Discrete LQ Rate Control for MPEG2 Video Streaming System

Xiaofei Zhou
Department of Electrical and Computer Engineering,
National University of Singapore
E4-05-48, 4 Engineering Drive 3, Singapore
xiaofei.zhou@gmail.com

Kenneth Ong
Department of Electrical and Computer Engineering,
National University of Singapore
E4-05-48, 4 Engineering Drive 3, Singapore.
eleongk@nus.edu.sg

Abstract — In this paper, we propose a novel rate control system by applying the discrete linear quadratic rate control method (DLQ) to transmissions of MPEG2 streams in IP networks. We give out the detail deduction for disturbed-DLQ-regulator problem and the corresponding methods to implement the results into media streaming rate control. Study shows that DLQ is superior to conventional rate control schemes especially in client buffer utilization. Besides the basic DLQ method, we also investigate DLQ under heavy-traffic scenarios and provide improvements for this special case. The influences of delay, with respect to different mean values and variances, are discussed carefully.

Index Terms—Scheduling, LQ, Multimedia streaming

I. INTRODUCTION

With the growth of multimedia communications and applications, a large amount of multidimensional media traffic swarms into the traditional network. Basically, after a stream is retrieved from the database, it encounters a complex environment where all parameters change continuously and unpredictably. Therefore, an efficient rate control system which can provide the satisfactory performance on client side is crucial. Meanwhile, service providers should maintain a favorable environment for other traffic on the intermediate network.

Generally, multimedia stream rate control methods can be classified into three groups according to the focus: server oriented, network oriented, and client oriented. Most conventional methods belong to the first two groups ([1][2][3][4]). Among them, researchers 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 ([5]), multi-channel data scheduling ([6]), and multi-thread distributed delivery [7] are widely used. In recent years, most researches focus on TCP-friendly rate control ([17]) or layer-coded rate control ([18]). All the mentioned methods are network oriented open loop control. Priority schemes can hardly match the requirements for each flow perfectly because the levels of priorities are limited. Multi-channel and multi-thread are better ways to schedule media traffic especially MPEG2 streams but the scheduler is quite complex with large run-time overhead. Even TCP-friendly rate control and layer-coded scheduling do not consider matching the client playback speed with the sending procedure. Therefore they are only friendly to the network, not end users.

In this paper, we consider an end-to-end client oriented close-loop control method for resource utilization called discrete linear quadratic (DLQ) rate control that can overcome the shortcomings for conventional schedule mechanisms. DLQ is selected

© 2008 ACADEMY PUBLISHER

based on its cost-effective, its accuracy, and its easy to implement. Previous works [8] and [9] used the similar method in media delivery rate control. But there are still questions unsolved. First, all previous approaches use continuous system model, which is not proper for network transmission system. Although multimedia can be delivered in a streamed way, it is handled discretely on the rate control layer. Second, network delay and noise are not taken into account. Third, the previous papers did not consider LQ rate control under various traffic environments. Last, no detail mathematical deductions are provided.

This paper tries to mend the mentioned gap and gives out a systematic design of DLQ transmission scheduler system. We want to apply DLQ into two typical scenarios: no congestion (*Normal Scenario*) and congested network (*Heavy-traffic Scenario*). For basic DLQ under *Normal Situation*, we adopt the formula for u^* derived in section II. For *Heavy-traffic Situation,* server can trade off the current client buffer usage in order to support more users as long as the decoding procedure is not disturbed.

This paper is based on our previous work in [16]. New system design with Kalman Filter and new results on different scenarios have been added in this paper. The paper is arranged as follows. Section II describes the mathematical deduction of the DLQ rate control scheduler. Results and discussion are presented in section III and IV. Section V concludes the paper.

II. DLQ RATE CONTROL SYSTEM DESIGN

We mathematically model the scheduler as shown in Figure 1. The client gives back the information of buffer vacancy x_i and buffer vacancy rate v_i. Buffer vacancy rate is the rate on which media data in client buffer are fetched for decoding. It is approximately a Gaussian variable randomly distributed between the max and min. playback rate of the transmitted movie. Q_r is the allocated client buffer size. Q_i is the instant client buffer usage. The difference between Q_r and Q_i gives the buffer vacancy x_i and it is feedback to server. The server calculates the optimal transmission rate u_{i_pre} according to the feedback x_i and control gain k_i, then takes into consider the decoding rate v_i with feedback gain k_{i_fb}. Finally the optimal sending rate u_i is decided.

