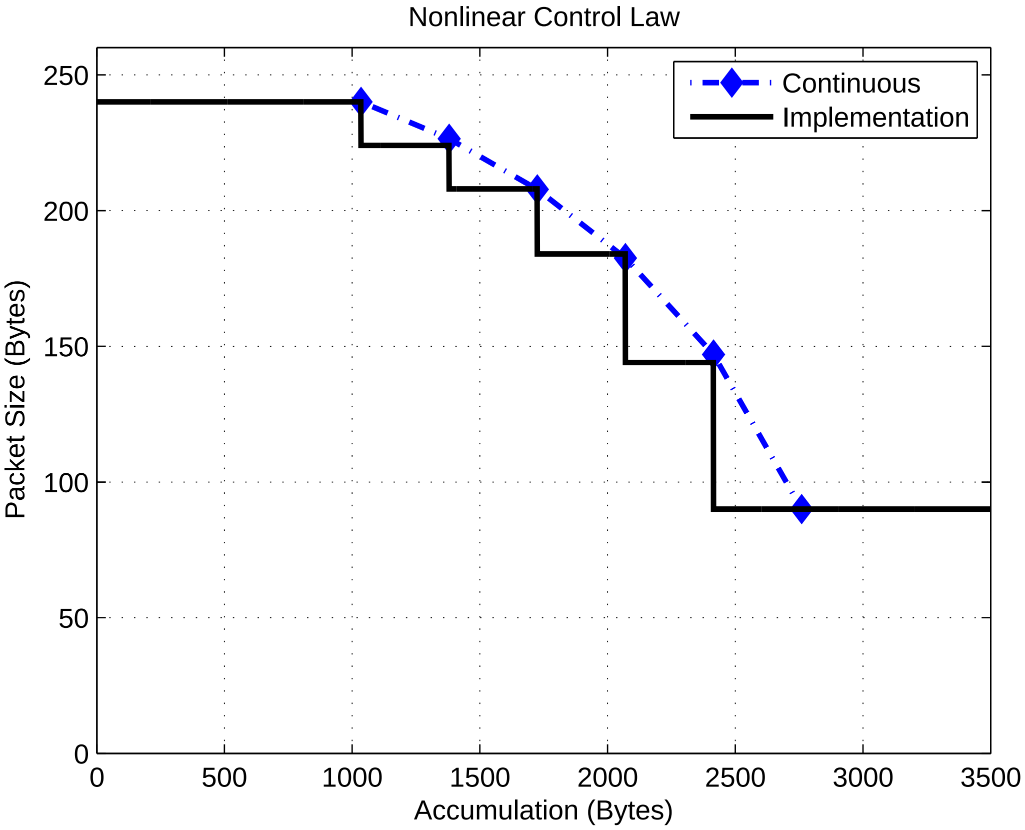


Fig. 38. Nonlinear control law.

3. The Model Predictive Control Law

The accumulation signal is directly correlated with the congestion level in the network. The control strategy that will be employed by the adaptive end-to-end flow control law is called regulation. Systems designed to hold an output steady against unknown disturbances are called regulators. The aim of the adaptive flow controller is to measure the state of the accumulation and change the bit-rate so that it can maintain the accumulation signal measured between the sender and the receiver to a desired reference level, irrespective of the congestion in the network.

MPC makes explicit use of models of the processes to calculate the control signal by minimizing an objective function. MPC helps in designing linear controllers. MPC is a strategy that can be summarized by the following points:

1. MPC uses a model of the process in order to predict the future output from the system.

2. MPC calculates the control signal by minimizing an objective function.

3. MPC uses a receding horizon strategy. In the receding horizon strategy, a series of future control inputs are calculated by predicting future outputs using the system model and minimizing a specified objective function. However, only the first control input of the calculated sequence is applied at the next time instant.

The MPC strategy [84] can be explained more clearly with the help of figure 39. MPC is implemented in three steps:

- Step 1: The future outputs for a certain prediction horizon N are predicted at instant k using the process model. The prediction horizon is 3 in figure 39. The predicted outputs $\{\hat{y}(k+n|k)\}$ for $n = 1, 2, 3$ depend on the known values of past inputs and outputs up to instant k, as well as on the future control inputs $\{u(k+n|k)\}$ for $n = 1, 2$.

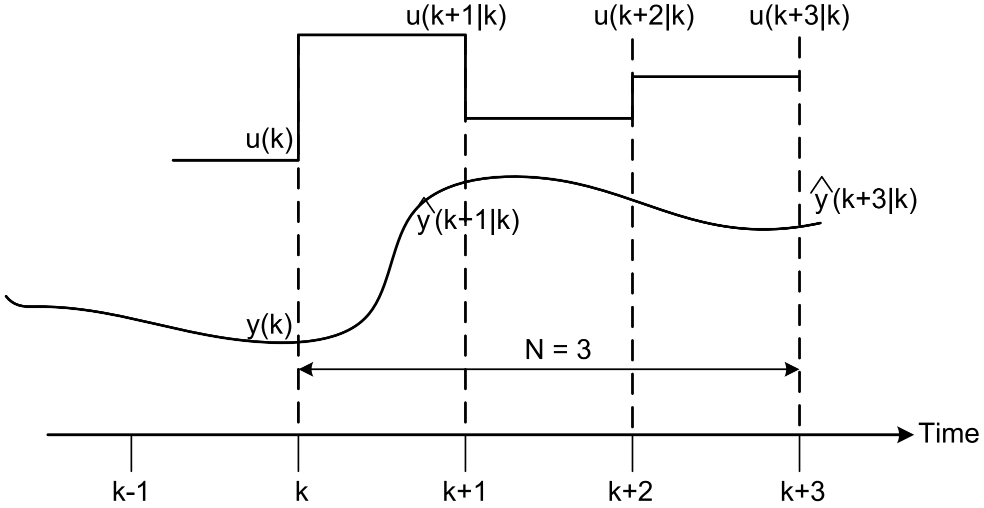


Fig. 39. Schematic of the MPC strategy.

- Step 2: The set of future control signals is calculated by optimizing a criterion in order to keep the process as close as possible to the reference trajectory $\{r(k+n)\}$. The objective function that represents the pre-determined criterion takes the form of a quadratic function of the errors between the predicted output signal and the reference trajectory. The control effort is included in the objective function in most cases.

- Step 3: Among all of the calculated control signals $u(k)$ is the only one that is applied to the process. All the others are rejected. At the next sampling instant step 1 and step 2 is repeated, with all the new measured values of the inputs and the outputs.

During the optimization process of step 2, if the cost function is quadratic, the minimum of the function can be obtained as an explicit function of past inputs, outputs, and the future reference trajectory. However, if constraints are imposed on the control inputs, the solution to the optimization problem is obtained by computationally intensive numerical algorithms. The size of the optimization problems depend on the number of variables and the prediction horizon used. The computational time needed for the calculation of the control inputs for constrained and robust cases can be quite large as compared to unconstrained cases.

4. Formulation of the Model Predictive Control

Let the linear difference equation describing the end-to-end single flow be

$$y(k) = \alpha_1 y(k-1) + \alpha_2 y(k-2) + \ldots + \alpha_n y(k-n) + b_1 u(k-1) + \ldots + b_m u(k-m) + \frac{d(k)}{\Delta}, \tag{7.8}$$

where $y(k)$ represents the output of the system (the accumulation), $u(k)$ represents the input to the system, $d(k)$ be the disturbance, and $\Delta = 1 - q^{-1}$. The term $\frac{d(k)}{\Delta}$ represents an integrated white noise term. This is an accepted model for disturbances as it allows non-zero mean with random changes.

The input signal to the system is the sizes of the packets sent across the network between two time instants. Similarly, the output signal of the system is the measured accumulation at a time instant. Although accumulation in the earlier sections of the dissertation has been denoted by the notation $A(k)$, yet $y(k)$ has been used in this section to represent accumulation. There are two reasons for this. The derivation of the MPC strategy contains many signals and parameters that can be denoted by A. In conventional control systems design terminology, $y(k)$ is used to represent the output of the system.

Equation 7.8 can be simplified by multiplying both sides with the operator $\Delta = 1 - q^{-1}$. We get

$$\begin{aligned} y(k) &= \alpha'_1 y(k-1) + \alpha'_2 y(k-2) \ldots + \alpha'_n y(k-n) + \alpha'_{n+1} y(k-n-1) + \\ &\quad b_1 \Delta u(k-1) + b_2 \Delta u(k-2) \ldots + b_m \Delta u(k-m) + d(k), \end{aligned} \tag{7.9}$$

where $\alpha'_1, \ldots, \alpha'_{n+1}$ are the new coefficients for the auto-regressive part of the system.

The predictor form of 7.9 is:

$$\begin{aligned} \widehat{y}(k|k-1) &= \alpha'_1 y(k-1) + \alpha'_2 y(k-2) \ldots + \alpha'_n y(k-n) + \alpha'_{n+1} y(k-n-1) + \\ &\quad b_1 \Delta u(k-1) + b_2 \Delta u(k-2) \ldots + b_m \Delta u(k-m). \end{aligned} \tag{7.10}$$

Equation 7.10 can also be written as:

$$\begin{aligned} &\widehat{y}(k|k-1) + a_1 y(k-1) + a_2 y(k-2) \ldots + a_n y(k-n) + a_{n+1} y(k-n-1) = \\ &\qquad b_1 \Delta u(k-1) + b_2 \Delta u(k-2) + \ldots + b_m \Delta u(k-m), \end{aligned} \tag{7.11}$$

where $a_i = -\alpha'_i$. This step is performed for simplicity.

The number of steps of prediction needed to formulate MPC is dependent on the nature of the problem. If the model of the system is accurate, more steps of prediction are used to optimize dynamic performance. However, if the model of the system is inaccurate, there is no advantage in using many step ahead predictions. For the current work, multi-step-ahead prediction from the designed SSPs deteriorate sharply with increase in number of steps of prediction. Four steps of prediction, i.e., 960 ms, is suitable to formulate MPC flow control. Therefore, the equations for the next three steps of the prediction are:

$$\begin{aligned} &\widehat{y}(k+1|k-1) + a_1 \widehat{y}(k|k-1) + a_2 y(k-1) \ldots + a_n y(k-n+1) + a_{n+1} y(k-n) = \\ &\qquad b_1 \Delta u(k) + b_2 \Delta u(k-1) + \ldots + b_m \Delta u(k-m+1), \end{aligned} \tag{7.12}$$

$$\widehat{y}(k+2|k-1) + a_1\widehat{y}(k+1|k-1) + a_2 y(k|k-1)\ldots + a_{n+1}y(k-n+1) = \\ b_1\Delta u(k+1) + b_2\Delta u(k) + \ldots + b_m\Delta u(k-m+2), \tag{7.13}$$

$$\widehat{y}(k+3|k-1) + a_1\widehat{y}(k+2|k-1) + a_2 y(k+1|k-1)\ldots + a_{n+1}y(k-n+2) = \\ b_1\Delta u(k+2) + b_2\Delta u(k+1) + \ldots + b_m\Delta u(k-m+3). \tag{7.14}$$

The above equations can be written in the following matrix form

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ a_1 & 1 & 0 & 0 \\ a_2 & a_1 & 0 & 0 \\ a_3 & a_2 & a_1 & 1 \end{pmatrix}}_{C_A} \underbrace{\begin{pmatrix} \widehat{y}(k|k-1) \\ \widehat{y}(k+1|k-1) \\ \widehat{y}(k+2|k-1) \\ \widehat{y}(k+3|k-1) \end{pmatrix}}_{\underline{\widehat{y}}} + \underbrace{\begin{pmatrix} a_1\, a_2\, a_3 \ldots & a_n & a_{n+1} \\ a_2\, a_3\, a_4 \ldots & a_{n+1} & 0 \\ a_3\, a_4\, a_5 \ldots & 0 & 0 \\ a_4\, a_5\, a_6 \ldots & 0 & 0 \end{pmatrix}}_{H_A} \underbrace{\begin{pmatrix} y(k-1) \\ y(k-2) \\ y(k-3) \\ \ldots \\ y(k-n+1) \\ y(k-n) \\ y(k-n-1) \end{pmatrix}}_{\underline{y}}$$

$$= \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ b_1 & 0 \\ b_2 & b_1 \end{pmatrix}}_{C_B} \underbrace{\begin{pmatrix} \Delta u(k+1) \\ \Delta u(k+2) \end{pmatrix}}_{\Delta\underline{u_F}} + \underbrace{\begin{pmatrix} 0\, b_1\, b_2 \ldots & b_{m-1} & b_m \\ b_1\, b_2\, b_3 \ldots & b_m & 0 \\ b_2\, b_3\, b_4 \ldots & 0 & 0 \\ b_3\, b_4\, b_5 \ldots & 0 & 0 \end{pmatrix}}_{H_B} \underbrace{\begin{pmatrix} \Delta u(k) \\ \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-m+1) \\ \Delta u(k-m) \end{pmatrix}}_{\Delta\underline{u}} \tag{7.15}$$

$\Delta\underline{u_F}$ is the vector of the unknown increments in the future control inputs. The solution for this unknown vector is derived by optimizing a suitable objective function. $\Delta\underline{u}$ is the vector of the known increments of the past control inputs. It should be noted that we already know the value of $\Delta u(k)$ because of the control action performed in the previous sampling instant. Hence, $\Delta u(k)$ has been included in the vector that represents known values of the inputs to the system. The shorter version of equation 7.15 is

$$\begin{aligned} C_A\underline{\widehat{y}} &= -H_A\underline{y} + C_B\Delta\underline{u}_F + H_B\Delta\underline{u}, \\ \Rightarrow \underline{\widehat{y}} &= -C_A^{-1}H_A\underline{y} + C_A^{-1}C_B\Delta\underline{u}_F + C_A^{-1}H_B\Delta\underline{u}. \end{aligned} \tag{7.16}$$

Let $H = C_A^{-1}C_B$, $P = C_A^{-1}H_B$, and $Q = -C_A^{-1}H_A$. Therefore, the predicted future values of $\widehat{y}(k|k-1)$, $\widehat{y}(k+1|k-1)$, $\widehat{y}(k+2|k-1)$, $\widehat{y}(k+3|k-1)$ are provided by

$$\underline{\widehat{y}} = H\Delta\underline{u}_F + P\Delta\underline{u} + Q\underline{y}. \tag{7.17}$$

The MPC control strategy is devised by minimizing the objective function $\mathbf{J}$ defined by the following expression w.r.t. the future control inputs Δu_F:

$$\begin{aligned} \min_{\Delta\underline{u}_F} \mathbf{J} &= \|\underline{\mathbf{e}}\|_2^2 + \lambda\|\Delta\underline{u}_F\|_2^2, \\ \Rightarrow \min_{\Delta\underline{u}_F} \mathbf{J} &= \|\underline{r} - \underline{\widehat{y}}\|_2^2 + \lambda\|\Delta\underline{u}_F\|_2^2, \end{aligned} \tag{7.18}$$

where $\underline{r}$ represents the value of the reference signal in the next few sampling instants. After optimizing $\mathbf{J}$ and getting the values of $\Delta\underline{u}_F$, only the first element, that is $\Delta u(k)$ is implemented as the optimization is updated at the next sampling instant. This is termed as *receding horizon* in predictive control schemes. In practical sense, it means that the horizon is moving away constantly at the same speed at which the observations about the system under consideration are moving through time.

