Estimating Buffer Size Using Positive Normal Distribution in Wireless Networks

Thesis submitted in partial fulfillment of the requirement for

the award of the degree of

Masters of Science

in

Mathematics and Computing

Submitted by

Richa Gupta

Roll No. – 300903020

Under

the guidance of

Dr. R. K. Sharma

JULY 2011

School of Mathematics and Computer Applications

Thapar University

Patiala – 147004(PUNJAB)

INDIA

Contents

Abstract

This thesis presents the use of positive normal distribution to estimate the minimum buffer size by estimating Cumulative distribution function of available bandwidth and streaming rate by using enhanced bandwidth estimation technique WBest for wireless networks. This has been implemented by using MATLAB environment. Evaluation and experimentation show that the simulation of BROS algorithm using positive normal distribution is more effective than using normal distribution for estimations in wireless networks. The algorithm which has been implemented in MATLAB is used by varying different parameters. Thus by using different parameters, namely, MTBBU, streaming rate, buffer size we have generated different results that seems to be effective in estimating buffer size which can further improve the performance of streaming multimedia applications in wireless networks.

Certificate

This is to certify that the thesis entitled "Estimating Buffer Size Using Positive Normal Distribution in Wireless Networks" submitted to the School of Mathematics and Computer Applications, Thapar University, Patiala for the award of the Degree of Masters of Mathematics and Computing is a bonafide record of my own work carried out under the supervision of Dr. R.K. Sharma. The contents of this thesis have not been submitted for the award of any other degree to this or any other university.

Richa Gupta

(Richa Gupta)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

13.7.11

(Dr. R.K. Sharma)

(Professor)

Thapar University

Patiala

Countersigned by:

Dr. S.S. Bhatia

(Professor and Head)

SMCA, Thapar University,

Patiala

Dr. S.K. Mohapatra

(Dean of Academic Affairs)

Thapar University,

Patiala

Acknowledgement

I have taken efforts in this thesis. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks and gratitude to all of them.

First and foremost, I am highly indebted to Dr. R.K. Sharma, for his guidance and constant supervision as well as for providing necessary information regarding the thesis work and also for his support in completing this thesis. I feel very much blessed to have such a wise, knowledgeable and kind person as my mentor.

I express my regards and gratitude to Dr. S. S. Bhatia, Head of Department School of Mathematics and Computer Applications, Thapar University, Patiala for his unfailing support, inspiration and necessary research facilities in the school.

I would also thank all the academic and administrative staff of School of Mathematics and Computer Applications, Thapar University, Patiala.

I would like to express my gratitude towards my parents & family for their kind co-operation and encouragement which helped me in completion of this project.

My thanks and appreciations also go to my friends and classmates in developing this thesis and people who have willingly helped me out with their abilities.

List of abbreviations

ACK: Acknowledgement

AP: Access Point

ARQ: Automatic Repeat Request

BROS: Buffer and Rate Optimization for Streaming

BTC: Bulk-Transfer Capacity

CDF: Cumulative Distribution Function

FEC: Forward Error Correction

FIFO: First In First Out

FTP: File Transfer Protocol

IEEE: Institute of Electrical and Electronics Engineers

IP: Internet Protocol

LAN: Local Area Network

MAC: Medium Access Control

MPEG: Motion Picture Expert Group

MTBBU: Mean Time between Buffer Underflows

PDF: Probability Distribution Function

PGM: Probe Gap Model

PTR: Packet Transmission Rate

RTT: Round Trip Time

SLOPS: Self-loading Periodic Streams

TCP: Transmission Control Protocol

TFRC: TCP Friendly Rate Control

TOPP: Train of Packet Pairs

UDP: User Datagram Protocol

UEP: Unequal Error Protection

VPS: Variable Packet Size Probing

WBest: Wireless Bandwidth Estimation Tool

WLAN: Wireless Local Area Network

WMAN: Wireless Metropolitan Area Network

WPAN: Wireless Personal Area Network

WWAN: Wireless Wide Area Network

List of Figures

Chapter – 1

Introduction

Prices of wireless LAN access points (AP) have decreased significantly in previous years. Also, wireless link capacities have significantly been increased during this period. As a result of these changes, number of wireless networks has been increased significantly. It has been noted that wireless capacity has been increased up to 54 Mbps capacity. At this capacity, we can seamlessly run applications such as streaming video from wired connected servers to wireless connected clients. It has also been noted that many streaming techniques are primarily designed for wired networks, streaming media applications may perform poorly when these are used in a hybrid network environment where the clients are connected via a wireless network.

In wired networks, streaming media quality is affected by packet delay, jitter and packet loss due to network congestion. To mitigate the impact of network congestion, various techniques have been used to improve streaming media quality, such as initial capacity estimation, media scaling and playout buffer optimization. Even though these techniques can reduce the degradation in quality when streaming over wireless networks, they do not perform as effectively as in wired networks. Recent research shows that the wireless network conditions, such as the wireless link layer rate adaptation, contention, and interference can significantly degrade the performance of streaming media applications by incurring re-buffer events and degraded perceptual quality [Kuang and Williamson 2004, Li et al. 2005a, 2005b, Bai and Williamson 2004]. This mismatch in design leads to significant degradation in the effectiveness of rate selection and playout buffer techniques.

Li et al. [2006] presented the evaluation of application layer solution for improving streaming multimedia application performance in IEEE 802.11 wireless networks by using a Buffer and Rate Optimization for Streaming (BROS) algorithm to guide the streaming rate selection and initial buffer optimization. Their evaluation shows that BROS algorithm selects the proper streaming rate and initial buffer size based on the bandwidth estimation done by Wireless Bandwidth Estimation Tool (WBest). By taking the available bandwidth distribution into consideration, streaming rate and buffer optimizer module selects the proper streaming rate and buffer size to mitigate

Chapter – 1

Introduction

Prices of wireless LAN access points (AP) have decreased significantly in previous years. Also, wireless link capacities have significantly been increased during this period. As a result of these changes, number of wireless networks has been increased significantly. It has been noted that wireless capacity has been increased up to 54 Mbps capacity. At this capacity, we can seamlessly run applications such as streaming video from wired connected servers to wireless connected clients. It has also been noted that many streaming techniques are primarily designed for wired networks, streaming media applications may perform poorly when these are used in a hybrid network environment where the clients are connected via a wireless network.

In wired networks, streaming media quality is affected by packet delay, jitter and packet loss due to network congestion. To mitigate the impact of network congestion, various techniques have been used to improve streaming media quality, such as initial capacity estimation, media scaling and playout buffer optimization. Even though these techniques can reduce the degradation in quality when streaming over wireless networks, they do not perform as effectively as in wired networks. Recent research shows that the wireless network conditions, such as the wireless link layer rate adaptation, contention, and interference can significantly degrade the performance of streaming media applications by incurring re-buffer events and degraded perceptual quality [Kuang and Williamson 2004, Li et al. 2005a, 2005b, Bai and Williamson 2004]. This mismatch in design leads to significant degradation in the effectiveness of rate selection and playout buffer techniques.

Li et al. [2006] presented the evaluation of application layer solution for improving streaming multimedia application performance in IEEE 802.11 wireless networks by using a Buffer and Rate Optimization for Streaming (BROS) algorithm to guide the streaming rate selection and initial buffer optimization. Their evaluation shows that BROS algorithm selects the proper streaming rate and initial buffer size based on the bandwidth estimation done by Wireless Bandwidth Estimation Tool (WBest). By taking the available bandwidth distribution into consideration, streaming rate and buffer optimizer module selects the proper streaming rate and buffer size to mitigate the probability of buffer underflow events. The advantage of the BROS algorithm over existing buffer optimizing approaches is that it takes the variation of available bandwidth into consideration, which usually has a greater impact on streaming performance than delay jitter in wireless networks. Their evaluation also shows that BROS can effectively select the best streaming rate and optimize the initial buffer size based on wireless network bandwidth conditions, thus achieving fewer buffer underflow events and lower initial delay than buffers based on static rate selection, static sizing, and jitter removal.

In their work, Li et al. [2006] used WBest algorithm to estimate the mean and standard deviation of available bandwidth. This was further used in BROS algorithm. Li et al. [2006] have considered that available bandwidth follows normal distribution with some mean and some standard deviation. In the present work, we have carried out the simulation study after taking positive normal distribution instead of normal distribution. These simulation experiments suggest that positive normal distribution is a better choice for deciding the distribution of available bandwidth in order to optimize the buffer size, than the normal distribution.

Chapter - 2
Literature Review

This chapter reviews the research work related to the work in this thesis. Three research areas covered in this review are: Wireless Networks, Streaming Rate Selection and Playout Buffer Techniques.

2.1 Wireless Networks

Wireless networks had been widely deployed over the last few decades. Most of the protocols and applications that were developed for wired networks have been transferred to wireless networks for their actual implementations. However, the wireless characteristics that differ from wired networks may affect the performance of these applications in wireless networks. To understand these characteristics of wireless networks, this section provides a general review of the wireless networks and IEEE 802.11 Wireless Networks.

The most important characteristics of wireless radio medium that differ from the wired network are as follows [Pahlavan and Krishnamurthy 2002].

- Shared Medium
- Propagation
- Bursty channel errors
- Location dependent carrier sensing

These wireless characteristics may degrade the wireless network performance extensively. Therefore, most of the wireless network standards implement a variety of error recover mechanisms, such as the Forward Error Correction (FEC), Automatic Request for retransmission (ARQ) and rate adaptation.

2.1.1 Wireless Network Categorization

One of the ways to categorize wireless data communication networks is based on the coverage range [Intel Corporation 2004]. This categorization is briefly presented below.

- **<u>Wireless Personal Area Networks (WPANs)</u>**

  WPANs are small networks operating within a confined space, such as an office workspace or room within the home. The coverage range is usually less than 30 feet. For example, Bluetooth can provide up to 720 Kbps capacity over less than 30 feet distance. Ultra Wideband (UWB), which is still under development, is proposed to be designed to provide up to 480 Mbps throughput over a short distance.

- **<u>Wireless Local Area Networks (WLANs)</u>**

  WLANs have broader range than WPANs, typically confined within office buildings, restaurants, stores, homes, etc. WLANs have become the most popular wireless data communication techniques as the production of the WLAN standards, such as IEEE 802.11 standard family.

- **<u>Wireless Metropolitan Area Networks (WMANs)</u>**

  WMANs cover a much greater distance than WLANs, connecting buildings to one another over a broader geographic area. For example, the emerging Wi-MAX technology will further enable mobility and reduce reliance on wired connections. Typical WMANs have a throughput up to 10-20 Mbps and cover a distance of approximately several miles.

