Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Receiver-Based Rate Control Scheme for Streaming Video over Wireless

Youn-Sik Hong, Hwa-Seok Lim,
Dept. of Computer Science and Engineering
University of Incheon
177 Dowha-dong Nam-gu 402-749 Incheon KOREA
{yshong, hepi83}@incheon.ac.kr

Cheol-Ho Lee, and Seung-Sik Choi
Dept. of Computer Science and Engineering
University of Incheon
177 Dowha-dong Nam-gu 402-749 Incheon KOREA
jullio@chol.com, sschoi@incheon.ac.kr

***Abstract*— We present a receiver-based rate control scheme in an infrastructure network. A mobile device to receive video stream over wireless plays an important role to decide its rate. As an inter-packet delay (IPD) at the sender decreases, the sending rate increases so that the IPD can be adjusted to control the rate, depending on the receiver buffer occupancy. At every round-trip time, such adjustments will be performed in order to achieve a responsive behavior to sudden changes in availability of a receiver buffer. In addition, the proposed method utilizes the discrete derivative of the buffer occupancy to reduce the occurrence of the buffer becoming full.**

Keywords*— *TFRC, video stream, PDA, receiver-based rate control, round-trip time (RTT), inter-packet delay (IPD).

I. INTRODUCTION

Delivering streaming video over a wireless network, to mobile users, is becoming a research topic of growing interest. With the emergence of small wireless handheld devices, it is expected that streaming video will be a major source of traffic to these mobile devices. In a typical VoD (Video on Demand) application, a mobile device, particularly a PDA (Personal Digital Assistant), is likely to be used for receiving video streaming from its stationary server.

A mobile device as a receiver needs some buffer space to smooth out the fluctuations in the data receiving rate from such a network. At the start of a session, or after a fast forward, the buffer needs to be filled up first, before the video can be viewed. This results in a startup delay, which depends on the buffer size at the mobile device, and the bandwidth available from the network. The buffer needs to be properly dimensioned so that the video display is continuous, without underflow at the client's buffer due to reduced bandwidth availability when the network is congested. Moreover, an inaccurately dimensioned buffer results in an expensive client system and a large startup delay.

To sustain uninterrupted video displaying at the mobile device end, continuous monitoring of the network's status is required to prevent transmission collapse. To address these concerns, real-time streaming needs a more intelligent rate adaptation, or advanced rate control mechanisms [1]. One popular rate control scheme, for streaming data in wired networks, is equation-based rate control, also known as TCP-friendly rate control (TFRC) [2]. In TFRC, the TCP-friendly rate is determined as a function of the packet loss rate, the round-trip time (RTT), and the packet size, to mimic the long-term steady performance of the TCP. One of the advantages of rate control using TFRC results in small rate fluctuations, making it attractive for streaming applications that require constant video playback quality.

A number of previous efforts have achieved improved performance of TFRC in wireless systems, but these also raise their own concerns. Firstly, mobile devices are often small and inexpensive with limited computation and buffering capacities. When a mobile client receives a video stream from its stationary server, the server and a mobile client act as a fast sender and a slow receiver, respectively, due to substantial differences in their computational capabilities. Therefore, even if there is no congestion and extremely low BER (bit error rate) over the wireless system, the mobile client cannot receive the streaming video as fast as the sending rate. Secondly, the status of the mobile device can be determined from the occupancy level of the buffer. For example, if the buffer level is less than the minimum fill level, the device stays in the state of *underflow* which ensures that sending rates are too low. If the buffer level is larger than some specified value, it enters a state of *overflow*. When this state is reached it prevents buffer overflow by ensuring that the sending rates are not too aggressive.