Kalman filter (KF) is introduced to handle the noise along the transmission path. In figure 1, we use dash-line rectangle for Kalman Filter module because this filter is not a must for all situations. In wired network where noise is negligible, this filter could be omitted for the simplicity of system. While in unstable wireless networks, it is highly recommended to get a more accurate state variable estimation. Additionally, the design of this module is comparatively independent. It exists or not does not influent the DLQ system design. We omit it in the following mathematical deduction and provide its algorithm in appendix B.

Here we did not consider the transmission delay between the client and server. The delay problem will be discussed in a separate section in this paper. With reference of the mathematical system model (Fig.1), the state and the index functions are:

$$x(i+1) = x(i) + t_s u(i) - t_s v(i) \quad i \geq 0$$

$$J = \sum_{i=0}^{N} [x^2(i) + u^2(i)]$$

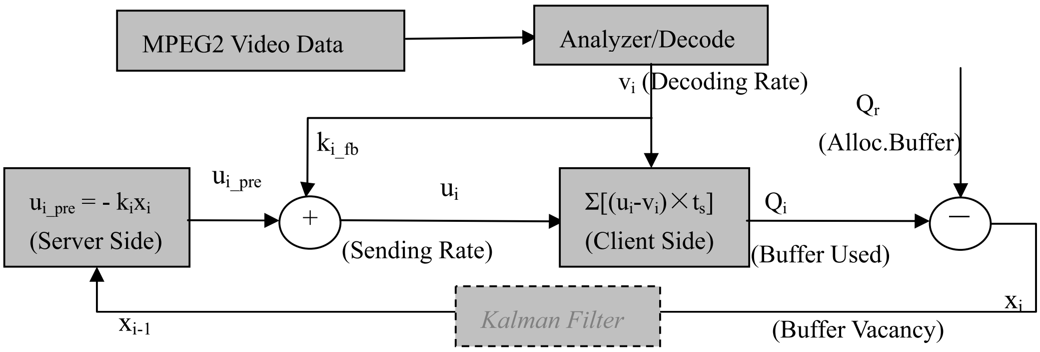

Fig.1 DLQ scheduler mathematical model

© 2008 ACADEMY PUBLISHER

The purpose of DLQ scheduler system is to minimize the bandwidth usage while maximize the client buffer usage. Thus, we take the sum of quadratic term of buffer vacancy x(i) and sending rate u(i) as the measurement criteria. The following deduction focuses on finding a $u^*=-kx$ that minimize the J. Parameters used in the functions are shown in table 1.

Table1. Definition of variables in scalar DLQ formula

Variables	Signification	Explanation
x	-Client Buffer vacancy	State Variable (Bytes)
u	Optimal Sending Rate	Control Effort (Bytes/s)
v	Client Buffer Vacancy Rate	Disturbance (Bytes/s)
t_s	Sample Interval	Sample Rate

Here we use (-client buffer vacancy) as the state variable to make its coefficient and the coefficients of control variable (u) to be positive. Lower case is used to differentiate from the formula of matrix solution, which will be discussed later in this section.

Let $J^*(x_i, i)$ denote the minimal value of performance measure starting at time $t=i^*t_s$ and state $x(i^*t_s)=x_i$. Then the optimality principle states that any input that is optimal over the interval (i, N) must necessarily be optimal over the interval (i+1, N). So that the following recursive relation must hold true:

$$J^*(x_i,i) = \min_{u_i}\{x_i^2 + u_i^2 + J^*(x_{i+1}, i+1)\}$$

Because J^* has the quadratic form, Let $J^*(x_i,i) = p_i x_i^2 + 2b_i x_i + c_i$, so that:

$$J^*(x_i,i) = \min_{ui}\{x_i^2 + u_i^2 + p_{i+1}x_{i+1}^2 + 2b_{i+1}x_{i+1} + c_{i+1}\} \quad \text{---(1)}$$

To find the u_i that minimize J^*, we let:

$$\frac{\partial J^*(x_i,i)}{\partial u_i} = 0$$

That is:

$$2u_i^* + 2t_s p_{i+1}(x_i + t_s u_i^* - t_s v_i) + 2t_s b_{i+1} = 0$$

$$u_i^* = \frac{-t_s p_{i+1} x_i + t_s^2 p_{i+1} v_i - t_s b_{i+1}}{1 + t_s^2 p_{i+1}} \quad \text{---(2)}$$

Rewrite (1) as:

$$p_i x_i^2 + 2b_i x_i + c_i = x_i^2 + u_i^{*2} + p_{i+1}x_{i+1}^2 + 2b_{i+1}x_{i+1} + c_{i+1}$$

Substitute (2) into this function, we have the following three functions (See Appendix A).

