Problems on implementation of LQ rate control schedule system for multimedia transmission

Xiaofei Zhou, Kenneth Ong

Department of Electrical and Computer Engineering
National Unversity of Singapore
4 Engineering Drive 3, Singapore 117576.
{xiaofei, eleongk} @ nus.edu.sg

Abstract

In this work, we investigated the linear quadratic rate control method (LQ) for the transmission of MPEG2 streams in IP networks. Our study shows that LQ is superior to conventional proxy schedule schemes in avoiding jitter. However, we also find LQ has problems in buffer allocation, overflow, and feedback rate selection. We studied these problems in depth and identified their influence on both the host and client. We propose improvements in client buffer allocation, overflow prevention and slow feedback rate compensation.

Keywords: Scheduler, LQ, Multimedia streaming.

1. Introduction

With the explosive growth of multimedia communications and applications, a large amount of multidimensional media traffic swarms into the traditional network. To deliver multimedia streams efficiently, many supporting technologies such as distributed server systems, proxy caching methodologies, session management and resource allocation, scheduling, and real time control are needed. Basically, after a stream is retrieved from the database, it encounters a complex environment where all parameters change continuously. The outgoing network may be DiffServ or IntServ; the backbone probably is ATM or wireless network; the receiver could be LAN users, PDA or hand phone. Different environments require different QoS levels [1]. These levels are evaluated by examining the dominating parameters for multimedia transmission such as delay, jitter, BW usage and reliability [2]. Service providers should guarantee QoS while at the same time maintaining a favorable environment for other traffics on the intermediate network.

Focusing on conventional transmission scheduler, people either use priority schemes to isolate timing sensitive flows from bursty ones or enable reservations to guarantee QoS. For example, separate priorities for different frames [3], multi-channel data scheduling [4], and multi-thread distributed delivery [5] are widely used. In this paper, we consider an end-to-end control method for resource utilization called LQ control [6]. Although there are many other rate-base control methods for multimedia stream delivery [7] [8], we select LQ for our implementation based on the following reasons.

Firstly, it is cost-effective. LQ adjusts the sending rate according to client buffer occupancy. It is considerate to users and friendly to other applications on the network. Secondly, it is fairly accurate. Linear Quadratic control has been used for decades in linear continuous control systems. The precise mathematical calculations behind the method allow it to trace and control the system accurately and effectively. Thirdly, it is easy to implement. End-to-end control neglects the complexity of intermediate networks. High client buffer occupancy spares us from handling delay and jitter problems because there is always enough data in the client buffer to be decoded.

The remainder of the paper will be organized as follows. In Section II, we provide some essential background information on the LQ method and our system model. In section III, problems corresponding to the implementation will be proposed and analyzed. Section IV shows our simulation results and gives the detailed discussions. Section V provides a summary and presents future work.

2. Background information

Linear Quadratic (LQ) control is not a new theory but its implementation in media packet schedulers started only in the last century. Its characteristics like accurate real-time control, low feedback overhead and easy implementation make it a good choice for engineers.

2.1. LQ resource utilization based control

LQR (Linear Quadratic Regulator) is a Min-Max problem. The aim is to find a state-feedback control law of the form u = -kx that minimizes the performance measurement function:

$$V = \int_0^T (x'Qx + u'Ru)dt + x'(T)Mx(T)$$

where x is the state vector, u is the input vector, Q and M is a positive semi-definite matrices, R is a positive definite matrix. Q, R, and M are weight coefficients for the state and input parameters. In the above function, x'Qx measures the control accuracy; u'Ru represents the control effort; x'(T)Mx(T) represents the terminal control accuracy. Let the performance index function be

$$J = 1/2 \int_0^{T_f} [e^2(t) + R^2(t)]\, dt$$

where e(t) = Q_r – Q(t), Qr is the allocated client buffer size and Q(t) is the buffer occupancy at a given time t. T_f is the finish time for a media stream. Using the Riccati equation and boundary condition $P(T_f) = Q_r$, we can get the optimal transmission rate

$$R^*(t) = Q_r + L(t) - e^t L(t)/e^{Tf} - Q(t)$$

Here L(t) is the playback rate at the receiver's side. The aim of the LQ tracker rate control scheduler is to maximize the client buffer usage without bringing high loss rate and to save as much network bandwidth as possible.

