An Adaptive Rate-Control Streaming Mechanism with Optimal Buffer Utilization

Shu-Ching Chen [a,*], Mei-Ling Shyu [b], Irina Gray [a], Hongli Luo [b]

[a] *Distributed Multimedia Information System Laboratory*
School of Computer Science, Florida International University
Miami, FL 33199, USA

[b] *Department of Electrical and Computer Engineering, University of Miami*
Coral Gables, FL 33124, USA

Abstract

In this paper, an end-to-end real-time adaptive protocol for multimedia transmission is presented. The bandwidth is dynamically allocated according to the network status, and the client buffer occupancy and playback requirement. The transmission rate is determined by the quadratic probing algorithm that can obtain the maximal utilization of the client buffer and minimal occupation of the network bandwidth. It is also coupled with a congestion control mechanism that can effectively decrease the packet loss rate during network congestion. We investigate the performance of our quadratic probing algorithm in different congestion levels under both the Local Area Net (LAN) and Internet environments. Performance analysis reveals that our approach is more robust in avoiding overflows and underflows in different network congestion levels, and adapting to the changing network delays. Comparisons are made with the fixed rate approach and the rate by playback requirement approach. The experimental results show that our proposed real time protocol with the rate adjusting quadratic probing algorithm is efficient in utilizing the network resources and decreasing the packet loss ratios.

Key words: Multimedia streaming, Protocol, Adaptive transmission rates, Buffer utilization

* Corresponding author. Tel.: +1-305-348-3480; Fax: +1-305-348-3549.
E-mail address: chens@cs.fiu.edu (S.-C. Chen).

1 Introduction

Efficient delivery of streaming media over the Internet poses many challenges. Distributed multimedia applications have different Quality of Service (QoS) requirements. For example, the transmission of real-time video requires interactivity, low jitter, low delay and higher bandwidth but can tolerate some transmission errors (Wu et al., 2000). However, the Internet is a best-effort network and dose not provide QoS guarantee for multimedia services. The application must be aware of the conditions of the network and adapts the multimedia transmission to the network conditions. Therefore, it is important to design an adaptive and reliable multimedia streaming protocol that can cope with varying Internet conditions.

Different approaches may be considered to address the QoS requirements. Adaptive rate control is to adjust the bandwidth used by an application according to the existing network conditions. This approach has the advantage of better utilizing available network resources (which change with time) compared to those approaches relying on resource reservation (e.g., RSVP) (Braden et al., 1997). RSVP requires that all intermediate routers have QoS supports. According to (Wang and Schulzrinne, 1999), adaptive control schemes presented in the literature can be generally classified into three categories: sender-driven, receiver-driven and transcoder-based. Sender-driven adaptation schemes fall into two categories: buffer based and loss based. Buffer based adaptation schemes use the occupancy of a buffer on the transmission path as a measure of congestion (Jacobs and Eleftheriadis, 1998; Kanakia et al., 1995). Loss based adaptation schemes adjust the rate based on the packet loss experienced by the receivers (Busse et al., 1996; Sisalem and Schulzrinne, 1998). In receiver-driven adaptation, the receivers individually select the transmission of a particular quality according to their needs and capabilities. A number of receiver-driven schemes use a combination of layered encoding, and a layered transmission scheme. The receiver selects a transmission quality appropriate to its requirements and constraints by subscribing to a certain number of multicast groups carrying different layers. The receiver monitors network congestion (based on the parameters such as packet loss and throughput), and adapts to the changes in the network conditions by adding or dropping layers accordingly. Transcoder-based approaches use multimedia gateways at the appropriate locations in the network, which convert through transcoding a high bandwidth transmission into a transmission with the appropriate bandwidth, thus to accommodate groups of poorly connected receivers. The gateway may also use an adaptive rate-control algorithm to adjust its transmission in response to the receiver feedback (Kouvelas et al., 1998).

Efficient delivery of multimedia streams over the Internet also requires that the media react to the network congestion by adapting their transmission rates.

Since the routers typically do not actively provide congestion control (Braden et al., 1998), end-to-end congestion control is more recommended for Internet multimedia transmission (Floyd and Fall, 1999). Real-time streams are also expected to share the network bandwidth with the dominant TCP flows to obtain the inter-protocol fairness. Some approaches adjust the transmission rate in an additive increase and multiplicative decrease (AIMD) way which is similar to TCP (Jacobs and Eleftheriadis, 1998; Rejaie et al., 1999). These approaches will result in rather fluctuant transmission rates, which may produce an annoying presentation quality at the receiver. The receiver is required to acknowledge every received packet for the sender to collect information of the network status. The frequent feedback packets will consume the bandwidth and degrade the network congestion. Some approaches use model-based flow control, where the models were developed to calculate the bandwidth as a function of the packet loss ratio and round trip time (Padhye et al., 1998). However, the estimated packet loss ratio may not be suitable for the next time interval, and thus affect the accuracy of the throughput calculation.

Common requirements for multimedia applications have led to the design of the general purposed protocol called the Real-Time Transport Protocol (RTP) (Schulzrinne et al., 1996). To satisfy those common requirements, RTP provides the functions such as (1) the ability to communicate the selected coding scheme, (2) the mechanisms to facilitate the application-specific handling of time-stamped data, which enables the receiver to play the data at the appropriate time, (3) the synchronization of multiple media, (4) the indication of packet loss, and (5) the notification to the sender when the loss occurs. Although RTP provides the functionality suited for carrying real-time content and is the primer protocol for real-time applications, it cannot provide any form of reliability or protocol-defined flow/congestion control. However, the existing RTP has a flexible mechanism that leaves many protocol details. This mechanism allows the designers to build up the functionality required by the particular application.

In this paper, we design and implement a novel real-time adaptive multimedia transmission protocol. The central component of the proposed scheme is an adaptive rate control mechanism. It dynamically changes the server transmission rates, considering the relationships among the server transmission rates, client buffer occupancies, playback rates, network delays and packet loss ratios (Chen et al., 2003; Shyu et al., 2002a,b,c). Since the network bandwidth is scarce in today's Internet, it is meaningful to use minimal bandwidth for each media streaming. Our approach is source rate adaptive, which aims at obtaining a minimal transmission rate according to the client buffer occupancy, network delay, packet loss rates and playback requirement. It can achieve an efficient utilization of network resources such as bandwidth and client buffer at the same time. To ensure the fairly bandwidth sharing with the dominant TCP flows, a congestion control mechanism is incorporated in our optimal

bandwidth allocation scheme. We also use the estimated network bandwidth as the upper bound limit for the transmission rates.

The paper is organized as follows. In the next section, the adaptive multimedia transmission protocol is presented and the optimal bandwidth allocation scheme is introduced. Experimental results for both LAN and Internet are given in Section 3. Conclusions are presented in Section 4.