- **<u>Wireless Wide Area Networks (WWANs)</u>**

  WWANs have the broadest coverage range and are most widely deployed today in the cellular voice infrastructure to provide the capability of transmitting data. The most popular WWAN techniques include the currently available cellular 2.5G (Generation) data services, such as General Packet Radio Service (GPRS) and Enhanced Data Rates for Global Evolution (EDGE), and the next-generation cellular services based on various 3G technologies.

Out of these wireless network techniques, WLANs are the most widely deployed wireless networks that are being used for streaming multimedia applications. The IEEE 802.11 Standard defines a family of Wireless Local Area Networks (WLANs), including 802.11b, 802.11a, 802.11g, etc. All the standards use the same MAC layer specification, but different physical layer specifications [IEEE committee 1999]. Therefore, the research focuses only on WLANs. WLANs implement a highly reliable MAC/link layer by using retransmission, error correction, or link adaptation techniques to reduce the impacts caused by the high loss rate, high dynamic physical layer conditions. These techniques provide the wireless network with better performance for traditional Internet applications, such as Web, Email service and FTP service. However, these techniques may affect rate based or time sensitive applications, such as streaming multimedia and interactive Internet telephone applications.

2.2 Streaming Rate Selection

It has been shown that the media scaling performance is limited when the optimal streaming rate is not correctly selected over wireless networks [Kuang and Williamson 2004, Li et al. 2005a, Bai and Williamson 2004, Cen et al. 2003]. The network effects on media scaling can be reduced by adjusting streaming data rate by estimating bandwidth or it can be reduced by typical streaming rate selection which is used in media scaling and is based on loss rate and round-trip time. But wireless network conditions and target rates cannot be directly provided by the above measured factors which are affecting media performance.

The streaming performance for wireless networks can be improved by different approaches. These approaches include:

- Novel transport layer protocols
- Cross-layer approaches
- Bandwidth estimation approaches

The review of the above related approaches is given below.

2.2.1 Novel Transport Protocols

In wireless networks, wireless loss differentiation techniques are used to improve the performance of TCP friendly protocols [Cen et al. 2003]. This technique gives a streaming application a right to select the proper rate for observed wireless network conditions according to the TCP friendly rate. New transport protocols were also proposed to control the streaming rate in wireless network using TCP friendly rate by Chen and Zakhor [2005] and [Yang et al. 2004]. To increase the utilization of wireless networks Chen and Zakhor [2005] also proposed MULTFRC which builds multiple TFRC connections. Many other researchers have contributed in increasing the streaming performance by decreasing the network delay factor. It has also been noted that “Trial and Error” scaling technique cannot be used for initial streaming rate selection as this technique is unreliable and converges slowly [Kazantzidis and Gerla 2003].

2.2.2 Cross Layer Approaches

Rate adaption, amount of forward error corrections and retransmissions is the information which is provided by MAC and physical layer information. This information is used by Cross-layer approaches to control the selection of the streaming rate. Link network feedback architecture is proposed to provide cross layer information like streaming rate for media applications by Kazantzidis and Gerla [2005]. To improve the TCP friendly rate control, Yang et al. [2004] proposed a new protocol that utilizes link layer loss information by combining the cross layer and TCP friendly rate control approach. Many other crossing layer approaches were proposed to provide Unequal Error Protection (UEP) for streaming traffic. As in Li et al. [2004], application layer information like frame type is made available to link layers. Therefore, these approaches can improve the quality of streaming video over wireless networks by applying different protection to different frame types. This protection is applied to reduce the packet loss and delay on important frames, such as I-frames. However, cross layer approaches require modification to end hosts and to protocol stacks, but this may need multiple vendor implementations which are difficult to deploy.

2.2.3 Bandwidth Estimation Approaches

The rate selection can be evaluated by using bandwidth estimation approaches which use various application measurements. Packet pair techniques are used by various streaming media applications such as windows media service to estimate the capacity and choosing a appropriate streaming rate [Birney 2004]. In wireless networks, packet pair or receiver side statistical bandwidth estimate techniques are used by Beek et al. [2004, 2005] to guide the rate selection.

Various advantages of this type of approaches are:

- It doesn't depend upon lower layer information or new protocol stacks as in the case of cross layer approaches.
- It can also avoid the "Trial and Error" problem caused by a TCP friendly approach for selecting streaming rate.

There are some disadvantages as well of these approaches:

- Traditional bandwidth techniques which are designed for wired networks were not able to estimate the bandwidth accurately for wireless networks. [Li et al. 2006, LakshmiNaraynan et al. 2004 and Angrisani et al. 2006].
- Also these techniques only provide the capacity estimation information while streaming rate selection also needs the other bandwidth information such as available bandwidth and variance in available bandwidth.

2.3 Playout Buffer Techniques

Buffer techniques are used on the server side, network (caching and proxy) side and on the client side to get better results for streaming multimedia over best effort networks such as the Internet and wireless networks. Client side buffering techniques play an important role in streaming multimedia as it removes the jitter effects and playback disruption caused by oscillations in the transmission rate which are caused by transport protocols such as TCP and TFRC at the cost of initial startup delay [Birney 2004, Li et al. 2001].

A number of strategies have been proposed to improve the effect of client side buffering:

- To prevent buffer underflow and to reduce the consumption rate, media playout rate is decreased at the client side [Yuang et al. 1996, Girod et al. 2001, Laoutaris and Stavrakakis 2001, Kalman et al. 2004].
- Various media scaling techniques are proposed.
- By optimizing the buffer size based on jitter removal [Mundur et al. 1999, Ramjee et al. 1994, Moon et al. 1998, and Fujimoto et al. 2002]. But jitter removal buffer algorithm is not sufficient to avoid buffer underflow because

the arrival of streaming traffic cannot be modeled as Poisson distribution as the change in capacity causes a variance in the streaming traffic arrival rate.

The buffer optimization approach proposed by Li et al. [2006] is based on estimation of the network conditions. It not only considers TCP-Friendly rate but can also be applied to some environments where the buffer underflow is not caused by transmission rate changes.

Chapter – 3
Bandwidth Estimation

If we have more information, we can estimate better bandwidth available for a connection. This estimation helps in a better and fairer utilization of network resources [Prasad et al. 2003]. This is the principle followed by several bandwidth estimation techniques**.** Bandwidth estimation refers to the end-to-end measurement of bandwidth-related metrics, such as capacity, available bandwidth and bulk TCP transfer capacity, performed by the end hosts of a path without requiring administrative access to intermediate routers along the path. There are different bandwidth-related metrics which are important as different applications need different information regarding bandwidth.

3.1 Bandwidth Related Metrics

Three bandwidth metrics that are defined for different aspects of bandwidth are: Capacity, Available Bandwidth, and TCP throughput and Bulk-Transfer Capacity (BTC) [Prasad et al. 2003]. The first two are defined both for individual links and end-to-end paths, while the BTC is usually defined only for an end-to-end path.

3.1.1 Capacity

The capacity of a link can be defined both for individual links and end-to-end paths. The transmission rate or the capacity of the segment is the rate at which a network segment can normally transfer data. The capacity C of a path is the maximum possible IP layer rate the path can transfer from the source to the receiver. The hop with the minimum capacity is called the narrow-link of the path. If C_h is the transfer rate at hop h,

Then the capacity of the path will be

$$C = min\{C_h\}, \quad h = 1,2,\dots,H \tag{3.1}$$

where the hop capacities are assumed to be constant.

At the IP layer, each hop delivers data at a rate lower than its nominal transmission rate due to the overhead of link layer encapsulation and framing. Prasad et al. [2003] defined the IP layer capacity as follows:

$$C_{L3} = C_{L2} \frac{1}{1+\frac{H_{L2}}{H_{L3}}} \quad (3.2)$$

where C_{L3} is the IP layer capacity, C_{L2} is the link layer capacity, H_{L2} is the total link layer overhead, and L_{L3} is the size of an IP packet. The capacity definitions in equations 3.1 and 3.2 can only be used for those techniques in which the capacity remains constant during the time intervals.

3.1.2 Available Bandwidth

Another important metric is the available bandwidth of a link or end-to-end path. The available bandwidth of a link relates to the unused, or "spare", capacity of the link during a certain time period. So even though the capacity of a link depends on the underlying transmission technology and propagation medium, the available bandwidth of a link additionally depends on the traffic load at that link, and is typically a time-varying metric [Prasad et al. 2003].

If C_i is the capacity of hop i and u_i is the average utilization of that hop in the given time interval, the average available bandwidth A_i of hop i is given by the unutilized fraction of capacity as follows:

$$A_i = (1 - u_i)C_i \quad (3.3)$$

Extending the previous definition to an H-hop path, the available bandwidth of the end-to-end path is the minimum available bandwidth of all H-hops,

$$\mathrm{A} = \min\{\mathrm{A_h}\}, \ \mathrm{h} = 1,2, \ldots, \mathrm{H} \quad (3.4)$$

The hop with the minimum available bandwidth is called the tight link of the end-to-end path.

3.1.3 TCP Throughput and Bulk Transfer Capacity (BTC)

Another key bandwidth-related metric in TCP/IP networks is the throughput of a TCP connection. TCP is the major transport protocol in the Internet, carrying almost 90% of the traffic. But it is not easy to define the expected throughput of a TCP connection. Several factors which are affecting TCP throughput are transfer size, type of cross traffic (UDP or TCP), number of competing TCP connections, TCP socket buffer sizes at both sender and receiver sides, congestion along reverse (ACK) path, as well as size of router buffers and capacity and load of each link in the network path.

The throughput of a small transfer such as a typical Web page depends on the initial congestion window, Round-Trip Time (RTT), and slow-start mechanism of TCP, rather than on available bandwidth of the path. The throughput of a large TCP transfer over a certain network path can vary significantly when using different versions of TCP even if the available bandwidth is the same.

The Bulk-Transfer-Capacity (BTC) [Prasad et al. 2003] defines a metric that represents the achievable throughput by a TCP connection. BTC is the maximum throughput obtainable by a single TCP connection.

The BTC of an end-to-end path is defined as:

$$BTC = \frac{data\ sent}{elapsed\ time} \tag{3.5}$$

where data sent represents the unique data bits transferred, which does not include header bits or emulated header bits or retransmitted data, and elapsed time is the measurement interval.