Many of the research works do not reflect the buffer status of a receiver, and have only considered the network status. Thus, the proposed scheme here aims to reformulate the rate in the TFRC by considering the buffer status of the end point mobile device. The advantages of this approach are as follows: first, it does not require any modifications to the network protocols, except at the application layer. To do this, the inter-packet delay at the sender can be adjusted to control the sending rate. Second, it has the potential to fully consider the processing capabilities of the mobile devices being used. The objective is to develop a simple, cost effective scheme which offers an acceptable video quality over a noisy wireless channel using a limited buffer size in the mobile device

This paper is organized as follows; related works are discussed in section 2, then the proposed methods for improving the performance are described in section 3. The experimental results are presented in section 4 and finally, our conclusions are given in section 5.

II. RELATED WORKS

This section briefly summarizes some of the related works. *Padhye* et al [4] proposed the following model of the steady state throughput for a TCP flow, as a function of packet loss rate and round trip time (RTT).

$$T = \frac{S}{rtt\sqrt{\frac{2p}{3}} + rto(3\sqrt{\frac{3p}{8}})p(1+32p^2)}$$

where T represents the sending rate, S is the packet size, rtt is the end-to-end RTT, rto is the retransmission time-out (RTO), and p is the end-to-end packet loss rate. One of the most popular end-to-end streaming protocol is presented in [2] and this works well for wired networks. However, in the presence of wireless errors, this protocol shows a degraded performance. *Chen* et al [3] uses a more simple model for TFRC, as shown in (2), which gives all the fundamental factors that affect the sending rate.

$$T = \frac{kS}{rtt\sqrt{p}}$$

where k is a constant factor. *Chen* et al proposed multiple TFRC connections to fully utilize the wireless channel.

Sisalem et al [5] proposed an LDA (Loss Delay-based Adaptation) which is a video streaming scheme based on RTP/RTCP [18]. LDA uses RR (Receiver Report) messages in RTP/RTCP to extract the network parameters such as RTT, packet loss, and RTO. Then the LDA method recalculates the sending rate by using (1). SRTP [6] is similar to LDA as it attempts to smooth out the variation of the sending rate with the network fluctuations. A typical problem of the above equation-based rate control approaches is an under-utilization of the wireless channel. For example, the calculated sending rate is about 60% of the measured sending rate. *Zheng* et al [7] derive a minimum handheld device buffer size which is proportional to the average retransmission time. As the channel BER increases, the minimum buffer size increases abruptly in order to maintain a continuous display at the mobile device.

Some of the works have focused on the issues to distinguish between packet loss caused by congestion, and that caused by wireless channel error. *Cen* et al [9] proposed and evaluated several differentiation algorithms. *Yang* et al [8] proposed a video transport protocol (VTP) based on achieved rate estimation and loss discrimination algorithm. *Krunz* et al [10] proposed an adaptive rate control scheme with the aim of reducing the bit rate of the transmitted video signal, and increasing the error protection when the channel is anticipated to be bad or the receiver playback buffer starvation is predicted. The receiver has two distinct states, *stable* and *underflow*, depending on the playback buffer threshold. The work of *Gualdi* et al [11] is somewhat similar to the proposed approach of this paper in the sense that they performed adaptive control by keeping the buffer occupancy at the decoder side between two given levels. However, they tried to control the playback frame rate as a function of the buffer occupancy. *Papadimitriou* et al [12] proposed a receiver-centric congestion control mechanism where the transmission rate is controlled by adjusting the IPG (inter-packet gap), which is similar to the IPD used in the proposed method taken here. If no congestion is sensed, IPG is reduced additively; otherwise, it is increased multiplicatively. However, the experimental results show that as an IPD becomes greater than 20ms, the packet transmission time increases exponentially. *Papadimitriou* et al did not give typical values for possible IPGs. In other words, a resulting IPD to control the sending rate falls into a specified range.

In IEEE 802.11-based wireless networks, a number of algorithms for rate adaptation have been proposed [13, 14]. Their basic idea is to estimate the channel quality and adjust the transmission rate accordingly. In the current 802.11 standard, a receiver does not provide explicit feedback information on the best rate or perceived SNR to the sender. Therefore, most practical rate adaptation algorithms make decisions solely based on ACK.