Quadratic terms in x:

$$p_i = 1 + \frac{p_{i+1}}{1 + t_s^2 p_{i+1}} \quad \text{---(3)}$$

Linear terms in x:

$$2b_i = \frac{2b_{i+1} - t_s p_{i+1} v_i}{1 + t_s^2 p_{i+1}} \quad \text{---(4)}$$

Terms independent of x:

$$c_i = t_s^2 p_{i+1} v_i^2 - \frac{t_s b_{i+1} v_i (2 + t_s^2 p_{i+1}) + t_s^2 b_{i+1}^2}{1 + t_s^2 p_{i+1}} + c_{i+1} \quad \text{---(5)}$$

Substitute (4) into (2), we get the final formula of u^*:

$$u_i^* = -\frac{t_s p_{i+1}}{1 + t_s^2 p_{i+1}} x_{i-1} + \frac{t_s^2 p_{i+1}}{2(1 + t_s^2 p_{i+1})} v_{i-1} - t_s b_i \quad \text{---(6)}$$

Given the terminal values of p_n and b_n, (3) and (4) will decide all p and b values recursively. In equation (6), the coefficient before x_{i-1}, that is $t_s p_{i+1}/(1+t_s^2 p_{i+1})$, is the k_i in figure 1 and the coefficient before v_{i-1}, that is $t_s^2 p_{i+1}/2(1+t_s^2 p_{i+1})$, is the k_{i_fb}. The aim of this DLQ tracker rate control scheduler is to minimize the client buffer vacancy (represented by x) while at the same time saving as much network bandwidth (represented by u) as possible.

III. RESULTS AND DISCUSSIONS

We obtain the results for two scenarios through a series of simulations using an (18,3) m2v video clip. The notation (18,3) is a MPEG encoding format that contains 18 frames on a Group of Picture (GOP), and two B frames between a pair of main frames (I or P). The parameters of the video are listed in table 2.

Table 2. Key parameters in simulation

Total Frames	*8760*	Video Length	*292s*
Minimum Frame	*2324B*	Playback time/frame	*0.033s*
Maximum Frame	*81340B*	Allocated Buffer	*1MB*
Playback Rate	*30f/s*	Sample Rate	*62.5/s*

A. Basic DLQ

The client buffer we allocate (1MB) can accommodate nearly 5 GOPs. Sample interval is 0.016s (Nyquist theory: minimum twice/frame). Then we attained the performance of client buffer occupancy in Figures 2. The statistics show that the client buffer is, on average, around 79.86% full. At the beginning of transmission, the server sends data more than the client can consume. Eventually, the client buffer occupancy becomes higher and higher. In our simulation, the maximum sending rate at very early beginning is 1.2MB/s. After the system reaches its steady state, the buffer occupancy stays between 0.7 MB and 0.86 MB with small fluctuations.

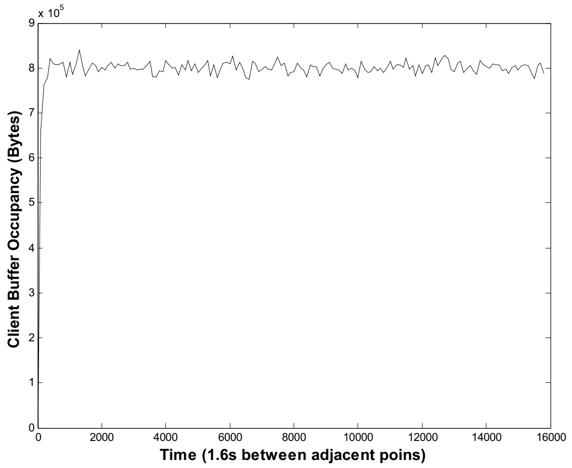

Fig2. Client buffer occupancy using DLQ

Here we do not limit the bandwidth of the outgoing network. If a maximum BW is set, for example 0.6 MB/s, the curve will climb up slower and take a longer time to reach the steady point.

Considering the network noise, we simulate the transmission with and without Kalman filter as shown in Figure 3. The average deviation of state variable x is set to 1.0×10^5 Bytes. Such noise degrades the performance of transmission by pulling down the client buffer occupancy and enlarging its fluctuation. Kalman filter lightens the problem and offers a stable delivery. To make the Kalman filter works more efficient, it is significant that the covariance of noise is properly probed. Techniques for online noise measurement could be used here together with the filter.