Equation 7.18 can be written as

$$\min_{\Delta\underline{u}_F} \mathbf{J} = \|\underline{r} - H\Delta\underline{u}_F - P\Delta\underline{u} - Q\underline{y}\|_2^2 + \lambda\|\Delta\underline{u}_F\|_2^2, \tag{7.19}$$

After substituting the value of $\widehat{\underline{y}}$ from equation 7.17 in equation 7.18. Let us now expand the RHS of equation 7.19 and rewrite the expression for the objective function $\mathbf{J}$ as

$$\begin{aligned}
\mathbf{J} &= (\underline{r} - H\Delta\underline{u}_F - P\Delta\underline{u} - Q\underline{y})^T(\underline{r} - H\Delta\underline{u}_F - P\Delta\underline{u} - Q\underline{y}) \\
&\quad + \lambda\Delta\underline{u}_F{}^T\Delta\underline{u}_F, \\
\Rightarrow \mathbf{J} &= (\underline{r}^T - (H\Delta\underline{u}_F)^T - (P\Delta\underline{u})^T - (Q\underline{y})^T)(\underline{r} - H\Delta\underline{u}_F - P\Delta\underline{u} - Q\underline{y}) \\
&\quad + \lambda\Delta\underline{u}_F{}^T\Delta\underline{u}_F.
\end{aligned} \tag{7.20}$$

In order to minimize $\mathbf{J}$ w.r.t. $\Delta\underline{u}_F$, we need to differentiate the expression for $\mathbf{J}$ with $\Delta\underline{u}_F$ and equate the resultant expression to zero. The terms of the equation not containing $\Delta\underline{u}_F$ will become zero during the differentiation process. Therefore,

$$\begin{aligned}
\mathbf{J} &= (H\Delta\underline{u}_F)^T(H\Delta\underline{u}_F) - \underline{r}^T(H\Delta\underline{u}_F) + (P\Delta\underline{u})^T(H\Delta\underline{u}_F) + (Q\underline{y})^T H\Delta\underline{u}_F \\
&\quad -(H\Delta\underline{u}_F)^T\underline{r} + (H\Delta\underline{u}_F)^T(P\Delta\underline{u}) + (H\Delta\underline{u}_F)^T(Q\underline{y}) + \lambda\Delta\underline{u}_F{}^T\Delta\underline{u}_F \\
&\quad +\mathcal{X}, \\
&= (\Delta\underline{u}_F^T H^T H\Delta\underline{u}_F) - (\underline{r}^T H\Delta\underline{u}_F) + (\Delta\underline{u}^T P^T H\Delta\underline{u}_F) + (\underline{y}^T Q^T H\Delta\underline{u}_F) \\
&\quad -(\Delta\underline{u}_F^T H^T\underline{r}) + (\Delta\underline{u}_F^T H^T P\Delta\underline{u}) + (\Delta\underline{u}_F^T H^T Q\underline{y}) + \lambda\Delta\underline{u}_F{}^T\Delta\underline{u}_F + \mathcal{X}, \\
&= -2(\Delta\underline{u}_F^T H^T\underline{r}) + 2(\Delta\underline{u}_F^T H^T P\Delta\underline{u}) + 2(\Delta\underline{u}_F^T H^T Q\underline{y}) + (\Delta\underline{u}_F^T H^T H\Delta\underline{u}_F) \\
&\quad +\lambda\Delta\underline{u}_F{}^T\Delta\underline{u}_F + \mathcal{X}, \\
\Rightarrow \mathbf{J} &= 2\Delta\underline{u}_F^T(-H^T\underline{r} + H^T P\Delta\underline{u} + H^T Q\underline{y}) + \Delta\underline{u}_F^T(H^T H + \lambda I)\Delta\underline{u}_F + \mathcal{X}.
\end{aligned} \tag{7.21}$$

where $\mathcal{X}$ is the sum of terms that do not contain $\Delta\underline{u}_F$. The terms in $\mathcal{X}$ will reduce to 0 when $\mathbf{J}$ is differentiated w.r.t. $\Delta\underline{u}_F$. The next step is to differentiate equation 7.21 w.r.t.$\Delta\underline{u}_F$ and equate it to 0. This results in

$$\begin{aligned}(-H^T\underline{r} + H^T P\Delta\underline{u} + H^T Q\underline{y}) + (H^T H + \lambda I)\Delta\underline{u}_F &= 0,\\ H^T(-\underline{r} + P\Delta\underline{u} + Q\underline{y}) + (H^T H + \lambda I)\Delta\underline{u}_F &= 0.\end{aligned} \tag{7.22}$$

From equation 7.22, we can find the final control law is:

$$\Delta\underline{u}_F = (H^T H + \lambda I)^{-1} H^T(\underline{r} - P\Delta\underline{u} - Q\underline{y}). \tag{7.23}$$

$\Delta\underline{u}_F$ contains two elements, $\Delta u(k+1) = u(k+1) - u(k)$, and $\Delta u(k+2) = u(k+2) - u(k+1)$. Only the first element in the vector containing the future inputs, $\Delta u(k+1)$, is implemented online. This implies that

$$u(k+1) = \Delta u(k+1) + u(k). \tag{7.24}$$

5. Development of Different Model Predictive Controllers

Two different MPC schemes are developed using the formulation shown in the subsection 4. The first scheme uses ARX [15 14 1] as the linear model representing the system flow in the best effort network. The second MPC scheme uses ARX [14 15 1] to model the network.

There are two parameters that need to be decided upon by trial and error before the MPC scheme based on linear prediction model can be finalized. Both these parameters are present in the final control law defined by equation 7.23. The first parameter is λ. This parameter is responsible for penalizing the change of input (bit-rate) in the objective function defined by equation 7.18. Higher the λ, the more costlier is the input change and vice versa.

The second parameter that needs to be decided upon is the vector $\underline{r}$. The current problem is a regulatory problem where the reference is kept constant at a certain level. The future control inputs are decided based on how far the current output is from the reference signal. The question that needs to be answered is what is the level of accumulation in the network that can be used as a reference signal. If the accumulation level chosen for the reference is too high, the network remains congested all the time because the control law can never distinguish between a congested system or a normal system based on the feedback. Similarly, if the accumulation level is set to be too low, the control law becomes too conservative and the lower bit-rates dominate during a session.

A series of experiments are performed to determine suitable values of λ and the vector $\underline{r}$ for each of the two MPC schemes based on two different predictors. The first set of experiments are conducted by fixing λ at 0.1 and varying the reference accumulation from 1725 bytes to 3105 bytes. After determining the accumulation that showed the best results in terms of the MOS, the parameter λ was varied from 0.1 to 10. For the MPC scheme designed with the help of ARX [15 14 1], the constant reference signal for accumulation within the network was chosen to be 2245 bytes. The λ was determined to be 0.1. The accumulation reference for the MPC scheme designed using ARX [14 15 1] did not change. However, the parameter, λ changed from 0.1 to 10 for the second MPC scheme. Table XIV provides a summary of the different parameters used for designing the two MPC strategies, suitably named MPC3 and MPC9.

The theoretical MPC control problem formulation has been done by assuming that the actuator, i.e. the codec generating information, possesses an unlimited range. However, this is not true in practice. All actuators have a limited range of action.

Table XIV. Parameters used to design the two MPC schemes.

	Control Scheme	Predictor Used	Reference Accum. (Bytes)	λ
1	MPC3	ARX [15 14 1]	2245	0.1
2	MPC9	ARX [14 15 1]	2245	10

The MPC based control system designed without taking care of the constraints of the actuator violates constraints frequently during the course of the operation. However, the input control signal can always be kept within the bounds imposed by the constraints of the actuator by clipping the control action to a value or a set of values.

In the current research, the control input generated by the MPC based controllers, MPC3 and MPC9, is quantized to six fixed bit-rates according to the following equation:

$$u(k+1) = \begin{cases} 90\ \ , \text{if } u(k+1) < 105 \\ 120\,, \text{if } 105 \geq u(k+1) < 135 \\ 150\,, \text{if } 135 \geq u(k+1) < 165 \\ 180\,, \text{if } 165 \geq u(k+1) < 195 \\ 210\,, \text{if } 195 \geq u(k+1) < 225 \\ 240\,, \text{if } u(k+1) \geq 225 \end{cases} \tag{7.25}$$

It can be noticed from the equation 7.25 that any control input generated by the MPC controller less than 105 bytes or greater than 225 bytes is saturated to 90 bytes and and 240 bytes, respectively. The controller has six levels of bit-rates. The quantization of the control input is uniform in the range of 90 bytes to 240 bytes.

The MPC controllers have not been designed by taking the constraints of the actuator into account. Therefore, they are not optimal. If the constraints are not included in the problem formulation, the objective function is not minimized theoretically and will lead to a poorer performance of the control system. Explicit handling of the constraints allow the process to operate closer to the optimal operating conditions. However, the implementation of the Generalized Predictive Controllers (GPC) for processes with bounded input and output signals requires the solution of a Quadratic Programming (QP) problem, an optimization problem with a quadratic objective function and linear constraints. The solution for these kind of problems depends on algorithms that take too much computing power and time to execute in real-time. Therefore, an end-to-end flow control scheme implemented at the application layer of the OSI, based on a QP problem solution at each time step, might be very difficult to implement in practice.

A. Implementation of the Flow Controllers

A separate module has been developed in C++ using the libraries and the interfaces provided by ns-2 in order to conduct the simulation experiments to validate the ideas presented in the research. The module is a realistic representation of the real-time multimedia applications transmitting information over best-effort networks. The developed application is attached to the UDP nodes. At least one node running the application is needed at both the ends of the network in order to establish a full-duplex multimedia flow through the network. A single instance of the application on a node has both the sender as well as the receiver. The sender sends the packets at the specified rate. The receiver receives the packets from the other end. The "Acknowledgements" of the packets received at the other side of the network are sent by piggybacking the "ACKS" with the information sent by the sender of the application at the other end. The feedback information used for calculating the accumulation signal at a specified sampling rate is sent using the “ACKS”. Figure 40 shows the flow chart that provides the schematic representation of the top level design of the algorithms of the sender of the real-time multimedia application. There are three clocks that handle the execution of a set of events. These three clocks control a set of events that are oriented towards accomplishing a particular task. Each of these set of events update certain variables that are accessed globally. However, the set of events should not be confused with threads. ns-2 is a Discrete Event Simulator (*DES*). Every time a set of events happen because of the expiry of one of the clocks, the clock in the simulator does not progress further unless all the events are completed. Therefore, there is no chance that the clock governing the calculation of input control expire at the same time and accesses the variable that are also being update by the clock governing the sending of the packets in to the network. Therefore, there is no need of taking care of locking and unlocking the shared variables.

The first clock controls the inter-departure time period of the sending rate of the packets. Every 20 ms one packet of a fixed size as determined by the controller is sent into the network. As soon as the packet is sent into the network with the appropriate header, the size of the packet is added to a variable keeping track of the amount of information that has been pushed into the network. This variable is used for the calculation of the accumulation, when needed. The header attached to the packet sent to the application at the other end also contains information about the sequence number of the last packet received from the other side and the total amount of information that has been received in the current session. This information is also used to calculate the accumulation of the network at the other end.

The second clock controls the frequency with which control input for the future is calculated. In the experiments for demonstrating the efficacy of the ideas proposed in the current research, control is calculated every 240 ms based on average measured

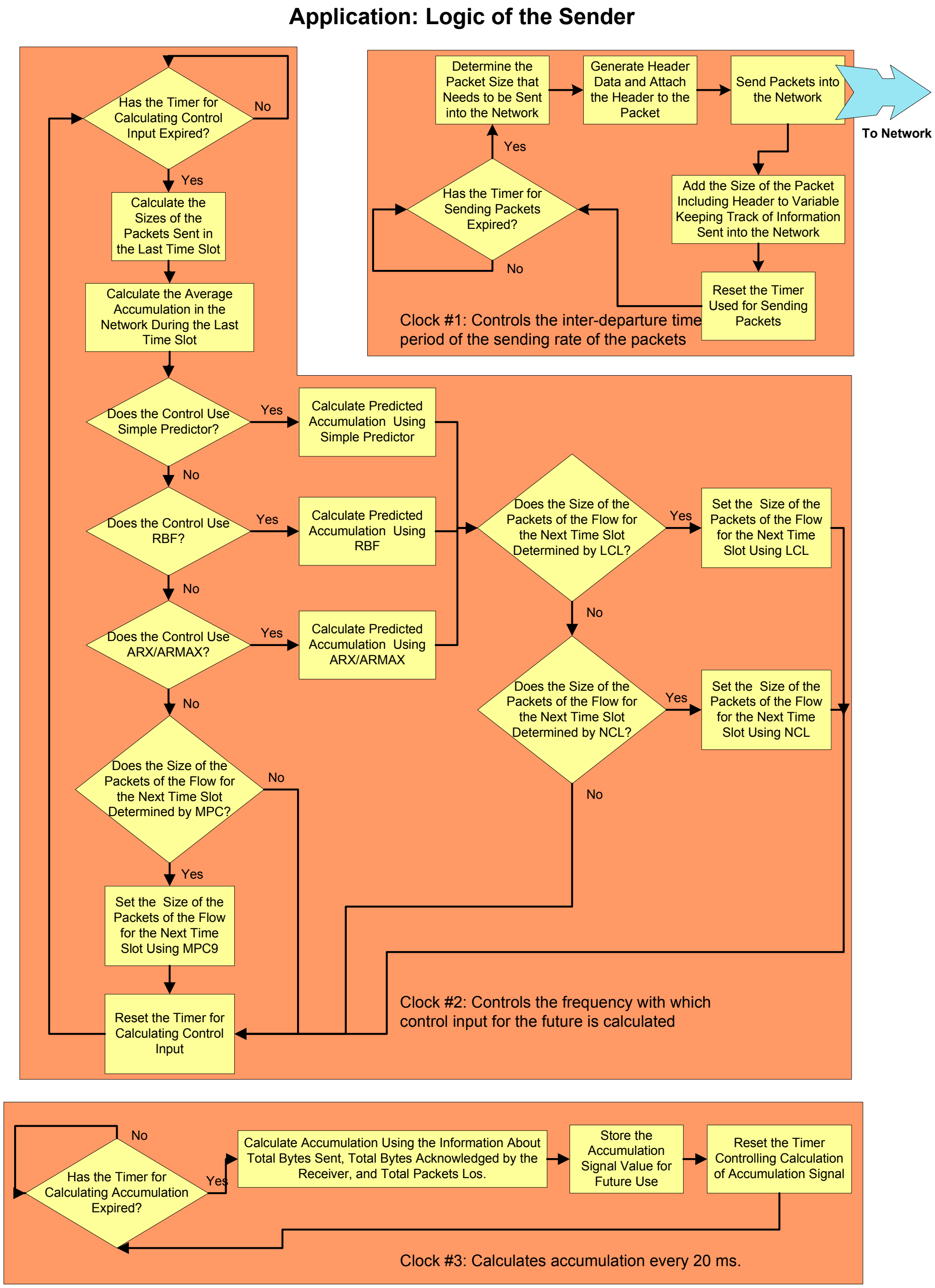

Application: Logic of the Sender
Has the Timer for Calculating Control Input Expired?
No
Yes
Calculate the Sizes of the Packets Sent in the Last Time Slot
Calculate the Average Accumulation in the Network During the Last Time Slot
Does the Control Use Simple Predictor?
Yes
Calculate Predicted Accumulation Using Simple Predictor
No
Does the Control Use RBF?
Yes
Calculate Predicted Accumulation Using RBF
No
Does the Control Use ARX/ARMAX?
Yes
Calculate Predicted Accumulation Using ARX/ARMAX
No
Does the Size of the Packets of the Flow for the Next Time Slot Determined by MPC?
No
Yes
Set the Size of the Packets of the Flow for the Next Time Slot Using MPC9
Reset the Timer for Calculating Control Input
Does the Size of the Packets of the Flow for the Next Time Slot Determined by LCL?
Yes
Set the Size of the Packets of the Flow for the Next Time Slot Using LCL
No
Does the Size of the Packets of the Flow for the Next Time Slot Determined by NCL?
Yes
Set the Size of the Packets of the Flow for the Next Time Slot Using NCL
No
Clock #2: Controls the frequency with which control input for the future is calculated
Determine the Packet Size that Needs to be Sent into the Network
Generate Header Data and Attach the Header to the Packet
Send Packets into the Network
To Network
Yes
Has the Timer for Sending Packets Expired?
No
Add the Size of the Packet Including Header to Variable Keeping Track of Information Sent into the Network
Reset the Timer Used for Sending Packets
Clock #1: Controls the inter-departure time period of the sending rate of the packets
No
Has the Timer for Calculating Accumulation Expired?
Yes
Calculate Accumulation Using the Information About Total Bytes Sent, Total Bytes Acknowledged by the Receiver, and Total Packets Los.
Store the Accumulation Signal Value for Future Use
Reset the Timer Controlling Calculation of Accumulation Signal
Clock #3: Calculates accumulation every 20 ms.

Fig. 40. Flowchart showing the schematic representation of the sender of real-time multimedia application used for ns-2 simulation experiments.

accumulation and the average bit-rate sent. As soon as this clock expires, a series of events take place that changes the bit-rate of the multimedia flow in the current session based on the type of predictor and the control scheme selected. The clock is reset after all the events.

The third clock calculates accumulation every 20 ms in all the experiments. Accumulation is calculated every 20 ms based on the amount of information that has been sent to the other side, amount of information that has been received by the other side and the acknowledgement sent back, and the amount of information that has been lost in the transit.

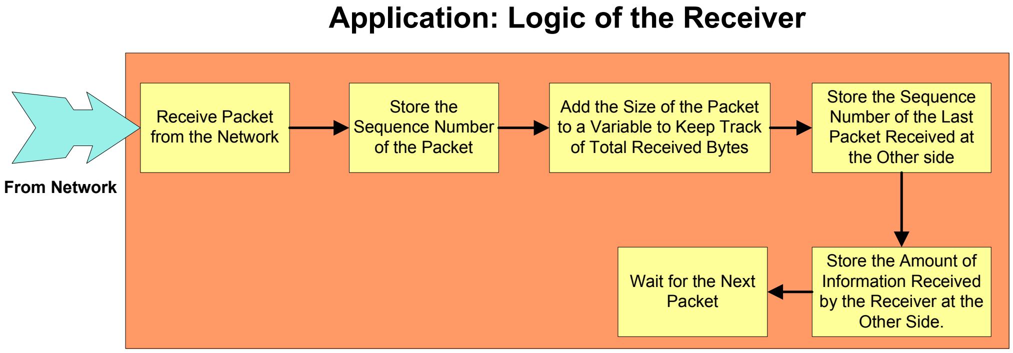

Application: Logic of the Receiver
From Network
Receive Packet from the Network
Store the Sequence Number of the Packet
Add the Size of the Packet to a Variable to Keep Track of Total Received Bytes
Store the Sequence Number of the Last Packet Received at the Other side
Store the Amount of Information Received by the Receiver at the Other Side.
Wait for the Next Packet

Fig. 41. Flowchart showing the schematic representation of the receiver of real-time multimedia application used for ns-2 simulation experiments..

Figure 41 shows the flow chart that provides the schematic representation of the top level design of the algorithms of the receiver of the real-time multimedia application. The receiver is responsible for accepting the packet being sent from the other side. It makes a note of the sequence number and the size of the packet. This information is used to generate the information for sending the feedback through the header of the next packet that is sent into the network by the sender.

B. Testing of the Proposed Control Strategies

The efficacy of the control strategies discussed in the previous chapter will be decided on the basis of the following factors:

- Quality of Service (QoS): The QoS of the real-time audio session is decided by the Mean Opinion Score (MOS) calculated by using E-Model. The E-Model uses the end-to-end packet delays and the packet loss rates to calculate the R-factor that is mapped to the MOS scale using a nonlinear relationship as explained in chapter IV.

- Bandwidth occupancy of the controlled multimedia flow: This factor is related to being a "good citizen of the Internet". Let us assume that a controlled UDP flow employing one of the control strategies devised in the current research has the same end-to-end QoS as compared to an uncontrolled UDP flow. However, if the controlled UDP flow occupied less bandwidth in the network as compared to the uncontrolled UDP flow in order to provide the same level of end-to-end QoS, then using the adaptive flow control strategy is considered to be better than employing no control at all.