III. A RECEIVER-CENTRIC RATE CONTROL

A. *Receiver buffer occupancy*

The buffer occupancy *B(t)* of a receiver at time *t* is computed by the following equation:

$$B(t) = B(t-1) + T(t,t-1) - C(t,t-1)$$

where $B(t\text{-}1)$ represents the buffered data at time *t-1*, $T(t,t\text{-}1)$ is the server sending rate during the time interval [*t, t-1*], and $C(t, t\text{-}1)$ is the amount consumed by the stream's playback during the time interval [*t, t-1*]. With the experimental results the buffer occupancy can be estimated as in (3). For simplicity, assume that the packet size S is constant. With the buffer size of 16,384 bytes, the measured data consumption rate per second is 1.56 S, provided the playback rate is also constant[16]. The sending rate per second is approximately 10.7 S. Thus, equation (3) becomes (4).

$$B(t) \approx B_{\min} + 10.7 \times t \times S - 1.56 \times t \times S = B_{\min} + 9.14 \times t \times S$$

where $B_{\min}$ represents the minimum fill level of the buffer (*fixed*) and the unit of the time t is *seconds*. Therefore, the client buffer must have $B_{\min}$ for no underflow while waiting for the new segment of the video to arrive. Measures were carried out on a test-bed which reproduces (on a small scale) a real prototype of an infrastructure network. The prototype system is an infrastructure network that integrates a wired LAN based on Ethernet with a WLAN based on the IEEE 802.11b standard. In the test-bed, the BS (*Base Station*) is simply an AP (*Access Point*). Notice that these empirical data have been obtained using a mobile device such as a PDA (with 400MHz CPU).

For streaming videos, a stationary server and a PDA act as a fast sender and a slow receiver, respectively, due to substantial differences in their computational capabilities. Since, in typical situations, the sending rate is faster than the data consumption rate, there is a higher probability of the buffer becoming full. The minimum buffer level required at the client to prevent underflow can be approximately expressed by (5).

$$B_{\min} \propto rtt \times C^{\max}$$

$C^{\max} = \max_{0 \le i \le N}(C(i, i-1))$, and N is the length of the video. In contrast, buffers in routers are designed to be larger than the bandwidth-delay product [15].

Assume that there is no user interaction and that the playback rate is to be kept constant. In addition, the minimum fill level is pre-determined by (5) and remains constant.

$$B(t) \approx B_{\min} + T(t, t-1) \propto T(t, t-1)$$

In that case, the buffer level can be affected by the sending rate. On the contrary, the sending rate can be controlled by the client's buffer so that we have

$$B(t) \propto T(t, t-1) \Leftrightarrow T(t, t-1) \propto B(t)$$

A receiver-based buffer control adjusts the server sending rate such that the probability of the buffer becomes full and the buffer's outage being minimized. Thus, the adaptation of the server sending rate is a function of the buffer occupancy, as shown in Fig.1. In practice, if the buffer occupancy is high, the source sending rate needs to be decreased to compensate for it. Similarly, if the buffer occupancy is too low, the sending rate needs to be increased.

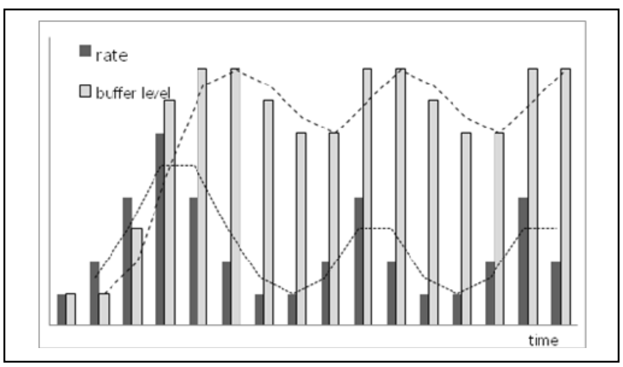

Figure 1. A server sending rate verus receiver buffer occupancy.