Also there is a noticeable difference between BTC and available bandwidth metrics. BTC is TCP-specific whereas the available bandwidth metric does not depend on a specific transport protocol. The BTC depends on how TCP shares bandwidth with other TCP flows, while the available bandwidth metric assumes that the average traffic load remains constant and estimates the additional bandwidth that a path can offer before its tight link is saturated.

The general relationship between the terminologies used as bandwidth related metrics can be presented as:

$$BTC < Available\ Bandwidth \leq Achievable\ Throughput < Capacity$$

Figure 3.1 illustrates the relationship with the presence of crossing traffic. The capacity is the maximum possible bandwidth that includes the volume used by the crossing traffic. The achievable throughput could get a higher volume than the Available Bandwidth, which is the Capacity−Crossing Traffic. The BTC only considers TCP traffic, which will take the congestion control into consideration.

Therefore, the BTC volume is likely to be smaller than Available Bandwidth since it needs to maintain a TCP-Friendly share with the crossing traffic in the network.

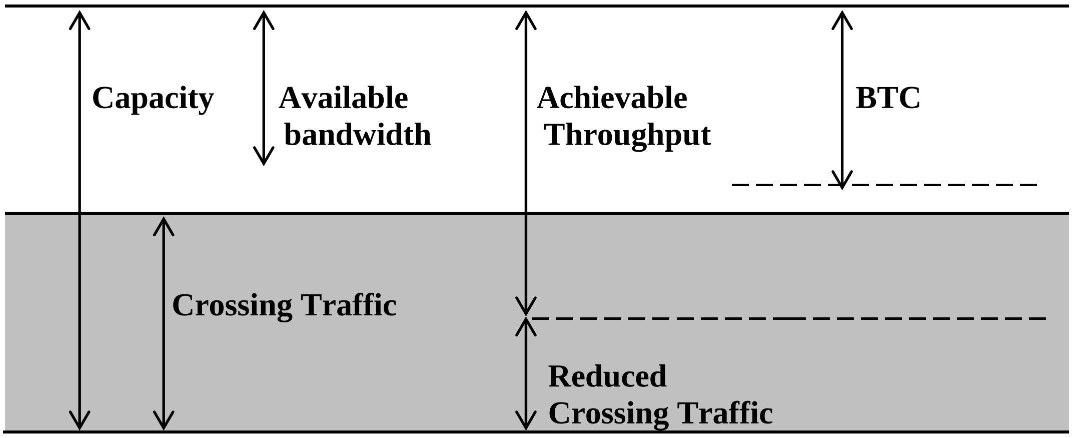


Figure 3.1 Bandwidth Related Terminology

3.2 Bandwidth Estimation Techniques

This section describes existing bandwidth measurement techniques for estimating capacity and available bandwidth in individual hops and end-to-end paths [Prasad et al. 2003]. There are four major techniques: Variable Packet Size (VPS) probing, Packet Dispersion, Self-Loading Probing and Probe Gap Model.

3.2.1 Variable Packet Size (VPS) Probing

VPS probing aims to measure the capacity of each hop along a path. Bellovin [1992] and Jacobson [1997] were the first to propose and explore the VPS methodology. The key element of the technique is to measure the RTT from the source to each hop of the path as a function of the probing packet size. The RTT to each hop consists of three delay components in the forward and reverse paths, namely, serialization delays, propagation delays, and queuing delays.

- VPS sends multiple probing packets of a given size from the sending host to each layer-3 device along the path.
- The technique assumes that at least one of these packets, together with the ICMP reply that it generates, will not encounter any queuing delays.

Therefore the minimum RTT that is measured for each packet size will consist of two terms:-

- A Delay that is independent of packet size and is mostly due to propagation delays.
- A term proportional to the packet size due to serialization delays at each link along the packet's path.

Specifically, the minimum RTT T_i *(L)* for a given packet size *L* up to hop *i* is represented as [Prasad et al. 2003]:

$$T_i(L) = \alpha + \sum_{k=1}^{i} \frac{L}{C_k} = \alpha + \beta_i L \tag{3.6}$$

where:

- α denotes the delays up to hop *i* that do not depend on the probing packet size *L*,
- C_k is the capacity of K^{th} hop,
- β_i is the slope of minimum RTT up to hop *i* against probing packet size, given by

$$\beta_i = \sum_{k=1}^{i} \frac{1}{C_k} \tag{3.7}$$

Therefore, by computing the serialization latency at each hop, the capacity of each hop *i* can be estimated as:

$$C_i = \frac{1}{\beta_i - \beta_{i-1}} \tag{3.8}$$

The VPS model has some advantages compared to other related bandwidth estimation techniques.

- VPS is able to measure the network capacity in an uncooperative environment, meaning it does not need special software on both the source and destination.
- The VPS technique can measure the entire network path at each hop along the path.
- VPS sends a large number of probing packets and records the minimum traversal times, so it can mitigate the effects caused by crossing traffic.

The important limitations of VPS model are:

- Most of VPS tools rely on a functional ICMP implementation at each router along the measured network path.

- Second, this technique measures bandwidth in a single direction, from the local host to the remote end host.
- The VPS tools may yield significant capacity underestimation errors if the measured path includes store-and-forward layer-2 switches.

3.2.2: Packet Dispersion

Packet pair probing is used to measure the end-to-end capacity of a path. The source sends multiple packet pairs to the receiver. Each packet pair consists of two packets of the same size which are sent back-to-back. The dispersion of a packet pair at a specific link of the path is the time distance between the last bit of each packet. Packet pair techniques originate from work done by Jacobson [1988], Keshav [1991], and Bolot [1993].

Figure 3.2 shows the dispersion of a packet pair before and after the packet pair goes through a link of capacity C_i assuming that the link does not carry other traffic.

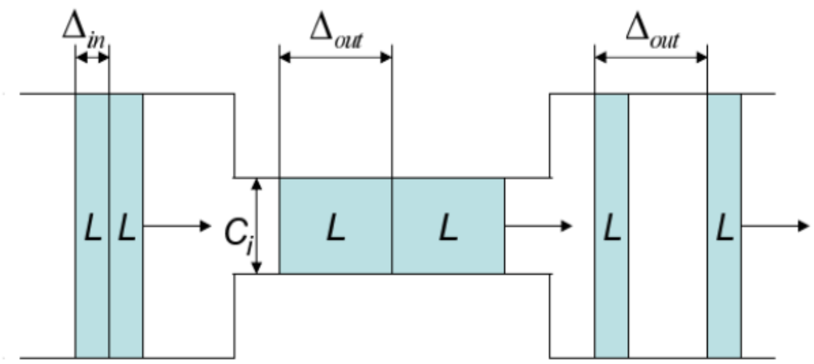


Figure 3.2 Packet Dispersion

If a link of capacity C_0 connects the source to the path and the probing packets are of size L, the dispersion of the packet pair at that first link is

$$\Delta_0 = \frac{L}{C_0} \tag{3.9}$$

In general if the dispersion prior to a link of capacity C_i is Δ_{in}, the dispersion after the link will be

$$\Delta_{out} = max\left(\Delta_{in}, \frac{L}{C_i}\right) \tag{3.10}$$

After packets go through each link along an *H* hop end-to-end path, the final dispersion Δ_R at the receiver is:

$$\Delta_{\mathrm{R}} = \max_{\mathrm{i}=1,\ldots\ldots,\mathrm{H}} \left(\frac{\mathrm{L}}{\mathrm{C_i}}\right) = \frac{\mathrm{L}}{\min_{\mathrm{i}=1,\ldots\ldots\ldots,\mathrm{H}} \mathrm{C_i}} = \frac{\mathrm{L}}{\mathrm{C}} \qquad (3.11)$$

where *C* is the end-to-end capacity. Therefore, the end-to-end path capacity can be estimated from

$$C = \frac{L}{\Delta_R} \qquad (3.12)$$

Packet Dispersion Technique has an advantage as compared to other bandwidth estimation techniques.

- Packet dispersion technique usually has a faster measurement time and induces less stress on the network path.

The limitations of Packet Dispersion Technique are:

- The effects caused by crossing traffic may significantly degrade the accuracy of the link capacity measurement [Prasad et al. 2003]. Several statistical filtering methodologies are proposed to mitigate the effects caused by crossing traffic.
- Another disadvantage of packet dispersion technique is that it requires the tool to be executed on both end-hosts of the network path, which makes it hard to apply to uncooperative environments.

Packet dispersion techniques have been used in some commercial applications. For instance, Windows Media Service uses a packet train probing of three packets to estimate the end-to-end capacity before beginning the streaming from server to client.

3.2.3 Self-Loading Probing

Self-loading techniques measure the available bandwidth of the end-to-end network path which includes the Self-loading Periodic Streams (SLoPS) [Jain and Dovrolis 2003] and Train of Packet Pairs (TOPP) [Melander et al. 2000, 2002]. Self-loading techniques are also known as self-induced congestion [Ribeiro et al. 2003]. There are several tools that implement a variety of self-Loading techniques, such as pathload [Jain and Dovrolis 2003], Packet Transmission Rate (PTR) and pathChirp [Ribeiro et al. 2003]. Self-loading probe techniques are also used to locate the bottleneck in an end-to-end network path with the help of hop by hop delay measurements. Self-loading techniques probe the end-to-end network path using multiple rate traffic.

- If the probing rate is grater than available bandwidth, the probing packets become queued at the tight link router, which results in an increased delay on the receiver side.
- If the probing rate is lower than available bandwidth at the tight link, the probing packets will go through the tight link without causing an increased delay.

By analyzing the packet delay at the receiver, the available bandwidth at the tight link can be obtained at the turning point probing rate, at which the queuing delay starts increasing. The change in the probing rate can be managed in different ways. For example, SLoPS [Jain and Dovrolis 2003] uses a binary search to adjust the probing rate, TOPP [Melander et al. 2000, 2002] use a linearly increased probing rate, while pathchirp [Ribeiro et al. 2003] uses an exponentially increased probing rate.

Most of the self-loading tools can detect a change in the available bandwidth by reporting a grey region during measurement [Jain and Dovrolis 2003]. The self-loading technique may stress the network path due to self-induced congestion. Depending on the implementation, the probing may take a long time to detect the available bandwidth.

3.2.4 Probe Gap Model

The Probe Gap Model (PGM) uses a concept similar to packet dispersion probing. PGM measures the available bandwidth by estimating the cross traffic at the tight link.

PGM assumes a single bottleneck which is both the narrow and tight link for that path and also the queue is not empty between the two packets in a probing packet pair. As shown in Figure 3.3 [Strauss et al. 2003], a probing pair is sent with initial time gap time Δ_{in}, and reaches the receiver with a receiving time gap Δ_{out}. Δ_{out} is the time taken by the bottleneck to transmit the second probing packet in the pair and the crossing traffic that arrived during Δ_{in}.

