End-to-End Congestion Control via Optimal Bandwidth Allocation for Multimedia Streams

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

[1]Department of Electrical and Computer Engineering, University of Miami
Coral Gables, FL 33124, USA
[2]Distributed Multimedia Information System Laboratory, School of Computer Science
Florida International University, Miami, FL 33199, USA

Abstract

In this paper, an end-to-end congestion controlled optimal bandwidth allocation scheme with a transmission rate control mechanism for multimedia transmission is proposed. This rate control mechanism aims at minimizing the allocated rate and maximizing the utilization of the client buffer. By adjusting the transmission rate in response to the buffer occupancy and the playback requirements in the client, and variations in the network delay and packet loss rate, an acceptable quality-of-service (QoS) level can be maintained at the end systems. Simulation results show that our proposed scheme can generally achieve high network utilization and low losses, and no network support is needed.

Index Terms – Congestion control, adaptive transmission, QoS, bandwidth allocation, network resource optimization.

1. Introduction

Streaming multimedia applications such as video and audio has become increasingly popular on the Internet. Transmission of real-time video has bandwidth, delay and loss requirements [8]. An efficient multimedia transmission scheme should be adaptive to packet losses, variable bandwidth, and variable network delays. It also should make fully utilization of the available network resources such as bandwidth and buffer (at the client or in the intermediate router), and satisfy the requirement of bandwidth, delay and packet loss. Congestion control should also be implemented to avoid inter-protocol unfairness and even network congestion collapse. Internet is a shared best-effort environment where the end systems are expected to be cooperative by responding to network congestion properly and promptly [3]. End-to-end congestion control can achieve higher utilization of network resources and improves inter-protocol fairness.

It has been shown in [2] that the Additive Increase and Multiplicative Decrease (AIMD) algorithm efficiently converges to a fair state. For real-time stream transmission, some rate adaptation approaches have been proposed to realize inter-protocol fairness and particularly TCP-friendly. A mechanism that dynamically adjusts the bandwidth requirement of multimedia applications was introduced in [1]. Based on the congestion states seen by the receivers, the bandwidth is adjusted by a linear regulator with dead zone. TCP-friendly protocols employing rate-based control to compete fairly with other TCP flows for bandwidth were proposed in [4], which tried to stabilize the throughput and reduce the jitter for multimedia streams. In [6], an end-to-end TCP-friendly Rate Adaptation Protocol was presented, which employs an AIMD algorithm. However, most of these approaches do not address the problem of optimally utilize the available bandwidth and the client buffer. In this paper, we propose an end-to-end congestion controlled optimal bandwidth allocation mechanism that can achieve the optimal utilization of bandwidth and client buffer, and at the same time does not starve TCP flows during congestion.

The paper is organized as follows. In the next section, the end-to-end congestion controlled optimal bandwidth allocation mechanism is presented. Simulations are given in Section 3. Conclusions are presented in Section 4.

2. End-to-End Congestion Controlled Optimal Bandwidth Allocation Mechanism

Assume there is an end-to-end transmission from the server to the client. When the transmission rate at the server and the playback rate at the client are different, the client buffer is used to accommodate the difference between them. With the buffering of some packets and slightly delaying the playback schedule, the client can decrease the playback jitter caused by the variations of network bandwidth and the end-to-end delay. Hence, we try to maximize the utilization of the client buffer to provide a better QoS for the playback.

Our proposed mechanism is an end-to-end adaptive rate control mechanism that is suitable for real-time multimedia streams. The server sends the packets with sequence numbers, and the client acknowledges the received packets at a regular time period to provide end-to-end feedback. Each acknowledgement (ACK) contains the percentage of the packets arriving at the client (indicating the network delay), the packet loss information and the client buffer occupancy. No explicit congestion signal from the network is needed to obtain the optimal transmission rate since it is based on the packet arrival pattern.