B. Buffer States

A buffer can be in one of the three states: *underflow*, *stable* (*normal*), or *overflow*. Since the buffer requirement for the client is related to its state, it is necessary to determine the stationary state probability of the client.

If the data production rate $T(t\text{-}1)$ never lags the consumption rate $C(t\text{-}1)$, there is never a net decrease in the amount of buffered data. Equation (8) gives a constraint to prevent starvation at the mobile device (resulting in frozen displays).

$$C(t) \le T(t), 0 \le t \le N$$

If this constraint is not satisfied, the status of the buffer remains in the *underflow* state. However, in a typical situation, the data production rate at the stationary server is higher than the consumption rate in the mobile device, as shown in (4). The maximum size of the buffer can be expressed as:

$$T(t) - C(t) \le B_{\max}, 0 \le t \le N$$

where $B_{\max}$ is the maximum buffer size. If this constraint is not satisfied, the status of the buffer remains in the *overflow* state. That means some packets may be lost or dropped. The larger the buffer size, the longer the consumption rate. Finally, if the buffer level is greater than the minimum fill level, and is also less than the overflow level, then the status of the buffer remains in the state *stable* (or *normal*).

The above three states for the client buffer can be summarized in (10). Note that α will be determined empirically and naturally speaking, it is proportional to the current sending rate.

Underflow state: $0 \le B(t) < B_{\min} = B_L$

Stable state: $B_L \le B(t) < B_H$

Overflow state: $B_H = B_{\max} - \alpha \le B(t) \le B_{\max}$

The buffer in a receiver plays an essential role in order to reduce the effects of the discrepancies between data production rate (packet generation rate at the encoder side) and data consumption rate (packet extraction rate at the decoder side). The playback latency time is directly related to the occupancy of the buffer. In addition, if the packet generation rate remains higher than the packet extraction rate for long enough, the buffer might fill up and every packet received thereon would be lost.

The system employs adaptive controls to achieve the best tradeoff between low-latency and good video fluency, by keeping the buffer occupancy at a receiver between two required levels B_L and B_H. In practice, two values are monitored to implement dynamic control; these are $B(t)$ that is the buffer and $\Delta B(t)$ that is the discrete derivative of the buffer's occupancy. In other words, if the buffer occupancy is high and continues to increase at a high rate, the sending rate needs to be decreased to compensate. Similarly, if the buffer occupancy is too low and is still decreasing, the sending rate needs to be increased.

C. Inter-packet delay

In an application layer, an inter packet delay (IPD) refers to the time interval between the transmission of two successive packets by any host, as shown in Fig.2. The transmission rate is controlled by properly adjusting the IPD. In general, the sending rate *T* increases as IPD decreases

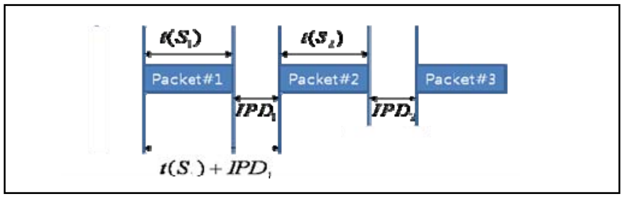

Figure 2. A definition of inter-packet delay

Consider a source that transmits *n* packets with packet lengths $S_1, S_2, \cdots, S_n$ during a time period *t,* where S_i represents the *i*th packet. The average throughput used by the connection is given by:

$$throughput = \sum_{i=1}^{n} \frac{S_i}{t} \approx \frac{n\overline{S}}{\sum_{i=1}^{n}(t(S_i) + IPD_i)} = \frac{\overline{S}}{t(\overline{S}) + \overline{IPD}}$$

where $\overline{S}$, $t(\overline{S})$, and $\overline{IPD}$ denotes the average packet length, the average time to transmit and the average inter-packet delay, respectively. It takes $t(S_i) + IPD_i$ time to transmit the *i*th packet frame. It performs the rate adjustment per RTT in order to achieve a relatively responsive behavior to the sudden changes of the buffer's availability. Assuming a measurement period of one RTT, the flow throughput is given by:

$$throughput \propto \left(\frac{\overline{S}}{\overline{RTT}}\right) \approx \frac{\overline{S}}{t(\overline{S}) + \overline{IPD}}$$