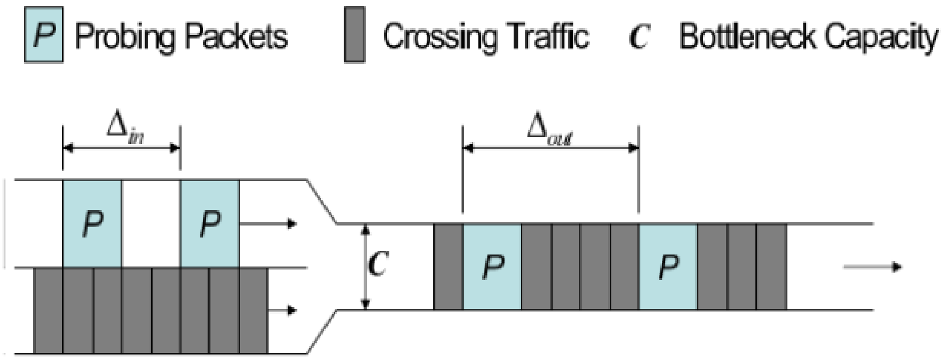


Figure 3.3 Probe Gap Model

Therefore, the time spend on transmitting the crossing traffic at the bottleneck is $\Delta_{out} - \Delta_{in}$. If the bottleneck capacity C is known, the rate of the crossing traffic R_C can be presented as [Strauss et al. 2003]:

$$R_C = \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}} * C \tag{3.13}$$

And the available bandwidth A can be computed as:

$$A = C * \left(1 - \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}}\right) \tag{3.14}$$

PGM usually has a fast measurement time and lower stress on the end-to-end network when compared with self-loading probe. PGM assumes that the narrow link capacity is known and constant but this is not true in case of wireless networks. Therefore, the unknown-capacity problem limits the usage of PGM in an uncooperative environment.

3.3 Wireless Bandwidth Estimation Tool

Most available bandwidth estimation techniques are designed to provide accurate bandwidth information for wired networks at the cost of long convergence times and high intrusiveness. Various researches have been proposed to improve the bandwidth estimation in wireless networks. ProbeGap estimates the fraction of time that a link is idle by probing for gaps in the one-way packet delays, and then use this idle fraction time to estimate the available bandwidth in broadband access networks including IEEE 802.11 networks [Lakshminarayanan 2004]. But ProbeGap does not provide capacity estimation and have to use capacity estimation tools, such as pathrate. DietTOPP uses a reduced TOPP [Melander et al. 2002] algorithm with a modified search algorithm to determine available bandwidth in wireless networks. ProbeGap and DietTOPP techniques have improved the accuracy of available bandwidth estimation in wireless networks, but they did not consider the improvement of convergence time and intrusiveness.

Due to various issues like inaccurate results, high intrusiveness and long convergence time, existing bandwidth estimation mechanisms were difficult to be applied to the applications, such as multimedia streaming, over wireless networks and this difficulty leads to the development of the Wireless Bandwidth Estimation tool (WBest)[Li et al. 2008]. WBest employs a two-step algorithm to determine available bandwidth. In the first step, a packet pair technique estimates the effective capacity of the wireless network. In the second step a packet train scheme determines achievable throughput and infers available bandwidth.

WBest applies a two-step algorithm to estimate both effective capacity and available bandwidth.

- In the first step, *n* packet pairs are sent to estimate the effective capacity C_e. Effective capacity, the maximum capability of the wireless network to deliver network layer traffic, is a function of time and the packet size:

  $$C_e = \frac{\int_{t_0}^{t_1} \frac{L}{T(t)} dt}{t_1 - t_0} \tag{3.15}$$

  where L is the packet size, and $T(t)$ is the packet dispersion at time t. To use packet dispersion in a discrete environment, T_i, the i^{th} packet dispersion at time t, is used to represent $T(t)$.

  To minimize the impact of crossing and contending traffic and capture the impact of rate adaptation on measurements of effective capacity, the median of the *n* packet pair capacity estimates is used to approximate C_e in the estimation time period:

  $$C_e = median(C_i), \quad i = 1,2, \ldots, n \tag{3.16}$$

  where C_i is the estimation result of packet pair i and $C_i = L/T_i$
- For the second step of the WBest algorithm, a packet train of length *m* is sent at rate C_e to estimate available bandwidth A:

$$A = C_e \left(2 - \frac{C_e}{R}\right) = 2C_e - \frac{C_e^2}{R} \qquad (3.17)$$

where R is the average dispersion rate of the probing traffic at the receiver.

Therefore, WBest is a new bandwidth estimation tool for wireless networks, designed to provide accurate bandwidth estimation in a small amount of time. One advantage of WBest is that it does not depend upon search algorithms to measure available bandwidth. WBest provides fast available bandwidth estimation, with more accurate estimations and lower intrusiveness when compared with other tools So, WBest is applied to multimedia streaming applications to improve the performance of streaming rate selection and buffer optimization in wireless networks.

Chapter – 4
Buffer and Rate Optimization

This chapter discusses the Buffer and Rate Optimization for Streaming (BROS) model that is used to improve streaming performance. BROS uses Wireless Bandwidth Estimation Tool (WBest) as discussed in Chapter 3 and models the relationship between buffer size, streaming data rate and available bandwidth distribution. BROS optimizes the streaming data rate and initial buffer size, which results in reduced frame losses and buffer underflow events. Streaming rate selection and playout buffers are used to improve the streaming performance by streaming media applications. But most of the streaming rate selection and buffer optimization algorithms are developed for wired networks. Their performance gets degraded when these are applied to wireless networks.

Based on a Markov Chain model of the buffer size, streaming data rate and available bandwidth distribution, Li et al. [2009] developed the BROS algorithm to optimize the streaming rate selection and the playout buffer size. BROS uses WBest, a low-cost bandwidth estimation approach, at the application layer to provide bandwidth information for the bottleneck wireless network. The evaluation shows that BROS can optimize the streaming rate and initial buffer size based on wireless network bandwidth conditions and can achieve better performance than static rate selection and static or jitter removal buffers.

4.1 BROS Model

A typical client-side playout buffer system with a buffer size of N frames, arrival rate λ, and playout rate μ is given in Figure 4.1

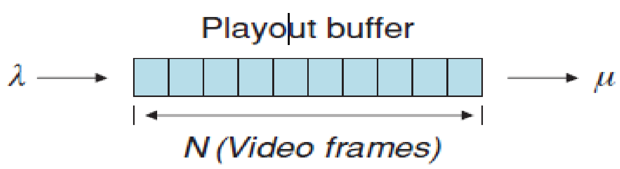


Figure 4.1 Buffer Model

Based on the buffer occupation, the Markov Model of *N + 1* states is as shown in Figure 4.2.

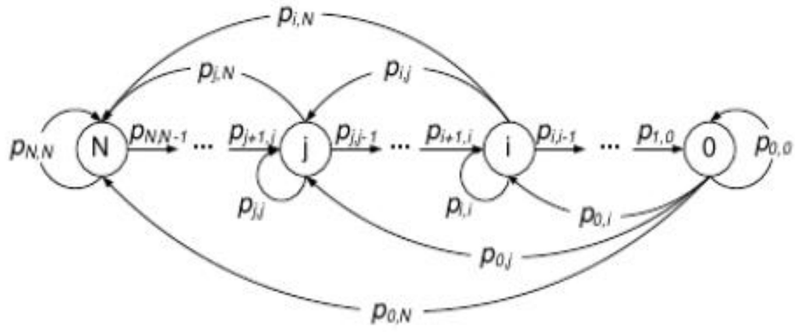


Figure 4.2 States of the Buffer Model

State N has a buffer with N frames, where state 0 is a buffer underflow. The matrix P in Equation 4.1 presents the transition probability of states i and j, where $0 \leq i, j \leq N$ [Li et al. 2009].

$$P_{i,j} = [p_{i,j}] = \begin{bmatrix} p_{0,0} & p_{0,1} & p_{0,2} & \cdots & p_{0,N} \\ p_{1,0} & p_{1,1} & p_{1,2} & \cdots & p_{1,N} \\ p_{2,0} & p_{2,1} & p_{2,2} & \cdots & p_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{N,0} & p_{N,1} & p_{N,} & \cdots & p_{N,N} \end{bmatrix} \tag{4.1}$$

Previously developed playout buffer models uses a Poisson arrival process [Yuang et al. 1996, 1997, Girod et al. 2001]. Poisson arrivals can be used as a lower bound on system performance when analyzing the buffer behavior [Laoutaris and Stavrakakis 2001]. Expected arrival rate of the streaming packets is impacted by the available bandwidth in networks with a large available bandwidth variance. For example, if the available bandwidth is less than the streaming rate, the expected arrival rate at the playout buffer will also be less than the streaming rate. Therefore, the transition probability model is based on both the available bandwidth and the streaming rate.

Following assumptions were made by Li et al. [2009] in order to define the probability matrix. First, packet loss is modeled as a reduction in available bandwidth

[Kalman et al. 2004, Lin and Costello Jr. 1984]. A lost packet will have multiple retransmission opportunities for given playout buffer of a few seconds. For typical inter-packet loss rates of less than 20% [Conklin et al. 2001], the probability that a packet is received after a few retransmission attempts is nearly one [Kalman et al. 2004].

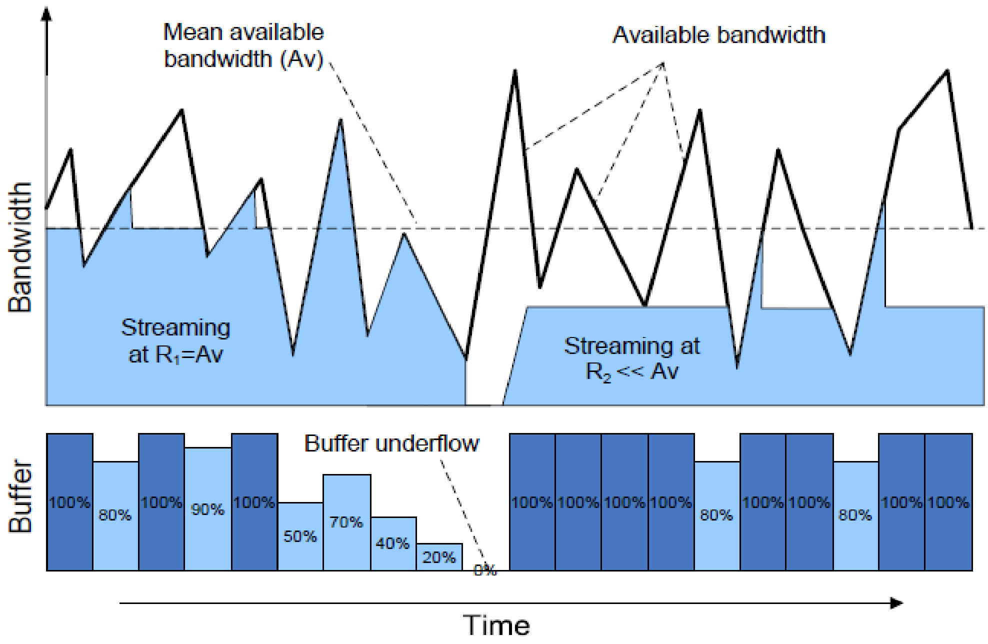


Figure 4.3 Streaming Rate and Avaliable Bandwidth.