2.1. Optimal Bandwidth Allocation Algorithm

Assume at time interval k, the buffer occupancy is Q_k, packets transmitted from the server is R_k, packets arriving at the client buffer is P_k, packets used for playback is L_k, and the allocated buffer size for each client at the setup of the connection is Q_r. Here a time interval can be one or several round trip times (RTTs). In our proposed framework, it is assumed that the values for L_k's are known. Let Q_{k+1} denote the buffer occupancy at time interval $k+1$, we have

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

However, because of the changeable network delays, P_k is not equal to R_k at time interval k. Assume that the packets arriving at the client buffer at time interval k comprise of packets transmitted from the server at the time interval k-d, k-d-1, …, and k-d-i+1, and let $b_{i,k}$ denote the percentage of the packets transmitted at $R_{k-d-i+1}$ that arrives at the client buffer at time interval k to capture the scenario of changing network delays. P_k can be represented as

$$P_k = b_{1,k} R_{k-d} + b_{2,k} R_{k-d-1} + \ldots + b_{i,k} R_{k-d-i+1} \tag{2}$$

where the subscript k-d denotes the closest time interval when the transmitted packet can arrive at the buffer, and k-d-i+1 denotes the farthest time interval when the transmitted packet can arrive at the buffer at time interval k. The values of $b_{i,k}$'s can be obtained through the protocol. Each packet is appended a timestamp when transmitted from the server. In order to fully utilize the network resources such as the network bandwidth and client buffers, we need to maximize the utilization of client buffers and minimize the bandwidth allocation. The quadratic performance index we try to minimize is set as

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

where w_p, w_q, and w_r are the weighting coefficients, and d_0 is referred to as the transmission control delay. Different weighting coefficients can be selected to specify a wide range of adaptive rate controllers that result in the desirable closed-loop behavior. The objective is to select a transmission rate sequence R_k to minimize J_k for a given buffer size Q_r. The optimal transmission rate at time interval k is

$$\begin{aligned}&(w_p^2 B(z^{-1})F(z^{-1}) + \frac{1}{b_1} w_r^2) R_k \\ &= -w_p^2 G(z^{-1}) Q_k + w_p w_q Q_r + w_p^2 F(z^{-1}) L_{k+d_0-1}\end{aligned} \tag{4}$$

The detail derivation of the optimal transmission rate is given in [7]. 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.

Assume there are m active clients requesting service from the server, and the available bandwidth at the server is BW. According to the corresponding Equation (3), let $J_{j,k}$, be the cost function of the j^{th} client, $e_{j,k}$ be the difference between the allocated buffer size and the buffer occupancy of the j^{th} client at time interval k, $R_{j,k}$ be the transmission rate of the j^{th} client at time interval k, $Q_{j,k+d_0}$ be the buffer packet of the j^{th} client at time interval $k+d_0$, and $Q_{j,r}$ be the allocated buffer size of the j^{th} client, we have

$$e_{j,k} = w_p Q_{j,k+d_0} - w_q Q_{j,r} \tag{5}$$

For each client, Equation (3) can be used to obtain the optimal transmission rate. The optimization function for resource allocation of the server should be

$$J = \sum_{j=1}^{m} J_{j,k} = \sum_{j=1}^{m} (e_{j,k}^2 + (w_r R_{j,k})^2) \tag{6}$$

Since the m clients are independent, if the $J_{j,k}$ value of the j^{th} client in Equation (6) is minimal, the sum of the $J_{j,k}$ functions (i.e., the J value) is also minimal. Hence, we can obtain the optimal transmission rate for each client and at the same time achieve the optimization of the performance index in Equation (6). If the sum of all the calculated bandwidth is greater than the network bandwidth, we need to reallocate the bandwidth to each client connection. To provide fairness among all clients, the bandwidth reallocated should be proportional to the actual requirement of each client.

2.2. End-to-End Congestion Control

In the Internet environment, all sources are expected to react to congestion by adapting their transmission rates. This can avoid the network congestion collapse, and at the same time keep the network utilization high. Real-time streams should also perform TCP-friendly congestion control. Real-time flows prefer smoother changes in the transmission rates so that a more stable presentation quality can be provided. For real-time flows, there are some AIMD congestion control mechanisms that do not use a decrease-by-half reduction in response to congestion. For example, [5] was based on AIMD, but the transmission rate was reduced to 7/8 of the previous value in response to a packet drop.