Then, the instantaneous transmission rate T_i is obtaned for the flow as:

$$T_i = \frac{S_i}{t(S_i) + \overline{IPD_i}} = f(\frac{1}{IPD_i}, S_i) \approx kf\left(\frac{1}{IPD_i}\right)$$

where *k* is used to adjust the resulting *IPD* (*t*) whose value falls into the possible range of the IPD. Fig.3 shows the downstream behavior of the transmission time with respect to IDP. Apparently, an increase in the IPG directly affects the transmission rate, as well as the flow throughput. Theoretically, IPD is chosen judiciously by considering the network parameters, such as *rtt*, the size of the receive buffer, and the capacity of the link, to avoid the degradation of the overall performance. In this test, the IPD was empirically chosen in the range of between 1ms and 20ms.

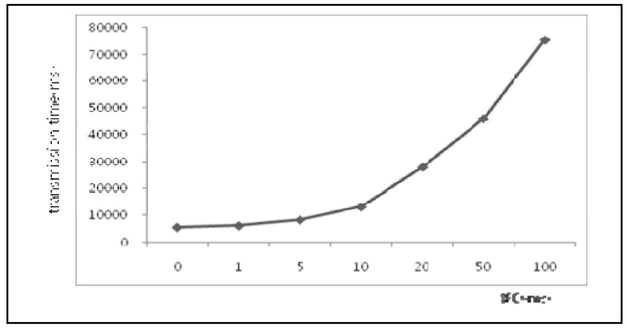

Figure 3. The transmission time with respect to inter-packet delay during downstream

The following equations show the resulting rate control scheme depending on the buffer's state. With the underflow state, the sending rate should be increased as

$$\text{Underflow state: } T(t) = \gamma_1 \times T(t-1), where, 1 < \gamma_1 < 2$$

A parameter γ_1 is computed empirically and heavily depends on $\Delta B(t)$.

$$\text{Stable state: } T(t) = \frac{T(t-1) \times rtt + 1}{rtt + \Delta rtt}$$

A parameter γ_2 between 0 and 1 is selected as the tolerable rate reduction ratio and also depends on the discrete derivative of the buffer occupancy ΔB (*t*). When the overflow state is entered, the sending rate should be reduced as

$$\text{Overflow state: } T(t) = \gamma_2 \times T(t-1), \text{ where } 0 < \gamma_2 < 1$$

IV. PERFORMANCE EVALUATIONS

A. Experiment Setup

Two different experiments were performed; one type of experiment was carried out with the ns-2 simulator [17] to analysis the efficiency of the proposed rate control in different error rate conditions. The topology of [16] representing a mixed wired-cum-wireless network is used throughout this paper. Another type of experiment is carried out on a test-bed which reproduces (on a small scale) a real prototype. In this experiment, the sending rate was traced to see whether it reacts quickly to buffer level changes.

B. The estimated minimum buffer size and maximum buffer size

First *rtt* was measured (on average) under different error rates through 10 runs to estimate the minimum buffer level and the maximum buffer level by using (5) and (9). These estimated parameters have been used to determine the adjustable constant α in (10) to classify the buffer levels. The estimated sending rates, with respect to the error rates and the measured *rtt* (in average), are summarized in Table I. Notice that the packet size *S* is kept constant.

TABLE I. THE ESTIMATED SENDING RATES UNDER DIFFERENT RANDOM ERROR RATES.