2.2. Network model for implementation

With reference to the following figure (Fig.1):

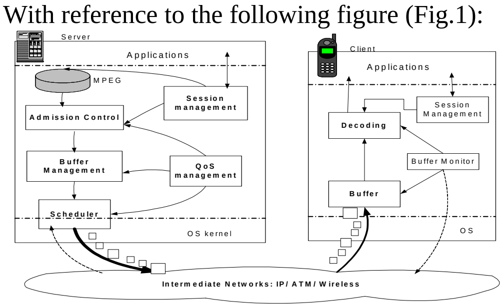

Fig. 1 Network Model for LQ Scheduling System

At the server side, the session manager receives a demand from the client and sets up a session if the admission controller still allows sessions to enter the network. Then multimedia data is retrieved from the MPEG database and flows into the server buffer. If congestion occurs, some of the low priority frames are discarded before entering the buffer.

At the client's side, media streams are received by the client buffer. The buffer monitor records the current buffer occupancy and the playback rate. Then it sends this information back to the source server.

For each flow, the calculated optimal rate is translated into a weight for the packet dispatcher and a QoS level for buffer management. The weight will be used to decide the transmission speed for current flow and the QoS level is used to decide the threshold for discarding packet during congestion. Also, the LQ controller will dynamically calculate a sub-optimal rate to limit the maximum transmission rate whenever it predicts a possible overflow in the client buffer.

In order to simplify the client device, we perform packet discarding on server buffer instead of client buffer. Thus we need a complex buffer manager on server side to make discarding decisions but only a simple buffer monitor on client side to record information.

3. Problems for implementation

Focusing on the LQ scheduler system, we propose the following questions that are pivotal during implementation.

Firstly, the client buffer should be allocated properly, which is economical and large enough. Secondly, feedback information may be delayed and cannot represent the current situation for client buffer accurately. Thus we should decide the acceptable feedback interval that can tolerate delay. Thirdly, the overflow problem on the client buffer should be considered seriously. Fourthly, too much feedback information will exhaust the bandwidth resource. Thus we need to investigate whether a satisfactory performance can be achieved under slow feedback rate.

Based on the problems described above, we got some useful results through conducting series of simulations. We also proposed our improvements.

4. Results and discussion

According to [9], I and P frames are divided into 7 separate packets with 986 Bytes and 353 Bytes per packet respectively in the simulation. The B frame is in a single packet with the size of 815 Bytes. Thus one GOP size is 7*986+7*353*3+815*8 = 20835 bytes. Here we take the (12,3) MPEG 2 sequence and the IP head is not counted.

4.1. Allocated buffer size vs Buffer occupancy

Using (12, 3) MPEG video, the client will play 2 GOPs per second with some fluctuation. For the PDA or hand phone applications, 15frames/s should be enough. We allocated the client buffer as integer multiples of the GOP size. Other parameters and overflow results are listed in the table below.

Qr\Overflow	max	Min	mean	standard
1GOP	718.7	0.427	237.6	183.1
2GOP	718.7	0.427	236.6	183.4
3GOP	718.7	0.427	237.3	183.7

Table. 1 Overflow under Different Allocated Buffer Size
Feedback: twice/s; Playback: 40~42KB/s (24frame/s);

Statistic shows that the client buffer occupancy is above 99.75% full along the transmission. The above table indicates that the LQ rate control method is not affected by the client buffer size. It tries to feed as much into the client buffer as possible after stabilization. With such a characteristic, we can allocate buffer size freely as long as it guarantees the normal playback rate of the client. The system performance depends on other parameters.