Thus lost packets are treated as delayed due to insufficient bandwidth as shown in Figure 4.3. When bandwidth becomes available, delayed packets are sent in bursts at a rate equal to the available bandwidth until the buffer is filled again. Second, once a streaming rate is selected, Li et al. [2009] modeled the multimedia content as a constant data rate R and a constant frame rate μ for both streaming and decoding. Frame sizes depend upon the encoding and type of encoded frame, such as I, B, and P frames in MPEG encoding. So, a constant frame size $S = R/\mu$ is used to simplify the model.

The transition probability matrix can be defined for the buffer model, henceforth referred to as the as the full model [Li et al. 2009]:

$$p_{i,j} = \begin{cases} F_A[(j-i+1)R] - F_A[(j-i)R]\,, & 0 \le i < j < N\,,\ j-1 \le {}^{A_i}/_{R}\,; \\ 1 - F_A[(j-i)R]\,, & 0 \le i < j\,,\ j = N\ ,\ j-i \le {}^{A_i}/_{R}\,; \\ 1 - \sum_{j=0,j\neq i}^{N} p_{i,j}\,, & 0 \le i \le N,\ \ j = i; \\ F_A(R)\,, & 0 < i \le N,\ j = i-1; \\ 0\,, & elsewhere \end{cases} \tag{4.2}$$

- $F_A(\cdot)$ is the Cumulative Distribution function of the available bandwidth A.
- A_i is the available bandwidth in bits per seconds at state i.
- R is the constant data rate.

But the full model characterization does not lend itself to a closed form solution, thus it is difficult to use this model for real systems due to the required massive matrix computations. As shown in the right side of Figure 4.3, lowering the streaming rate below the average available bandwidth reduces the buffer requirement to avoid buffer underflow.

Therefore the full model can be simplified to closed form solution for the buffer underflow probability and the transition matrix can be reduced to simplified buffer model as:

$$p_{i,j} = \begin{cases} 1 - F_A(R)\,, & 0 \le i < N\,, j = i+1\,,\ j-1 \le {}^{A_i}/_{R}\,; \\ 1 - \sum_{j=0,j\neq i}^{N} p_{i,j}\,, & 0 \le i \le N,\ \ j = i; \\ F_A(R)\,, & 0 < i \le N,\ j = i-1; \\ 0\,, & elsewhere \end{cases} \tag{4.3}$$

Therefore, the closed form solution for the simplified buffer model becomes [Li et al. 2009] :

$$\pi_i = \gamma^i \pi_0 \tag{4.4}$$

where $\gamma = {}^{\left(1 - F_A(R)\right)}\!/_{F_A(R)}$

The buffer underflow probability π_0 for a given buffer size of N frames is:

$$\pi_0 = \frac{1-\gamma}{1-\gamma^{1+N}}, \quad \gamma \neq 1 \tag{4.5}$$

A streaming system with $\gamma \leq 1$ means streaming at a rate greater than the average available bandwidth and usually results in a high buffer underflow probability. Therefore, the streaming rate selection algorithm selects the initial streaming rate such that $\gamma > 1$. Moreover, if a streaming application demands a upper bound on the buffer underflow in terms of π_0 , γ can be computed from π_0 using Equation 4.5, and the streaming data rate can be calculated from the inverse CDF of the available bandwidth $F_A^{-1}(.)$ by

$$R = F_A^{-1}\left(\frac{1}{\gamma+1}\right) \tag{4.6}$$

Mean Time Between Buffer Underflows (MTBBU) can be used as a measure of performance for playout buffering [Kalman et al. 2004]. MTBBU, M_U , is distributed geometrically over the succession of frame slots as:

$$M_U = \left(\frac{1}{\pi_0}\right).\left(\frac{1}{\mu.60}\right) \tag{4.7}$$

where μ is the playout rate in frames per second and 60 is the number of seconds in a minute. Given M_U and the CDF of the available bandwidth, Equation 4.8 is the required buffer size N in frames:

$$N = \left\lceil \frac{\log\left(1+(\gamma-1)(M_U.\mu.60)\right)}{\log \gamma} - 1 \right\rceil \tag{4.8}$$

The minimum client side buffer, N', may also include an extra buffer space required for video decoding or playback. For example, an extra buffer B_{min} may be needed for handling VBR video or encoding dependencies. Li et al. [2009] assumes $B_{min} = 1$, which means only the frame that is currently being played out is considered.

$$N' = N + B_{min} = N + 1 \tag{4.9}$$

Equation 4.9 gives the resultant minimum buffer size required.

Chapter 5
Simulation Experiments

In Chapter 4, BROS model [Li et al. 2009] was discussed to minimize the buffer size for given streaming rate and CDF of available bandwidth. This streaming rate and CDF can be calculated by WBest model as discussed in chapter 3. Combination of both these models results in a required Buffer size. Matlab software has been used for the simulation purpose for BROS. Matlab program is developed for the simulation purposes to estimate the minimum buffer required to achieve a MTBBU with a given rate and known network condtitions, such as the CDF of the available bandwidth $F_A(.)$. Also this model can be used to explore the relationship between MTBBU, buffer size, and streaming rate from a different aspects which are discussed in this chapter. Li et al. [2009] used the normal distribution to estimate the available bandwidth distribution $F_{\hat{A}}(.)$ in wireless networks. In our work, we have used positive normal distribution to estimate the available bandwidth distribution which is further used in estimation of Buffer size.

5.1 Simulation using Normal Distribution

Li et al. [2009] used normal distribution to estimate the CDF of available bandwidth which is further used to estimate buffer size required for streaming in multimedia applications.

If X follows normal distribution, the probability density function (p.d.f.) of the normal distribution is given as:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \quad -\infty < x < \infty, -\infty < \mu < \infty, \sigma > 0 \qquad (5.1)$$

where, μ & σ^2 are the parameters of this distribution and this is denoted by $X \sim N(\mu, \sigma^2)$.

Mean of normal distribution = $E(X) = \int_{-\infty}^{\infty} x f(x)\, dx = \mu$

Variance of normal distribution = $V(X) = \int_{-\infty}^{\infty} (x - mean)^2 f(x)\, dx = \sigma^2$

Using the buffer model discussed in chapter 4 and WBest combined with normal distribution, a Matlab program was developed for the simulation which is given below.

Program 1: Simulation Program using Normal Distribution

```
clc;
clear;
faR = input('enter initial distribution streaming rate ')
maxbw = input('enter maximum bandwidth possible')
mean = input('enter mean of normal distribution ')
var = input('enter variance of normal distribution ')
stdev= sqrt(var)
gammaint = (1-faR)/faR
R = mean+sqrt(-2*var*log((stdev*sqrt(2*pi))/(gammaint+1)))
frate = maxbw-R
Nmax = input('enter max buffer size ')
fid = fopen('data1.txt','a');
fprintf(fid,'\n mtbbu(sec)     buffer(sec)\n');
index=1;
while(index<=7)
Mu = input('enter mtbbu ')
PI = 1/(Mu*frate*60)
gamma = gam(PI,Nmax)
lim = 1/(gamma+1)
THr = mean+sqrt(-2*var*log(stdev*sqrt(2*pi)*lim))
n = input('enter no of streaming rates to be selected ');
fprintf('enter streaming rates ');
for i=1:1:n;
    r(i) = input('');
end


disp(r)
for i=1:1:n-1
  if THr > r(i)
      if THr <=r(i+1)
         hsr  = r(i)
```

```
    end
  end
end
if THr <= r(1)
  stop
else
   fprintf(' to calculate new gamma calculate dist fun of hsr');
   a=input('input a (starting value)->');
   b= hsr;
   n=input('input number of intervals (n) ->');
   h=(b-a)/n;
   fa=exp(-(((a-mean)/var)^2)/2);
   fb=exp(-(((b-mean)/var)^2)/2);
   ff=0;
   for i=2:2:n;
     x = (a+(i-1)*h);
     fx = exp(-(((x-mean)/var)^2)/2);
     ff = ff + 4*fx;
  end
  for i=3:2:n;
     x = (a+(i-1)*h);
     fx = exp(-(((x-mean)/var)^2)/2);
     ff = ff + 2*fx;
  end
 result=((h/3)*(fa+fb+ff))/sqrt(2*pi)
 gammanew = (1-result)/result
 N1 = (log(1+(gammanew-1)*Mu*frate*60)/log(gammanew))    %final buffer size
end
fprintf(fid,'%8.2f      %8.2f\n',Mu,N1);
index = index+1;
end
fclose(fid);
```

The results obtained using this program with various input parameters are shown in Figure 5.1, Figure 5.2 and Figure 5.3.

