An Adaptive Multimedia Transmission Protocol for Distributed Multimedia Applications

Shu-Ching Chen[1], Mei-Ling Shyu[2], Irina Gray[1], Hongli Luo[2]

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

[2]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 transmission rate is determined by the quadratic probing algorithm that can obtain the maximal utilization of the client buffer and minimal allocation 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 Local Area Net (LAN) and Internet environment. Performance analysis reveals that our approach is more robust in avoiding overflow and underflow 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. Experimental results show that the proposed real time protocol with rate adjusting quadratic probing algorithm is efficient in utilizing the network resources and decreasing the packet loss ratios.

Keywords: Multimedia streaming, protocol, Quality of Service (QoS), adaptive transmission rates.

1. Introduction

Distributed multimedia applications have different Quality of Service (QoS) requirements. For example, a video conferencing service requires interactivity, low jitter, low delay and higher bandwidth but can tolerate some transmission errors. Even in a pure video transmission system, different parts of the video data need different transmission capabilities according to loss and delay sensitivity. The application must be aware of the conditions of the network.

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 the available network resources (which change with time) compared to those approaches relying on resource reservation [9]. According to [8], 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 that are discussed here fall into two strategies: buffer based and loss based. Buffer based adaptation schemes use the occupancy of a buffer on the transmission path as a measure of congestion [10][11]. Loss based adaptation schemes adjust the rate based on the packet loss experienced by the receivers [5][6].

Moreover, common requirements for multimedia applications have led to the design of the general purposed protocol such as the Real-Time Transport Protocol (RTP). The requirements include the ability to communicate the coding scheme, mechanisms that facilitate the application-specific handling of time-stamped data so that the receiver can play it at the appropriate time, the synchronization of multiple media, the indication of packet losses, and notification of the sender that the losses are occurring. Although RTP provides the functionality suitable for carrying real-time content [1] and is the primer protocol for real-time applications, it cannot provide any form of reliability or a protocol-defined flow/congestion control. However, on the other hand, the existing RTP has a flexible mechanism that allows the designers to build up the functionality required by the particular application.

In this paper, we implement a real-time multimedia transmission protocol for multimedia transmitting. It is source rate adaptive, where the transmission rate is adjusted according to the number of packets in the client buffer, available bandwidth, network delay and packet loss rate. At the same time, it can achieve an efficient utilization of network resources such as bandwidth and client buffer.

The paper is organized as follows. In the next section, the adaptive multimedia transmission protocol with quadratic probing algorithm is presented. 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

In this section, we will present the architecture of the system, data packet format and the control flow part of our protocol, and then a short introduction of the quadratic probing algorithm is given.

2.1 Protocol Design

The proposed protocol consists of two parts: a data part and a control part. The control part provides feedback on the performance of the application and the network by periodically sending the control information associated with a certain data flow. Our system is a distributed multimedia system, which is constructed upon the client/server architecture. The client and the server, implemented in C++, use UDP APIs for the connections between the UNIX Sockets and run on Sun SPARC/SunOS platform. The server and client both use Posix threads [3] for the creation of the multithreaded environment.

Each packet for sending data 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 detection and measurement of the lost and misordered packets. 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 sent this packet (i.e., 6 bytes).
- The time period to which this packet belongs (3 bytes): 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).

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

2.2 Quadratic Probing Algorithm with Congestion Control

The transmission rate is determined by the quadratic probing algorithm. This algorithm aims at achieving maximal utilization of the client buffer and minimal allocation of the bandwidth. It is also coupled with a congestion control scheme 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. For the time interval k, R_k is the transmission rate of the multimedia server, P_k is an incoming packet rate at the client buffer, Q_r is an allocated buffer size for each client, Q_k is the total size of the packets in the client buffer, and L_k is the playback rate. For the total size of the packets in the client buffer Q_{k+1} at the time interval $k+1$, we can describe the relationships among Q_k, R_k, P_k, and L_k with the following equation [2].

$$Q_{k+1} = Q_k + P_k - L_k \qquad (1)$$

Because the network delays tend to vary with time, P_k can be described as a function of R_{k-d}, R_{k-d-1}, …, and $R_{k-d-i+1}$, as displayed in Equation (2).

$$P_k = b_{1,k} R_{k-d} + b_{2,k} R_{k-d-1} + ... + b_{i,k} R_{k-d-i+1} \qquad (2)$$

where for the packets that arrived at the client buffer at time period k:

- $k\text{-}d$ is the closest time period that a packet is transmitted from the server;
- $k\text{-}d\text{-}i\text{+}1$ is the farthest time period that a packet is transmitted from the server; and
- b_i is the percentage of the packets, transmitted at $k\text{-}d$, $k\text{-}d\text{-}1$,…$k\text{-}d\text{-}i\text{+}1$ time periods and arrived at time period k.

