Implementation of Video Transfer with TCP-friendly Rate Control Protocol

Masaki Miyabayashi, Naoki Wakamiya, Masayuki Murata, Hideo Miyahara
Department of Informatics and Mathematical Science,
Graduate School of Engineering Science, Osaka University, Japan
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
Tel: +81-6-6850-6588, Fax: +81-6-6850-6589
E-mail: {miyabays, wakamiya, murata, miyahara}@ics.es.osaka-u.ac.jp

Abstract: As the use of real-time multimedia applications increases, a considerable amount of "greedy" UDP traffic would easily dominate network bandwidth and packet loss. As a result, bandwidth available to TCP connections is oppressed and their performance extremely deteriorates. In order that both TCP and UDP sessions fairly co-exist in the Internet, it is vital that we consider the fairness among both protocols. In this work, we implement a "TCP-friendly" rate control mechanism suitable to video applications and consider its applicability to a real system through observation of the video quality at the receiver and the connection state. It is shown that we can achieve high-quality and stable video transfer fairly sharing the network bandwidth with TCP by applying our rate control at a control interval of 32 times as long as RTT.

1 Introduction

Since the current Internet does not provide QoS (Quality of Service) guarantee mechanisms, each application chooses the preferable transport protocol to achieve the required QoS level. For example, traditional data applications such as `http`, `ftp`, `telnet` employ TCP which accomplishes the loss-free data transfer by means of window-based flow control and retransmission mechanisms. On the other hand, real-time multimedia applications such as video conferencing prefer UDP in order to avoid the unacceptable delay introduced by packet retransmissions. Against the network congestion TCP throttles its transmission rate, whereas UDP does not have such control mechanisms. As real-time multimedia applications increase, a considerable amount of "greedy" UDP traffic would dominate network bandwidth. As a result, the available bandwidth to TCP connections is oppressed and their performance extremely deteriorates.

In order that both TCP and UDP sessions fairly co-exist in the Internet, it is meaningful to consider the fairness among protocols. In recent years, several researches have been focused on the investigation of the "TCP-friendly" rate control [1-7]. "TCP-friendly" is defined as "a non-TCP connection should receive the same share of bandwidth as a TCP connection if they traverse the same path" [4]. With the rate control based on the concept of "TCP-friendly", a UDP connection achieves a fair share of the network bandwidth with a TCP connection on the same path by regulating its sending rate according to the network condition.

Some researches on TCP-friendly rate control have been devoted to the applicability to real-time MPEG-2 video communications, and the effective control mechanisms have been proposed [6, 7]. However, the effectiveness of those mechanisms is evaluated through simulation experiments assuming the ideal network system environment. That is, they do not take into account several factors which would affect the performance of the rate control. Those include the fluctuation of control interval or the quality degradation during playback of a video sequence. In this work, we implement TCP-friendly rate control protocol suitable to video applications (called MPEG-TFRCP), which is proposed by our research group. We demonstrate its applicability to a real system through evaluation of the perceived video quality and observation of the traffic on the link.

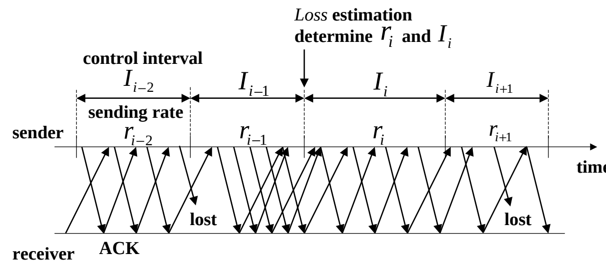

Figure 1: TFRCP video transfer

The paper is organized as follows. In Section 2, we introduce MPEG-TFRCP and show some results of our real system experiments. In Section 3, we consider the mechanism to improve our proposed MPEG-TFRCP algorithm. Finally, we summarize our paper and outline our future work in Section 4.