C. Determining I_e Curves for Different Modes of the Ideal Codec

The end-to-end QoS of real-time multimedia flows simulated using ns-2 is measured using E-Model. The E-Model has been explained in chapter IV, section C. Measuring the QoS using E-Model is dependent on the I_e curve of the specific codec that is used for compressing the audio signals. The I_e is the effect of impairments caused by packet losses in the network and is dependent on the codec in use. The other parameter in the E-Model, I_d, reflects the effect of impairments because of the delay suffered by the packets, is not dependent on the specific codec in use for compressing audio signal information.

In the current research, MOS of the voice transmitted over the best-effort network is calculated by assuming a desirable codec with six different bit-rates. Therefore, I_e curves for this multi-mode codec with six different bit-rate modes has to be determined in order to calculate the MOS of the real-time VoIP streams in the ns-2 simulations.

As stated earlier in chapter IV, the relation between I_e and overall packet loss rate e can be expressed as:

$$I_e = \gamma_1 + \gamma_2 ln(1 + \gamma_3 e).$$

Ye [77] determined the values of the parameters, γ_1, γ_2, and γ_3, for eight bit-rate modes of Speex. The values of the three parameters for each mode of the Speex codec have already been provided in Table IV. The Speex codec has eleven quality modes, each corresponding to a particular bit-rate. The frame size of Speex is constant at 20 ms irrespective of the "Quality" mode of operation. This is equal to the inter-departure time of the packets generated by the VoIP application employing the ideal codec that is intended to be used for the current research work. The I_e curves of the Speex codec corresponding to the "Quality" modes of 4, 8, and 10 are used to emulate the I_e curves of the desirable codec sending information at the bit-rates of 36 kbps, 48 kbps, and 60 kbps, respectively. This is a valid choice as the bit-rates of these three modes at constant bit-rate without the packet headers are very similar to the bit-rates generated by the desirable codec.

The remaining three I_e curves of the codec for the bit-rates of 72 kbps, 84 kbps, and 96 kbps are determined by extrapolating from the three I_e curves for the bit-rates of 36 kbps, 48 kbps, and 60 kbps. The polynomials representing the I_e curves for three highest bit-rates (72 kbps, 84 kbps, 96 kbps) of the desirable codec used for determining the QoS of the VoIP flows are:

$$\begin{aligned} I_{e_{72}} = {} & 2.187 \times 10^{-12} e^9 - 6.447 \times 10^{-10} e^8 + 8.129 \times 10^{-8} e^7 \\ & -5.736 \times 10{-}6 e^6 + 0.002491 e^5 - 0.006914 e^4 + 0.124 e^3 \\ & -1.444 e^2 + 11.69 e + 5.162, \end{aligned} \tag{7.26}$$

$$\begin{aligned} I_{e_{84}} = {} & 1.704 \times 10^{-9} e^7 - 3.963 \times 10^{-7} e^6 + 3.778 \times 10^{-5} e^5 \\ & -0.001913 e^4 + 0.05589 e^3 - 0.9706 e^2 + 10.52 e + 3.443, \end{aligned} \tag{7.27}$$

$$\begin{aligned} I_{e_{96}} = {} & 1.3 \times 10^{-9} e^7 - 3.118 \times 10^{-7} e^6 + 3.086 \times 10^{-5} e^5 \\ & -0.001633 e^4 + 0.05025 e^3 - 0.9234 e^2 + 10.47 e + 1.722. \end{aligned} \tag{7.28}$$

In equations 7.26, 7.27, and 7.28 the loss rate e is measured in percentage. These equations have been derived by using the curve-fitting toolbox of Matlab.

The six I_e curves for the desirable codec, each corresponding to a particular bit-rate, are shown in the figure 42. The bit-rates are referred to by the sizes of the packets (including headers) sent every 20 ms in the figure. The packet sizes of 90 bytes, 120 bytes, 150 bytes, 180 bytes, 210 bytes, and 240 bytes correspond to flow bit-rates of 36 kbps, 48 kbps, 60 kbps, 72 kbps, 84 kbps, and 96 kbps, respectively. These I_e curves will be used in the E-Model framework to determine the MOS of the transmitted voice signals in the simulations.

D. Calculation of the Mean Opinion Score to Determine QoS in the Simulations

Each of the simulated experiments to determine the efficacy of the developed adaptive flow control strategies run for 120 seconds (two minutes). The MOS of the controlled UDP flows in the simulations are calculated every 10 seconds using the CLR and

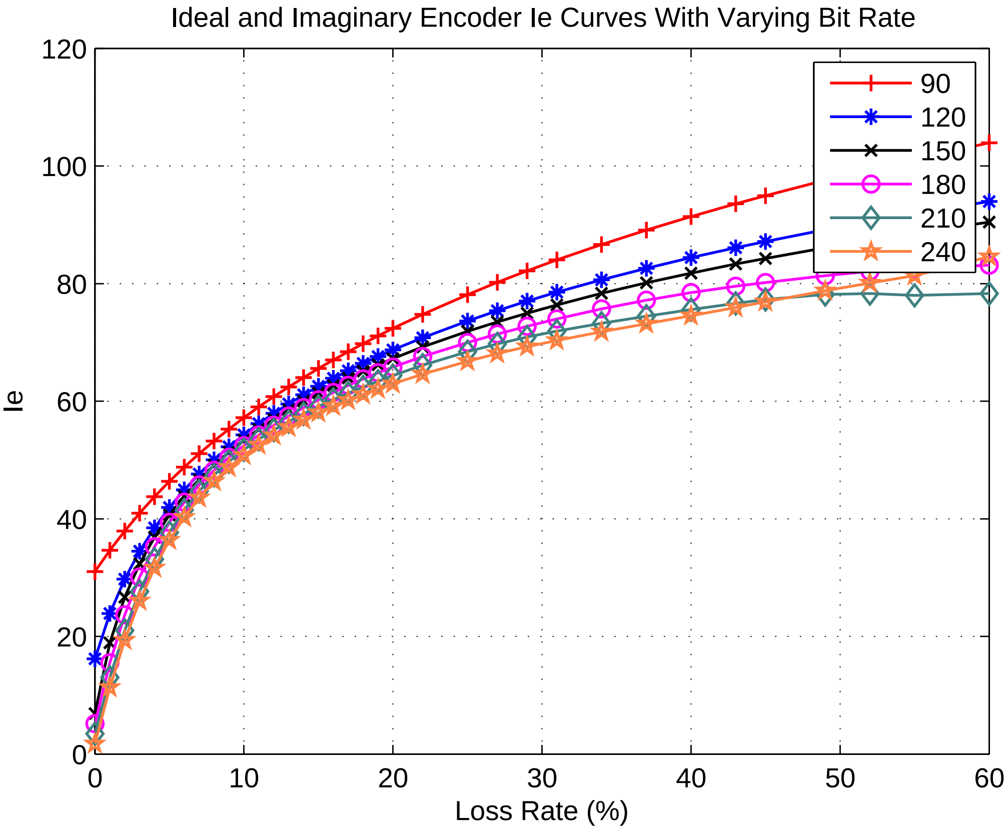


Fig. 42. I_e curves of the desirable six mode codec used for determining MOS of the controlled VoIP flows in ns-2 simulations.

the mean end-to-end packet delays in the specific time slot. The E-Model framework based on parameters R, I_d, and I_e is used to accomplish this. The final MOS of the controlled UDP flows is determined by taking a mean of all the 12 Mean Opinion Scores of the observed flow, calculated every 10 seconds.

E. Performance of the Flow Control Schemes in the Network Topology Used for Validation

In chapter V, two different ns-2 network topologies are discussed. The first topology is used to design and validate the flow control strategies. In this topology five nodes employing controlled UDP flows in the forward direction, from left to right, are attached to router R0. Similarly, five nodes employing controlled UDP flows in the backward direction, from right to left, are attached to router R2. Six sets of experiments are conducted in order to determine the performance of the four designed control strategies. Each set of experiments is conducted by varying the comprehensive loss percentage of the packets in the network. This is done by changing the bandwidth capacity of the link between router R0 and router R1. Table XI shows the bandwidth capacities of the link R0-R1 to perform each set of experiment.

Figure 43 has two plots. The first plot shows the relative performance of the control schemes in terms of the mean MOS of the five "Controlled UDP" flows in the network topology #1 with a CLR of $\sim 3\%$. The second part of the plot shows the mean bandwidth utilized by the five flows.

In the control strategies used High Bit Rate (*HBR*) stands for the experiment in which all the five UDP flows under observation send information at a constant bit rate of 96 kbps. HBR is performed in order to investigate what will happen if the codec sending information at $96kbps$ (best quality) does not respond to congestion

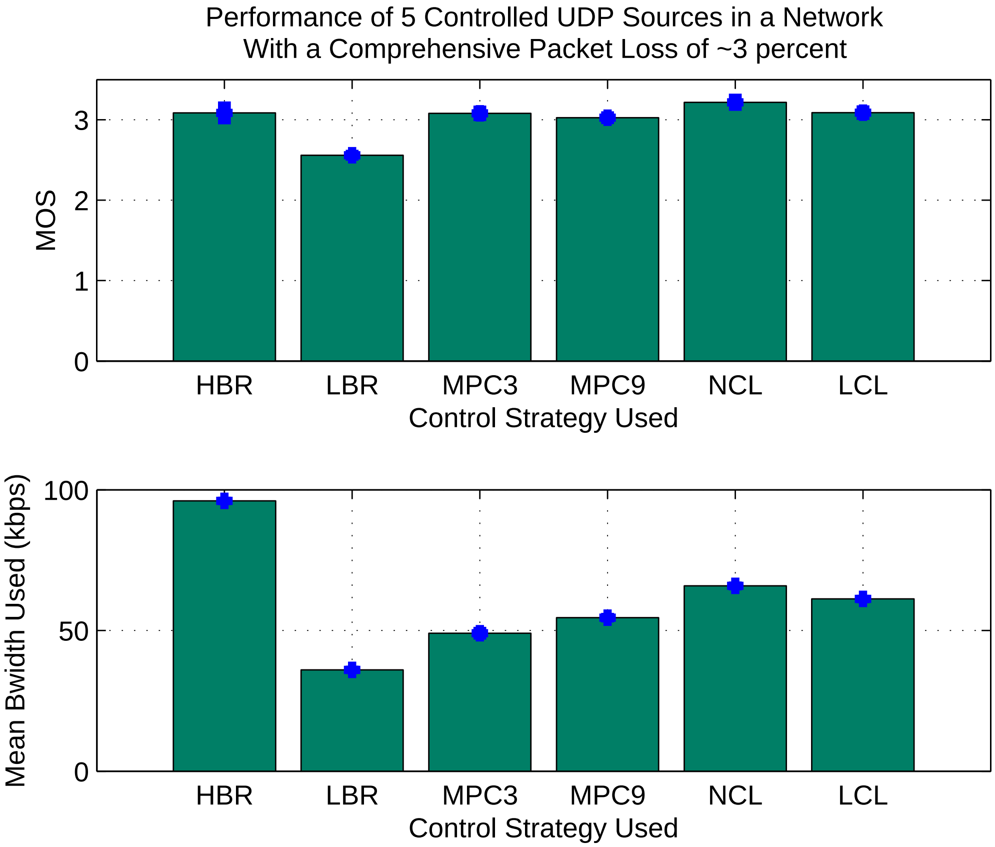


Fig. 43. Relative performance of the control schemes in terms of the mean MOS of the five “Controlled UDP” flows in the network topology #1 with a CLR of $\sim 3\%$.

in the network. Low Bit Rate (*LBR*) stands for the experiment in which all the five UDP flows under observation send information at a constant bit rate of 36 kbps. LBR investigates the effect on the QoS of the VoIP flows if the codec sends flow at the lowest bit rate of 36 kbps (worst quality). MPC3 and MPC9 depict the performance of the five VoIP flows in the forward direction employing MPC control schemes designed using ARX[15 14 1] and ARX[14 15 1], respectively. NCL refers to the performance of the VoIP flows employing nonlinear control law. LCL shows the performance of the VoIP flows employing linear control law.

The plots in figure 43 clearly shows the advantage of adapting the bit-rate of the codec based on the feedback about the congestion level in the network. VoIP flows employing MPC3 and NCL flow control schemes perform better as compared to the cases when the flows do not adapt to the congestion. The flows employing control schemes not only show either similar or better end-to-end QoS but also utilize less bandwidth.

Figure 44 shows the measured accumulation and the control input generated by the NCL controller for "controlled UDP" flow # 1. The dashed lines in the control input plot show the packet sizes related to the six modes of bit-rate of the NCL controller.

In the set of experiments done with the network topology having an average CLR of around 5%, the average MOS of the five flows employing MPC9 control is almost same as that of the average MOS of the same flows when their bit-rate is equal to 96 kbps. This is shown in the figure 45. However, the average bandwidth capacity utilized by each controlled flow is 42.2 kbps less than 96 kbps, i.e. 53.8 kbps.

Figure 46 shows the measured accumulation and the control input generated by the MPC9 controller for "controlled UDP" flow # 1. The red line in the accumulation plot of the figure shows the reference accumulation used for the MPC9 to determine

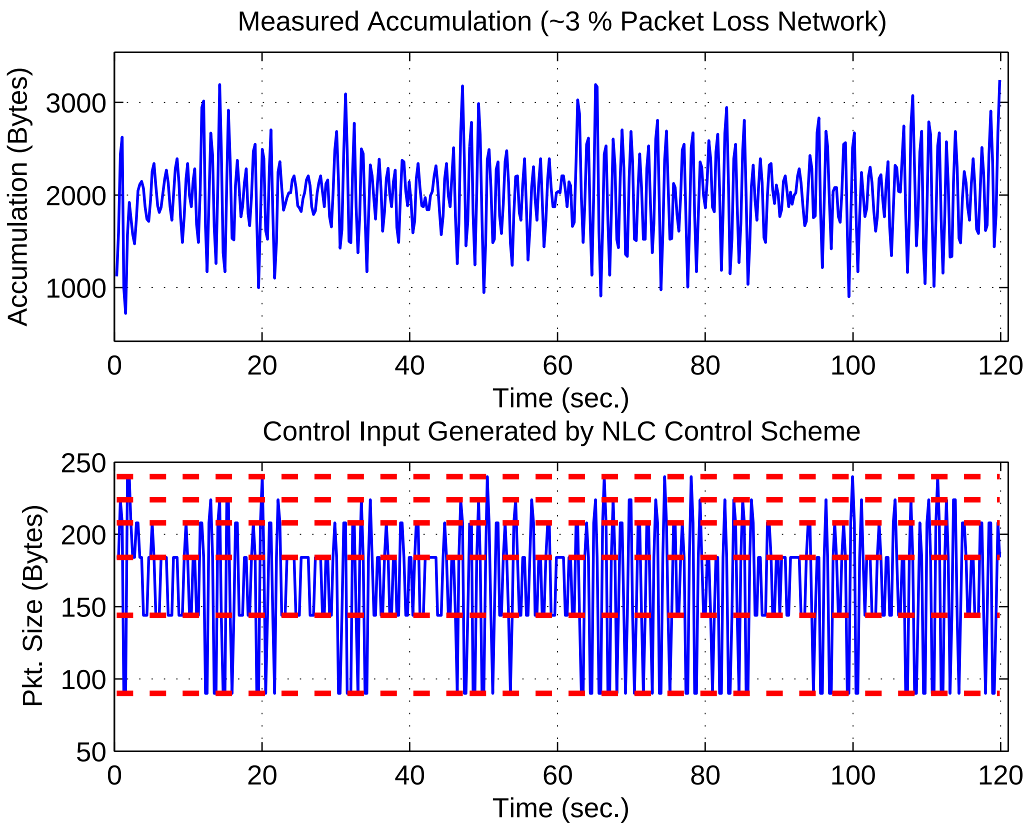


Fig. 44. Measured accumulation and the control input generated by NCL for the network with $\sim 3\%$ CLR.

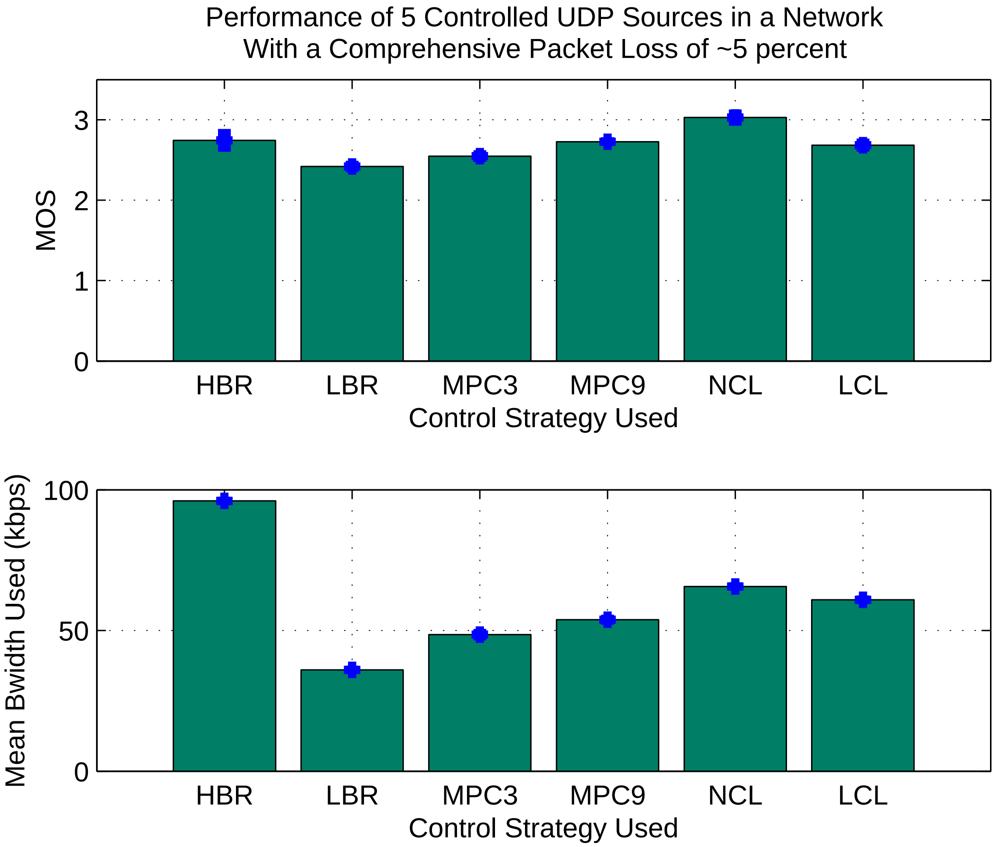


Fig. 45. Relative performance of the control schemes in terms of the mean MOS of the five "Controlled UDP" flows in the network topology #1 with a CLR of $\sim 5\%$.

the future control input. The dashed lines in the control input plot show the packet sizes related to the six modes of bit-rate of the controller. The MPC generates control inputs that are quantized to the six levels as shown in the figure. The quantization is uniform. It can be seen from the figure that the control responds to the accumulation increase as well as decrease in the network.