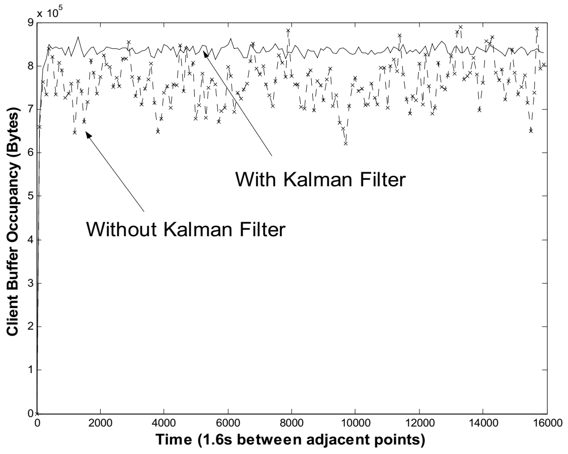

Fig.3 Buffer occupancy under noise with/without KF

B. Modified DLQ for heavy-traffic scenario

The discussion in previous paragraphs is based on the hypothesis that the bandwidth is enough to support each stream sending at its calculated optimal speed. If the server wants to accept more clients, this hypothesis may be violated. Fortunately, a good advantage of DLQ is that it always holds the client buffer occupancy at a certain level with small fluctuations. Figure 4 demonstrate clearly this good characteristic of DLQ.

In Figure 4, the fluctuation of client buffer occupancy (dotted line) keeps small even when the sending rate is reduced to 43%u^*. Thus we can trade off the current client buffer usage for supporting more users using the DLQ mechanism. When the steady buffer usage is dragged down from around 85% to 43% or less, the sending rate is also slowed. The recovered BW can be allocated to other incoming streams.

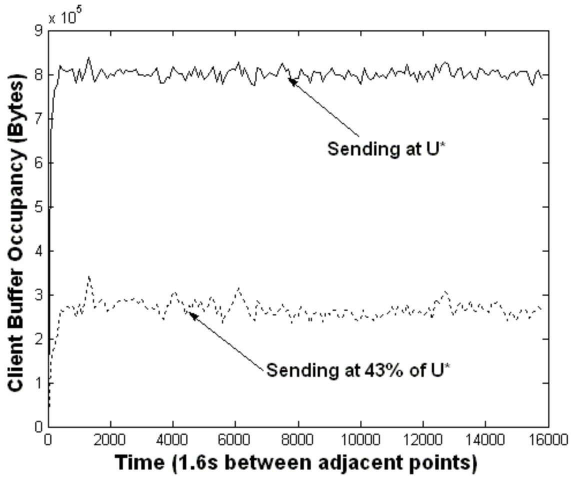

Fig.4 Client buffer occupancy with BW limitation

Now we determine theoretically how much BW can be saved for a single stream. Using the method for heavy-traffic scenario discussed in section I, we cannot reduce the sending rate too much in order to maintain at least one GOP in client buffer all the time. Here we select 6 different MTV videos and check the lowest possible sending rate.

Normally, the above six videos consume 1.8815 Mbps all together. Sending at a minimum rate can save 22.3 Kbps. From the table, there is no relationship between the fluctuation range of v_i and the BW be saved. Also, Max. GOP size does not influence the saved BW. DLQ provides a relatively stable performance and can save several Kbps bandwidth in average for each video.

© 2008 ACADEMY PUBLISHER

Table3 Results of modified DLQ with BW limitation

	Fluctuate Range of v_i	Avg. u^* (bps)	Min.αu^* (bps)	BW saved (bps)
Video1	*0.69 Mbps*	1.462×10^5	1.438×10^5	*2400*
Video2	*0.5 Mbps*	1.466×10^5	1.438×10^5	*2800*
Video3	*0.63 Mbps*	1.468×10^5	1.437×10^5	*3100*
Video4	*1.89 Mbps*	5.519×10^5	5.474×10^5	*4500*
Video5	*2.39 Mbps*	4.024×10^5	3.993×10^5	*3100*
Video6	*0.52 Mbps*	4.876×10^5	4.812×10^5	*6400*

V. DELAY IMPACTS ON DLQ SCHEDULE SYSTEM

Now we take delay into consideration to investigate its impact on system performance. We assume the delay follow a normal distribution with mean value μ and variance σ^2. The following rules are used to simulate the behavior of basic DLQ under delayed feedbacks.

1) Feedback packets between two adjacent sampling points will be retained for a decision later on.

2) If no new feedback comes within a sample interval, the scheduler maintains the previous sending rate.

3) If several feedback packets come together during an interval, the scheduler takes the latest one for calculation and discards all the others.

A. Delay impacts

Firstly, we fix the σ^2 to 144ms and investigate situations with mean values (μ) of delay to be 0ms, 32ms, and 128ms ([14]) respectively, which are 0, 2, and 8 times the sampling interval (16ms). Choosing basic DLQ for simulation, the client buffer occupancy with and without delay is shown in figure 5.