2 The real-time adaptive multimedia transmission protocol

As mentioned earlier, though RTP cannot provide any form of reliability or protocol-defined flow/congestion control, it gives the designers the opportunity to add reliability, flow/congestion control, and functionality for efficient use of the bandwidth. In this paper, a real-time adaptive multimedia transmission protocol is developed with the implementation of the rate adaptive algorithm. This protocol can provide the transmission of the requested video files from the sender to multiple receivers using transport-layer protocol UDP. At the receiver side, the control information such as the packet loss ratio, playback requirement and buffer occupancy are collected and fed back to the sender periodically. The sender uses the feedback information to calculate the optimal transmission rate based on the network model (presented in Section 2.2). In this section, the architecture of the system, data packet format and control flow part of our protocol are first presented and then a short introduction of the quadratic probing algorithm is given.

2.1 Protocol design

Our system is a distributed multimedia system, which is constructed upon the client/server architecture. Like the RTP and RTCP protocols, our protocol is running over UDP/IP. Unlike the TCP protocol, the UDP protocol is not checked for delivery and packets may be lost. So the compressed video stream must be divided into small packets before being sent to the network. Here we choose the packet size of 1024 bytes or 1 KB. At the other end of the network, the packets received by the receivers are decompressed to reconstruct the video stream, which may be degraded due to the packet loss. The proposed protocol consists of two parts: a data part and a control part. The control part provides the feedback information on the performance of the application and the status of the network by periodically sending the control information to the server in the report. Multimedia data is to be carried on an even UDP port number. The corresponding feedback packets with reports are to be carried on the next higher (odd) port number. The client and the

server applications, implemented in C++, use UDP APIs for the connections between the UNIX Sockets and run on Sun SPARC/SunOS platform. The server and the client both use Posix threads (Lewis and Berg, 1998) for the creation of the multithreaded environment.

The packet sent from the server to the clients consists of the header and the payload parts. The header has 15 bytes and includes the following fields:

- Packet sequence number (3 bytes), which provides the detection and measurement of the lost and misordered packets. The sequence numbers increase by one for each packet transmitted.
- Packet length (2 bytes), which provides a length of the payload part of the transmitted packet.
- Seconds (3 bytes) and microseconds (3 bytes) of the time when the server sends this packet (i.e., 6 bytes).
- The time period to which this packet belongs (3 bytes): The time periods increase by one when one second passes. Several video packets may have the same time period.
- Reserved byte for future use (1 byte).

For multimedia applications, it is necessary to keep the header short to reduce the overhead and make efficient uses of the bandwidth. Also, due to different data representations on different computer architectures, marshalling and unmarshalling of the header is performed every time when a packet is sent or received.

The receivers send to the server a control message report that includes the information such as time interval (time period), current playback rate, buffer usage, a detailed report of the packets arrived at this time period, the number of the discarded packets for this time period (fail number), and the number of the packets lost due to network congestion (lost number).

2.2 Quadratic probing algorithm with congestion control

A client requests continuous media streams from the server. The server starts the media stream by sending the data packets across the Internet. The client needs to buffer some data before the playback. To ensure continuous playback of the media stream, there should be always some packets buffered. That is, buffer underflows should be avoided. The congestion status and the available network bandwidth will vary in the course of the playback. The client buffer can help to alleviate the mismatch between the available bandwidth and the required transmission rate. Buffering at the client has been studied for the reason of smoothening the multimedia stream since it can reduce the jitter (Salehi et al., 1996). However, their proposed scheme assumes that a large

bandwidth is available and is not adaptive to the variations of the network traffic and the changes of the bandwidth. In order to maintain a near constant playback rate, we want to minimize the packet jitter. The fully utilization of the client buffer can minimize the playback degradations or the changes perceived by the users that are caused by the changes of the network.

Since the network bandwidth of the Internet is scarce, it is meaningful to allocate the bandwidth as minimal as possible for media streaming under the same playback requirement. Our goal is to obtain a suitable minimal transmission rate while at the same time to maximize the utilization of the buffer capacity. The transmission rate is determined by our proposed quadratic probing algorithm. This algorithm aims at achieving maximal utilization of the client buffer and minimal allocation of the bandwidth. Since the multimedia stream should also share bandwidth fairly with the dominant TCP traffic, our algorithm is also coupled with a congestion control mechanism that can effectively reduce the packet loss ratio during network congestion. A brief description of this algorithm is given below.

Assume there is an end-to-end transmission between a server and a client with the following variables:

- k: time interval;
- Q_k: buffer occupancy at the beginning of time interval k;
- R_k: the transmission rate at time interval k in terms of the number of packets transmitted from the server during per time interval;
- P_k: the arrival rate at time interval k in terms of the number of packets arriving at the client buffer during per time interval;
- L_k: the playback rate at time interval k in terms of the number of packets used for playback during per time interval; and
- Q_r: the allocated buffer size for each client, which is determined at the time when the client and the server setup their connection.

Consider the relationships among Q_k, R_k, P_k, and L_k. Let Q_{k+1} denote the buffer occupancy at time interval *k+1*. We have the following equation:

$$Q_{k+1} = Q_k + P_k - L_k \tag{1}$$

However, because of the changeable network delays, P_k is not equal to R_k at a certain time interval. Assume that the packets arriving at the client buffer at time interval k comprise of the packets transmitted from the server at the time intervals k-d, k-d-1, ..., k-d-i+1, ..., and k-d-m+1. Let $b_{i,k}$ denote the percentage of the packets transmitted at $R_{k-d-i+1}$ that arrive at the client buffer at time interval k. Therefore, P_k can be represented as a function of R_{k-d}, R_{k-d-1}, ..., $R_{k-d-i+1}$, ... and $R_{k-d-m+1}$ as given in Equation 2.

$$P_k = b_{1,k}R_{k-d} + b_{2,k}R_{k-d-1} + \ldots + b_{i,k}R_{k-d-i+1} + \ldots + b_{m,k}R_{k-d-m+1} \quad (2)$$

where the subscript k-d is to denote the closest time interval when the transmitted packet can arrive at the buffer, and k-d-m+1 denotes the farthest time interval when the transmitted packet can arrive at the buffer at time interval k.

As can be seen from Equation 2, our model can effectively capture the effects of changing network delays. First, the larger the value of d, the larger the network delay and the more congested the network is. Second, the larger the value of m, the larger the network delay is since it takes longer for all of the packets transmitted at a previous time interval to arrive at the client buffer. Third, the value of $b_{i,k}$ can reflect the changing network delays. The larger the value of $b_{i,k}$, the larger percentage of packets transmitted at time interval k-d-i+1 can arrive, which indicates the network is less congested. We can use the total number of $b_{i,k}$ considered and the position of the first nonzero $b_{i,k}$ value to indicate the network delay during the packet transmission. Different combinations can be used to capture different network delay situations. Every packet sent from the server is attached with a timestamp that indicates the time it is transmitted. At each time interval, the client checks the timestamp information of all the arriving packets. The client counts the number of arriving packets that are sent from the server at the same time interval separately, and then calculates the corresponding $b_{i,k}$ according to the transmission rate at a different time interval . The packet loss information can also be reflected by the value of $b_{i,k}$. The larger the value, the smaller the packet loss ratio is. When the value of $b_{i,k}$ is used as the feedback information, the server can decide the optimal transmission rate while also considering the packet loss ratio.