To avoid the jitter and to use the client buffer effectively, the difference between the size of the packets in the client buffer and the allocated buffer size 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 [2] is defined as follows.

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

where

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

The obtained R_k is the optimal transmission rate sequence that satisfies Equation (3). Our congestion control scheme 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, which results in a gradual increase of the transmission rate. The detail of our congestion control scheme is described in [3]. If the total of the transmission rates of all the clients exceeds the available bandwidth, the transmission rates should be reallocated according to the available bandwidth.

3. Experimental Results

Our quadratic probing algorithm is compared to the fixed rate and the rate by playback requirement algorithms, the experiments are conducted on two different network infrastructures, the Internet and local area network (LAN).

3.1 General Scenario of Conducted Experiments

Since the playback rate 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 a constant bandwidth. For the rate by playback requirement, the server allocates the bandwidth to the users according to their playback requirements. In our experiments, our rate adjustment approach is denoted as Approach A, the rate by playback requirements approach is denoted as Approach B, and the fixed rate transmission approach is denoted as Approach C.

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 size at each client is 200 packets or 200 KB (Kbytes), and the playback rates are generated randomly between [0.1×10^5, 0.8×10^5] MBps for the experiments. One 40 MBytes video file and one 1 GB video file were sent from the server to the clients.

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 playback rates (given in Table 2).

Table 1. Experiment Parameters

Data size in one packet	1024 Bytes
Client buffer occupancy	200 Kbytes
Number of clients for LAN experiments	55
Number of clients for Internet experiments	15
Playback rates	0.1×10^5 - 0.8×10^5 MBps

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

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 pathload tool in [7], we had minimum available bandwidth of the link 67 Mbps. Since the data rate supported by the Ethernet card is larger 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 report from the clients. In comparing Approach A with Approach C, the packets were sent with the fixed rate.

The playback requirement (0.1×10^5 Bps ~ 0.3×10^5 Bps) given in Table 2 is used to simulate the low range playback rate scenario. Figure 1 shows how the transmission rate is adjusted according to the playback rates and the number of packets 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.

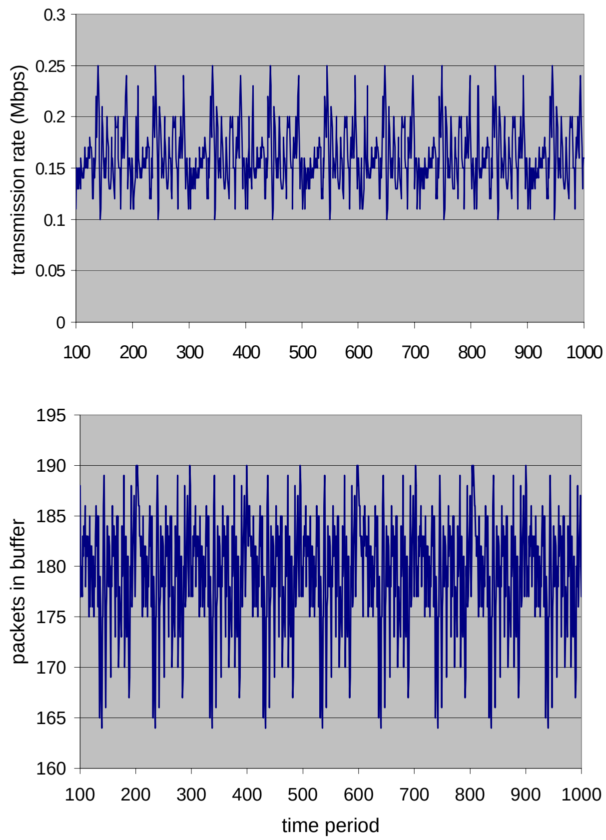

Figure 1. Transmission rate changes with numbers of packets in the client buffer and playback rates in the low range of playback rates during time intervals [100, 1000] for Approach A.