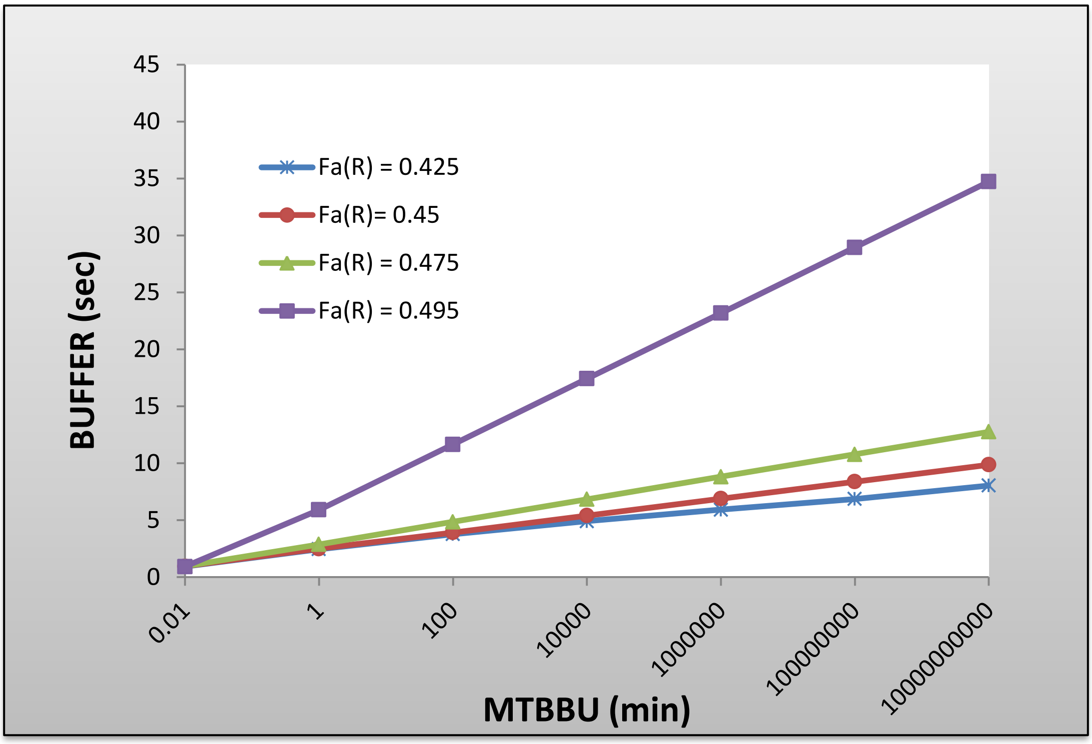


Figure 5.1 Required buffer size versus MTBBU

Figure 5.1 shows the minimum buffer size required for various values of the streaming rate to achieve MTBBU. This shows that larger buffer size is needed to achieve desired MTBBU for higher streaming rate values.

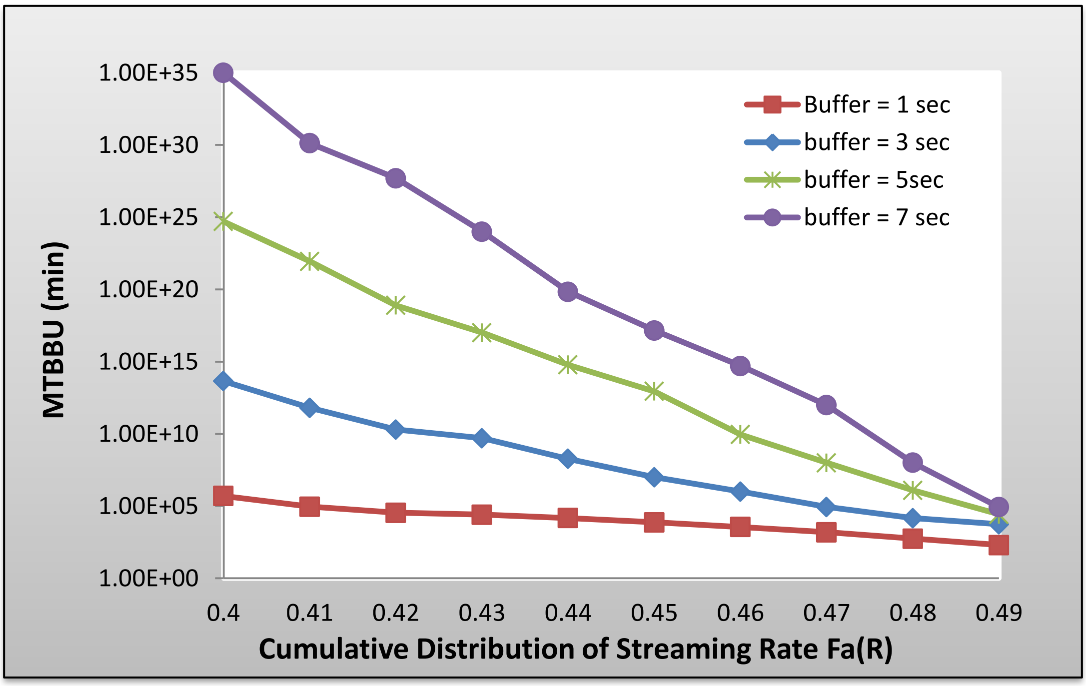


Figure 5.2 MTBBU versus Streaming rate

Figure 5.2 gives the relationship of MTBBU and streaming rate for fixed-sized buffers. This figure indicates that decrease in streaming rate results in increase of MTBBU for a given buffer size.

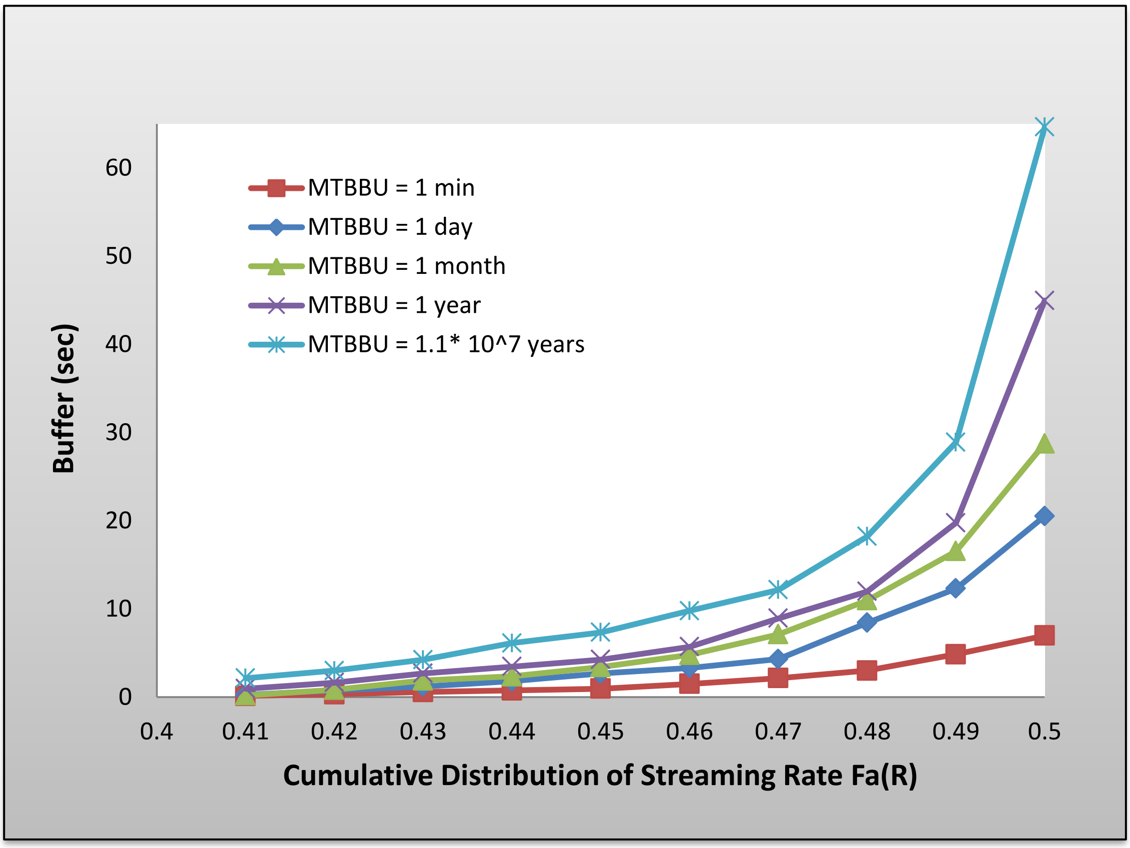


Figure 5.3 Buffer size versus streaming rate

Figure 5.3 depicts the relationship between buffer size and streaming rate for a given MTBBU. Increase in streaming rate increases the buffer size after a particular value of streaming rate for a given MTBBU.

5.2 Simulation using Positive Normal Distribution

In this work we have also used positive normal distribution to estimate the CDF of available bandwidth which is further used to estimate buffer size.

If X follows positive normal distribution, the probability density function (p.d.f.) of the distribution is given as:

$$f(x) = \frac{2}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \quad x \geq 0, \ \mu \geq 0, \ \sigma > 0 \tag{5.2}$$

Mean of positive normal distribution = $E(X) = \int_{-\infty}^{\infty} xf(x)\,dx = 2\mu$

Variance of positive normal distribution = $V(X) = \int_{-\infty}^{\infty} (x - mean)^2 f(x)\,dx = 4\sigma^2$

Program 2: Simulation using Positive Normal Distribution

```
global PI
clc;
clear;
faR = input('enter initial distribution streaming rate ')
maxbw = input('enter maximum bandwidth possible')
mean = input('enter mean of  positive normal distribution ')
var = input('enter variance of  positive normal distribution ')
stdev= sqrt(var)
gammaint = (1-faR)/faR
R = mean+sqrt(-2*var*log((stdev*sqrt(2*pi))/(2*(gammaint+1))))
frate = maxbw-R
Nmax = input('enter max buffer size ')
fid=fopen('data2.txt','a');
fprintf(fid,'\n mtbbu(sec)     buffer(sec)\n');
index=1;
while(index<=7)
Mu = input('enter mtbbu ')
PI = 1/(Mu*frate*60)
gamma = gam(PI,Nmax)
lim = 1/(2*(gamma+1))
THr = mean+sqrt(-2*var*log(stdev*sqrt(2*pi)*lim))
n = input('enter no of streaming rates to be selected ');
fprintf('enter streaming rates ');
for i=1:1:n;
    r(i) = input('');
```

```
end
disp(r)
for i=1:1:n-1
  if THr > r(i)
      if THr <=r(i+1)
        hsr  = r(i)
      end
  end
end
if THr <= r(1)
  stop
else
   fprintf(' to calculate new gamma calculate dist fun of hsr');
   a=input('input a (starting value)->');
   b= hsr;
   n=input('input number of intervals (n) ->');
   h=(b-a)/n;
   fa=exp(-(((a-mean)/var)^2)/2);
  fb=exp(-(((b-mean)/var)^2)/2);
  ff=0;
  for i=2:2:n;
     x = (a+(i-1)*h);
     fx = exp(-(((x-mean)/var)^2)/2);
     ff = ff + 4*fx;
  end
  for i=3:2:n;
     x = (a+(i-1)*h);
     fx = exp(-(((x-mean)/var)^2)/2);
     ff = ff + 2*fx;
```

```
 end
result=(((2*h)/3)*(fa+fb+ff))/sqrt(2*pi)
gammanew = (1-result)/result
 N1 = (log(1+(gammanew-1)*Mu*frate*60)/log(gammanew))
end
fprintf(fid,'%8.2f     %8.2f\n',Mu,N1);
index=index+1;
end
fclose(fid);
```

The results obtained using this program with various inputs is shown in Figure 5.4, Figure 5.5 and Figure 5.6.