2 MPEG-TFRCP

In this section, we introduce MPEG-TFRCP suitable for video transfer, which is proposed by our research group [6, 7]. In order to transmit a video sequence with high and stable quality while fairly sharing the network bandwidth with TCP, our MPEG-TFRCP behaves as illustrated in Fig. 1; at the end of the control interval $i-1$ whose duration is I_{i-1}, the sender estimates the network condition from information gathered within the interval, then derives the throughput of a TCP session r_{TCP}, and finally regulates its sending rate r_i for the next interval i.

2.1 MPEG-TFRCP mechanism

In [6, 7], the MPEG-TFRCP mechanism achieves fairness among TCP and UDP connections by adjusting the sending rate to the estimated TCP throughput at the regular interval of 32 times as long as RTT. The target rate of interval i (denoted as r_i), is determined as

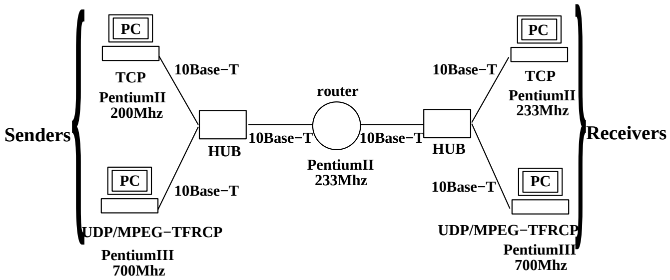

Figure 2: System configuration

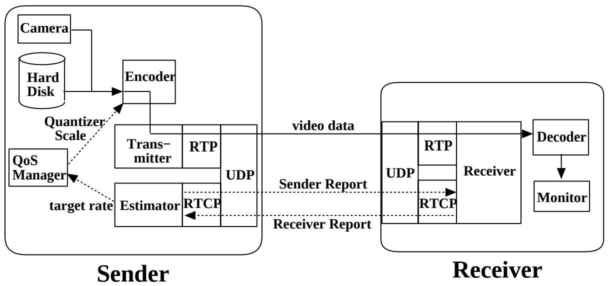

Figure 3: MPEG-TFRCP sender & receiver

$$r_i = \begin{cases} r_{TCP} \approx \dfrac{MTU}{RTT\sqrt{\frac{2p}{3}} + T_0 \min(1, 3\sqrt{\frac{3p}{8}})p(1+32p^2)}, & \\ & \text{if } p > 0 \quad (1) \\ 2 \times r_{i-1}, & \\ & \text{if } p = 0 \quad (2) \end{cases}$$

where *MTU* stands for the maximum transfer unit size, *p* is the packet loss probability, *RTT* and T_0 are for the round trip time and the retransmission timeout, respectively. The network condition (conjectured by RTT and packet loss probability) is estimated from the feedback information obtained by means of ACK packets. Then, the video sending rate is effectively adjusted the target rate r_i by choosing an appropriate quantizer scale [8].

2.2 Implementation of MPEG-TFRCP

To investigate the applicability of the MPEG-TFRCP to the actual system, we built the small-scale 10Base-T network as illustrated in Fig. 2. The network consists of four personal computers, two HUBs and one PC router. Two PCs communicate with each other on TCP connection. The others employ UDP or MPEG-TFRCP. Our MPEG-TFRCP sender transmits the video data using RTP (Real-time Transport Protocol) on the UDP protocol stack and utilizes the RTCP (Real-Time Control Protocol) mechanism to obtain the feedback information (See Fig. 3). The sender regularly emits a RTCP Sender-Report packet to the receiver every 5 pictures. On receiving the RTCP packet, the receiver sends back a RTCP Receiver-Report packet which contains the number of packets it has received since the previous RTCP packet. Then the sender can derive the packet loss probability from the reported number and the round trip time by observing the timestamp.