An example is given below to illustrate how to obtain the values of $b_{i,k}$. At time interval k, the packets arriving at this time interval consist of 1/2 of packets transmitted from the server at time interval *k-7*, 1/4 of packets transmitted at time interval *k-8*, 1/8 of packets transmitted at time interval *k-9*, and 1/8 of packets transmitted at time interval *k-10*. In this example, though only four $b_{i,k}$ values in Equation 2 are considered, it is enough to capture the network congestion due to the following two reasons. First, those packets sent earlier than time interval *k-10* may have been lost in the Internet because of the network congestion, and will never arrive. Second, even these packets arrive later, they will be useless and be discarded since they arrive after the playback schedule. So we have

$$b_{1,k}=1/2,\ b_{2,k}=1/4,\ b_{3,k}=1/8,\ b_{4,k}=1/8,\ d=7,\ m=4.$$

Thus the arriving packets can be represented as

$$P_k = \tfrac{1}{2}R_{k-7} + \tfrac{1}{4}R_{k-8} + \tfrac{1}{8}R_{k-9} + \tfrac{1}{8}R_{k-10}$$

To avoid the jitter and use the client buffer effectively, the difference between the buffer packet and the allocated buffer capacity should be minimized. On the other hand, the transmission rate R_k should be minimized for bandwidth optimization. The quadratic performance index J that requires minimization (Shyu et al., 2002b) is defined as follows.

$$J_k = (w_p Q_{k+d_0} - w_q Q_r)^2 + (w_r R_k)^2 \tag{3}$$

where

- w_p, w_q, and w_r are the weighting coefficients that can be chosen differently;
- d_0 is the transmission control delay.

Rewrite Equation 1 with one time interval shift, we have

$$Q_k = Q_{k-1} + P_{k-1} - L_{k-1} \tag{4}$$

Combine Equations 2 and 4 in the time domain (Lewis and Syrmos, 1995) to obtain the following equation

$$\begin{aligned}(1 - z^{-1})Q_k = {} & b_{1,k-1}R_k z^{-d-1} + b_{2,k-1}R_k z^{-d-2} \\ & + \ldots + b_{i,k-1}R_k z^{-d-i} + \ldots - L_k z^{-1}\end{aligned} \tag{5}$$

where z^{-1} is the delay operator (i.e., $z^{-1}Q_k = Q_{k-1}$). In addition, let $d_0 = d+1$, and $A(z^{-1}) = 1 - z^{-1}$ and $B(z^{-1}) = b_{1,k} + b_{2,k}z^{-1} + \ldots + b_{m,k}z^{-(m-1)}$ be two polynomials. We can have

$$A(z^{-1})Q_k = z^{-d_0}B(z^{-1})R_k - L_k \tag{6}$$

Here we assume that $b_{1,k} \neq 0$, and as defined earlier, d_0 is referred to as the transmission control delay. When 1 is divided by $A(z^{-1})$, we can get the following quotient $F(z^{-1})$ and remainder $z^{-d_0}G(z^{-1})$.

$$1 = A(z^{-1})F(z^{-1}) + z^{-d_0}G(z^{-1}), \tag{7}$$

where

$$\begin{aligned}F(z^{-1}) &= 1 + f_1 z^{-1} + \ldots + f_{d_0-1}z^{-(d_0-1)} \\ G(z^{-1}) &= g_0 + g_1 z^{-1} + \ldots + g_{n-1}z^{-(n-1)}\end{aligned} \tag{8}$$

From Equation 6, we get

$$\begin{aligned} Q_{k+d_0} &= \frac{1}{A(z^{-1})}(B(z^{-1})R_k - L_{k+d_0-1}) \\ &= B(z^{-1})F(z^{-1})R_k - F(z^{-1})L_{k+d_0-1} \\ &+ z^{-d_0}\frac{1}{A(z^{-1})}(B(z^{-1})R_k - L_{k+d_0-1}) \end{aligned}$$

or

$$Q_{k+d_0} = B(z^{-1})F(z^{-1})R_k - F(z^{-1})L_{k+d_0-1} + G(z^{-1})Q_k \tag{9}$$

As can be seen from Equation 9, the number of packets in the buffer at time interval $k + d_0$ (denoted as Q_{k+d_0}) can be represented as a combination of transmission rates and buffer occupancies of time interval k and the time intervals before k. It is also related to the playback rates of time interval k and the time intervals after k. Hence, this is a predictive formulation that can be used to predict the buffer occupancy at time interval $k + d_0$.

With this transformation, an optimal transmission rate can be obtained. When the packets are transmitted with this optimal transmission rate at time interval k, we can expect the number of packets that will arrive at the client at time intervals *k+1*, *k+2*, ..., etc. For an individual packet, we can schedule its transmission time instant to ensure that it can arrive at the client before the playback, since the delay that it will experience is considered in this model. This rate is said to be optimal because with this transmission rate, the performance index function in Equation 3 can be minimized. That is, the transmission rate is minimal and the buffer occupancy is maximal in the meaning of the sum of the values at all the time intervals. Hence, the objective function can be defined as follows.

$$\begin{aligned} J_k = &(w_p B(z^{-1})F(z^{-1})R_k - w_p F(z^{-1})L_{k+d_0-1} \\ &+ w_p G(z^{-1})Q_k - w_q Q_r)^2 + (w_r R_k)^2 \end{aligned} \tag{10}$$

Differentiate J_k with respect to R_k to obtain the optimal transmission rate R_k.

$$\begin{aligned} \frac{\partial J_k}{\partial R_k} = &2w_p b_{1,k}(w_p B(z^{-1})F(z^{-1})R_k - w_p F(z^{-1})L_{k+d_0-1} \\ &+ w_p G(z^{-1})Q_k - w_q Q_r) + 2(w_r)^2 R_k = 0. \end{aligned} \tag{11}$$

Solving for the optimal transmission rate at time interval k yields

$$((w_p)^2 B(z^{-1})F(z^{-1}) + \frac{1}{b_{1,k}}(w_r)^2)R_k =$$
$$-(w_p)^2 G(z^{-1})Q_k + w_p w_q Q_r + (w_p)^2 F(z^{-1})L_{k+d_0-1} \quad (12)$$

The above equation is a recursive equation for R_k represented in terms of R_{k-1}, R_{k-2}, ..., Q_k, Q_{k-1}, ..., and Q_r. All of them are known variables. In other words, the optimal transmission rate R_k depends on the buffer occupancy, the allocated buffer size, and the previous transmission rates.