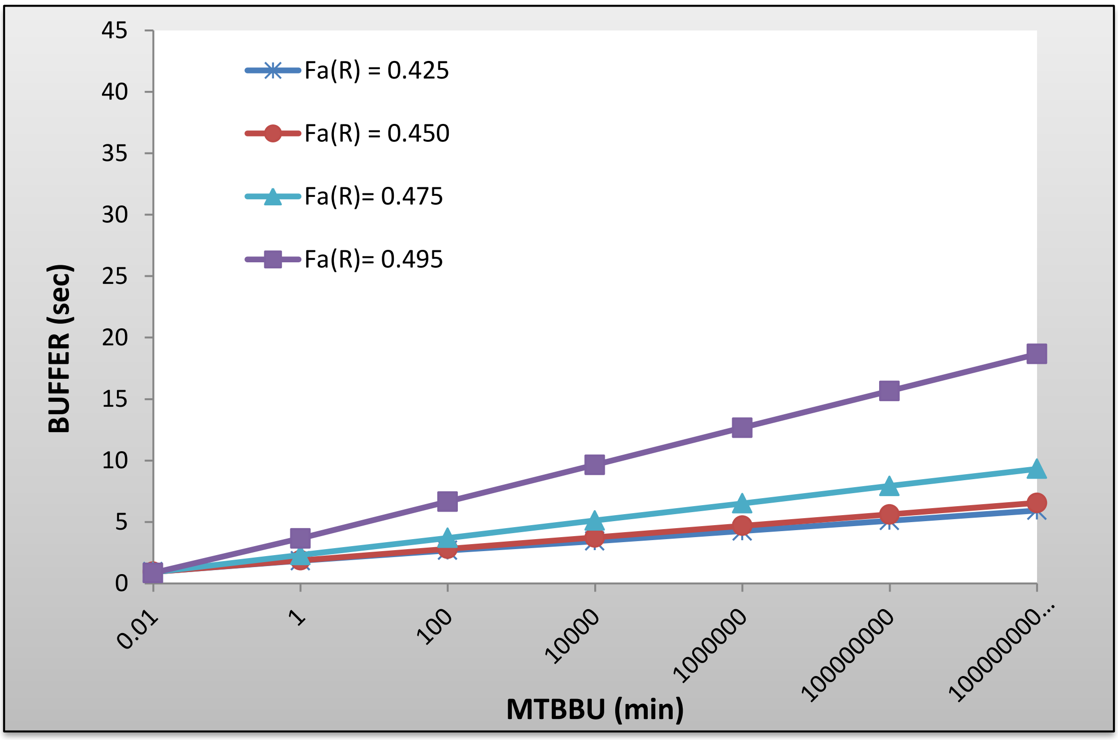


Figure 5.4 Required buffer size versus MTBBU

Figure 5.4 shows the minimum buffer size required for various values of the streaming rate to achieve a given MTBBU.

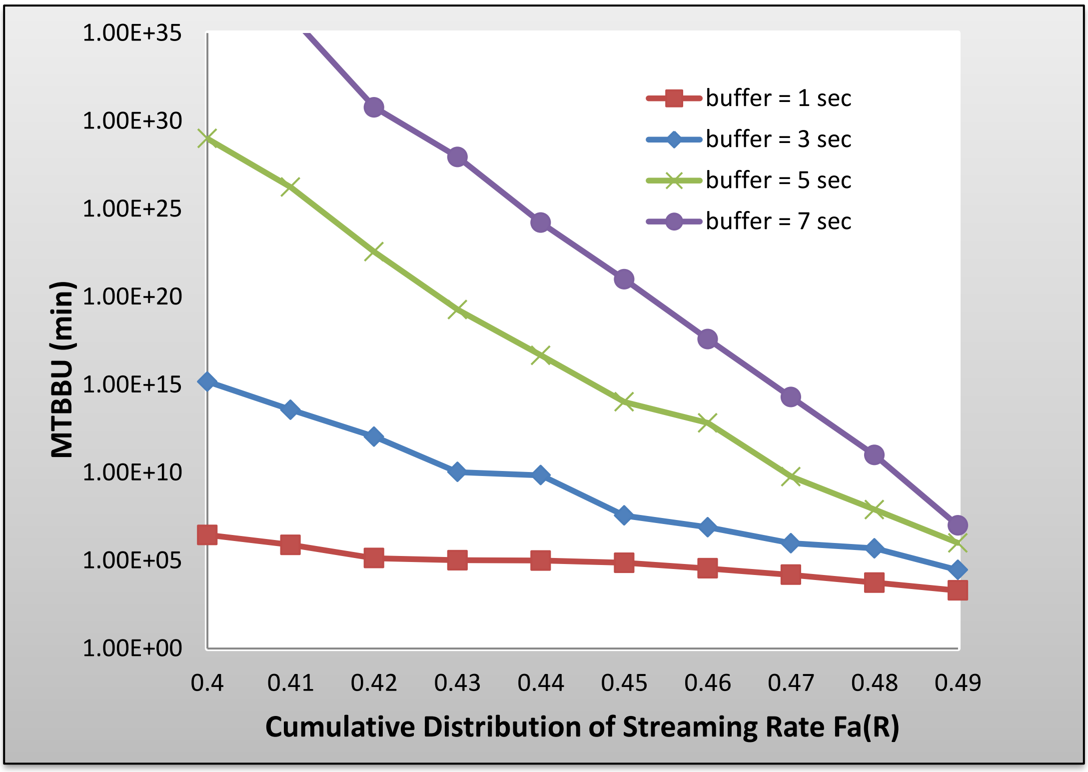
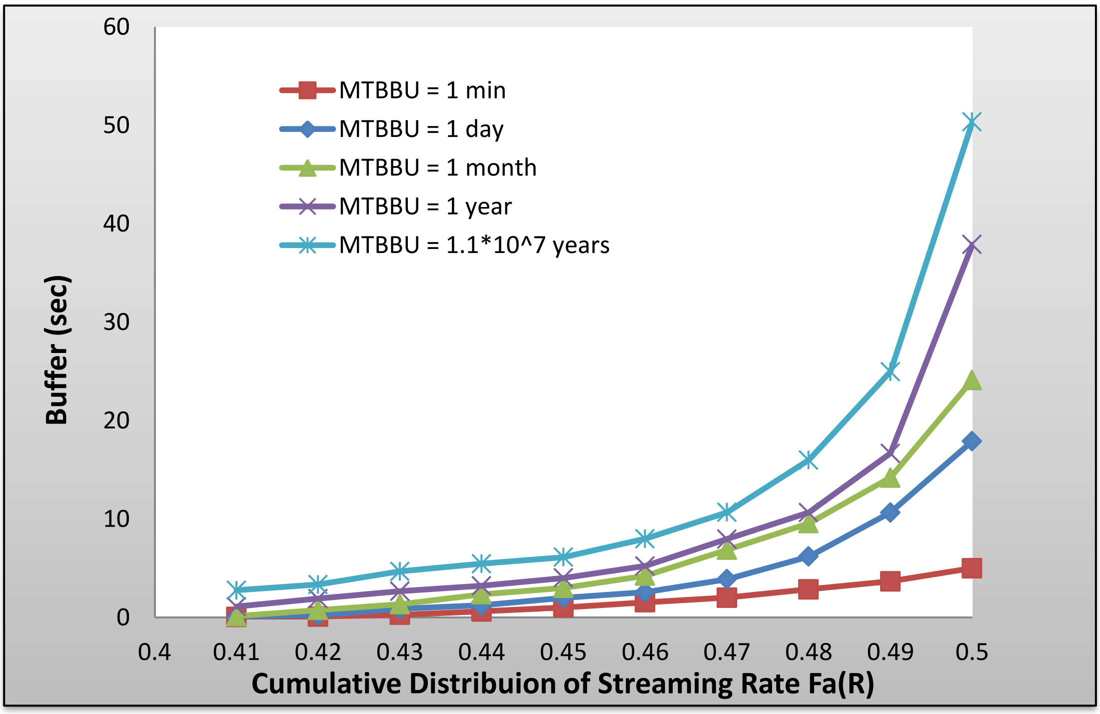


Figure 5.5 MTBBU versus Streaming rate

Figure 5.5 gives the relationship between MTBBU and streaming rate for fixed-sized buffers. Figure 5.6 shows the relationship between buffer size and streaming rate for a given MTBBU.



Chapter 6

Results, Discussion and Future Scope

This chapter contains the discussion on the results obtained during the work carried out in this presentation and the possible directions for future work. The discussions are divided into two parts. First part contains the result for normal distribution and second part contains the results for positive normal distribution.

The simulation results for normal distribution include:

- Larger the value of the streaming rate, the larger is the buffer size needed to achieve required MTBBU. For larger value of distribution function like $F_a(R) = 0.495$, buffer size ranges from 0 to 35 sec for MTBBU ranging from 0.01 to 1e+10.

- For a given buffer size, decreasing the streaming rate increases the MTBBU, thus providing fewer buffer underflow events. For streaming rate ranging from 0.5 to 0.4, MTBBU ranges from 1e+35 to 1e+04 for given buffer = 7 sec.

- The higher MTBBU values are more sensitive to the buffer size. As the streaming rate is increased, required buffer size increases for given MTBBU. For larger value of MTBBU, buffer size ranges from 2 to 65sec for streaming rate ranging from 0.41 to 0.5.

The simulation results for positive normal distribution include:

- Larger buffer size is needed to achieve required MTBBU for large values of the streaming rate, but as compared to normal distribution this increase is much less. In this, buffer size ranges from 0 to 20 sec for $F_a(R) = 0.495$ which is less than that for normal distribution. This shows that we need less buffer size in positive distribution for particular value of MTBBU.

- For a given buffer size, decreasing the streaming rate increases the MTBBU, thus providing fewer buffer underflow events. In case of positive normal distribution, MTBBU ranges from 1e+06 to 1e+41 for buffer size = 7 sec. this shows that there will be less buffer underflow events in this case as compared to normal distribution.

- The higher MTBBU values are more sensitive to the buffer size. As the streaming rate are increased, required buffer size increases. But this increase is less than the increase in normal distribution as buffer size ranges from 2 to 50 sec whereas in normal this ranges from 2 to 65 sec.