4.2. Performance with overflow prediction and prevention

In [6], the authors declared a fixed R_{max} to save BW by limiting the maximum transmission rate. This is unnecessary. It cannot represent the changing situation for the network. Also, when competing with other UDP flows in the network, such a self-effacing behavior may make our stream weaker than other flows and lose the BW it deserves. LQ has already tried to prevent wasting BW, so we do not need to add another parameter to confine it.

Through feedback packets, the source will get information on current playback rate and the buffer occupancy. We propose that the source carries out a security test after calculating the optimal transmission rate. If $R^* \times \triangle t > Qr - Q(t) + \min\{\text{Playback rate}\} \times \triangle t$, calculated optimal transmission rate is very likely to cause overflow and should be deduced. We limited it within $(Qr-Q(t))/ \triangle t$, that is, $R^*_{sub} = (Qr-Q(t))/ \triangle t$. Results show that overflow problem is solved after performing such a test. The assigned sub-optimal transmission rate R^*_{sub} replaces the R_{max}. This online test predicts possible overflow and prevents it from happening. Meanwhile, it reduces the BW usage as R_{max} did.

4.3. Pseudo-continuous control

The linearly increasing overflow size problem is caused by the target system, which is not strictly continuous or discrete. In history, such a problem (continuous state parameters with discrete control) has been studied for years. The authors of [10] solved this problem by dividing the performance index J into two parts: continuous part and discrete part. But the solutions are quite complex and difficult to implement for engineering purpose. So here we propose a pseudo-continuous control method which is reasonable, simple, and good enough for media transmission system with low feedback rate.

Now we set the feedback rate as low as Once/S. That is, only one feedback packet is sent back for every two GOPs. Under such settings, the LQ scheduler performs quite inaccurate and non-responsive. The substantial reason is that the control input is held constant across each feedback interval. Such a "zero-order hold" type control weakens the efforts of LQ controller. We realized that although the feedback rate is Once/S, this doesn't necessarily mean our control effort must be Once/S. With the characteristics of multimedia flows, continuous (we call it pseudo-continuous) optimal control can still be performed with comparatively high accuracy during feedback interval.

When the feedback packet comes, the system calculate optimal transmission rate according to the information inside the packet. After that, system records what has been sent out for each flow and predicts how much space should left on client side under such a sending rate. Then it calculates an "optimal" transmission rate using recorded and predicted information on every 0.01s (the interval is adjustable). The playback rate will take the mean value calculated from previous feedbacks. The simulation result is given below. The red dashed line represents client buffer occupancy with a feedback rate of 1 time/S, without pseudo-continuous control. The blue line shows the buffer occupancy under the same settings but with pseudo-continuous control.

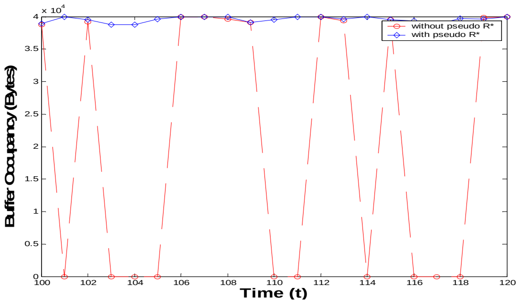

Fig. 2 Client Buffer Occupancy with or without Pseudo-continuous Rate Control

The above figure shows clearly the great improvement using pseudo-continuous control method. Without this method, the system encountered great oscillation due to the slow feedback rate. Using pseudo-continuous control method, the system performs like the one who has a feedback rate of 100 times/S. The mean buffer occupancy is 3.956×10^4 Bytes (allocated buffer size is 4.0×10^4 Bytes).

Pseudo-continuous control method solved the confliction between satisfactory performance and precious bandwidth. In our implementation, we need to send only one packet back for each second to get a very good performance, which spares the network resource to a great extent.