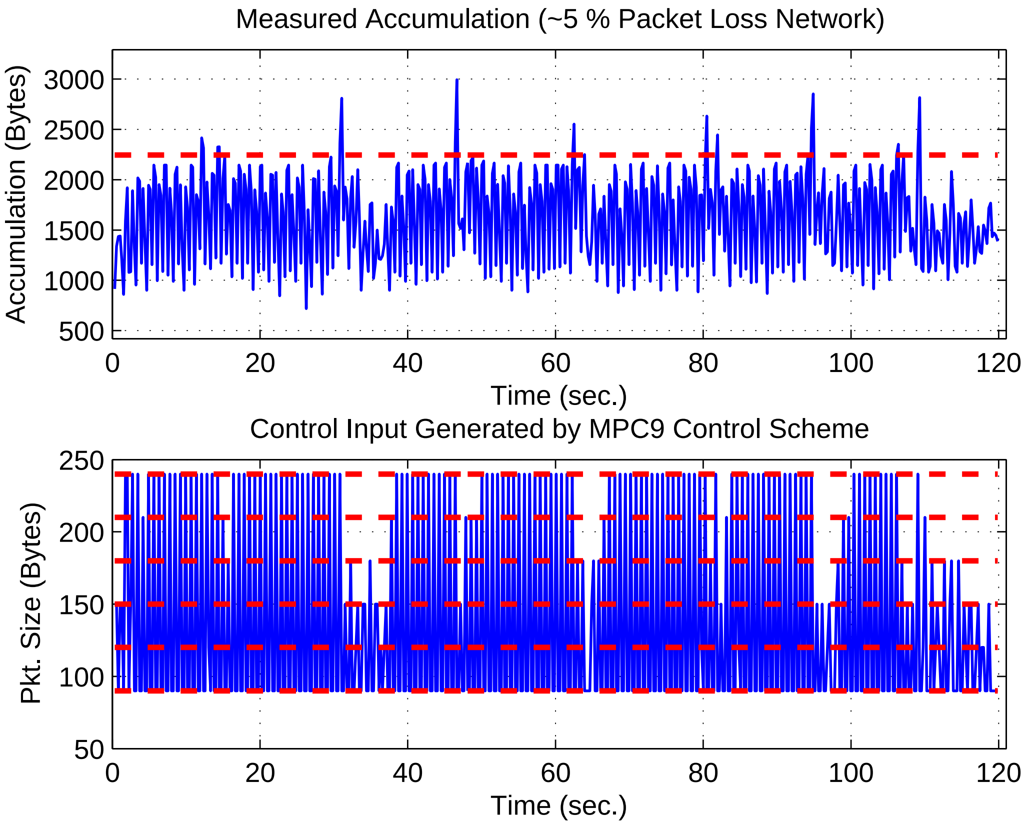


Fig. 46. Measured accumulation and the control input generated by MPC9 for the network with $\sim 5\%$ CLR.

The CLR of the packets of the flows under observation is changed to around 7% in the next set of experiments, as shown in figure 47. It can be seen that QoS of the five UDP flows employing LCL with a simple predictor is better than when these flows did not employ any control and sent information at a constant bit-rate of 96 kbps.

The average bandwidth used up by these controlled flows is 35.43 kbps less than 96 kbps. The other cases in which flows utilize control schemes like MPC3, MPC9, and NCL also perform well while consuming far less bandwidth than the HBR flows.

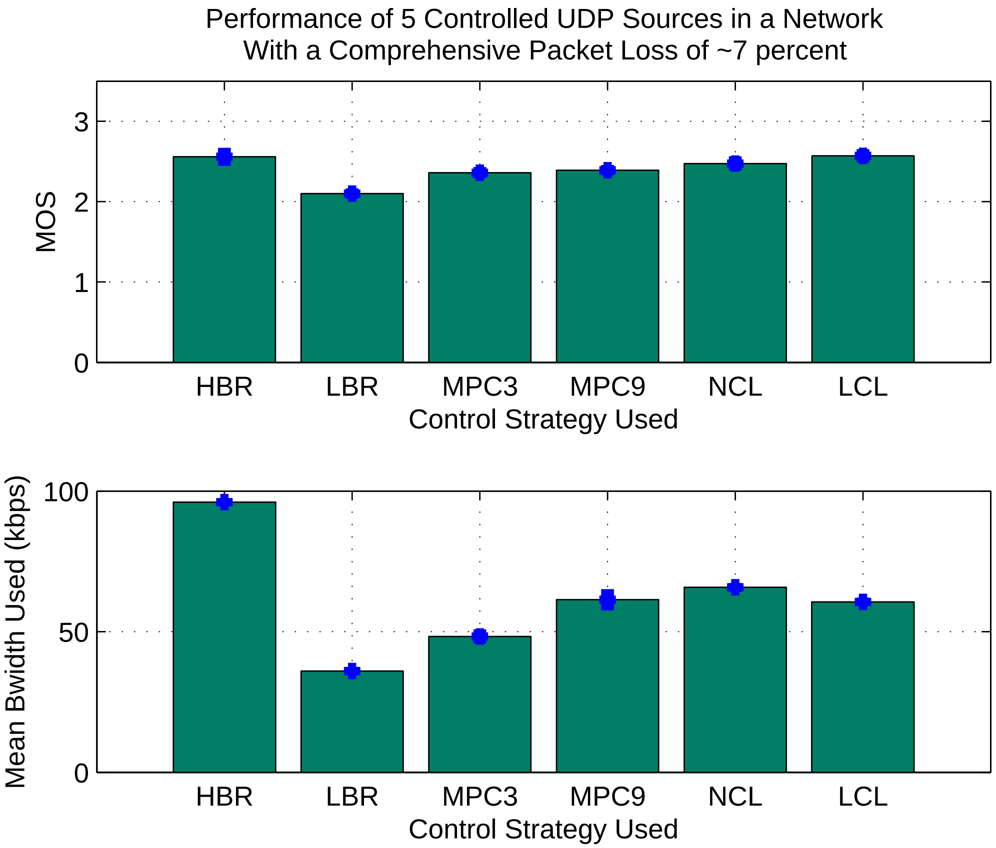


Fig. 47. Relative performance of the control schemes in terms of the mean MOS of the five "Controlled UDP" flows in the network topology #1 with a CLR of $\sim 7\%$.

Figures 48,49, and 50 show the relative performance of the control schemes in network topologies with average CLR of the observed flows equal to 9%, 11%, and 15%. One common trend that emerges from these figures is that the "controlled UDP" flows perform better or as good as the high constant bit-rate flows without employing any kind of control. However, these "controlled UDP" flows utilize far less bandwidth. Another observation that can be made after observing the performance of the control strategies in network topologies with different CLR is that no single control scheme consistently performs better than the others.

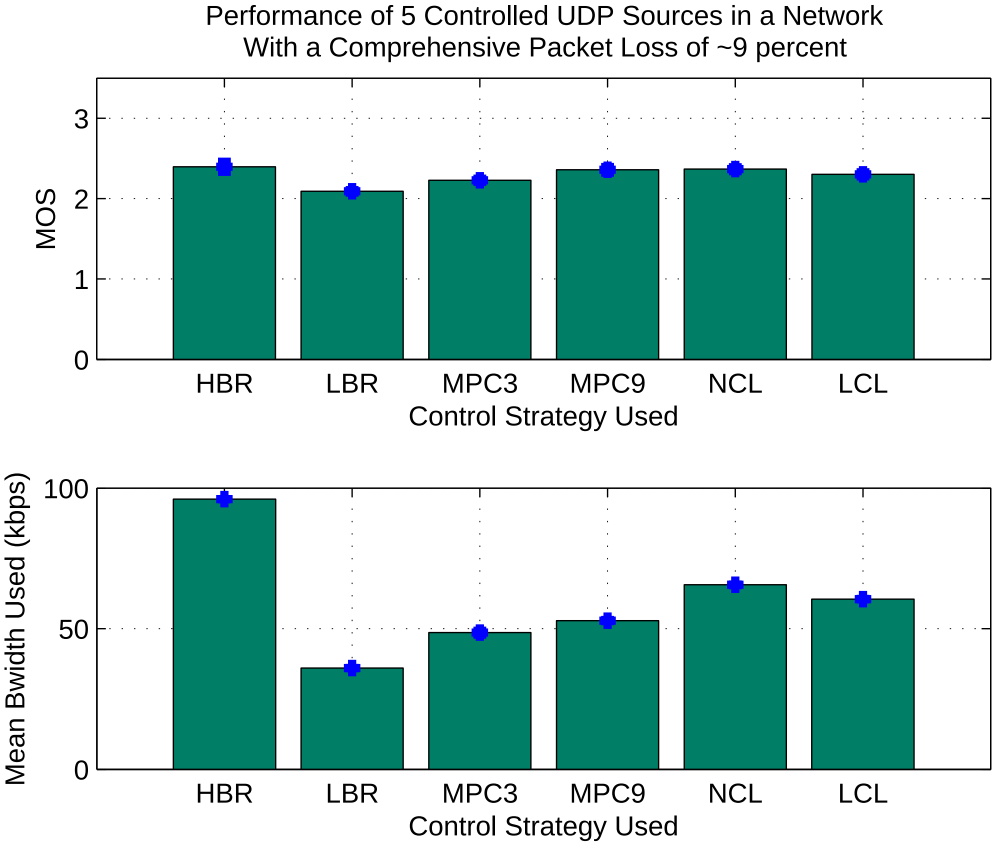


Fig. 48. Relative performance of the control schemes in terms of the mean MOS of the five “Controlled UDP” flows in the network topology #1 with a CLR of $\sim 9\%$.

F. Summary of the Performance of the Control Schemes

The ideal result while deploying end-to-end flow control schemes for real-time multimedia flows in best-effort networks is that the ”controlled UDP” flows use as little bandwidth as possible while delivering highest QoS measured in terms of MOS. How-

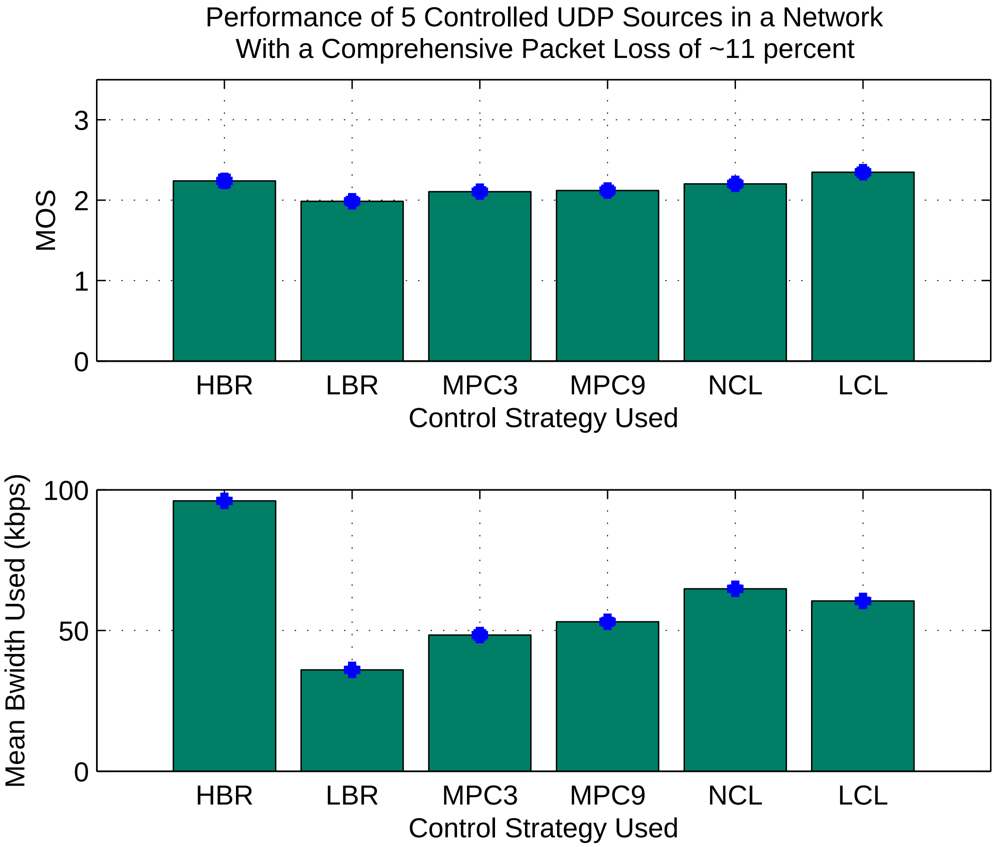


Fig. 49. Relative performance of the control schemes in terms of the mean MOS of the five "Controlled UDP" flows in the network topology #1 with a CLR of $\sim 11\%$.

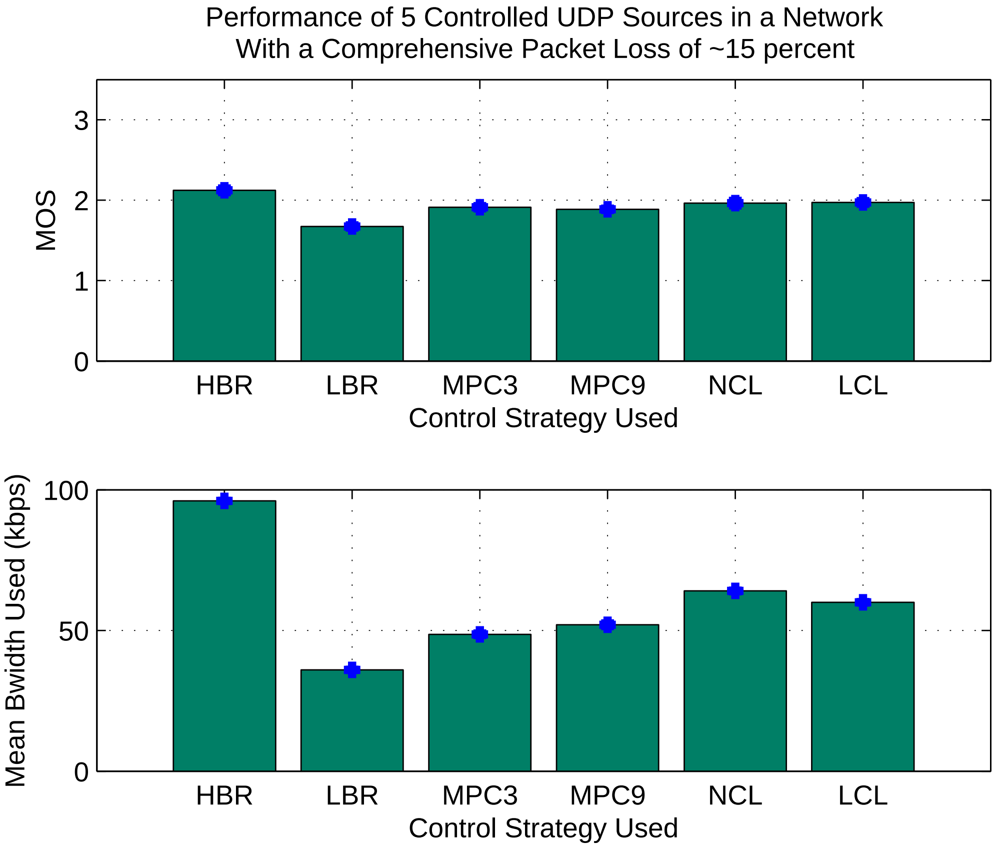


Fig. 50. Relative performance of the control schemes in terms of the mean MOS of the five "Controlled UDP" flows in the network topology #1 with a CLR of $\sim 15\%$.

ever, in practice, there is a limit to which this trade-off between using less bandwidth while delivering high QoS can be achieved. This is because of the limitations of the voice encoding algorithms. If two metrics, MOS^{-1} and the bandwidth utilized by the real-time multimedia flows are plotted with respect to each other, the plot would look somewhat similar to that shown in figure 51. This figure shows that MOS^{-1} decreases as the bandwidth increases and the function showing the relationship between them is convex and nonlinear. Assuming that the cost of bandwidth used by a real-time multimedia flow in a congested network and the end-to-end QoS are equally important, then the most optimal result of the problem of trade-off between bandwidth utilized while delivering good end-to-end QoS can be achieved at the point where the

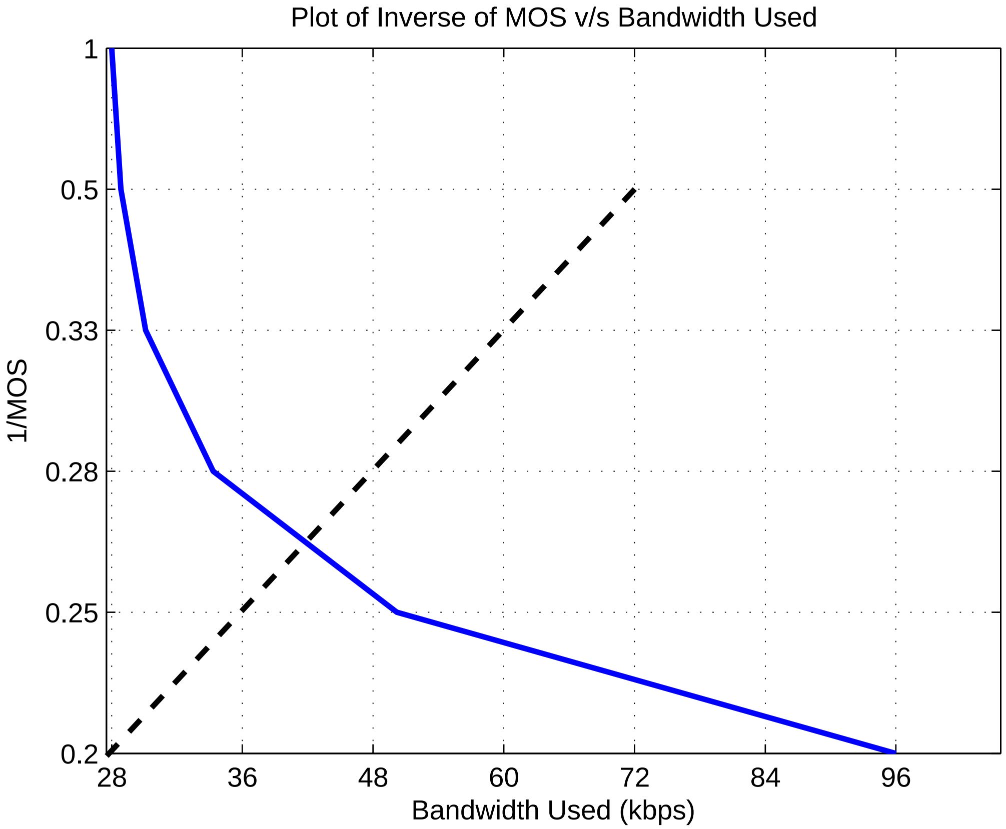


Fig. 51. Ideal plot of MOS^{-1} versus bandwidth utilized by the flows.

straight line intersects the convex curve. If we go above this point while following the convex curve, bandwidth utilization by the flows will decrease. However, the MOS of the flows will decrease simultaneously. Similarly, if we go below this point, the MOS of the flows will increase but the bandwidth utilized by them will increase.