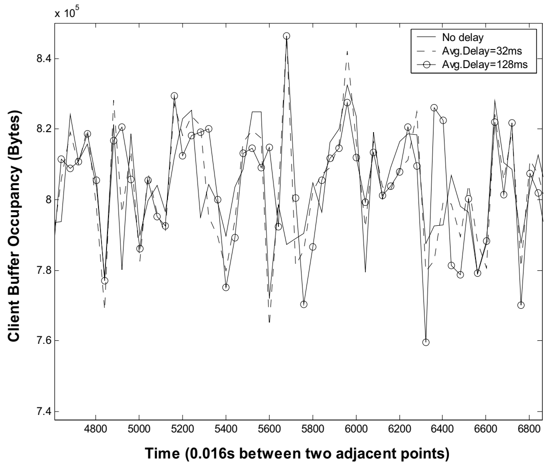

Fig.5 Buffer occupancy with mean delay of 0ms, 32ms, 128ms

Figure 5 is an enlarged picture to show the differences more clearly. From the figure, delay brings larger fluctuations to the buffer occupancy and the changes are not significant. For example, the buffer usage with no delay rises after 5600 points from 0.77 MB to 0.8 MB and then decreases (solid line), but for the delayed situation, the curve rises from 0.765 MB to 0.85 MB before decreasing (dotted and solid with circle line). Thus the delayed situation is more likely to encounter overflow. Moreover, the buffer occupancy with larger delay (solid with circle line) swings away from no-delay situation (solid line) more seriously than the one with smaller delay (dotted line).

Secondly, we fixed μ to 96 ms and investigated the variance (σ) of 144 ms and 1024 ms. Comparing it with no delay situation, we get the statistics in the following table 4.

Table 4. Buffer occupancies with different σ of delay

Delay\Client buffer occupancy	Max (MB)	Mean (MB)	Median (MB)	Std (KB)
No delay	0.8395	0.7949	0.8008	64.69
Σ^2=144ms	0.8370	0.7943	0.8013	64.69
Σ^2=1024ms	0.8443	0.7948	0.8019	64.71

From the table, delay variance also has small influence on buffer occupancy. Far from what is expected, delay does not bring hazardous disruption to the DLQ system. Reasons lie on the characteristics of MPEG2 video and the DLQ system itself. Delay affects two feedback variables: the current client buffer occupancy and the current buffer vacancy rate. However, MPEG2 video contains IBBP frames in repetition, and the sizes of the same type of frames are close. If the feedback packet for a frame is delayed to the sampling point for a following frame of the same type, the information of buffer vacancy rate it contains is near to the exact value. On the other hand, since the DLQ adjustments are at a fine-grain level, the high sampling rate ensures the outdated buffer occupancy value will not mislead the schedule decision to a great extent. In other words, DLQ system reacts fast enough to correct its deflection.

B. Analysis of delay impact

© 2008 ACADEMY PUBLISHER

After shown the revealed phenomenon, we found that all troubles were caused by two types of problems brought by the delay.

1) Time reverse problem

If several feedbacks came within a sample period in sequence, DLQ takes the one that arrived latest as the reference for decision. A feedback sent at time slot 10 may reach the server earlier than the feedback sent at slot 9. Suppose these two feedbacks come within the same sample interval, DLQ will discard the former one (sent at slot 10) but take the later one (sent at slot 9). This problem can be solved using a time stamp mechanism introduced later.

2) Outdated information problem

Cases where no feedback come within a sample period or feedback came late are considered as the outdated problem. This problem can be solved by enhancing the network transmission speed which is not the scope of this paper.

C. Time stamp solution

Within the two parameters influenced by the delay, buffer occupancy and buffer vacancy rate, making prediction on buffer vacancy rate during run-time will increase the system's complexity greatly without significant performance improvement. So the simple one-step prediction mechanism proposed here makes prediction only on actual buffer occupancy. We propose to give each feedback packet a time stamp when sent out. Receiving a new feedback, the DLQ scheduler compares it with the current time and predicts the current buffer occupancy using:

Buffer occupancy = Buffer occupancy in current feedback packet + Sent data during ([(Time stamp-Current time)/Ts]+1) steps – Playback data during this period.

Here, [(Time stamp-Current time)/Ts] means selecting the integer part of (Time stamp – Current time)/Ts and Ts is the sample interval. The playback data is calculated using the buffer vacancy rate in the current feedback packet. Of course if the system receives several feedbacks in a sample period, it compares their time stamps and chooses the latest one for calculation.