Error rate	5%	10%	15%	20%
rtt(ms)	21.4	22.05	23.75	29.15
sending rate	313.47×S	215.1209 × S	163.073×S	115.0635 × S

The simulation results show that *rtt* has changed little within the error rates of less than 10%. In the absence of random errors, the typical *rtt* has been measured as 19ms. Assuming that the error rate is less than 10% and the data consumption rate is to be kept constant $B_{\min}$ depends on the consumption rate; $B_{\min} > rtt \times C^{\max} \approx rtt \times C_{const} \approx 20ms \times C_{const}$

The empirical results give the following intuition: $\frac{C_{masured}}{T_{measured}} = \frac{1.56 \times S}{10.7 \times S} = 0.15 < 0.2$

Thus, the estimated lower bound on the buffer level is about 20% of the fixed buffer size. Because a larger buffer size results in a shorter processing time, the parameter α in (10) becomes smaller with a larger buffer size. That means the data consumption rate also increases as the buffer size increases.

C. *The adaptive rate control and its effectiveness*

The rate variation to IPD changes were tested under different packet sizes. This test is indicative of the agility of the proposed method, and also of its efficiency. As shown in Fig.4, with smaller packet size the rate variation to IPD changes quickly. By setting the IPD to 20ms, then the rate will be drastically decreased to 24% of that of 1ms.

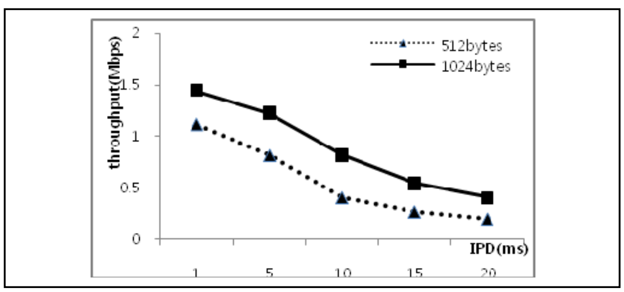

Figure 4. The transmission time with respect to IPD

Simulations were conducted varying each of the parameters: $B_{\min}$, B_H, γ_1, and γ_2. From the previous work [16], $B_{\min}$ and B_H are selected based on the least standard deviation. These parameter value settings are summarized in Table II.

TABLE II. THE PARAMETER VALUE SETTINGS FOR OUR SIMULATION

Parameter	B_{min}	B_H	γ_1	γ_2
value	0.2× B_{max}	0.9× B_{max}	1.2	0.9

As the discrepancy in the buffer level becomes relatively low, the sending rate is changed slowly. With the large discrepancy of the buffer level, the rate is changed rapidly. As mentioned earlier, it performs the rate adjustment per RTT so that the estimated latency to adjust the rate is RTT. With the ns-2 simulation typical value of RTT is 20~30ms but in a real prototype system the maximum RTT is 55ms.

For streaming video between a stationary server and a mobile station such as a PDA, the server sending rate is about 5 times faster than the consuming rate at the mobile receiver. Therefore if the system fails to adjust the rate at the correct time, the possibility of entering the overflow state increases at a high rate. In addition, if the buffer becomes full, returning to a stable state by reducing the sending rate is quite difficult.

With the previous work [16], called *Type-II* method, the discrete derivative of the buffer occupancy ΔB (*t*) was not considered to adjust the sending rate. For the proposed method taken here, called *Type-I* method, the values of γ_1 and γ_2 can be slightly altered depending on the ΔB (*t*). For example, if the buffer level remains at the underflow state and continues to increase (i.e., ΔB (*t*) >0), γ_1 is set to 1.3 instead of 1.2. On the contrary, in the case of the overflow state, if the buffer level continues to decrease (i.e., ΔB (*t*) < 0), γ_2 is set to 0.8 instead of 0.9 for reducing the rate at a higher rate.