Figure 52 summarizes the results of all the six sets of experiments involving five "controlled UDP" flows using network topologies with different CLRs. There are three distinct regions in the figure. The first region on the left corresponds to the average performance of the five UDP sources sending traffic at the lowest bit rate of 36 kbps in different network scenarios. The last region on the right shows the performance of the five UDP sources when they sent traffic at the highest bit rate of 96 kbps. The region in the center depicts the performance of the control schemes devised during this research. The green band shows the range of bandwidth used by flows employing NCL control scheme. NCL performs a bit worse than the LCL. The red band shows the range of bandwidth used by flows employing LCL control scheme. Among the control schemes based on MPC, MPC9 performs the best. However, the bandwidth capacity utilized by the UDP flows using MPC9 control scheme is less when compared to the bandwidth utilized by the same flows using NCL or LCL control schemes.

None of the control schemes perform consistently well irrespective of the network conditions. Table XV shows the best and the worst performances of control schemes for different CLR networks. LCL and NCL are the best performers. Among the MPC schemes, MPC9 performs the best. The bandwidth saved because of the control laws while delivering equal, if not better, performance in terms of MOS varies from 31.43% to 43.96%. This shows that the adaptive flow control helps to maintain the QoS of real-time multimedia flows and make them better citizens of the Internet.

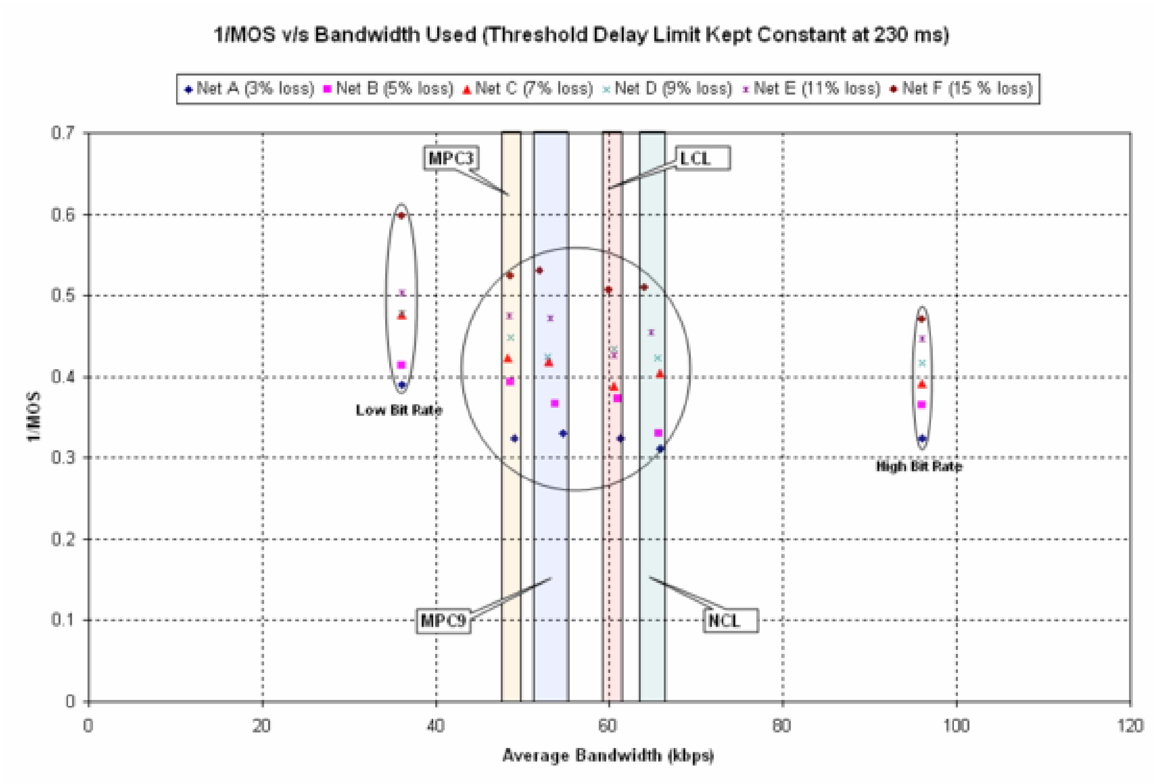


Fig. 52. Plot of inverse of mean MOS of 5 UDP flows employing different control schemes versus mean bandwidth utilized by them during each experiment.

Table XV. Summary of the best and worst performance of the control laws.

CLR (%)	MOS (HBR)	MOS (LBR)	Best	MOS (Best)	BW (kbps)	Worst	MOS (Worst)	BW (kbps)
3	3.09	2.56	NCL	3.22	65.83	MPC9	3.03	54.56
5	2.74	2.42	NCL	3.03	53.80	MPC3	2.55	48.54
7	2.56	2.09	LCL	2.57	60.57	MPC3	2.36	48.31
9	2.39	2.09	NCL	2.37	65.60	LCL	2.30	60.48
11	2.24	1.98	LCL	2.35	60.48	MPC3	2.10	48.39
15	2.12	1.67	LCL	1.97	59.99	MPC9	1.88	52.02

G. Scalability of the Control Strategies

In the previous section, the efficacy of the proposed control schemes is discussed. The ability of the control schemes to provide good quality flows when the network is under congestion and the usage of bandwidth capacity by the adaptive flows are the two aspects of the control schemes presented in the previous section. One of the important aspects of the study of flow control schemes for real-time multimedia flows is to investigate the scalability issues. The question that need to be answered are: (a) what happens when more number of multimedia flows in the network start employing the flow control schemes? (b) how friendly are these control schemes to the flows using TCP?

1. Increasing the Percentage of "Controlled UDP" Flow Traffic in the UDP Traffic Mix While Keeping the Overall Percent of UDP in the Total Traffic Constant

As mentioned in chapter V, the questions posed in the previous paragraph will be answered in two steps. In the first step, a set of experiments is carried out with the contribution of the UDP flows in the total network traffic remaining constant. The ratio of the traffic contributed by the "controlled UDP" with respect to the traffic contributed by the uncontrolled UDP flows is gradually increased. The effect of this increase in "controlled UDP" flow percentage is investigated by observing the goodput of the HTTP connections that exist between link R0-R1. Goodput is defined as the number of useful bits per unit of time from a source to a destination. This excludes protocol overhead and retransmitted data packets. The topology shown in figure 15 is used for the purpose of this investigation.

The variation of the traffic composition in the network is achieved by reducing the average bit-rate of the UDP nodes with exponential application that are attached to the routers R0 and R2. The bandwidth gained by reducing the traffic from exponential flows using UDP is allocated to the "controlled UDP" flows by increasing the number of UDP nodes that use end-to-end flow control schemes. Table XVI shows the variations of the different aspects of the topology of the network #2 in order to conduct this set of experiments. The percentage ratio of the controlled UDP traffic to the total UDP traffic varies from 0.67% to 62.77%. Total UDP contribution to the traffic does not undergo much change. The scalability studies have been performed

Table XVI. Details of topology #2 to see the effect of increasing the percentage contribution of "controlled UDP" flows in the network traffic while keeping the overall contribution of the UDP flows to the total traffic constant.

Experiment	#1	#2	#3	#4
No. of Controlled UDP Flows (R0-R2)	5	36	72	144
No. of Controlled UDP Flows (R2-R0)	5	36	72	144
Max. Bit-Rate/Control. UDP Flow (kbps)	96	96	96	96
Max. Total Bit-Rate of Control. UDP Flows (kbps)	960	6912	13824	27648
Min. Bit-Rate/Control. UDP Flow (kbps)	36	36	36	36
Min. Total Bit-Rate of Control. UDP Flows (kbps)	360	2592	5184	10368
No. of UDP Nodes (R0-R1)	5	5	5	5
Size of Pkts. (Bytes)	512	512	512	512
Inter-Dept. Time of Pkts. (sec.)	0.002	0.002	0.002	0.002
Total Bit-Rate (kbps)	10240	10240	10240	10240
No. of UDP Nodes (R1-R0)	5	5	5	5
Size of Pkts. (Bytes)	512	512	512	512
Inter-Dept. Time of Pkts. (sec.)	0.002	0.002	0.002	0.002
Total Bit-Rate (kbps)	10240	10240	10240	10240
No. of UDP Nodes (R0-R2)	20	20	20	20
Mean Bit-Rate/Flow (kbps)	800	700	600	400
Total Bit-Rate (kbps)	16000	14000	12000	8000
No. of UDP Nodes (R2-R0)	20	20	20	20
Mean Bit-Rate/Flow (kbps)	820	720	620	420
Total Bit-Rate (kbps)	16400	14400	12400	8400
Max of $\frac{Control.UDP}{TotalUDP}$ × 100 (with bursty UDP flows)	1.78	12.39	23.55	42.85
Min of $\frac{Control.UDP}{TotalUDP}$ × 100 (with bursty UDP flows)	0.67	5.04	10.36	21.95
Max of $\frac{Control.UDP}{TotalUDP}$ × 100 (without bursty UDP flows)	2.87	19.57	36.17	62.77
Min of $\frac{Control.UDP}{TotalUDP}$ × 100 (without bursty UDP flows)	1.09	8.36	15.53	38.74

using NCL and MPC9 control schemes. In the controller validation stage with networks of different packet loss percentages, NCL performed the best along with LCL. However, NCL is selected over LCL for scalability because NCL reacts more aggressively to reduce bit-rate of the flow during congestion. MPC9 represents the group of controllers that are designed using unconstrained MPC framework. MPC9 performed far better than MPC3 in the earlier tests. Figures 53, 54, 55, and 56 show the results of the current set of scalability experiments. It can be observed from the figures that the goodput of the hundred HTTP flows between the HTTP nodes connected to routers R0 and R1 has not been affected significantly by changing the percentage of the controlled UDP flows in the UDP flow mix. The control strategies do not affect the HTTP flows adversely.

Overall, the total percentage of the UDP traffic in the total traffic mix remains approximately constant. Although the NCL control scheme performed the best with five controlled UDP sources transmitting in the network topology #1, yet flows deploying MPC9 control scheme start performing better than the NCL consistently, as the percentage of controlled UDP flows go up in the mix of total UDP traffic. The flows deploying MPC9 end-to-end flow control strategy utilize bandwidth in the range of 53 kbps to 55 kbps to deliver better QoS, as compared to high bit-rate UDP flows transmitting information at the rate of 96 kbps. These figures show the advantages of deploying predictive control in real-time multimedia flows.

In the experiments mentioned in this section, the total contribution of UDP to the traffic mix is kept constant. The ratio of controlled UDP traffic to uncontrolled UDP traffic is gradually increased. Table XVII shows that as the number of controlled UDP flows increase in the network, the performance of the MPC9 flow control becomes better. On an average a real-time multimedia flow using MPC9 consumes 45.81% bandwidth less as compared to when it flows uncontrolled at the highest bit-rate.

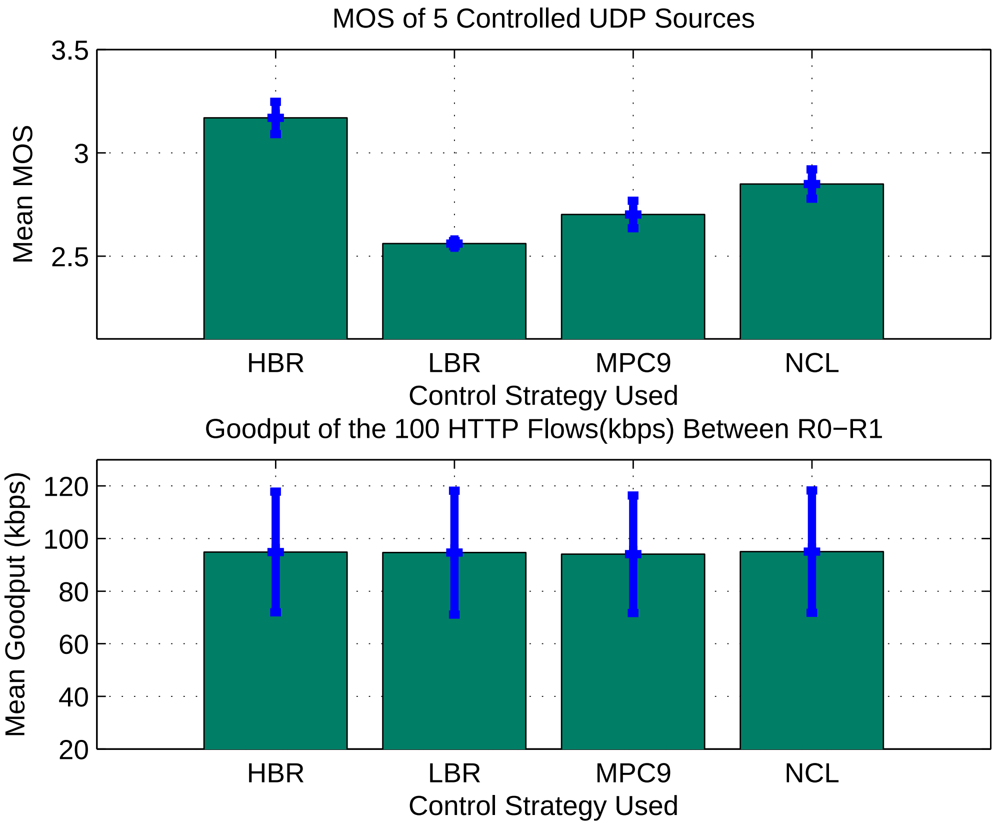


Fig. 53. Effect of 5 controlled UDP sources on the goodput of the HTTP flows between the nodes attached to routers, R0 and R1.

Table XVII. Summary of the performance of the control laws for scalability tests when the contribution of UDP flows to the total traffic is kept constant.

Cont. UDP Srcs.	MOS (HBR)	MOS (LBR)	MOS (MPC9)	MOS (NCL)
5	3.17	2.56	2.70	2.85
36	2.88	2.37	2.89	2.72
72	2.73	2.23	2.83	2.71
144	2.76	2.45	2.76	2.55

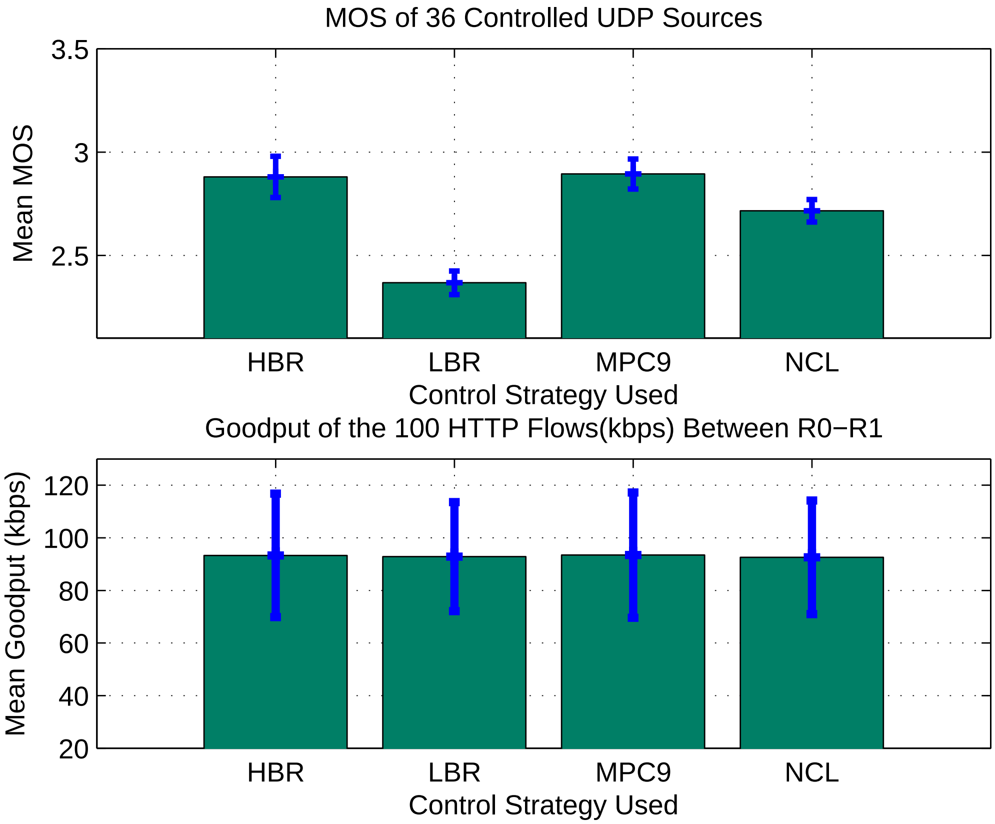


Fig. 54. Effect of 36 controlled UDP sources on the goodput of the HTTP flows between the nodes attached to routers, R0 and R1. Percentage of UDP traffic comprising the total traffic in the network has been kept approximately constant.