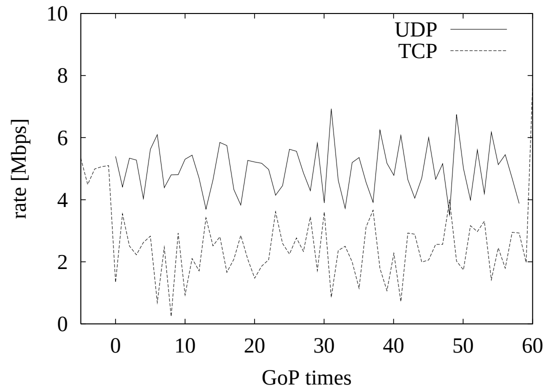

Figure 4: Rate variation (UDP vs. TCP)

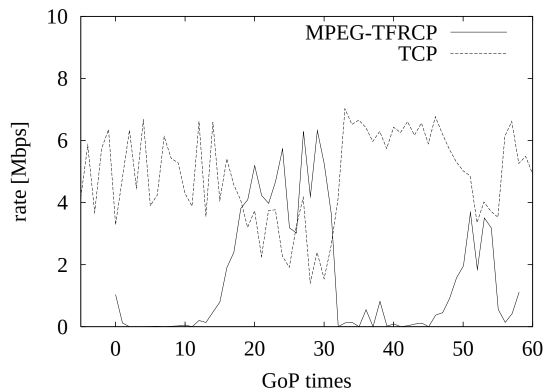

Figure 5: Rate variation (MPEG-TFRCP)

2.3 Evaluation of MPEG-TFRCP

When both TCP and UDP sessions co-exist in the network, the performance of TCP connection is heavily affected by the UDP traffic as shown in Fig. 4. In the experiment, both sessions transfer the MPEG-2 video data of the average rate 5 Mbps. The experiment time on an x-axis is expressed by the GoP time of MPEG-2 which is equal to 1.001 sec (GoP size/frame rate = 30/29.97). Y-axis express the actual rate measured on the link with `tcpdump`. In this experimental environment, we observe that RTT is about 10 ms.

On the contrary to Fig. 4, Fig. 5 shows the experimental result where TCP and MPEG-TFRCP sessions compete for the bandwidth. It can be observed that the MPEG-TFRCP connection regulates its data sending rate during the session. However, Fig. 5 is quite different from results in [6, 7] owing to the large packet loss probability and the unstable RTT introduced by the processing delay at the PC router and the end systems. For instance, Fig. 6 shows the packet loss probability p ("loss"), the target rate r_i ("target rate") estimated by Eqs. (1) and (2) and the average of the actual rate ("video rate") of MPEG-TFRCP. Here we should note that there are some durations where the video rate is higher than the target rate because video data we employed in the experiments ranges from 1.138 Mbps to 27.260 Mbps. In those cases, the sender first sends as much data as possible, then

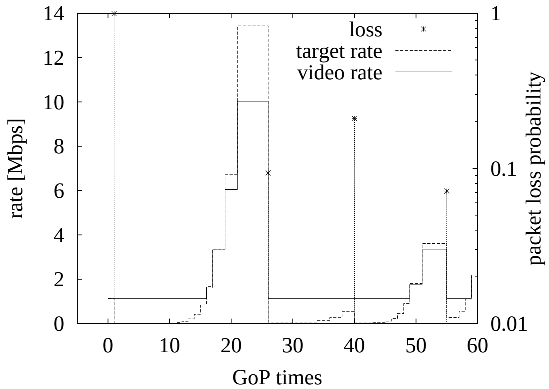

Figure 6: Packet loss probability variation (MPEG-TFRCP)

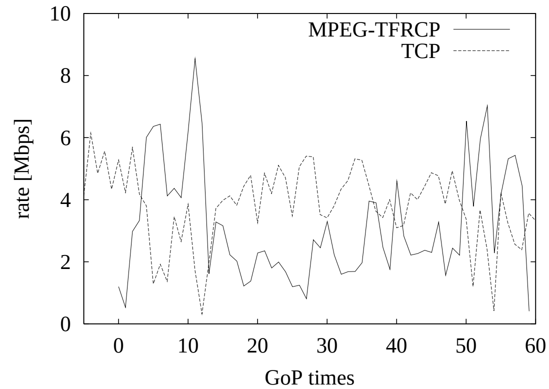

Figure 7: Rate variation (Improved)