Adopting μ =128ms and σ^2 =1024, we redo the previous delay-influence simulation and add the curve with prediction mechanism. From figure 6, the simple prediction mechanism (dotted line with cross marker) helps the buffer occupancy perform nearly as good as the no delay situation (solid line), much better than the no prediction situation (solid line with diamond marker).

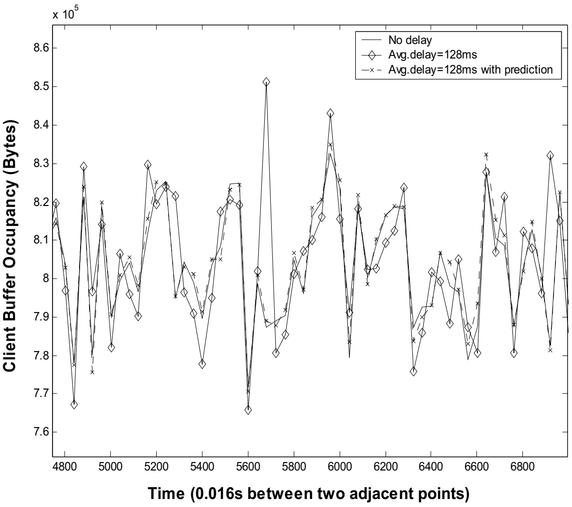

Fig 6. Buffer occupancy with delay prediction

VI CONCLUSION AND FUTURE WORK

In this paper, we demonstrated a close loop client-oriented rate control system using the discrete linear quadratic control. After mathematical deduction and implementing methods introduction, we presented the experiment results for the basic DLQ method. The results showed that basic DLQ can obtain stably nearly 80% of client buffer occupancy. The limitation of the maximum sending rate merely slows down the stabilization time. It will not influent the performance of steady state. Under heavy traffic, we can sacrifice the buffer usage level to support more streams. The discussion of delay problem indicated that DLQ methods are slightly disturbed by delays. With all these characteristics, the DLQ rate control mechanism could be a practical, simple but efficient method to support multimedia streaming transmission.

APPENDIX A: DEDUCTION OF FORMULA (3), (4), (5)

Function (1) is rewritten as:

$$p_i x_i^2 + 2b_i x_i + c_i = x_i^2 + u_i^{*2} + p_{i+1} x_{i+1}^2 + 2b_{i+1} x_{i+1} + c_{i+1}$$

© 2008 ACADEMY PUBLISHER

Substitute state function $x(i+1)=x(i)+t_s u(i)-t_s v(i)$ into it, we have:

$$p_i x_i^2+2b_i x_i+c_i=x_i^2+u_i^{*2}+p_{i+1}(x_i+t_s u_i^*-t_s v_i)^2$$
$$+2b_{i+1}(x_i+t_s u_i^*-t_s v_i)+c_{i+1}$$

After manipulation:

$$p_i x_i^2+2b_i x_i+c_i=(1+p_{i+1})x_i^2+2t_s p_{i+1}x_i u_i^*+(1+t_s^2 p_{i+1})u_i^{*2}$$
$$+(2b_{i+1}-t_s p_{i+1}v_i)x_i+(2t_s b_{i+1}-t_s^2 p_{i+1}v_i)u_i^*$$
$$+t_s^2 p_{i+1}v_i^2-2t_s b_{i+1}v_i+c_{i+1} \qquad \text{---(*)}$$

Now we need to replace the u_i^* in (*) with:

$$u_i^*=\frac{-t_s p_{i+1}x_i+t_s^2 p_{i+1}v_i-t_s b_{i+1}}{1+t_s^2 p_{i+1}}$$

To make things clearer, we first neglect the terms without u_i^* in the right hand side of (*) and substitute u_i^* into u_i^* related terms $2t_s p_{i+1}x_i u_i^*, (1+t_s^2 p_{i+1})u_i^{*2}$, and $(2t_s b_{i+1}-t_s^2 p_{i+1}v_i)u_i^*$ respectively.

$$2t_s p_{i+1}x_i u_i^*$$
$$=2t_s p_{i+1}x_i\frac{-t_s p_{i+1}x_i+t_s^2 p_{i+1}v_i-t_s b_{i+1}}{1+t_s^2 p_{i+1}}$$
$$=-\frac{(2t_s^2 p_{i+1}^2)}{1+t_s^2 p_{i+1}}x_i^2+\frac{2t_s^3 p_{i+1}^2 v_i-2t_s^2 p_{i+1}b_{i+1}}{1+t_s^2 p_{i+1}}x_i \qquad \text{---(*-1)}$$