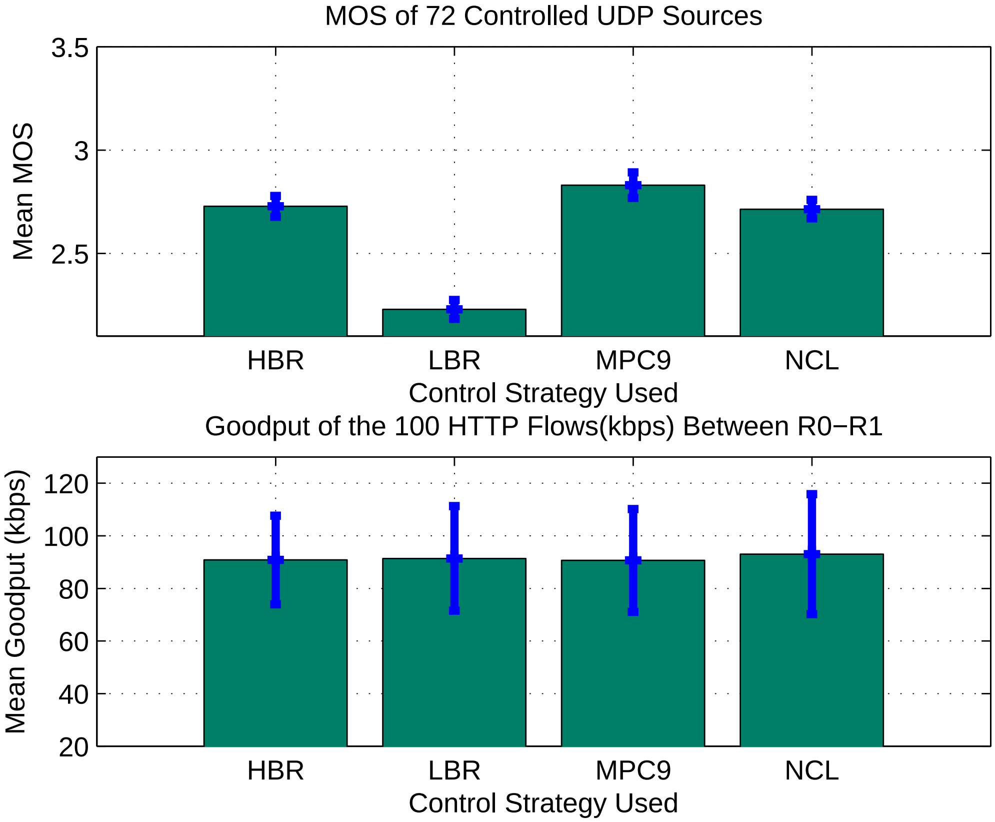


Fig. 55. Effect of 72 controlled UDP sources on the goodput of the HTTP flows between the nodes attached to routers, R0 and R1. Percentage of UDP traffic comprising the total traffic in the network has been kept approximately constant.

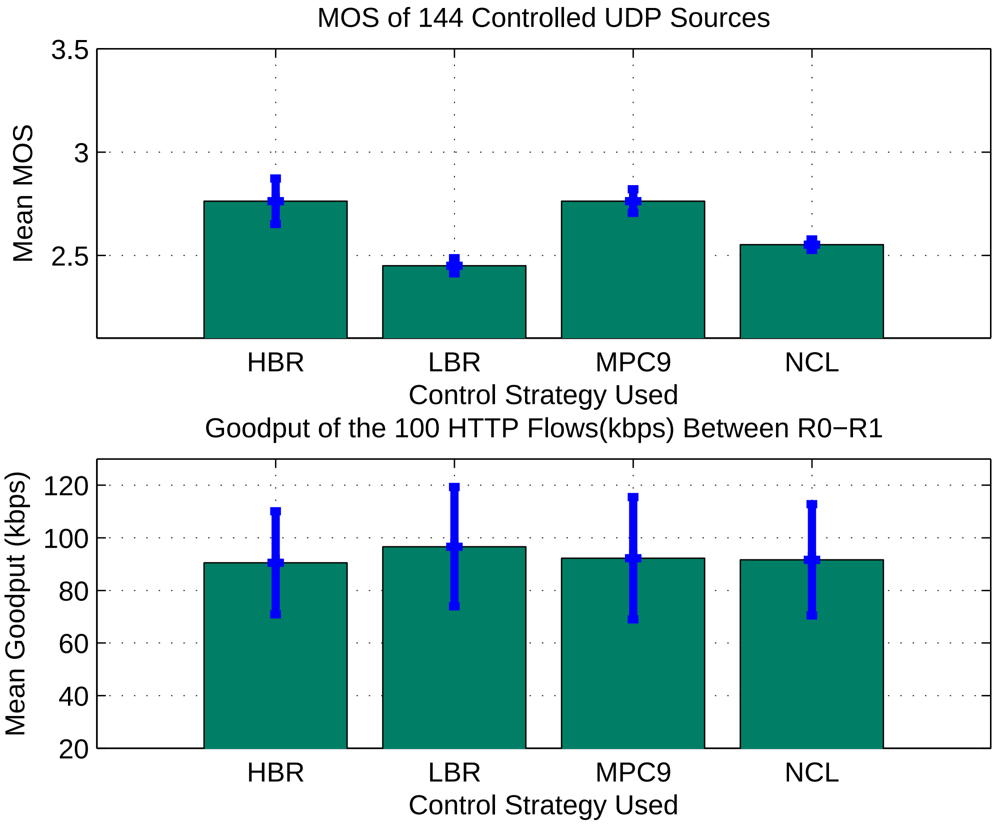


Fig. 56. Effect of 144 controlled UDP sources on the goodput of the HTTP flows between the nodes attached to routers, R0 and R1. Percentage of UDP traffic comprising the total traffic in the network has been kept approximately constant.

2. Increasing the Percentage of "Controlled UDP" Traffic in the Total Traffic Mix

The last part of the current research is to determine the scalability of the new control schemes when the percentage of UDP composition in the total network traffic goes up. The increase in the percentage of the UDP composition in the network traffic happens because of the increase in the number of UDP nodes using the designed control schemes. This is achieved by increasing the number of nodes using the flow control schemes over UDP attached to the routers, R0 and R2. No other parameter of the network topology used for studying the scalability of the control schemes is changed. Four sets of experiments are performed by changing the number of "controlled UDP" flows to 5, 20, 50, and 80.

Figure 57 shows the percentage of TCP traffic constituting total traffic in the network during the experiments to determine the performance of the control schemes when the percentage of the UDP content of the total traffic increases because of the increase in "controlled UDP" flows. As the number of "controlled UDP" sources are increased from 5 to 80, the traffic component comprising of the TCP flows drops from 81.31% to 73.46% percent when all the "controlled UDP" nodes transmit at the highest constant bit-rate of 96 kbps without responding to the network congestion.

The corresponding bar graph for the percentage of UDP traffic in the network as the number of "controlled UDP" flows are increased from 5 to 80 is shown in figure 58. The percentage of UDP traffic increases from 18.69% to 26.54% when all the "controlled UDP" nodes transmit at the highest constant bit-rate of 96 kbps without responding to the network congestion. As the number of "controlled UDP" nodes in the simulation are increased, the amount of bandwidth saved by deploying the control schemes MPC9 and NCL becomes more evident. The savings in the bandwidth is not visible when the number of "controlled UDP" flows is small. However, this

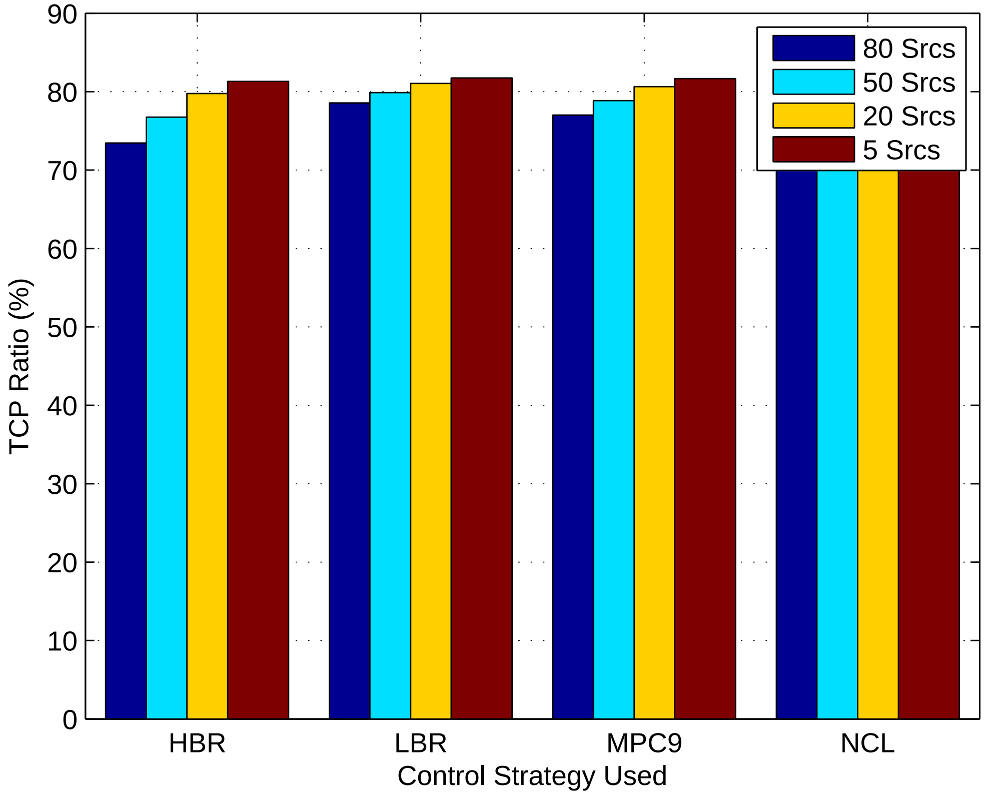


Fig. 57. Percentage of TCP traffic as proportion of the total traffic in the network topology used for scalability studies.

saving increases as the number of flows deploying the end-to-end adaptive flow control schemes increases. This means that real-time multimedia flows deploying adaptive flow control schemes in the network will help the network managers adapt to the increase in the UDP usage in the future. Figure 59 depicts the percentage of the controlled UDP flow traffic as compared to the total UDP traffic in the network topology used for studying scalability.

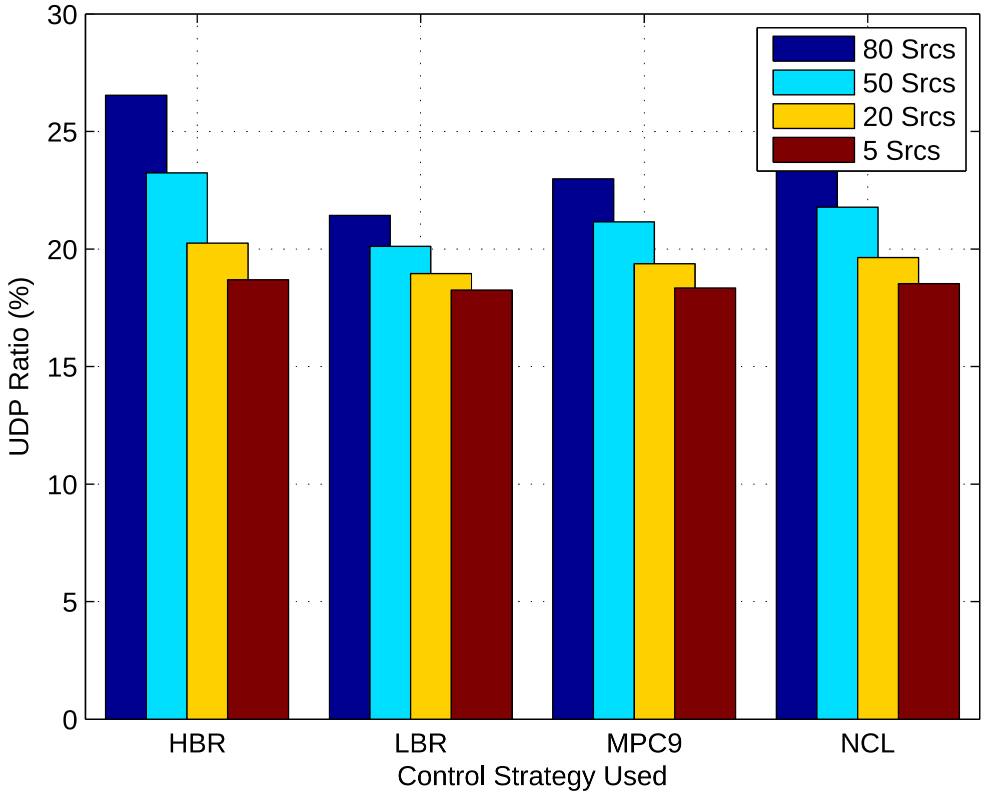


Fig. 58. Percentage of UDP traffic as proportion of the total traffic in the network topology used for scalability studies.

The quality of the real-time multimedia sessions during the different experiments need to be examined in order to see the efficacy of the flow control schemes when the percentage of the UDP flow contribution increases in the total traffic.

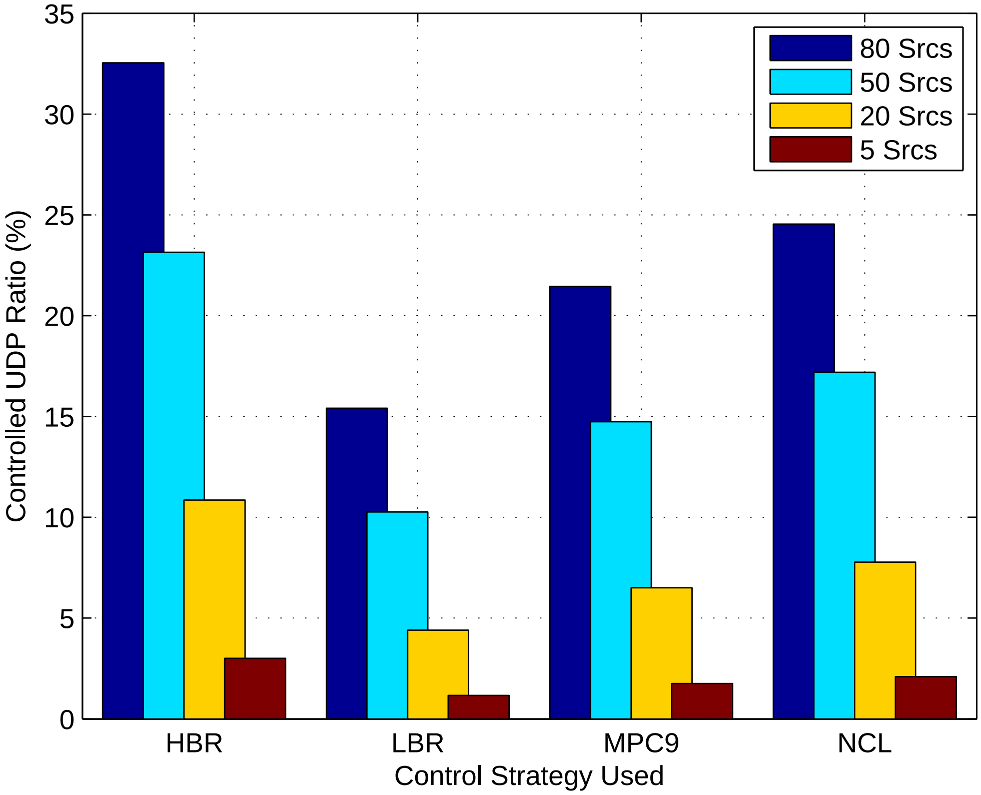


Fig. 59. Percentage of "controlled UDP" traffic as proportion of the total UDP traffic in the network topology used for scalability studies.