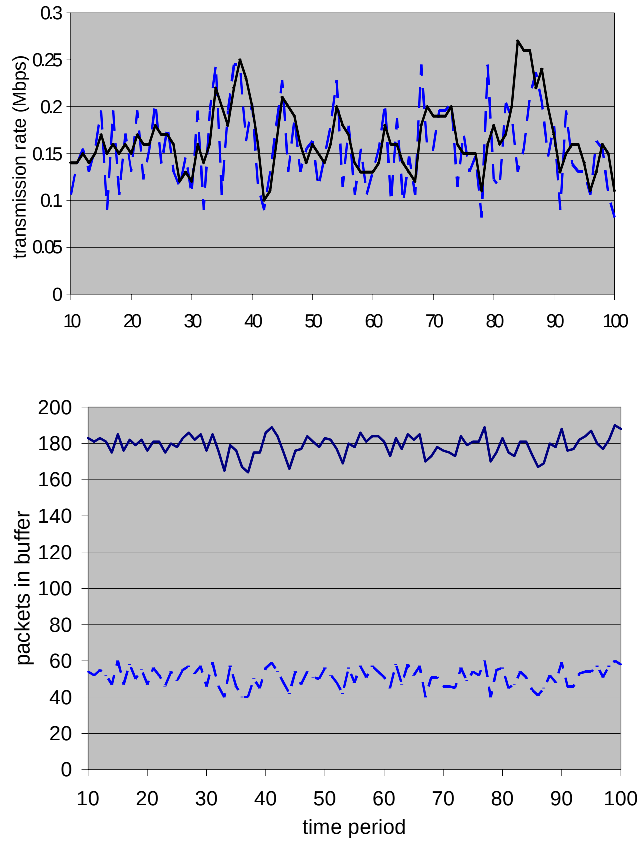

Figure 2. Comparison of Approach A and Approach B in transmission rates and number of packets in the client buffer, where Approach A is denoted as a solid line and Approach B is denoted as the dashed line in the low range of playback rates during time intervals [10,100].

To illustrate the efficiency of our approach, we compare the transmission rates and numbers of packets in the client buffer with the rate by playback requirements approach (Approach B) under the same playback requirements in Figure 2. As can be seen from Figure 2, generally the buffer utilization is much better in our approach than in Approach B. Compared with Approach B, the 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.

Experiments were also run under the medium range and high range playback scenarios (described in Table 2). Due to space limitation, the figures are not shown in this paper. However, the experimental results also indicated that our approach performs better than Approach B and Approach C in terms of bandwidth utilization and the occurrence of overflow/underflow.

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 Department of Electrical and Computer Engineering at the University of Miami (UM). The same scenario used in 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 rate was generated. The playback requirement ($0.1\text{x}10^5$ Bps ~ $0.3\text{x}10^5$ Bps) given in Table 2 is also used to test the Internet network scenario. The number of users requesting a video file was 15.