4.4. Buffer occupancy with delay

Now we take delay into consideration. Here, we only consider the delay involved in feedback packets. According to [11], network delay between US states ranges from 20ms to 76ms with an average of 35ms. In our simulation, we generated delay randomly between 30ms and 80ms. Under these settings, delays are within one feedback cycle, if they occur.

When the feedback interval is comparable with the propagation delay, the influence is obvious.

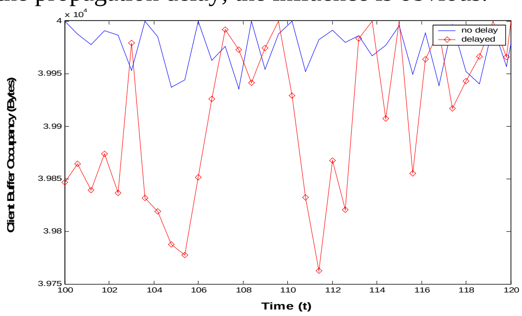

Fig. 3 Effects of Delay on Buffer Occupancy (Feedback Rate @ 40 times per sec.)

Here the feedback rate is fixed at 40 times per sec, which is comparable to mean network delay. The figures above show that the system performance degraded significantly due to the outdated information. At the same time, such a frequently non-full buffer also has greater chances to suffer overflow. This phenomenon is caused by the unresponsive source. When the client buffer is already full, the source cannot sense it quickly, and will continue to inject regular streaming data to the client. From statistics, the maximum overflow size in the delayed situation can reach up to 90 bytes with a feedback rate as high as 25 times per second. This is even worse than the overflow size with a feedback rate at twice per second, which is around 70 Bytes. Thus we should always employ a feedback interval that is much larger than the network delay so as to guarantee efficient buffer management. Adopting pseudo continuous control, this is easy to achieve.

5. Conclusion and future work

In this paper, we analyze the problems on client buffer allocation, overflow, feedback rate, and delay effects on the linear quadratic (LQ) transmission rate control scheduler system. We investigated how these problems affect the system performance. With proper parameters, the LQ scheduler system is a practical, efficient and reliable method to support continuous media transmission.

Our future work will focus on implementing the whole design into our proxy server. QoS levels will also be considered after the per-flow design is completed.

6. Reference:

[1] *Denise J. Ecklund at al,* Dynamic end-to-end QoS management middleware for distributed multimedia systems, Multimedia Systems 8: 431 – 442 (2002)

[2] *Wenyu Jiang and Henning Schulzrinne,* QoS measurement of Internet Real time Multimedia Services, Dept. of Computer Science in Columbia University, Dec. 1999.

[3] *Markus Fidler,* Real-Time Multimedia Streams in a Differentiated Services Network, Dept. of Computer Science, Informatik IV, Aachen Univ. of Tech., 2001

[4] *Randy Chow at al,* Traffic Dispersion Strategies for Multimedia Streaming, Proceedings of the Eighth IEEE Workshop on Future Trends of Distributed Computing Systems(FTDCS'01), 2001

[5] *Gwang S.Jung,Kyung W.Kang, and Qutaibah Malluhi,* Multithreaded Distributed MPEG1-Video Delivery in the Internet Environment

[6] *Mei-Ling Shyu1 et al,* Optimal Resource Utilization in Multimedia Transmission, 2001

[7] *Yuko Onoe et al,* Network information based Rate Controls on Multimedia Streaming Servers, IEEE Computer Society, 2003

[8] *Kang Li at al,* A Rate-Matching Packet Scheduler for Real-Rate Applications, Dept. of Computer Science and Engineering, Oregon Graduate Institute

[9] *Jill M.Boyce and Robert D. Gaglianello,* Packet Loss Effects on MPEG Video Sent Over the Public Internet, Bell laboratories, Lucent Technologies

[10] *Mohamed Abdel-Haleem & C.D.Johnson,* Optimal Discrete-Continuous Control for the Linear-Quadratic Regulator Problem, 1996 IEEE

[11] US network delay, http://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html