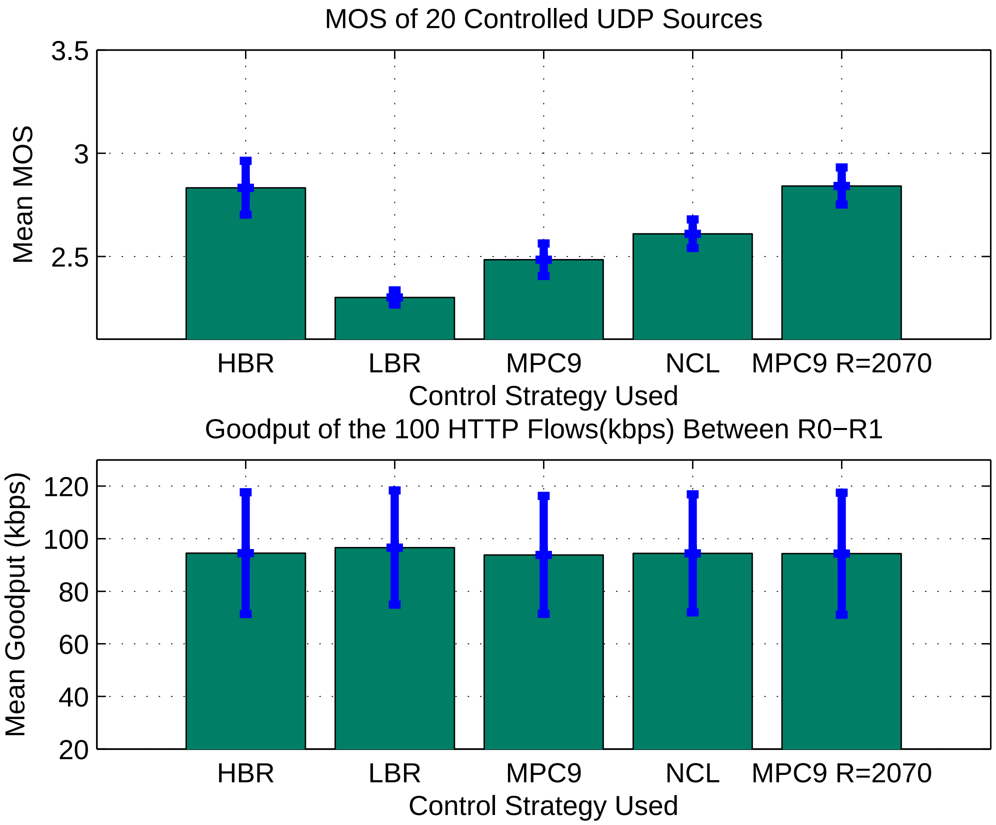


Fig. 60. MOS of 20 “controlled UDP” flows and their effect on the goodput of the 100 HTTP flows between routers, R0 and R1.

Average MOS of the five “controlled UDP” sources and their effect on the goodput of the 100 HTTP flows between routers, R0 and R1 has been shown in figure 53. The figure shows that the quality achieved by the five flows when they transmit at the highest constant bit-rate of 96 kbps can not be matched by the same flows when they are subjected to the end-to-end adaptive flow control schemes. Figure 60 shows the mean MOS of the twenty “controlled UDP” sources. The case with 20 “controlled UDP” flows using MPC9 and NCL control scheme does not perform as well as when they are uncontrolled. MPC9 uses a reference control signal of 2240 bytes in order to determine the control inputs. This reference control is changed to 2070 bytes. The 20 UDP flows with control inputs determined using the lower reference signal outperform the 20 UDP flows transmitting at the highest constant bit rate of 96 kbps. The congestion in the network has increased with the addition of 15 more flows in the forward direction. This means that a higher reference accumulation signal is not able to distinguish between a normal system and a congested system. A more conservative control scheme using a lower reference signal takes care of the congestion better while delivering better quality flows.

Figure 61 shows the mean MOS of the fifty “controlled UDP” sources. MPC9 outperforms others in terms of performance. On an average, the 50 flows employing MPC9 control strategy utilize 41.2 kbps less bandwidth than the highest constant bit rate of 96 kbps while delivering better quality. This shows the benefits of using adaptive end-to-end flow control in order to deliver better quality while saving bandwidth.

Figure 62 shows the mean MOS of the eighty “controlled UDP” sources. The performance of the eighty flows employing MPC9 controller is similar to the unconstrained constant high bit rate 96 kbps UDP flows. However, the bandwidth savings due to flow control schemes is substantial for 80 UDP sources. If each flow transmits

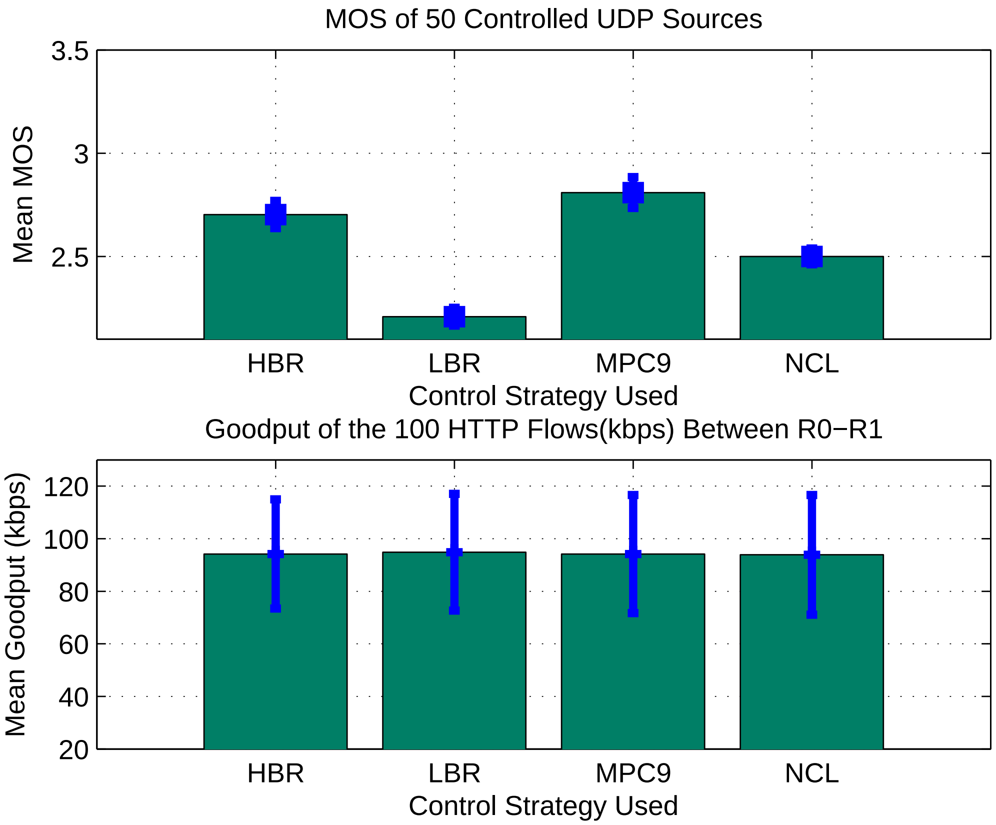


Fig. 61. MOS of 50 “controlled UDP” flows and their effect on the goodput of the 100 HTTP flows between routers, R0 and R1.

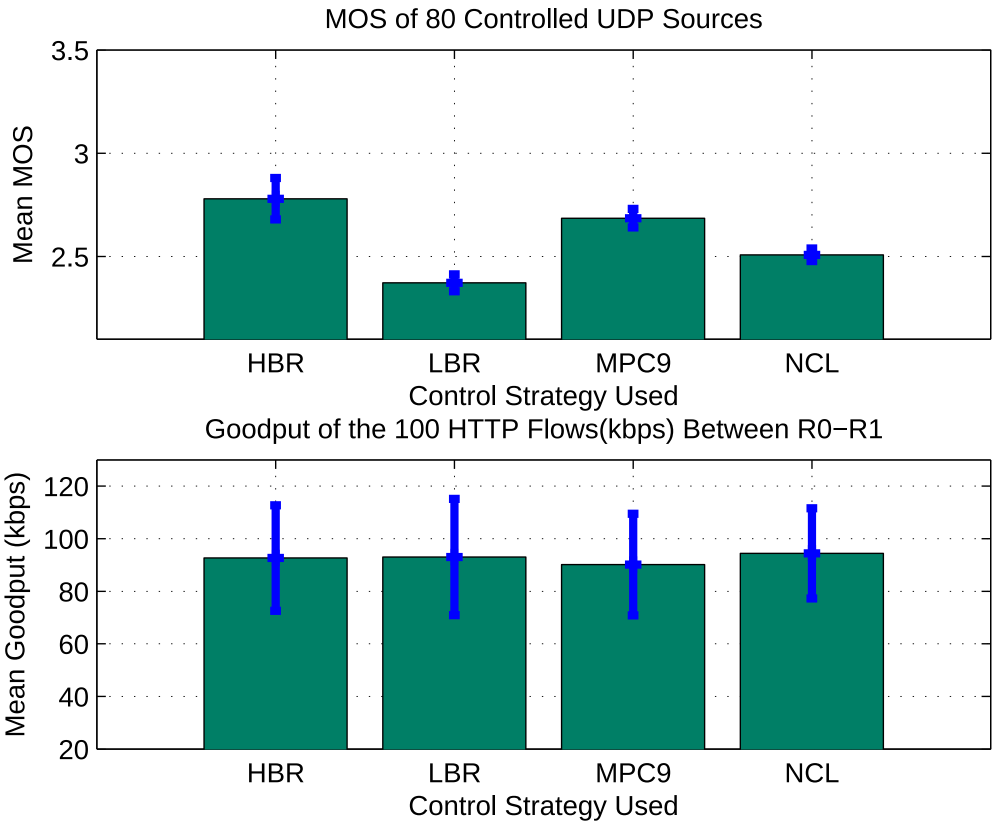


Fig. 62. MOS of 80 "controlled UDP" flows and their effect on the goodput of the 100 HTTP flows between routers, R0 and R1.

at a constant bit rate of 96 kbps, then eighty flows consume 7.68 Mbps of available bandwidth. However, when these eighty UDP flows employ MPC9 control schemes they consume 4.34 Mbps of the available bandwidth. This results into a saving of 3.34 Mbps of bandwidth that can be used by other flows. Table XVIII summarizes the mean MOS achieved by the UDP sources as the contribution of the UDP traffic to the total traffic is increased. Flows using MPC9 and NCL controllers scale up well as the number of flows deploying them increase.

Table XVIII. Summary of the performance of the control laws for scalability tests when the contribution of UDP flows to the total traffic increases.

Cont. UDP Srcs.	MOS (HBR)	MOS (LBR)	MOS (MPC9)	MOS (NCL)	MOS (MPC9) Ref. = 2070
5	3.17	2.56	2.70	2.85	N.A.
20	2.83	2.30	2.48	2.61	2.84
50	2.70	2.21	2.81	2.50	N.A.
80	2.78	2.37	2.68	2.51	N.A.

H. Chapter Summary

The first part of the chapter deals with the theoretical formulation and the development of control strategies for alleviating end-to-end congestion in a best-effort IP network. Four control strategies are developed. The first two - Linear Control Law (*LCL*) and Nonlinear Control Law (*NCL*), are very simple and intuitive control laws developed mostly from empirical data and observations. The last two control strategies - MPC3 and MPC9, are based on Model Predictive Control (*MPC*) formulation. These control laws use the framework provided by the two ARX predictors, developed earlier in the research.

The later part of the chapter discusses the results achieved by deploying the proposed control strategies in simulated environments. The specifications of the desirable codec used for measuring the MOS of all the flows is proposed. The performances of different control schemes are determined by changing the network topology to increase the percentage of comprehensive packet losses of the observed flows from 3% to 15%. MPC9 and NCL show the best performance as compared to the other controllers. These two control schemes are selected and put through a series of tests that aim to study their behavior as the characteristics of traffic change in the network. The first set of experiments in conducted to test the performance of the two control schemes when the percentage of the "controlled UDP" traffic in the network as compared to the uncontrolled UDP traffic increases steadily. The proportion of UDP to TCP traffic is kept constant. In most cases, MPC9 based on the ARX model of the network outperforms NCL not only in terms of the average quality of the voice delivered by the flows but also in terms of the average bandwidth consumed by the flows. At higher percentage of "controlled UDP" traffic, the bandwidth saved by the deployment of the control schemes is substantial and the performance is at par with the flows that deliver information at the highest constant bit-rate of 96 kbps.

The last section of the chapter investigates the performance of the MPC9 and NCL control schemes when the composition of traffic in the network topology is changed by increasing the ratio of UDP traffic with respect to the TCP traffic. Flows using MPC9 deliver QoS comparable to constant high bit-rate flows of 96 kbps. The controllers designed are friendly to the TCP flows as they do not lead to any abnormal and unexplained decline of the goodput.

CHAPTER VIII

SUMMARY AND CONCLUSIONS

This chapter summarizes the work presented in earlier chapters. Finally, research conclusions and suggestions for future work are presented.

A. Summary of Research

Chapter I provided an introduction to the problem. Congestion control can be achieved by developing suitable buffer management and buffer scheduling algorithms in the routers. They can also be achieved by using new frameworks like IntServ and DiffServ on top of the best-effort IP networks. However, all these approaches need substantial financial investment by varied parties to change the basic nature of the best-effort IP networks. End-to-end flow control schemes deployed in the application layer of the real-time multimedia applications is the most cost-effective and practical way to deal with the flow control problem. Many researchers have proposed various schemes for end-to-end flow control of real-time multimedia flows over the best-effort networks. In the current research, audio conferencing applications have been chosen among different types of multimedia applications for investigations. The objective of current research is to improve the quality of the voice at the receiver in audio conferencing applications when they communicate through best-effort networks while trying to remain friendly to other competing flows.

Chapter II provides an overview of the existing literature regarding the current topic at hand. Many notable researchers around the world have proposed different schemes for real-time multimedia applications using UDP. The chapter has been divided into two parts. The first part deals with the research papers related to audio applications. The second part deals with the research papers related to video applications. Although video applications form an integral part of the multimedia applications, the current research work focuses on audio applications because of their simplicity. The intention of this research is to demonstrate the feasibility of the flow control algorithms for real-time multimedia applications.

Chapter III discusses different types of codecs used for encoding voice in real-time audio applications. The codecs generate audio flows (without headers) that vary in their bit-rate from 4.75 kbps to 64 kbps. If the headers due to various layers of the network architecture are added the bit-rate of the codecs vary from 26.1 kbps to 96 kbps. A large majority of the codecs use 20 ms as the frame size.

Chapter IV presents the different tests used for judging the quality of the voice transmitted by the audio applications. The subjective tests involve the audio being played out under certain specific conditions to a set of listeners. These listeners rate the audio quality on a predetermined scale. An average of these scores is finally determined to be the quality of the audio sample. Some objective tests to determine the quality of the audio have also been developed. They are more easier to perform and rely on signal processing algorithms. There is a direct causality between packet losses and large end-to-end delays of the packets of an audio application to a lower quality of the audio that has been transmitted over the network. The relationship is effectively modeled using a framework called E-Model. E-Model is used extensively to calculate all the results in the research

Chapter V talks about the unsuccessful attempts to demonstrate the ideas proposed in the current research in a real world scenario and the development of ns-2 network topologies. Although the PlanetLab overlay network provides a useful test bed for many disruptive network technologies, the traffic in the network between the nodes is not large enough to perform the experiments needed to show the feasibility of the control schemes. Two network topologies are developed in ns-2 environment for this purpose. Each topology has a significant amount of traffic comprising of HTTP, FTP, and UDP flows. There are two backbone links. The cross-traffic in the first link creates the variation in congestion as seen over the real world networks. The cross traffic in the second link performs the role of building up normal levels of congestion within the network. The first network topology is used for selecting the best performing controllers. The second network topology is used for testing the scalability of the selected best performing control schemes.

Chapter VI shows the derivation of the state space equations to model a real-time flow using accumulation. Justification is provided for the use of accumulation and bit-rate signals as the output and the input signals, respectively. Two ARX models are proposed to model the real-time flows, ARX[15 14 1] and ARX[14 15 1]. Both of them perform equivalently under varying network conditions. SSPs based on these ARX models are used to develop MPC laws in the next chapter.

Chapter VII is divided into two parts. Theoretical formulation of the control schemes are proposed in the first part. Two simple control schemes - Linear Control Law and Nonlinear Control Law are proposed. The equations for developing unconstrained MPC controllers are presented. The first MPC controller MPC3 is based on the ARX model developed using the signals derived form the network topology #1 with CLR of 3%. The second MPC controller is based on the ARX model developed using the signals derived form the network topology #1 with CLR of 9%.

The later part of chapter VII demonstrates the performance of the various control schemes. Although the NCL scheme performs better than the other control schemes during the experiments performed to select the best controllers, yet the MPC9 scheme definitely outperforms all the other control schemes during the scalability studies. NCL and MPC9 were selected for the scalability experiments. Many times the performance delivered by the “controlled UDP” flows measured in terms of the average

MOS of the flows, is almost the same as that of the performance delivered by the same flows transmitting information at the highest constant bit-rate of 96 kbps. However, the bandwidth consumed by the "controlled UDP" flows is significantly less. This proves that adaptive real-time multimedia flows that respond to congestion in the Internet not only are more friendly to the other flows in the network but also deliver either comparable or better performance in terms of QoS.

B. Conclusions

The current research proposes novel end-to-end flow control algorithms for unicast real-time multimedia application transmitting over best-effort networks. Proposed real-time multimedia flow control schemes help to cut down the bandwidth needed by the flows anywhere between 31.43% and 43.96%. Simultaneously, they deliver QoS as good as, if not better, than that of the highest bit rate VoIP flow.

The prediction control framework is exploited to introduce MPC based control schemes that perform well under varying network conditions. MPC9 controller does not perform well as compared to the other schemes when the proportion of "controlled UDP" flows in the traffic is very less. However, MPC9 does well as the proportion of "controlled UDP" flows in the network is increased.

Two other simple flow control schemes, LCL and NCL, are proposed. They are inherently reactive control algorithms. These simple schemes perform well when the network is lightly loaded. However, their relative performance as compared to MPC9 deteriorates as the proportion of "controlled UDP" flows in the network is increased.

The following conclusions can be derived from the current research:

1. Active control of hard real-time flows delivers the same or somewhat better QoS as HBR (no control), but with a lower average bit rate. Consequently, it helps

reduce bandwidth use of controlled real-time flows.

2. Proposed reactive control schemes can be used by real-time multimedia applications for reducing the bandwidth consumed by the flows. These schemes deliver good QoS. They do not scale up as well as the predictive control schemes.

3. Proposed predictive control schemes are better in delivering good quality QoS while using up even less bandwidth. They scale up well as more real-time multimedia flows start employing them.