In addition, the combination of different values of w_p, w_q, and w_r can result in different bandwidth allocation and buffer occupancies. For example, a higher value of w_r results in a smaller transmission rate but lower buffer occupancy, while a lower value of w_r results in a larger transmission rate but better buffer occupancy. The values of w_p, w_q, and w_r can be dynamically changed under different network congestion situations. Since this protocol is designed to operate on the Internet, it should incorporate robust end-to-end congestion control mechanisms that can ensure the fair sharing of the network bandwidth with the existing TCP traffic. Our congestion control is implemented via adjusting w_r in Equation 3. The value of w_r is adjusted (either doubled or decreased by one) according to the packet loss ratio observed at the client. When relatively a high packet loss rate is detected, w_r is doubled. Hence, the transmission rate can be decreased quickly to reduce the network traffic. However, if the new w_r value is greater than an upper bound (say, w_r_bound), then w_r is set to w_r_bound. On the other hand, when the packet loss rate is decreased, w_r is decreased by one in each adjustment period. However, if w_r is less than a lower bound (say, 1), then w_r is set to 1. In this way, the transmission rate can be increased slowly when the network is not congested. A packet loss rate threshold value can be defined to determine whether w_r needs to be doubled or decreased by one. Compared with those congestion control mechanisms that are similar to TCP, our proposed approach will result in a gradual increase of the transmission rate when w_r is decreased by one, and a less aggressive reduction of the rate when w_r is doubled (Shyu et al., 2002a). In addition, our approach does not need to change the transmission rate too frequently, and at the same time can achieve a smoother transmission rate. The advantage is that under different congestion statuses, an optimal utilization of the network bandwidth and the client buffer can still be achieved for all those clients connecting to the server.

After the transmission rate is obtained, it needs to be compared with the available network bandwidth to avoid congestion. Available bandwidth depends on two things. One is the underlying capacity of the path from each client to the server, which is limited by the bottleneck link. The other is the amount of other traffic competing for the links on the path. If the sum of the calculated transmission rates for all the clients exceeds the available bandwidth, the transmission rates should be reallocated according to the available

network bandwidth.

3 Experimental results

In order to test our quadratic probing algorithm, experiments are conducted to compare it with the fixed rate and the rate by playback requirement algorithms under two different network infrastructures, namely the Internet and the local area network (LAN).

3.1 General scenario of conducted experiments

Since the playback rate for a compressed video is highly unpredictable, the playback rates in our experiments were generated randomly. Different playback requirements are also considered in our experiments. For the fixed rate transmission, the bandwidth allocated to each user is constant. For the rate by playback requirement, the server allocates the bandwidth to the users according to their playback requirements. In our experiments, our proposed approach is denoted as Approach A, the rate by playback requirements approach as Approach B, and the fixed rate transmission approach as Approach C.

Table 1
Experiment Parameters.

Data size in one packet	1024 Bytes
Buffer capacity of each client	200 KBytes
Number of clients for LAN experiments	55
Number of clients for Internet experiments	15
Playback rates	0.1×10^5Bps - 0.8×10^5Bps

The parameters for our experiments are given in Table 1. As shown in this table, the size of the transmitted data in one packet is 1024 bytes, the maximum buffer capacity at each client is 200 packets or 200 KB (KBytes), and the playback rates are generated randomly between $[0.1 \times 10^5, 0.8 \times 10^5]$ Bps. In fact, experiments were also conducted for different sizes of the client buffer including 200, 300, 400, 500, 800, and 1000. In this paper, only the experimental results for the buffer capacity 200 KB are presented. One 40 MB (MBytes) video file and one 1 GB video file were sent from the server to the clients. In both of the experiments on LAN and the Internet, we prefetch some packets first. When the buffer occupancy reaches a certain threshold value, the playback starts.

Table 2
Different Levels of Network Congestion.

Playback Requirement	Range of playback rates
0.1×10^5 Bps - 0.3×10^5 Bps	Low range
0.4×10^5 Bps - 0.6×10^5 Bps	Medium range
0.7×10^5 Bps - 0.8×10^5 Bps	High range

3.2 Experiments on LAN

The experiments were conducted on ETHERNET at the School of Computer Science (SCS), Florida International University (FIU). The server and clients were on the machines running SunOS 5.8. The scenario of video on demand (VOD) was taken as an example, where multiple users were requesting a movie (MPEG file) from the video server. The experiments were run under three different ranges of the playback rates (given in Table 2).

With a higher playback requirement, the traffic will be larger because the users need to request more data from the server. Hence, the congestion level of the network is considered higher. After estimating the available bandwidth of our LAN with the pathload tool in (HTTP, 2003), we had a minimum available bandwidth of the link 67 Mbps. This pathload tool is based on the approach in (Jain and Dovrolis, 2002), and can estimate the available end-to-end bandwidth of the Internet paths. For a network path, the end-to-end available bandwidth is defined as the maximum rate that path can provide to a flow without reducing the rate of the rest of the traffic in this network path (Jain and Dovrolis, 2002). Since the data rate supported by the Ethernet card is large enough to satisfy the playback requirement, the bandwidth bottleneck does not occur at our LAN.

In order to compare Approach A with Approach B and Approach C under the same condition, the same video file was transmitted and the same random sequence of playback rates was generated. The number of users requesting the video file was 55. In Approach A, the server starts sending the video data to the clients with the rate 50 packets per second or 0.41 Mbps. In Approach B, the server starts sending the video data to the clients with the rate set up according to the peak playback requirements. The bandwidth in Approach C is allocated according to the peak playback requirements. In comparing Approach A with Approach B, in Approach B, we adjust the rates according to the playback requirements obtained from the feedback reports from the clients. In comparing Approach A with Approach C, the packets were sent with the fixed rate to all clients.

The playback requirement (0.1×10^5 Bps - 0.3×10^5 Bps) given in Table 2 is

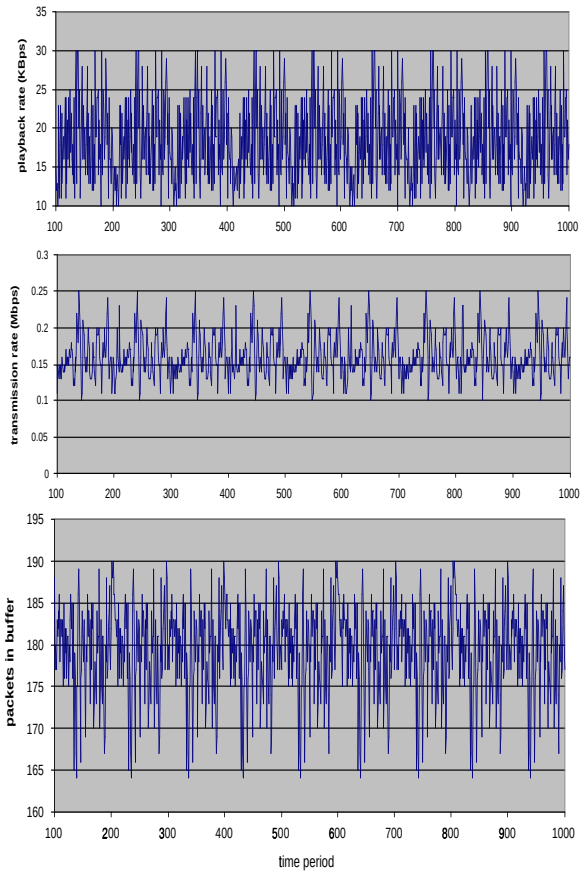

Fig. 1. Transmission rate changes with the numbers of packets in the buffer and the playback rates in the low range of playback rates during time intervals [100, 1000] for Approach A.