Our approach is TCP-friendly, which means that the transmission rate will decrease when network congestion is detected. The transmission rate is adjusted in response to the congestion status in a way similar to the idea of AIMD. The value of the weight w_r in Equation (6) is adjusted corresponding to different packet loss rates perceived by the clients. 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 and alleviate the congestion status of the whole network. A packet loss rate threshold value can be defined to determine whether w_r needs to be doubled or decreased by one. 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. Our approach can also achieve a smoother transmission rate since we do not directly reduce the rate to half during congestion. Instead, we reduce the weight value for the transmission rate to half. This results in a rate that is normally a little larger than half of the previous rate after the adjustment.

3. Simulation Results

The ns2 simulator is used in our simulations. The objective of the simulations is to demonstrate that our approach is TCP-friendly that can adjust the transmission rate in response to network congestion in a way similar to TCP. In other words, the adaptive flow can share the network bandwidth fairly with the traditional TCP flow in a best-effort network environment, e.g., the Internet. The simulation parameters are summarized in the Table 1.

Table 1. Simulation Parameters

Packet Size	1000Bytes
Bottleneck Bandwidth	2Mbps
Bottleneck Delay	10ms
Delay of Other Links	3ms
w_r bound	8
Adjust Period	50ms
ACK Size	40Bytes
Playback Rate	1Mbps
Size of Client Buffer	1/8MBbytes
Bandwidth of Other Links	5Mb
Packet_loss_rate_threshold	0.95

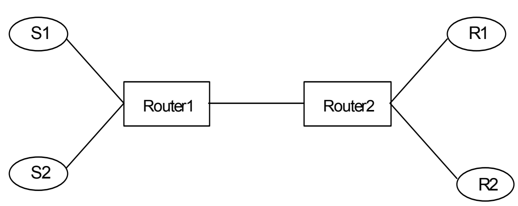

Figure 1. Simulation Topology

Figure 1 shows the topology of our simulations. The link between Router1 and Router2 is the bottleneck link, and Router1 is the bottleneck point where most of the packet drops occur. The switches implement FIFO scheduling and DropTail queuing. The properties of our mechanism are demonstrated when competing with the background TCP traffic. To better show how our approach can adapt the rate according to the network congestion status, a fixed playback rate is used in the simulation. Here we use the FTP session as the TCP flow, sending from S2 to R2. The adaptive real-time multimedia flow is sent from S1 to R1. The TCP flow and adaptive multimedia flow share the bottleneck bandwidth from Router1 to Router2. For a fair comparison, the end-to-end delay for the TCP flow and adaptive multimedia flow is the same. Link delays are also the same except the bottleneck link.

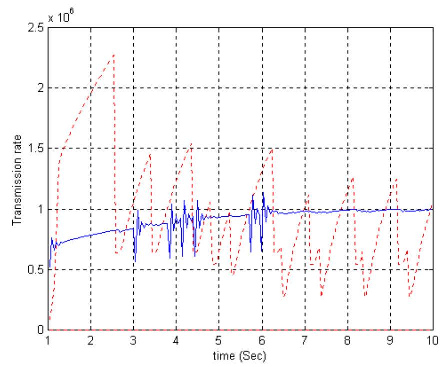

Figure 2. Transmission rates of the TCP flow (dotted line) and the multimedia flow (solid line)

Figure 2 compares the transmission rates of the multimedia flow and TCP flow. As can be seen from the figure, the transmission rate of the multimedia flow varies in the range of about 30% of the average rate. The multimedia flow adjusts its rate in a smoother way compared with AIMD since the adjusted rate is not reduced to the half of its previous rate in response to congestion. Instead, the adjusted rate is a little larger than half of the previous rate. It also occilates less frequently than the TCP flow. The occilation results from the congestion status feedbacked by the client. The source adjusts the transmission rates when the feedback information arrives at each interval. Figure 3 shows the packet loss rate perceived at the client. When the packet loss occurs (e.g., at 4^{th} second and 6.5^{th} second), the transmission rate is reduced in response to the packet loss. At the same time, TCP connection also reduces its transmission window, which leads to a congestion reduction. The packet loss rate peceived at the client is therefore reduced. This reduced congestion state will then be feedbacked to the server. If the packet loss rate is reduced to a threshold value, the servr will calculate a transmission rate with the new weight (decreased by one). In this way, the transmission rate will be increased slowly. The simulation results demonstrate that our approach can dynamically adjust the transmission rate according to the congestion level observed by the client.