Thus, the type-I method employs a pre-adjustment using ΔB (t) to reduce the occurrences of the buffer's fullness. The differences between the Type-I and the Type-II method are summarized in Table III. Since the type-I method can predict the tendency of the buffer occupancy using ΔB (t), the possibility of over-utilization of the buffer can be low.

TABLE III. THE SUMMARIZATION OF THE TWO PROPOSED METHODS: *TYPE-I* AND *TYPE-II*

State	Buffer occupancy	Sending rate	Proposed method	
			Type- I	*Type-II*
under flow	0≤B(t) ≤B_{min}	$\gamma_1 \times T(t-1)$	$\gamma_1 = 1.2$ *if* $\Delta B(t) > 0$ $\gamma_1 = 1.3$ *otherwise*	$\gamma_1 = 1.2$
normal	B_{min}≤ B(t)≤B_H	$\frac{T(t-1) \times rtt + 1}{rtt + \Delta rtt}$	the same as left	the same as left
over flow	B(t)≥ B_H	$\gamma_2 \times T(t-1)$	$\gamma_2 = 0.8$ *if* $\Delta B(t) < 0$ $\gamma_2 = 0.9$ *otherwise*	$\gamma_2 = 0.9$

Fig.5 compares the results over time of the sending rate for the type-I and type-II methods. In the simulation time of greater than 5.6 seconds with the type-II method the rate changes too rapidly. That means the system did not escape easily from the overflow state. If the sending rate is too high, the possibility of packet losses becomes high. With the overflowed socket buffer for a mobile device, it reacts to bandwidth changes slowly and then result in a performance degradation.

Fig.6 shows the sending rate with various buffer sizes. With smaller buffer size it maintains a smoother rate and reacts to bandwidth changes slowly but it results in under-utilization of the wireless channel.

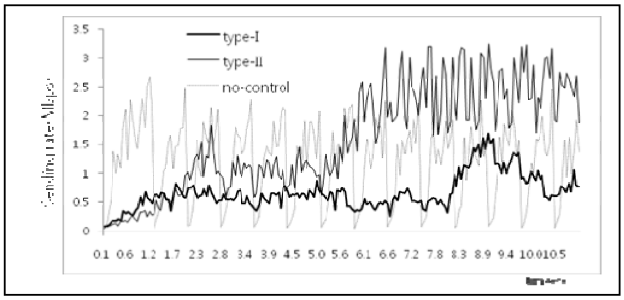

Figure 5. The comparison result of time evolution of the sending rate for Type-I and Type-II method (with the buffer size of 32,768bytes).

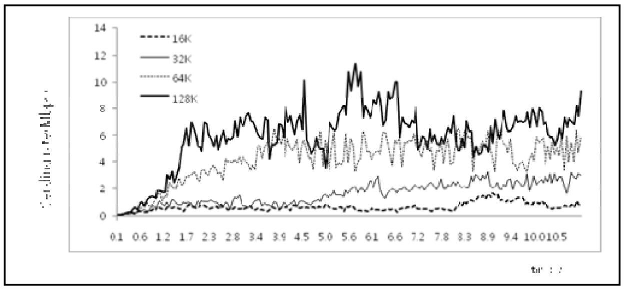

Figure 6. The sending rate under different buffer sizes.

The system was built and tested in a real prototype system with the proposed adaptive rate control algorithm. For this test, the resolution of a PDA for displaying video is set to 240x160. The measured frame rate per second is, at most, 20 with the socket buffer size of 16,384bytes. Notice that, for simplicity, only one PDA is used as the receiver. The bandwidth and the round trip time are obtained as 426,300bps and 19ms, respectively, in the case of the packet size of 1,500 bytes. The overall shape of the measured discrete buffer level is similar to Fig.7. However, the shape of the rate variance is somewhat different from Fig.8. The reason is that the duration to adjust the rate is large than was expected.