used to simulate the low range playback rate scenario. Fig. 1 shows how the transmission rate is adjusted according to the playback rates and buffer occupancies in Approach A during time intervals [100, 1000]. For each approach, a certain warm-up period is considered. It is evident from the figure that there are no overflows or underflows at the client buffer. Experiments were also run under the medium range and high range playback scenarios (described in Table 2). Fig. 2 shows how the transmission rate is adjusted according to the playback rates and buffer packets in Approach A during time intervals [1, 100] in the high range playback rate scenario. As can be seen from the figure, the transmission rate in Approach A is dynamically adjusted according to the packet loss requirements. For example, at the 43^{rd} time interval, we adapt the transmission rate to the congestion level of the network. Since at this time interval, we have a severe packet loss rate, the transmission rate is decreased. We do not have underflows at the client buffer because there are enough packets in the buffer to satisfy the playback requirements.

To illustrate the efficiency of our approach, we compare the transmission rates

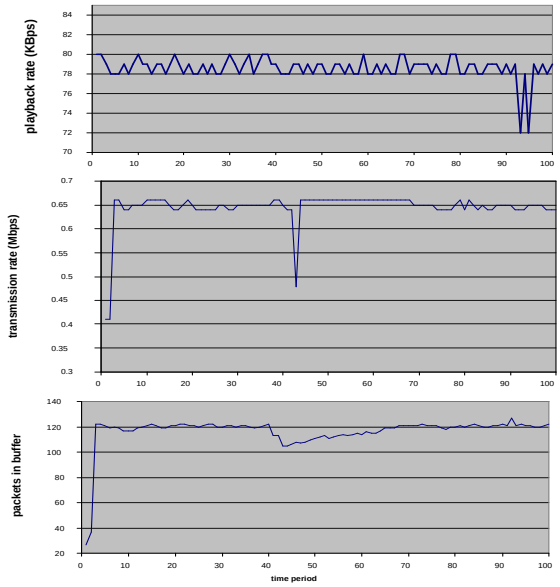

Fig. 2. Transmission rate changes with with the numbers of packets in the buffer and the playback rates in the high range of playback rates during time intervals [1, 100] for Approach A.

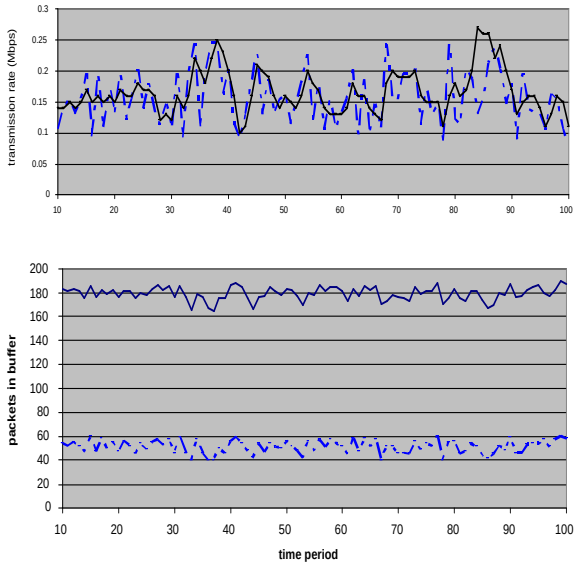

Fig. 3. Comparison of Approach A and Approach B in the transmission rates and the numbers of packets in the buffer, where Approach A is denoted as a solid line and Approach B is denoted as the dashed line in the low range of playback rates during time intervals [10,100].

and the numbers of packets in the client buffer with the rate by playback requirements approach (Approach B) under the same playback requirements in Fig. 3. As can be seen from Fig. 3, generally the buffer utilization is much better in our approach than in Approach B. Compared with Approach B, the

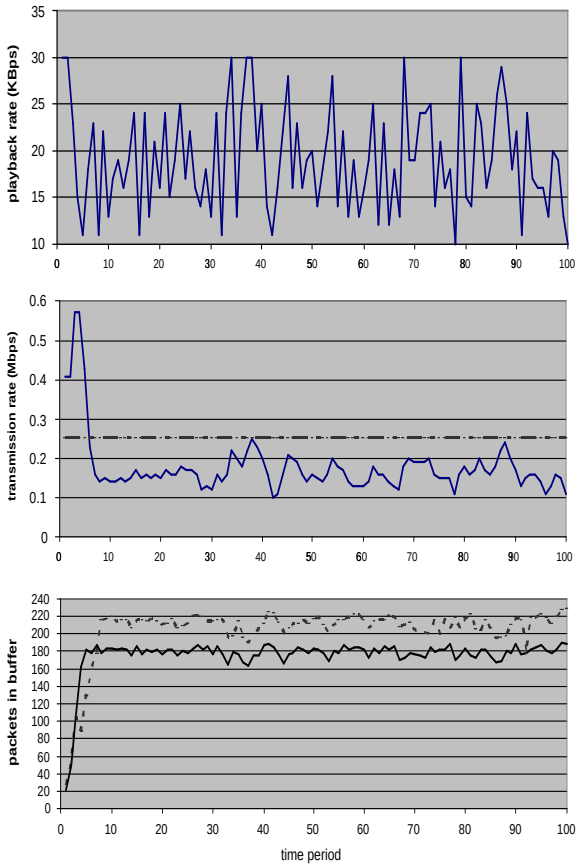

Fig. 4. Comparison of Approach A and Approach C in the transmission rates and the numbers of packets in the buffer, where Approach A is denoted as a solid line and Approach C is denoted as the dashed line in the low range of playback rates during time intervals [1, 100].

transmission rates in Approach A change more smoothly and less frequently. The transmission rates in Approach A also change in a smaller scope, which results in more stable arriving packet rates at the client side and therefore can produce a more stable presentation quality. Therefore, our approach is more robust in bandwidth adaptive allocation.