So, we can conclude from the above observations that in case of positive normal distribution we need less buffer size for streaming as compared to normal distribution. As such, optimizing playout buffer using positive normal distribution is more effective.

Future Scope

Now, the possible directions for future work in this area are presented. These directions may include the following.

- We have used positive normal distribution to estimate CDF of available bandwidth which is further used to estimate the required buffer size. Some other suitable probability distribution which might give better results can be used to improve the streaming in multimedia applications for wireless networks.

- In the proposed work, BROS model is used for streaming in multimedia applications for wireless networks. One can develop some other model which can improve the streaming in wireless networks for different network conditions and assumptions.

- In this model, BROS doesn't stream the data when there is no available bandwidth in the wireless network calculated by WBest. So, one can make this model to work when bandwidth information is missing by changing different parameters used in this model.

References

1. Angrisani, L., Botta, A., Pescape, A., and Vadursi, M. 2006. *Measuring wireless links capacity*. In Proceedings of the 1st International Symposium on Wireless Pervasive Computing, 1–5.
2. Bai, G. and Williamson, C. 2004. *The effects of mobility on wireless media streaming performance.* In Proceedings of Wireless Networks and Emerging Technologies (WNET), 596–601.
3. Bellovin Steven M., "*A Best-Case Network Performance Model*," Tech. Rep., ATT Research, Feb. 1992.
4. Birney, B. 2004. Reducing start-up latency with Windows Media 9 series. Online Documents. http://www.microsoft.com/windows/windowsmedia/howto/articles/reducingstartuplatency.aspx.
5. Bolot J.C., "*Characterizing End-to-End Packet Delay and Loss in the Internet*," in Proceedings of ACM SIGCOMM, 1993, pp. 289–298.
6. Cen, S., Cosman, P. C., and Voelker, G. M. 2003. *End-to-end differentiation of congestion and wireless losses*. IEEE/ACM Trans. Netw. 11, 5(1), 703–717.
7. Chen, M. and Zakhor, A. 2005. *Rate control for streaming video over wireless*. IEEE Wirel. Comm. 12, 4, 32–41.
8. Conklin, G. J., Greenbaum, G. S., Lillevold, K. O., Lippman, A. F., and Reznik, Y. A. 2001. *Video coding for streaming media delivery on the Internet*. IEEE Trans. Circ. Syst. Video Technol. 11, 3, 269–281.
9. Demircin, M. U. and Van Beek, P. 2005. *Bandwidth estimation and robust video streaming over 802.11E wireless LAN'S.* In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), 1250–1253.
10. Fujimoto, K., Ata, S., and Murata, M. 2002. *Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications*. In Proceedings of the IEEE Conference and Exhibition on Global Telecommunications (GlobeCom) vol. 3, 2451–2457.
11. Hsu, C. Y., Ortega, A., and Khansari, M. 1999. *Rate control for robust video transmission over burst-error wireless channels*. IEEE J. Select. Areas Comm. 17, 5, 756–773.

12. Intel Corporation, "White Paper: The New Era in Communications," 2004, Online at: http://www.intel.com/netcomms/bbw/302026.htm.
13. IEEE Computer Society LAN MAN Standard Committee, "*IEEE 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*," Standard, Aug. 1999.
14. Jacob Strauss, Dina Katabi, and Frans Kaashoek, "*A Measurement Study of Available Bandwidth Estimation Tools*," in IMC '03: Proceedings of the 3rd ACM SIG-COMM Conference on Internet Measurement, Miami Beach, FL, USA, 2003, pp. 39–44.
15. Jacobson Van, "*pathchar: A Tool to Infer Characteristics of Internet Paths*," Apr. 1997, Online at: ftp://ftp.ee.lbl.gov/pathchar/.
16. Jacobson Van, "*Congestion Avoidance and Control*," in Proceedings of ACM SIGCOMM, Sept. 1988, pp. 314–329.
17. Jain M. and Dovrolis C., "*End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput*," IEEE/ACM Transactions in Networking, pp. no. 295-308, Aug. 2003.
18. Kalman, M., Steinbach, E., Andgirod, B. 2004. *Adaptive media playout for low delay video streaming over error-prone channels*. IEEE Trans. Circ. Syst. Video Technol. 14, 6, 841–851.
19. Kaveh Pahlavan and Prashant Krishnamurthy, *Principles of Wireless Networks – A Unified Approach*, chapter 2-4, Prentice Hall PTR, 2002.
20. Kazantzidis, M. and Gerla, M. 2003. *The impact of link layer assisted multimedia adaptation in wireless network*. In Proceedings of the IEEE International Conference on Information Technology, Research and Education (ITRE), 326–330.
21. Keshav S., "*A Control-Theoretic Approach to Flow Control*," in Proceedings of ACM SIGCOMM, Sept. 1991, pp. 3–15.
22. Kuang, T. and Williamson, C. L. 2004. *Hierarchical analysis of real-media streaming traffic on an IEEE 802.11b wireless LAN*. Computer. Comm. 27, 6, 538–548.
23. LakshmiNaraynan, K., Padmanabhan, V. N., and Padhye, J. 2004. *Bandwidth estimation in broadband access networks*. In Proceedings of the Internet Measurement Conference (IMC 2004), 314–321.

24. Laoutaris, N. and Stavrakakis, I. 2001. *Adaptive playout strategies for packet video receivers with finite buffer capacity*. In Proceedings of the IEEE International Conference on Communications (ICC). Vol. 3, 969–973.
25. Li, F., Chung, J., Li, M., Claypool, M., and Kinicki, R. 2005a. *Application, network and link layer measurements of streaming video over a wireless campus network.* In Proceedings of the 6th Passive and Active Measurement Workshop (PAM), 189–202.
26. Li, K., Krasic, C., Walpole, J., Shor, M. H., and Pu, C. 2001. *The minimal buffering requirements of congestion controlled interactive multimedia applications.* In Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems (IDMS '01), 181–192.
27. Li M., Mark Claypool, and Robert Kinicki 2009"*Playout Buffer and Rate Optimization for Streaming over IEEE 802.11 Wireless Networks*," Vol. 5, Article 26.
28. Li, M., Claypool, M., and Kinicki, R. 2006. *WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks*. In Proceedings of the 33rd IEEE Conference on Local Computer Networks (LCN).
29. Li, M., Claypool, M., Kinicki, R., and Nichols, J. 2005b. *Characteristics of streaming media stored on the Web*. ACM Trans. Internet Technol. 5, 4, 601–626.
30. Li, M., Li, F., Claypool, M., and Kinicki, R. 2005c. *Weather forecasting: Predicting performance for streaming video over wireless LANs*. In Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), 33–38.
31. Li, Q. and Van Der Schaar, M. 2004. *Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation*. IEEE Trans. Multimedia 6, 2, 278–290.
32. Li, A. Markopoulos, N.Bambos, and J.Apostolopoulos, "*Joint Power/Playout Control Schemes for Media Streaming over Wireless Links*," in Proceedings of Inter-national Packet Video Workshop (PV), Irvine, CA, USA, Dec. 2004.
33. Lin, S., Costello, D. J., Jr., and Miller, M. J. 1984. *Automatic-repeat-request error-control schemes*. IEEE Comm. Mag. 22, 12, 5–17.
34. Melander B., Mats Bjorkman, and Per Gunningberg, "*A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottleneck*s," in IEEE

GLOBECOM–Global Internet Symposium, San Francisco, CA, USA, Nov. 2000, pp. 415–420.

35. Melander B., Mats Bjorkman, and Per Gunningberg, "*Regression-Based Available Bandwidth Measurements*," in International Symposium on Performance Evaluation of Computer and Telecommunications Systems, 2002, July 2002.
36. Moon, S. B., Kurose, J., and Towsley, D. 1998. *Packet audio playout delay adjustment: Performance bounds and algorithms*. Multimedia Syst. 6, 1, 17–28.
37. Mundur, P., Sood, A., and Simon, R. 1999. *Network delay jitter and client buffer requirements in distributed video-on-demand systems. Tech. rep*. Department of Computer Science at George Mason University, Fairfax, VA.
38. Prasad, R., Murray, M., Dovrolis, C., and Claffy, K. 2003. *Bandwidth estimation: Metrics, measurement techniques, and tools.* IEEE Netw. 17, 6, 27–35.
39. Ramjee, R., Kurose, J., Towsley, D., and Schulzrinne, H. 1994. *Adaptive playout mechanism for packetized audio applications in wide-area networks.* In Proceedings of the IEEE Infocom, 680–688.
40. Ribeiro V., R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "*pathChirp: Efficient Available Bandwidth Estimation for Network Paths*," in Proceedings of Passive and Active Measurements (PAM) Workshop, La Jolla, CA, USA, Apr. 2003.
41. Steinbach, E., Faerber, N., and Girod, B. 2001. *Adaptive playout for low-latency video streaming.* In Proceedings of the International Conference on Image Processing (ICIP), 962–965.
42. Van Beek, P. and Demircin, M. U. 2005. *Delay-constrained rate adaptation for robust video transmission over home networks*. In Proceedings of the IEEE International Conference on Image Processing (ICIP). Vol. 2, 173–176.
43. Van Beek, P., Deshpande, S., Pan, H., and Sezan, I. 2004. *Adaptive streaming of high-quality video over wireless LANs.* In Proceedings of the Conference on Visual Communications and Image Processing, 647–660.
44. Yang, F., Zhang, Q., Zhu, W., and Zhang, Y.Q. 2004. *End-to-end TCP-friendly streaming protocol and bit allocation for scalable video over wireless internet*. IEEE J. Select. Areas Comm. 22, 4, 777–790.

45. Yang, G., Gerla, M., and Sanadidi, M. Y. 2004. *Adaptive video streaming in presence of wireless errors*. In Proceedings of IFIP/IEEE Management of Multimedia Networks and Services (MMNS), 26–38.
46. Yuang, M., Liang, S., Chen, Y., and Shen, C. 1996. *Dynamic video playout smoothing method for multimedia applications*. In Proceedings of the IEEE International Conference on Communications (ICC). vol. 3, 1365–1369.
47. Yuang, M. C., Tien, P. L., and Liang, S. T. 1997. *Intelligent video smoother for multimedia communications*. IEEE J. Select. Areas Comm. 15, 2, 136–146.
48. Zink, M., Kunzel, O., Schmitt, J., and Steinmetz, R. 2003. *Subjective impression of variations in layer encoded videos*. In Proceedings of the 11th International Workshop on Quality of Service (IWQoS), 137–154.