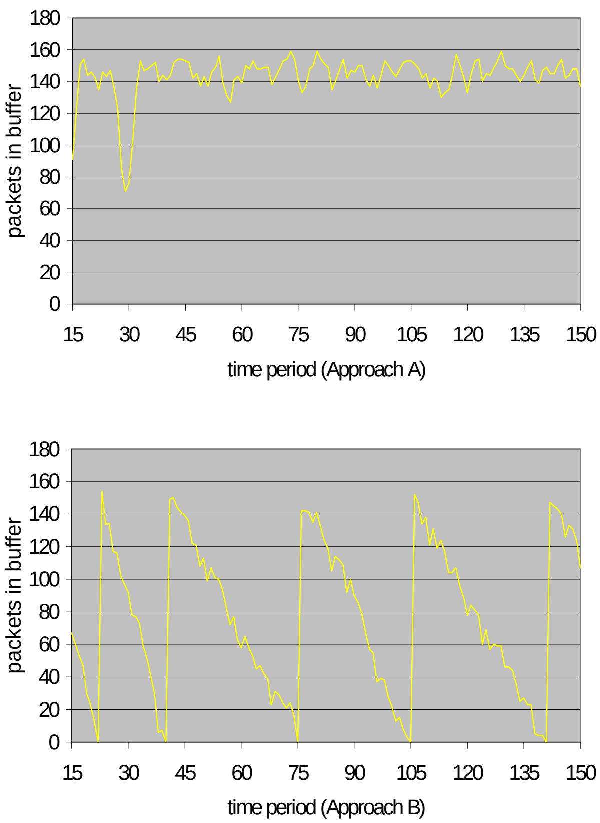

Figure 3. Comparison of Approach A (top) and Approach B (bottom) in number of packets in the client buffer in the medium range of playback rates under the Internet network during time intervals [15, 150].

Due to the compressed nature of each video stream, packet loss may significantly degrade the video quality. Since the packet loss is unavoidable in the Internet and excessive data loss will cause 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 described in Section 2. 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 Internet via effectively avoiding underflow, the buffer occupancies of Approach A and Approach B are displayed in Figure 3. As can be seen from Figure 3, there is no underflow in Approach A while underflows occur frequently in Approach B. When there are the same numbers 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 underflow.

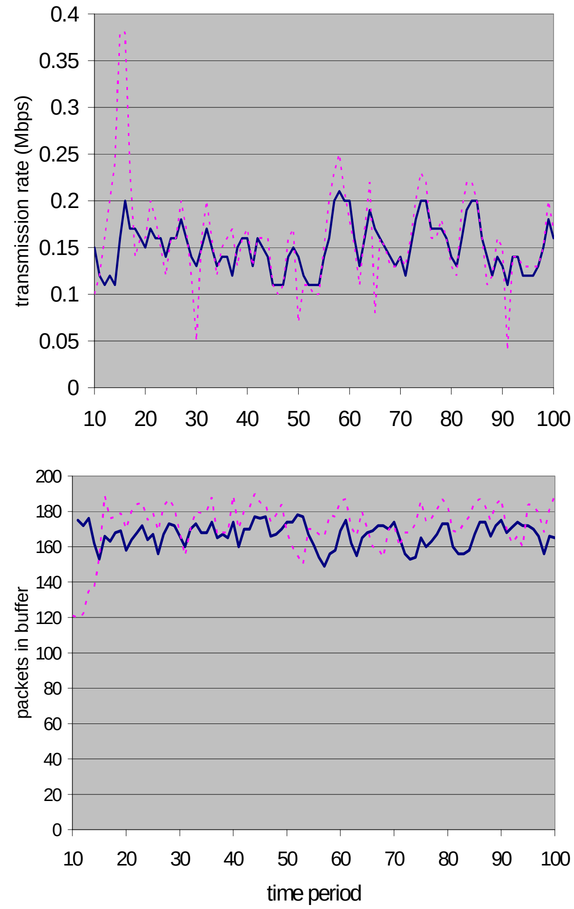

Figure 4. Comparison of experiments with available bandwidth control (solid line) and without available bandwidth control (dashed line) in transmission rates and numbers of packets in the client buffer under Internet network during time intervals [10,100] for Approach A.

Experiments were also conducted to illustrate the efficiency of our congestion control mechanism. After estimating the available bandwidth of the link between the FIU server and UM clients with the pathload tool in [7], we had minimum available bandwidth of the link 1.17 Mbps. Our algorithm prevents packet loss by matching the rate of video streams to the available bandwidth in the network. Figure 4 shows the comparison of the experiments with available bandwidth control and without bandwidth control in Approach A, where the experiment with available bandwidth control is denoted as a solid line and the experiment without bandwidth control is denoted as the dashed line.

We also compare the packet lose rates under various experiments and the results are shown in Figure 5. In this figure, the experiment without bandwidth control and the experiment with available bandwidth control for Approach A are denoted as experiment 1 and experiment 2, respectively. Experiments for approaches B and C under the same playback requirements are denoted as experiment 3 and experiment 4. Our proposed approach (Approach A with bandwidth control) performs the best with respect to the packet loss rates, which is less than 0.5%. Experiment 1 (Approach A without bandwidth control) has packet loss rate of more than 3.5%, which is still better than approach B (with packet loss rate almost 4.5%) and approach C (with packet loss rate almost 5%). From the results in Figure 4 and Figure 5, with the bandwidth control, even the buffer occupancy is smaller than that without bandwidth control, still no underflow occurs. At the same time, the packet loss rate is greatly decreased.