We also compare our approach with the fixed rate approach (Approach C). The bandwidth in Approach C is allocated according to the peak playback requirements, so the fixed rate is set as 30 packets per second or 0.25Mbps. The changes of the playback rates and buffer packets for Approach C are displayed in Fig. 4. It can easily be seen that the numbers of packets in the buffer in Approach C are continually increasing, so that an overflow occurs and it cannot be recovered. As soon as the buffer is filled with the packets, it has a severe overflow and a large number of packets will be discarded due to the buffer overflow at the client. On the other hand, as also shown in Fig. 4, the buffer utilization in Approach A is much better than that in Approach

C since there is no buffer overflow or underflow in Approach A, while there is a constant overflow in Approach C. In addition, the transmission rates in Approach C are generally higher than those in Approach A, which results in the waste of the network bandwidth.

The LAN experimental results also indicate that our approach performs better than Approach B and Approach C in terms of the bandwidth utilization and the occurrence of overflows/underflows.

3.3 Experiments on the Internet

For the Internet experiments, the server was run on one SunOS 5.8 machine in SCS at FIU and the clients were run on another SunOS 5.8 machine in the Department of Electrical and Computer Engineering at the University of Miami (UM). The same scenario used in the LAN experiments was considered in these experiments. The distance between the sender and receiver is 30 hops. Our experiments were conducted in the afternoon between 12 p.m. and 5 p.m. when the link was heavily loaded. In order to compare Approach A with Approach B and Approach C under the same condition, the same video file was transmitted and the same random sequence of playback rates was generated. The playback requirement (0.1×10^5 Bps - 0.3×10^5 Bps) given in Table 2 is also used to test the Internet network scenario. The number of users requesting a video file was 15.

Due to the compressed nature of each video stream, packet losses may significantly degrade the video quality. Since the packet loss is unavoidable in the Internet and excessive data loss will cause the pictures to jump or the audio stream to be lost, we focused our attention on the rate/congestion control and available bandwidth in the network as described in Section 2. In our experiments, the threshold value for the packet loss rate was chosen to be 8%.

To illustrate how our approach provides a better presentation quality at the client in the Internet via effectively avoiding underflows, the buffer occupancy of Approach A and Approach B under the medium range of playback rates is displayed in Fig. 5. As can be seen from Fig. 5, there is no underflow in Approach A while underflows occur frequently in Approach B. When there are the same number of clients requesting services from the server under the bandwidth limit, Approach A can provide a better service in terms of the occurrence of buffer underflows. We also display the buffer occupancy of Approach A and Approach B under the high range of playback rates in Fig. 6. Fig. 6 shows that the buffer occupancy of Approach B decreased with time; while the buffer occupancy of Approach A was kept at a more stable and much higher level compared with that of Approach B. To examine how our

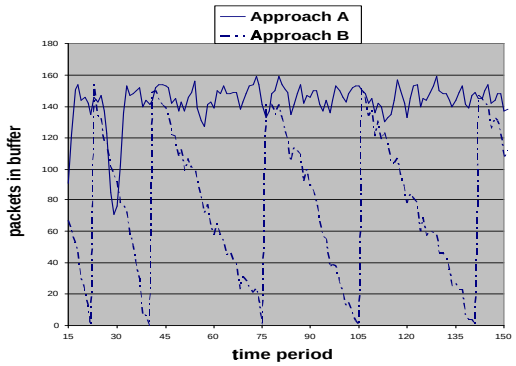

Fig. 5. Comparison of Approach A and Approach B in the numbers of packets in the buffer, where Approach A is denoted as a solid line and Approach B is denoted as the dashed line in the medium range of playback rates under the Internet during time intervals [15, 150].

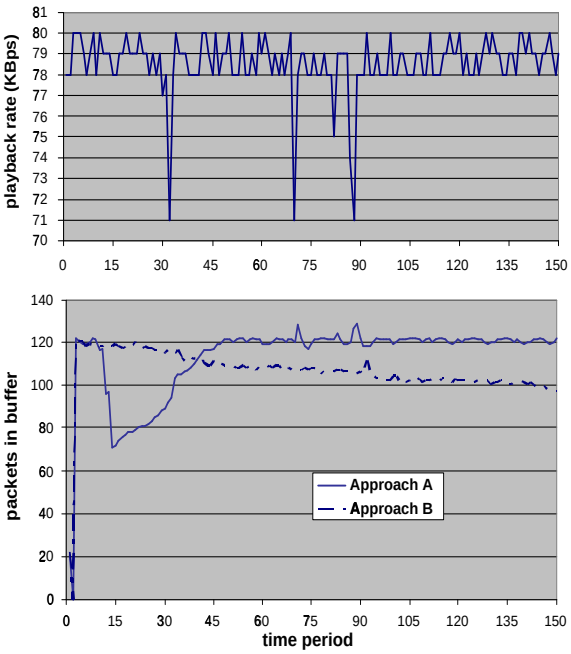

Fig. 6. Comparison of Approach A and Approach B in the numbers of packets in the buffer, where Approach A is denoted as a solid line and Approach B is denoted as the dashed line in the high range of playback rates under the Internet during time intervals [0, 150].

approach works under a broad range of playback rates (0.1×10^5 Bps - 0.8×10^5 Bps), the buffer occupancies of Approach A and Approach B are compared in Fig. 7. The buffer occupancy of Approach A in the broad range of playback rates changes more frequently than that in the high range of playback rates, but the buffer occupancy still maintains at a relative stable level. The buffer occupancy of Approach B also changes more frequently and has a decreasing trend.

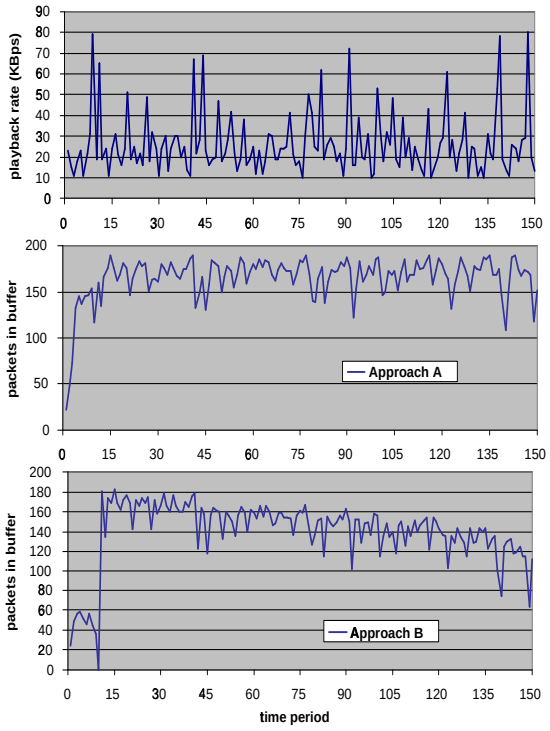

Fig. 7. Comparison of Approach A and Approach B in the numbers of packets in the buffer in the broad range of playback rates (0.1×10^5 Bps - 0.8×10^5 Bps) under the Internet during time intervals [0, 150].