$$(1+t_s^2 p_{i+1})u_i^{*2}$$
$$=(1+t_s^2 p_{i+1})\frac{(-t_s p_{i+1}x_i+t_s^2 p_{i+1}v_i-t_s b_{i+1})^2}{(1+t_s^2 p_{i+1})^2}$$
$$=\frac{t_s^2 p_{i+1}^2}{1+t_s^2 p_{i+1}}x_i^2+\frac{2t_s^2 p_{i+1}(b_{i+1}-t_s p_{i+1}v_i)}{1+t_s^2 p_{i+1}}x_i \qquad \text{---(*-2)}$$
$$+\frac{t_s^4 p_{i+1}^2 v_i^2+t_s^2 b_{i+1}^2-2t_s^3 p_{i+1}b_{i+1}v_i}{1+t_s^2 p_{i+1}}$$

$$(2t_s b_{i+1}-t_s^2 p_{i+1}v_i)u_i^*$$
$$=(2t_s b_{i+1}-t_s^2 p_{i+1}v_i)\frac{-t_s p_{i+1}x_i+t_s^2 p_{i+1}v_i-t_s b_{i+1}}{1+t_s^2 p_{i+1}}$$
$$=\frac{-(2t_s b_{i+1}-t_s^2 p_{i+1}v_i)t_s p_{i+1}}{1+t_s^2 p_{i+1}}x_i \qquad \text{---(*-3)}$$
$$+\frac{(2t_s b_{i+1}-t_s^2 p_{i+1}v_i)(t_s^2 p_{i+1}v_i-t_s b_{i+1})}{1+t_s^2 p_{i+1}}$$

Substitute (*-1), (*-2), (*-3) back into function (*) and equalize the coefficients of x^2, x, and constant on both side of the equation (*):

Quadratic terms in x:

$$p_i=1+p_{i+1}-\frac{t_s^2 p_{i+1}^2}{1+t_s^2 p_{i+1}}$$
$$\Rightarrow p_i=1+\frac{p_{i+1}}{1+t_s^2 p_{i+1}} \qquad \text{---(3)}$$

Linear terms in *x*:

$$2b_i=2b_{i+1}-t_s p_{i+1}v_i+\frac{2t_s^3 p_{i+1}^2 v_i-2t_s^2 p_{i+1}b_{i+1}}{1+t_s^2 p_{i+1}}$$
$$+\frac{2t_s^2 p_{i+1}(b_{i+1}-t_s p_{i+1}v_i)}{1+t_s^2 p_{i+1}}-\frac{(2t_s b_{i+1}-t_s^2 p_{i+1}v_i)t_s p_{i+1}}{1+t_s^2 p_{i+1}}$$
$$\Rightarrow 2b_i=\frac{2b_{i+1}-t_s p_{i+1}v_i}{1+t_s^2 p_{i+1}} \qquad \text{---(4)}$$

Terms independent of *x*:

$$c_i=\frac{t_s^4 p_{i+1}^2 v_i^2+t_s^2 b_{i+1}^2-2t_s^3 p_{i+1}b_{i+1}v_i}{1+t_s^2 p_{i+1}}$$
$$+\frac{(2t_s b_{i+1}-t_s^2 p_{i+1}v_i)(t_s^2 p_{i+1}v_i-t_s b_{i+1})}{1+t_s^2 p_{i+1}}$$
$$+t_s^2 p_{i+1}v_i^2-2t_s b_{i+1}v_i+c_{i+1}$$
$$\Rightarrow c_i=t_s^2 p_{i+1}v_i^2-\frac{t_s b_{i+1}v_i(2+t_s^2 p_{i+1})+t_s^2 b_{i+1}^2}{1+t_s^2 p_{i+1}}+c_{i+1} \qquad \text{---(5)}$$

APPENDIX B: KALMAN FILTER FOR DLQ SYSTEM

A. Filter Problem Definition

$$x(i+1)=x(i)+t_s u(i)+m+w(i) \quad i\geq 0 \qquad \text{---(B.1)}$$
$$y(i)=x(i)+n(i) \qquad \text{---(B.2)}$$

In state function (B.1), we divide the playback disturbance $t_s v(i)$ into average playback rate m plus a zero mean random variable w(i). The accumulate effect of m+w(i) performs the same as $t_s v(i)$ in initial state function (1). In observation function (B.2), y(i) is the observed state variable under network noise n(i). Where {n(i)}, similar to{w(i)}, is a sequences of white Gaussian noise with zero mean comes from network.