The source adopts a smaller weight value during the less congestion periods, which results in a larger transmission rate. During the more severe congestion periods, the source adopts a larger weight value, which decreases the transmission rate.

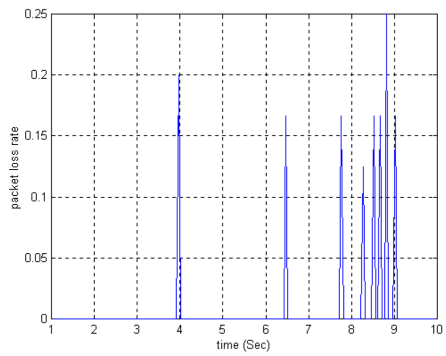

Figure 3. Packet loss rate of the multimedia flow at the client

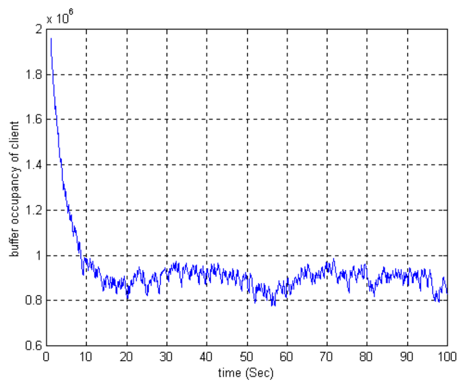

Figure 4. Client buffer occupancy during [1, 100] seconds

In order to demonstrate our approach can also maximally utilize the client buffers, the client buffer occupancy during time period [1, 100] seconds is shown in Figure 4. As can be seen from this figure, there is no overflow and underflow at the client buffer. Comparing with the simple AIMD rate adjustment algorithm, the transmission rate is adjusted more smoothly, which can better provide a stable presentation quality at the client. Another advantage of our approach is that when there are multiple clients experiencing different network congestion scenarios, the server can still allocate the minimal bandwidth to each client based on the congestion level, end-to-end delay and client buffer occupancy.

4. Conclusions

When there are multiple clients requesting data from a server simultaneously, how to efficiently allocate the bandwidth to each client to satisfy the QoS requirements of the application is a challenging task. In this paper, we presented an end-to-end congestion controlled optimal bandwidth allocation mechanism for multimedia transmission. The proposed mechanism can achieve the maximal utilization of the client buffer and the minimal allocation of the bandwidth. The optimized transmission rate can be adjusted dynamically based on the congestion level observed by the client to satisfy the constraint of the network bandwidth and to avoid overflows and underflows. Our approach is self-adjusted and can avoid overflows and underflows to guarantee QoS. Simulation results showed that our approach is adaptive to packet loss, variable bandwidth and variable network delays, and our rate is adjusted in a TCP-friendly way.

5. Acknowledgment

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

6. References

[1] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS Control of Multimedia Applications Based on RTP," First International Workshop on High Speed Networks and Open Distributed Platforms, pp. 831-843, St. Petersburg, Russia, June 1995.

[2] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithm for Congestion Avoidance in Computer Networks," *Journal of Computer Networks and ISDN*, 17(1): 1-14, June 1989.

[3] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," IEEE/ACM Transactions on Networking, vol. 7, no. 4, pp. 458-472, August 1999.

[4] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based Congestion Control for Unicast Applications," Applications, Technologies, Architectures and Protocols for Computer Communication, pp. 43-56, October 2000.

[5] R. Jain, K. Ramakrishnam, and D. Chiu, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer," Tech. Rep. DEC-TR-506, Digital Equipment Corporation, August 1987.

[6] R. Rejaie, M. Handley, and D. Estrin, "RAP: An End-to-End Rate-based Congestion Control Mechanism for Realtime Streams in the Internet," Proc. IEEE INFOCOM, vol. 3, pp. 1337--1345, March 1999.

[7] 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 (ICME2002), pp. 537-540, Lausanne, Switzerland, August 26-29, 2002.

[8] D. Wu, Y. Hou, and Y.-Q., Zhang, "Transporting Real-time Video over the Internet: Challenges and Approaches," Proceedings of the IEEE, vol. 88, no. 12, pp. 1-19, December 2000.