To examine to what degree the buffer occupancy of Approach B will decrease, the comparison of the buffer occupancy of Approach B with that of Approach A in a longer time interval [1, 500] is displayed in Fig. 8. As shown in Fig. 8, the occupancy of Approach B decreases to zero periodically, which means that underflows are unavoidable and frequent in Approach B. While Approach A can still maintain a stable and high level of buffer occupancy without underflows. The adaptive advantages of our approach can still be obviously observed from the figures when the playback rates change in a broad range. Hence, our approach is more robust and adaptive compared to Approach B.

Experiments were also conducted to illustrate the performance when the transmission rate is adjusted according to the available bandwidth as the upper limit. After estimating the available bandwidth of the link between the FIU server and UM clients with the pathload tool we mentioned before, we had a minimum available bandwidth of the link 1.17 Mbps. The available bandwidth is then dynamically allocated among the clients, and our algorithm attempts to prevent packet losses by matching the rate of video streams to the available bandwidth in the network. Fig. 9 shows the comparison of the experiments with available bandwidth adjustment and without bandwidth adjustment in Approach A, where the experiment with available bandwidth adjustment is denoted as a solid line and the experiment without available bandwidth ad-

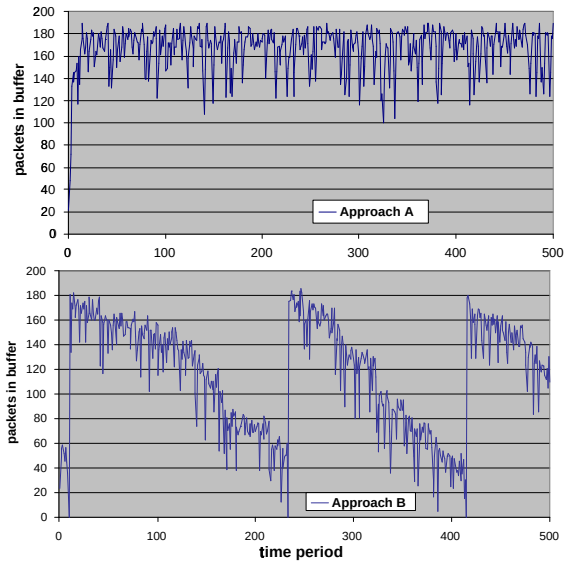

Fig. 8. Comparison of Approach A and Approach B in the numbers of packets in the buffer in the broad range of playback rates under the Internet during time intervals [1, 500].

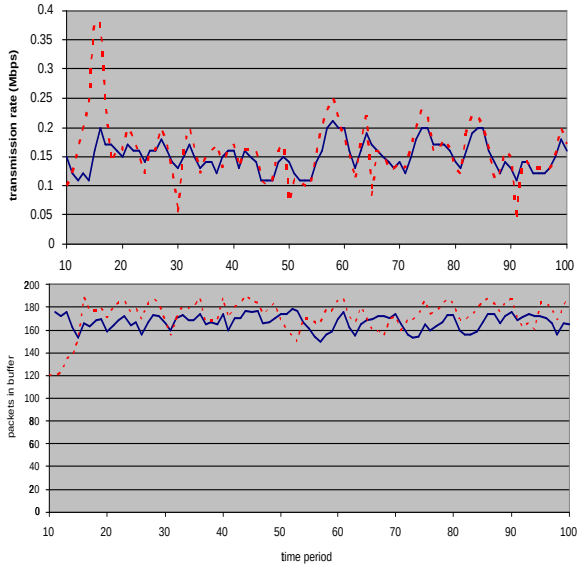

Fig. 9. Comparison of the experiments with available bandwidth adjustment (solid line) and without available bandwidth adjustment (dashed line) in the transmission rates and numbers of packets in the client buffer in the low range of playback rates under the Internet during time intervals [10,100] for Approach A.

justment is denoted as the dashed line.

To demonstrate the efficiency of our approach, we compare the packet loss rates under various experiments and the results are shown in Fig. 10. In this

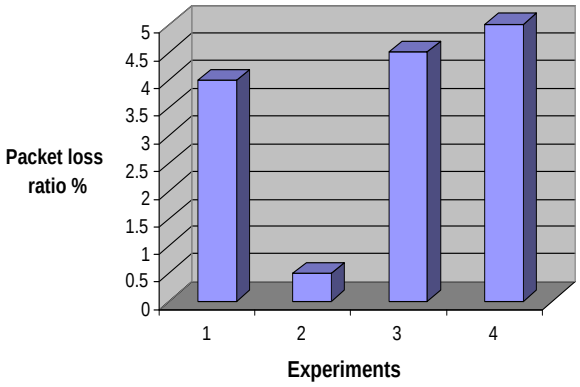

Fig. 10. Comparison of packet loss ratios in Experiment 1 (Approach A without bandwidth adjustment), Experiment 2 (Approach A with bandwidth adjustment), Experiment 3 (Approach B), and Experiment 4 (Approach C).

figure, the experiment without bandwidth adjustment and the experiment with available bandwidth adjustment for Approach A are denoted as Experiment 1 and Experiment 2, respectively. Experiments for Approach B and Approach C under the same playback requirements are denoted as Experiment 3 and Experiment 4. As can be seen from this figure, Experiment 2 (our proposed approach) performs the best with respect to the packet loss rates, which is less than 0.5%. Experiment 1 has a packet loss rate of more than 3.5%, which is still better than Appraoach B (with a packet loss rate almost 4.5%) and Approach C (with a packet loss rate almost 5%). It can be observed from Fig. 9 and Fig. 10 that available bandwidth adjustment can effectively reduce the packet loss ratio and obtain a more smoothly changed transmission rate, with only a slightly decreased buffer occupancy. Generally speaking, available bandwidth adjustment improves the performance in terms of packet loss ratios and smooth transmission rates.

From all these experiments, it is clearly shown that our proposed approach can achieve good performance in versatile network conditions. It is adaptive in the severe network traffic and congestion situations when the playback requirements change in the high range of playback rates. It is also adaptive when the network traffic changes drastically where the playback requirements change in a broad range of values. Experimental results also demonstrate that our proposed approach works well on both LAN and Internet.

4 Conclusions

In this paper, we proposed an end-to-end adaptive real-time multimedia transmission protocol with a quadratic probing algorithm for distributed multi-

media applications. The quadratic probing algorithm adjusts the transmission rate adaptively according to the client buffer occupancy, playback rates, changing network delay and packet loss ratio. The transmission rate obtained is minimal, and at the same time the client buffer utilization is maximized. When the packet loss ratio is high during network congestion, the congestion control mechanism can decrease the transmission rate, and thus decrease the packet loss ratio. Finally, the transmission rate is adjusted according to the network bandwidth so that our multimedia stream can share the bandwidth fairly with the existing TCP traffic.