B. Kalman Filter Algorithm

```
Given the initial values of $x(0)$ and $\Phi(0)$
WHILE (transmitting){
    /*Predict the state x(i|i-1)*/
    $x(i|i-1)=x(i-1|i-1)+t_s u(i-1)+m$
    /*Predict the error covariance*/
    $\Phi(i|i-1)=\Phi(i-1|i-1)+Q(i)$
    /*Compute the Kalman Gain*/
    $G(i)=\frac{\Phi(i|i-1)}{\Phi(i|i-1)+R(i)}$
    Waiting for y(i) from client feedback…
    /*Estimate x(i) with measured y(i)*/
    $x(i|i)=x(i|i-1)+G(i)[y(i)-x(i|i-1)]$
    /*Update the error covariance */
    $\Phi(i|i)=(1-G(i))\Phi(i))-1)$
    Increase i
    }
```

REFERENCES

[1]. Vieron, J.; Guillemot, C., “Real-time constrained TCP-compatible rate control for video over the Internet,” *IEEE Transactions on Multimedia, Volume 6, Issue 4, Aug. 2004 Page(s):634 – 646*

[2]. Jiang, M.; Ling, N., “Low-Delay Rate Control for Real-time H.264/AVC Video Coding,” *IEEE Transactions on Multimedia, Volume 8, Issue 3, June 2006 Page(s):467 - 477*

[3]. Yuko Onoe et al, “Network information based Rate Controls on Multimedia Streaming Servers,” *23rd International Conference on Distributed Computing Systems, 2003, pp.543~548*

[4]. JeongEun Lim; Kim, J.; Ngan, K.N.; Kwanghoon Sohn, “Advanced rate control technologies for 3D-HDTV,” *IEEE Transactions on Consumer Electronics, Volume 49, Issue 4, Nov. 2003 Page(s):1498 - 1507*

[5]. Markus Fidler, “Real-Time Multimedia Streams in a Differentiated Services Network,” Dept. of Computer Science, Informatik IV, Aachen Univ. of Tech., *Proc. 10th international conference on computer and communications and Networks, 2001, pp.308~385*.

[6]. Randy Chow at al, “Traffic Dispersion Strategies for Multimedia Streaming,” *Proceedings of the 8th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS’01), 2001, pp.18~24.*

[7]. Gwang S.Jung,Kyung W.Kang, and Qutaibah Malluhi, “Multithreaded Distributed MPEG1-Video Delivery in the Internet Environment,” *Proceedings of the 2000 ACM symposium on Applied computing - Volume 2*

[8]. Mei-Ling Shyu1 et al, “Optimal Resource Utilization in Multimedia Transmission,” *IEEE international conference on Multimedia and Expo, 2001. pp.673~676.*

[9]. N.Uchida, K.Takahata and Y.Shibata, “Optimal video stream transmission control over wireless network,” *IEEE international Conference on Multimedia and Expo, Vol.3, pp.1791-1794, 27-30 June 2004*

[10]. PeterDorato, Chaouki Abdallah, and Vito Cerone, “Linear-Quadratic Control, An Introduction,” *Prentice Hall, 0-02-329962-2*

[11]. Desineni Subbaram Naidu, “Optimal control systems,” *Boca Raton, FL: CRC Press, c2003.*

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

[13]. O.Rose, “Simple and Efficient Models for Variable Bit Rate MPEG Video Traffic,” *Performance Evaluation, vol.30, pp.69~85, 1997.*

[14].http://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html

[15]. Nelson Tang, Sonia Tsui, and Lan Wang, “A Survey of Admission Control Algorithms,” *Computer Science Department, UCLA, December 16, 1998*

[16]. X.Zhou and K.Ong, “Discrete LQ rate control schedule system for multimedia transmission,” *International Conference on Advanced Information Networking and Applications, Vol. 1, Page(s):6, 18-20 April 2006.*

[17]. Jinyao Yan; Katrinis, K.; May, M.; Plattner, B.; “Media- and TCP-friendly congestion control for scalable video streams”, *IEEE Transactions on Multimedia, Volume8, Issue2, April 2006, Page(s):196 – 206*

[18]. Pilgyu Shin; Kwangsue Chung; “A Cross-Layer Based Rate Control Scheme for MPEG-4 Video Transmission by Using Efficient Bandwidth Estimation in IEEE 802.11e”, *ICOIN 2008, 23-25 Jan. 2008 Page(s):1 - 5*

Xiaofei. Zhou is a Ph.D. student in the Electrical and Computer Engineering Department of National University of Singapore. She received her Bachelor degree on Electrical and Information Engineering from Huazhong University of Science & Technology of China at 2003. Her research interests focus on multimedia technologies, multimedia streaming networks, and real time scheduling.

Kenneth. Ong is currently an associate professor of Electrical and Computer Engineering in National University of Singapore. He received his Ph.D. degree in Cambridge University on computer engineering. His research interests are real-time systems, knowledge acquisition techniques for expert systems, and distributed computing.