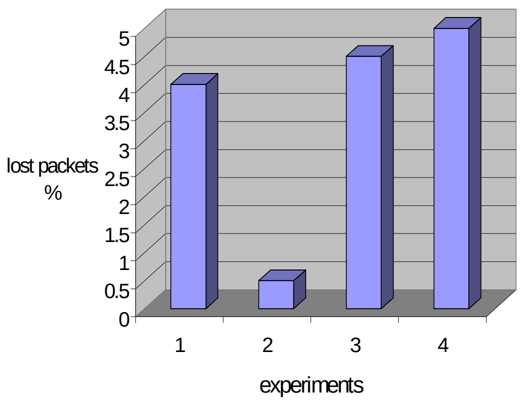

Figure 5. Comparison of packet loss rates in experiment 1 (Approach A without bandwidth control), experiment 2 (Approach A with bandwidth control), experiment 3 (Approach B), and experiment 4 (Approach C).

4. Conclusions

In this paper, we proposed an end-to-end adaptive real-time multimedia transmission protocol with a quadratic robing algorithm for distributed multimedia applications. The quadratic probing algorithm adjusts the transmission rate adaptively according to the number of packets in the client buffer, playback rates, changing network delay and packet loss rate. The transmission rate achieved is minimal, and at the same time the client buffer utilization is maximized. Congestion control scheme is used to decrease the packet loss rates caused by network congestion. Comparisons were made with the fixed rate algorithm and the rate by playback requirements algorithm. Experiments were run in the different ranges of playback rates scenarios in real network (LAN and Internet) to show how our approach outperforms the other two approaches under different network congestion levels. Experimental results show that our proposed protocol can provide efficient utilization of network resources and reliable delivery of multimedia data for distributed multimedia applications.

5. Acknowledgement

This work was supported in part by NSF EIA-0220562 and Telecommunications & Information Technology Institute (IT2)/FIU under IT2 BA01.

6. References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real –Time Applications," *RFC 1889*, January 1996.

[2] M.-L Shyu, S.-C. Chen, and H. Luo, "Optimal Bandwidth Allocation Scheme with Delay Awareness in Multimedia Transmission," *IEEE International Conference on Multimedia and Expo (ICME)*, Lausanne, Switzerland, August 26-29, 2002, pp. 537-540.

[3] M.-L Shyu, S.-C. Chen, and H. Luo, "End-to-End Congestion Control via Optimal Bandwidth Allocation for Multimedia Streams," *15th International Conference on Computer Applications in Industry and Engineering*, San Diego, California, USA, November 7-9, 2002, pp. 57-60.

[4] B. Lewis and D.J. Berg "Multithreaded Programming with Pthreads," *Sun Microsystems Press*, 1998.

[5] D. Sisalem and H. Schulzrinne, "The Loss-Delay Adjustment Algorithm: A TCP-friendly Adaptation Scheme," *Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Cambridge, UK, July 8-10, 1998.

[6] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS Control of Multimedia Applications Based on RTP," *Computer Communications*, Jan. 1996, vol. 19, pp. 49-58.

[7] http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/bw.html

[8] X. Wang and H. Schulzrinne, "Comparison of Adaptive Internet Multimedia Applications," *IEICE Trans. Commun.*, June 1999, Vol. E82-B, No.6, pp. 806-818.

[9] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVationProtocol (RSVP) – version 1 Functional Specification," *Internet Engineering Task Force, Internet Draft*, June 1997.

[10] H. Kanakia, P. Mishra, and A. Reibman, "An Adaptive Congestion Control Scheme for Real-time Packet Video Transport," in *SIGCOMM Symposium on Communications Architectures and Protocols* (San Francisco, California), Sept. 1993, pp. 20-31.

[11] S. Jacobs and A. Eleftheriadis, "Streaming Video using Dynamic Rate Shaping and TCP Congestion Control," *Journal of Visual Communication and Image Representation*, Jan. 1998, Vol. 9, No. 3, pp. 211-222.