C. Limitations

Some of the limitations of the current work are:

1. Lack of real-world data: All the experiments have been done under the ns-2 simulation framework. Lot of time has been spent in developing network topologies that reflect the conditions of the Internet. In spite of wide acceptability of ns-2 in the networking community, a simulator can not replicate the real world conditions completely.

2. Robustness of predictive controllers: The SSPs designed to formulate the MPC strategies have been developed using data generated by simulations using Topology #1. The scalability tests of the the MPC controllers have been performed using Topology #2. The two network topologies are very similar in nature. This might be one of the reasons why the predictive control strategies scale up well. If the predictive controllers are used in a different network topology, the controllers might not perform as well. There is a possibility that LCL and NCL might turn out to be more robust in terms of their ability to handle varied

network topologies as compared to the MPCs that are dependent on the flow models.

3. MPC formulation does not consider constraints on the codec: The MPC formulation has been done without taking into account the limitations of the codec. This results into a non-optimal controller that does not perform as well as it should.

4. Lack of real-world implementation: Multi-mode variable bit-rate codecs that can change their bit-rate dynamically during a VoIP session need to be developed. This is a very difficult task. Without the availability of such codecs, it is not feasible to implement the flow control algorithms suggested in the current work.

D. Recommendations for Future Work

This research shows the feasibility of developing end-to-end flow control schemes for real-time unicast multimedia applications based on predictive framework. Some of the recommendations for future work are:

1. Develop suitable real-world test-beds to perform controlled experiments to collect data. Modeling using real-world data is far more acceptable than data obtained from simulations. All the flow control algorithms need to be tested using these test-beds.

2. Perform comparison studies of the current set of proposed control schemes with the other proposed popular control schemes for multimedia applications like TCP Friendly Rate Control- Packet Size (*TFRC-PS*).

3. Develop and study the performance of more complicated end-to-end flow controllers based on MPC framework that incorporate the constraints of the actuator in the theoretical formulation.

REFERENCES

[1] D. Clark, "The design philosophy of the DARPA internet protocols," *SIGCOMM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 106–114, 1988.

[2] D. Wu, Y.T. Hou, W. Zhu, Y-Q Zhang, and J.M. Peha, "Streaming video over the Internet: approaches and directions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 282–300, March 2001.

[3] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, pp. 40–48, September/October 1998.

[4] B.W. Wah, X. Su, and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the Internet," in *International Symposium on Multimedia Software Engineering*, Taipei, Taiwan, December 2000, pp. 17–24.

[5] J. Liebeherr, "Overlay networks," http://www.cs.virginia.edu/ cs757/slidespdf/757-09-overlay.pdf, October 2004.

[6] W. Stallings, *Data and Computer Communications*, Upper Saddle River, NJ: Pearson Education Inc., eighth edition, 2007.

[7] D. Minoli and E. Minoli, *Delivering* V*oice over* IP N*etworks*, New York: Wiley, second edition, 2002.

[8] T. Yensen, R. Goubran, and I. Lambadaris, "Synthetic stereo acoustic echo cancellation structure for multiple participant VoIP conferences," *IEEE Transactions on Speech Audio Processing*, vol. 9, pp. 168–174, February 2001.

[9] X. Chen, C. Wang, D. Xuan, Z. Li, Y. Min, and W. Zhao, "Survey on QoS management of VoIP," in *International Conf. on Computer Networks and Mobile Computing (ICCNMC)*, Shanghai, China, October 2003, pp. 69–77.

[10] J. Glasmann, W. Kellerer, and H. Müller, "Service architectures in H.323 and SIP: a comparison," *IEEE Communications Surveys and Tutorials*, vol. 5, no. 2, pp. 2–17, Fourth Quarter 2003.

[11] M. Gough, *Skype* Me*!*, Rockland MA: Syngress Publishing Inc., 2006.

[12] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, pp. 47–58, April 1991.

[13] B. Fufht, "A survey of multimedia compression techniques and standards. Part I. JPEG standard," *Real Time Imaging*, vol. 1, pp. 49–67, 1995.

[14] "ITU-T, Recommendation H.261," http://www.itu.int/rec/T-REC-H.261/en, March 1993.

[15] M. Baldi and Y. Ofek, "End-to-end delay analysis of videoconferencing over packet-switched networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 479–492, 2000.

[16] A. M. Adas, "Supporting real time VBR using dynamic reservation based on linear prediction," Tech. Rep. Report GIT-CC-95/26, Georgia Institute of Technology, Atlanta, GA, August 1995.

[17] M. Krunz and S. K. Tripathi, "On the characterization of VBR MPEG streams," *ACM SIGMETRICS Performance Evaluation Review*, vol. 25, pp. 192–202, June 1997.

[18] P.R. Chang and J.T. Hu, "Optimal nonlinear adaptive prediction and modeling of MPEG video in ATM networks using pipelined recurrent neural networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 1087–1100, August 1997.

[19] A.D. Doulamis, N.D. Doulamis, and S.D. Kollias, "Nonlinear traffic modeling of VBR MPEG-2 video sources," in *IEEE International Conference on Multimedia and Expo. ICME*, New York, NY, USA, July-August 2000, IEEE, vol. 3, pp. 1318–1321.

[20] D. Liu, E.I. Sára, and W. Sun, "Nested auto-regressive processes for MPEG-encoded video traffic modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 169–183, February 2001.

[21] S.J. Yoo, "Efficient traffic prediction scheme for real-time VBR MPEG video transmission over high-speed networks," *IEEE Transactions on Broadcasting*, vol. 48, no. 1, pp. 10–18, March 2002.

[22] A. Chodorek and R.R. Chodorek, "An MPEG-2 video traffic prediction based on phase space analysis and its application to on-line dynamic bandwidth allocation," in *2nd European Conference on Universal Multiservice Networks. EDUMN*, April 8-10 2002, pp. 44–55.

[23] A. Bhattacharya, A. Parlos, and A. Atiya, "Prediction of MPEG-coded video source traffic using recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2177–2190, 2003.

[24] S. Poretsky, J. Perser, S. Erramilli, and S. Khurana, "Terminology for benchmarking network-layer traffic control mechanisms," IETF RFC 4689 (Informational), October 2006.

[25] R. Ramjee, J. F. Kurose, D. F. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," in *INFOCOM (2)*, Toronto, Canada, 1994, pp. 680–688.

[26] A. P. Markopoulou, F. A. Tobagi, and M. J. Karam, "Assessing the quality of voice communications over Internet backbones," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, October 2003.

[27] ITU-T Rec. P.800, "Methods for subjective determination of transmission quality," http://www.itu.int/rec/T-REC-P.800/en, August 1996.

[28] L. Peterson and B. S. Davie, *Computer Networks:* A *systems approach*, San Francisco, CA: Morgan Kaufmann, second edition, 2002.

[29] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[30] B.E. Carpenter and K. Nichols, "Differentiated services in the Internet," *Proc. of the IEEE*, vol. 90, no. 9, pp. 1479–1494, September 2002.

[31] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, August 1999.

[32] M.-S. Kim, Y. J. Won, and J. W. Hong, "Characteristic analysis of Internet traffic from the perspective of flows," *Computer Communications*, vol. 29, no. 10, pp. 1639–1652, June 2006.

[33] M. Fomenkov, K. Keys, D. Moore, and K. Claffy, "Longitudinal study of Internet traffic in 1998-2003," in *WISICT '04: Proc. of the Winter International Sympo-*

sium on Information and Communication Technologies, Cancun, Mexico, January 2004, ACM, pp. 1–6.

[34] C. Krasic, K. Li, and J. Walpole, "The case for streaming multimedia with TCP," in *Interactive Distributed Multimedia Systems: 8th International Workshop,* IDMS *2001*, Lancaster UK, September 2001, vol. 2158/2001 of *Lecture Notes in Computer Science*, pp. 213–218, Springer Verlag.

[35] J. Martin, A. Nilsson, and I. Rhee, "Delay-based congestion avoidance for TCP," *IEEE Transactions on Networking*, vol. 11, no. 3, pp. 356–369, June 2003.

[36] J. Pinto and K.J. Christensen, "An algorithm for playout of packet voice based on adaptive adjustment of talkspurt silence periods," in *24th Conference on Local Computer Networks*, Lowell, Massachusetts, October 1999, pp. 224–231.

[37] H. Sanneck, A. Stenger, K. Ben Younes, and B. Girod, "A new technique for audio packet loss concealment," in *IEEE GLOBECOM*, London, UK, November 1996, IEEE, pp. 48–52.

[38] Y.J. Liang, N. Färber, and B. Girod, "Adaptive playout scheduling using time-scale modification in packet voice communications," in *ICASSP 2001: IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, 2001, vol. 3, pp. 1445–1448.

[39] M. K. Ranganathan and L. Kilmartin, "Neural and fuzzy computation techniques for playout delay adaptation in VoIP networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1174–1194, September 2005.

[40] D. Sisalem and H. Schulzrinne, "The loss-delay based adjustment algorithm:

A TCP-friendly adaptation scheme," in *Proc. of NOSSDAV*, Cambridge, UK., 1998.

[41] P. DeLeon and C.J. Sreenan, "An adaptive predictor for media playout buffering," in *International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, March 1999, IEEE, vol. 6, pp. 3097–3100.

[42] C. Casetti, J.C. De Martin, and M. Meo, "A framework for the analysis of adaptive voice over IP," in *International Conference on Communications*, New Orleans, USA, June 2000, vol. 2, pp. 821–826.

[43] C. J. Sreenan, J.C. Chen, P. Agrawal, and B. Narendran, "Delay reduction techniques for playout buffering," *IEEE Transactions on Multimedia*, vol. 2, no. 2, pp. 100–112, June 2000.

[44] C.H. Wang, J.M. Ho, R.I. Chang, and S.C. Hsu, "A control-theoretic mechanism for rate-based flow control of real-time multimedia communication," in *Multimedia, Internet, Video Technologies WSES/IEEE International Multi-conference*, September 2001.

[45] F. Beritelli, G. Ruggeri, and G. Schembra, "TCP-friendly transmission of voice over IP," in *ICC 2002: IEEE International Conference on Communications*, Anchorage, Alaska, April–May 2002, vol. 2, pp. 1204–1208.

[46] K. Homayounfar, "Rate adaptive speech coding for universal multimedia access," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 30–39, March 2003.

[47] J. W. Seo, S.J. Woo, and K.S. Bae, "A study on the application of an AMR speech codec to VoIP," in *Proc. IEEE ICASSP*, Salt Lake City, Utah, 2001, vol. 3, pp. 1373–1376.

[48] V. Abreu-Sernandez and C. Garcia-Mateo, “Adaptive multi-rate speech coder for VoIP transmission,” *IEE Electronic Letters*, vol. 36, no. 23, pp. 1978–1980, November 2000.

[49] Y. Bai and M.R. Ito, “QoS control for video and audio communication in conventional and active networks: approaches and comparison,” *IEEE Communication Surveys*, vol. 6, no. 1, pp. 42–49, 2004.

[50] I. Akyildiz, Y. Altunbasak, F. Fekri, and R. Sivakumar, “AdaptNet: An adaptive protocol suite for the next-generation wireless Internet,” *IEEE Communications Magazine*, pp. 128–136, March 2004.

[51] L. Roychoudhury and E.S. Al-Shaer, “Adaptive rate control for real-time packet audio based on loss prediction,” in *GLOBECOM 2004: IEEE Global Telecommunications Conference*, Dallas, Texas, November–December 2004, vol. 2, pp. 634–638.

[52] J.C. Bolot and T. Turletti, “A rate control mechanism for packet video in the Internet,” in *INFOCOM (3)*, Toronto, Canada, 1994, pp. 1216–1223.

[53] J-C. Bolot and T. Turletti, “Experience with control mechanisms for packet video in the Internet,” *Computer Communication Review*, vol. 28, no. 1, pp. 4–15, January 1998.

[54] J.F. Gibbon and T.D.C. Little, “The use of network delay estimation for multimedia data retrieval,” *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1376–1387, September 1996.

[55] S. Jacobs and A. Eleftheriadis, “Real-time dynamic rate shaping and control for Internet video applications,” in *Workshop on Multimedia Signal Processing*,

Princeton. NJ, June 1997, pp. 23–25.

[56] S. Jacobs and A. Eleftheriadis, "Streaming video using dynamic rate shaping and TCP congestion control," *Journal of Visual Communication and Image Representation*, vol. 9, no. 3, pp. 211–222, September 1998.

[57] Y. Chung, J.W. Kim, and J. Kuo, "Network friendly video streaming via adaptive LMS bandwidth control," in *SPIE Proc. of International Symposium on Optical Science, Engineering, and Instrumentation*, A. G. Tescher, Ed., San Diego, CA., October 1998, vol. 3460, pp. 448–456.

[58] R. Rejaie, D. Estrin, and M. Handley, "Quality adaptation for congestion controlled video playback over the internet," in *SIGCOMM*, Cambridge, MA, August-September 1999, pp. 189–200.

[59] H.J. Song, J.W. Kim, and C.-C. Jay Kuo, "Real-time encoding frame rate control for H.263+ video over the Internet," *Signal Processing: Image Communication, vol. 15, Sept. 1999*, vol. 15, September 1999.

[60] H.J. Song and C.C.J. Kuo, "Rate control for low-bit-rate video via variable-encoding frame rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 4, pp. 512–521, April 2001.

[61] H.J. Song and K.M. Lee, "Adaptive rate control algorithms for low bit rate video under networks supporting bandwidth renegotiation," *Signal Processing: Image Communication*, vol. 17, no. 10, pp. 759–779, November 2002.

[62] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the

internet," *IEEE Transactions on Circuits Syst. Video Technol.*, vol. 10, pp. 923–941, September 2000.

[63] D. Wu, Y.T. Hou, B. Li, W. Zhu, Y.Q. Zhang, and H.J. Chao, "An end-to-end approach for optimal mode selection in Internet video communication: theory and application," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 977–995, June 2000.

[64] N.G. Feamster, "Adaptive delivery of real-time streaming video," M.S. thesis, MIT, Cambridge, MA, May 2001.

[65] L.C. Wai, "Sender-adaptive rate control for layered video multicast," M.S. thesis, Hong Kong Polytechnique University, Hung Hom, Kowloon, Hong Kong, May 2001.

[66] X. Lu, R.O. Morando, and M. El Zarki, "Understanding video quality and its use in feedback control," International Packet Video Workshop, Pittsburgh, PA, USA., 2002.

[67] W. C. Chu, *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*, Hoboken, NJ: Wiley Interscience, 2003.

[68] J.-M. Valin, *The Speex Codec Manual*, Xiph.org, 1.2-beta1 edition, August 2006.

[69] S. Andersen, A. Duric, H. Astrom, W. Kleijn, and J. Linden, "RFC 3951: Internet Low Bit Rate Codec (iLBC)," http://www.ietf.org/rfc/rfc3951.txt, December 2004.

[70] ITU-T Rec. G.107, "The E-model, a computational model for use in transmission planning," http://www.itu.int/rec/T-REC-G.107/en, March 2003.

[71] Newport Networks Limited. UK, "VoIP bandwidth calculation," http://www.newport-networks.com/cust-docs/52-VoIP-Bandwidth.pdf, December 2006.

[72] B. Goode, "Voice over Internet Protocol," *Proc. of the IEEE*, vol. 90, no. 9, pp. 1485–1517, September 2002.

[73] R. G. Cole and J. H. Rosenbluth, "Voice over IP performance monitoring," *Computer Communication Review*, vol. 31, no. 2, pp. 9–24, April 2001.

[74] L. Ding and R. Goubran, "Speech quality prediction in VoIP using the extended E-Model," in *IEEE GLOBECOM*, San Francisco, CA, December 2003, vol. 7, pp. 3974–3978.

[75] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z. Zhang, "Improving VoIP quality through path switching," in *IEEE INFOCOM 2005: 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, Florida, March 2005, vol. 4, pp. 2268–2278.

[76] Lingfen Sun and Emmanuel C. Ifeachor, "Voice quality prediction models and their application in VoIP networks," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 809–820, August 2006.

[77] D. Ye, "Control of real-time multimedia applications in best-effort networks," Ph.D. dissertation, Texas A&M University, College Station, TX, 2006.

[78] ANSI T1.521-1999, "Packet loss concealment for use with ITU-T recommendation G.711," December 1999.

[79] "PLANETLAB: An open platform for developing, deploying, and accessing planetary-scale services," http://www.planet-lab.org.

[80] C. Dovrolis and R. Prasad, “Pathrate: A measurement tool for the capacity of network paths,” http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/pathrate .html.

[81] K. Park and W. Willinger, Eds., *Self-Similar Network Traffic and Performance Evaluation*, New York: John Wiley and Sons Inc., 2000.

[82] L. Ljung, *System Identification: Theory for the User*, Prentice Hall Information and System Sciences. Upper Saddle River, NJ: Prentice Hall PTR, second edition, 1999.

[83] H. Ohsaki, M. Morita, and M. Murata, “On modeling round-trip time dynamics of the Internet using system identification,” in *ICOIN ’02: Revised Papers from the International Conference on Information Networking, Wireless Communications Technologies and Network Applications-Part I*, London, UK, 2002, pp. 359–371, Springer-Verlag.

[84] E. F. Camacho and C. Bordons, *Model Predictive Control*, Advanced Textbooks in Control and Signal Processing. London, UK: Springer-Verlag, 1998.

VITA

Aninda Bhattacharya received his Bachelor of Engineering (BE) degree in mechanical engineering from Sardar Vallabhbhai National Institute of Technology, Surat, India, in 1997. He worked as a Senior Software Engineer for two years in Tata Infotech Limited, SEEPZ, Mumbai, India from 1997 - 1999. He joined Texas A&M University, College Station, TX, USA as a graduate student in the fall of 1999. Aninda obtained his Master of Science (MS) degree in mechanical engineering in 2002. He received his PhD in 2007 from the department of mechanical engineering at the same university. He will be joining GE John F. Welch Technology Center, Bangalore, India as a "Research Engineer" from August 2007.

The typist for this thesis was Aninda Bhattacharya.