V. Conclusions

In this paper, a buffer controlled adaptive video streaming method for wireless systems has been proposed. In a prototype VoD system, a VoD server delivers video streams to a mobile device such as a PDA. However, since a mobile device has limited computation and less buffer capacities, the socket buffer for it can be frequently overflowed even if there is no network congestion or low BER over a wireless channel. Therefore the rate at the server can be strongly affected by the receiver's state. A simple prediction has been proposed for controlling the sending rate at the server by inspecting the current buffer level. In addition, a simple cost effective rate control was presented by varying the inter-packet delays at the server. It does not require any modifications to network protocols. With the adaptation of the proposed scheme the occurrences of the overflowed socket buffer are kept to be low. Thus, the proposed method maintains a smoother rate and reacts to bandwidth changes more quickly.

References

[1] Boudec, J.-Y. L., Rate adaptation, Congestion Control and Fairness: A Tutorial, 2008, pp 1-44.

[2] S. Floyed et al., "Equation-based Congestion Control for Unicast Applications", In Proc. of ACM SIGCOMM 2000, 2000, pp 43-56.

[3] Chen, M., Zakhor, A., "Rate Control for Streaming Video Over Wireless", IEEE Wireless Communications, 2005, pp32-41.

[4] . Padhye, V. Firoiu, D. Towsley, and J. Kurpose, "Modeling TCP throughput: A simple model and its empirical validation", In Proc. of ACM SIGCOMM, 1998, pp.303-314.

[5] D. Sisalem, and H. Schulzrinne, "The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme", Int. Workshop on Network and Operating System Support for Digital Audio and Video(NOSSDAV), 1998.

[6] B. Song, K. Chung, and Y. Shin, "SRTP: TCP-friendly congestion control for multimedia streaming", 16th Int. Conf. on Information Networking, 2002.

[7] B. Zheng, and M. Atiquzzaman, "A Novel scheme for streaming multimedia to personal wireless handheld devices", IEEE Trans. On Consumer Electronics, vol.49, no.1, 2003, pp32-40.

[8] G. Yang, M. Gerla and M. Y. Sanadidi, "Adaptive Video Streaming in Presence of Wireless Errors", In. Proc. of IPIF/IEEE MMNS, Oct., 2004.

[9] S. Cen, C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses", IEEE/ACM Trans. On Networking, Vol.11, No.5, 2003, pp703-717.

[10] M. M. Krunz and M. Hasssn, "Adaptive rate control scheme for video streaming over wireless channels", in Proc. of Data Compression Conference (DCC'04), 2004.

[11] G. Gualdi, R. Cucchiara and A. Prati, "Low-latency Live Video Streaming over Low-Capacity Networks", in Proc. of 8th Int. Symp. on Multimedia(ISM'06), 2006.

[12] P. Papadimitriou, V. Tsaoussidis and C. Zhang, "Enhancing video streaming delivery over wired/wireless networks", in Proc. of 13th European Wireless Conference(EW2007), 2007.

[13] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach", in Proc. of 7th ACM Int. Symp. on Modeling, analysis and Simulation of Wireless and Mobile Systems, Oct., 2004.

[14] S. H. Y. Wong, H. Yang, S. Lu, and W. Bharghavan, "Robust rate adaptation for 802.11 wireless networks", in Proc. of 7th ACM Int. Conf. on Mobile Computing and Networking, 2006, pp. 146-157.

[15] Lachlan L. H. Andrew, et al, "Buffer Sizing for Non-homogeneous TCP Sources", IEEE Communication Letters, Vol. 9, No. 6, 2005, pp. 567-569.

[16] Youn-Sik Hong, Hwa-Seok Lim and Jun-Sik Hong, "A Cost effective rate control for streaming video dedicated to wireless handheld devices", in Proc. of 2nd Int. Conf. on Multimedia and Ubiquitous Engineering (MUE2008), May 2008, pp. 537-542.

[17] The network simulator ns-2, http://www.isi.edu/nanam/ns/

[18] RTP: A Transport Protocol for Real-Time Application, http://www.ietf.org/rfc/rfc3550.txt