To evaluate the efficiency of our algorithm, comparisons were made with the fixed rate algorithm and the rate by playback requirement algorithm. Experiments were run in different ranges of the playback rate scenarios in real networks (LAN and Internet) to show how our approach outperforms the other two approaches. The experimental results show that our proposed transmission protocol can provide better utilization of the client buffer and network bandwidth under different network congestion situations in both LAN and Internet. A much smoother transmission rate can be maintained, which is advantageous for a stable presentation quality at the client. That is, our proposed approach can provide efficient utilization of the network resources and reliable delivery of the multimedia data for distributed multimedia applications.

Acknowledgements

For Shu-Ching Chen, this work was supported in part by NSF EIA-0220562, NSF HRD-0317692, the office of the Provost/FIU Foundation, and Telecommunications & Information Technology Institute (IT2)/FIU under IT2 BA01. For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260.

References

Braden, R., Clark, D., Crowcroft, J., Davie, B., Deering, S., et al., 1998. Recommendation on Queue Management and Congestion Avoidance in the Internet, RFC 2309, Internet Engineering Task Force, April.

Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S., 1997. Resource ReSerVationProtocol (RSVP) - version 1 Functional Specification, Internet Engineering Task Force, Internet Draft, June.

Busse, I., Deffner, B., Schulzrinne, H., 1996. Dynamic QoS control of multimedia applications based on RTP, Computer Communications 19, January, 49-58.

Chen, S.-C., Shyu, M.-L., Gray, I., Luo, H., 2003. An adaptive multimedia transmission protocol for distributed multimedia applications. In: Proceedings of the 5th International Workshop on Multimedia Network Systems and Applications (MNSA'2003), in conjunction with The 23rd International Conference on Distributed Computing Systems (ICDCS-2003), Providence, Rhode Island, USA, May/June, 537-542.

Floyd, S., Fall, K., 1999. Promoting the use of end-to-end congestion control in the Internet, IEEE/ACM Transactions on Networking 7(4), August, 458-472.

Jacobs, S., Eleftheriadis, A., 1998. Streaming video using dynamic rate shaping and TCP congestion control, Journal of Visual Communication and Image Representation 9(3), January, 211-222.

Jain, M., Dovrolis, C., 2002. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In: Proceedings of ACM SIGCOMM, Pittsburgh, PA, USA, August, 295-308.

Kanakia, H., Mishra, P., Reibman, A., 1995. An adaptive congestion control scheme for real time packet video transport, IEEE/ACM Transactions on Networking 3(6), December, 671-682.

Kouvelas, I., Hardman, V., Crowcroft, J., 1998. Network adaptive continuous-media applications through self organized transcoding. In: Proceedings of Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98), Cambridge, UK, July 8-10.

Lewis, B., Berg, D. J., 1998. Multithreaded Programming with Pthreads, Sun Microsystems Press.

Lewis, F., Syrmos, L., 1995. Optimal Control, John Wiley & Sons, INC.

Padhye, J., Firoiu, V., Towsley, D., Krusoe, J., 1998. Modeling TCP throughput: A simple model and its empirical validation. In: Proceedings of SIGCOMM98, August, 303-314.

Rejaie, R., Handley, M., Estrin, D., 1999. Quality adaptation for congestion controlled video playback over the Internet. In: Proceedings of SIGCOMM99, August, 189-200.

Salehi, J., Zhang, Z., Kurose, J., Towsley, D., 1996. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing, In: ACM Sigmetrics Conference, Philadelphia, PA, USA, May, 222-231.

Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., 1996. RTP: A transport protocol for real-time applications, RFC 1889, January.

Shyu, M.-L., Chen, S.-C., Luo, H., 2002. End-to-end congestion control via optimal bandwidth allocation for multimedia streams. In: Proceedings of the 15th International Conference on Computer Applications in Industry and Engineering, San Diego, California, USA, November 7-9, 57-60.

Shyu, M.-L., Chen, S.-C., Luo, H., 2002. Optimal bandwidth allocation scheme with delay awareness in multimedia transmission. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Lausanne, Switzerland, August 26-29, 537-540.

Shyu, M.-L., Chen, S.-C., Luo, H., 2002. Self-adjusted network transmission for multimedia data. In: Proceedings of the Third IEEE Conference on Information Technology: Coding and Computing (ITCC-2002), Las Vegas, Nevada, USA, April 8-10, 128-133.

Sisalem, D., Schulzrinne, H., 1998. The loss-delay adjustment algorithm: A TCP-friendly adaptation scheme. In: Proceedings of Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98), Cambridge, UK, July 8-10.

Wang, X., Schulzrinne, H., 1999. Comparison of adaptive Internet multimedia applications, IEICE Trans. Commun. E82-B(6), June, 806-818.

Wu, D., Hou, Y.T., Zhang, Y.-Q., 2000. Transporting real-time video over the Internet: Challenges and approaches, Proceedings of the IEEE 88(12), December, 1-19.

http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/bw.html (Available in December 2003).

Shu-Ching Chen received his Ph.D. from the School of Electrical and Computer Engineering at Purdue University, West Lafayette, IN, USA in December 1998. He also received Masters degrees in Computer Science, Electrical Engineering, and Civil Engineering from Purdue University, West Lafayette, IN, USA. He has been an Assistant Professor in the School of Computer Science (SCS), Florida International University (FIU) since August 1999. He is currently the director of Distributed Multimedia Information System Laboratory and the Associate Director of Center for Advanced Distributed System Engineering (CADSE). His research interests include distributed multimedia database systems and information systems, data mining, databases, and multimedia communications and networking. Dr. Chen is authored and co-authored more than 100 research papers in various prestigious journals, refereed conference/symposium/workshop proceedings and book chapters. He received outstanding faculty research award from SCS at FIU in 2002. He was the General co-chair of the 2003 IEEE International Conference on Information Reuse and Integration, and the program co-chairs of the 10th ACM International Symposium on Advances in Geographic Information Systems and the First ACM International Workshop on Multimedia Databases.

Mei-Ling Shyu received her Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA in 1999, and her three master degrees from Computer Science, Electrical Engineering, and Restaurant, Hotel, Institutional, and Tourism Management from Purdue University. She has been an Assistant Professor at the Department of Electrical and Computer Engineering, University of Miami since January 2000. Her research interests include data mining, multimedia communications and networking, multimedia database systems, and multimedia information systems. She has authored and co-authored more than 80 technical papers published

in various prestigious journals, referred conference/symposium/workshop proceedings and book chapters. She was the program co-chair of the First ACM International Workshop on Multimedia Databases.

Irina Gray received her Master of Science from the School of Computer Science, Florida International University, Miami, FL, USA in December 2002, and her Master of Science in Electrical and Mechanical Engineering from Volgograd State Technical University, Volgograd, Russia in June 1981. Her research interest includes multimedia networking, Software Engineering, and Operating Systems.

Hongli Luo received her B.S. and M.S. degrees in electrical engineering from Hunan University, Hunan, China, in 1993 and 1996. She is currently pursuing the Ph.D. degree at the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL, USA. Her main research interest is in multimedia networking.