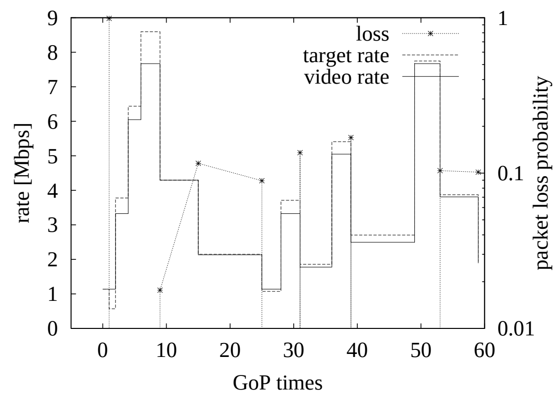

Figure 8: Packet loss probability variation (Improved)

stops video transmission as shown in Fig. 5.

The MPEG-TFRCP sender doubles its data sending rate during a loss-free period. As it encounters packet losses, it suddenly shrinks the sending rate because the loss probability is high due to the network congestion caused by itself. Although not shown in figures, the perceived video quality considerably fluctuates as the video rate changes. Especially when congestion occurs, decreased target rate becomes far below the minimum video rate, and no picture can be displayed at the receiver. To solve the drastic rate variation, we improve our MPEG-TFRCP in the next section. The control intervals are not the same during the session as shown in Fig. 6, because the observed RTT fluctuate in an actual system environment. To smooth the RTT variation, we should employ some filtering algorithm.

3 Improving MPEG-TFRCP

In this section, we consider several methods to improve the MPEG-TFRCP mechanism for achieving the higher friendliness and the stable video quality.

3.1 Investigation into rate control algorithm

The previous work [7] proposes a new rate increase algorithm, Eq. (4), which imitates the window-based flow control of TCP, instead of the rate determination algorithm Eq. (1). By the new algorithm, the additive rate increase as in TCP can be performed in MPEG-TFRCP. We also replace Eq. (2) with the TCP rate decrease algorithm, Eq. (3). Although Eq. (1) assumes that statistics are derived from the long-term observation and the network condition is relatively steady, those are not true on the actual system and it leads to the unexpected result (Fig. 5).

$$r_i = \begin{cases} \dfrac{r_{i-1}}{2}, & \text{if } p > 0 \qquad (3) \\ r_{i-1} + \dfrac{MTU \times I_{i-1}}{RTT^2}, & \text{if } p = 0 \qquad (4) \end{cases}$$

The results are depicted in Figs. 7 and 8. As shown in those figures, the rate variation becomes relatively small, while our MPEG-TFRCP connection can receive the almost same throughput as a TCP connection. The subjective video quality of Fig. 7 measured by MOS (Mean Opinion Score) evaluation is 2.6 and higher than Fig. 5 (1.8). Further, the video quality is still higher than that of UDP case (2.4).

We also evaluate the other combination of rate control strategies, i.e. Eqs. (1) and (4), or Eqs. (3) and (2). Although we do not show those results due to the space limitation, the algorithm with Eqs. (3) and (4) provides the most preferable control among them.

3.2 Investigation into appropriate control interval

As we mentioned in Section 2, our MPEG-TFRCP estimates the TCP throughput from the network condition conjectured by the obtained feedback information, and then regulates the data sending rate at the periodic interval as long as 32×RTT (we call this strategy as 32-RTT). The duration of each control interval must be carefully determined to attain the effective rate control. When the control interval is too short, the sending rate changes greatly owing to the frequent rate control, then the perceived video quality becomes unstable. Conversely, infrequent control leads to the unfairness because video applications cannot follow changes of network condition.

In Figs. 9 and 10, employing the rate control mechanism in Section 3.1, we depicts variations of the averaged rate for a few settings of control interval such as 16×RTT rounded by GoP-time and 64×RTT rounded by GoP-time, respectively.

In Fig. 9, we depict experimental results of MPEG-TFRCP with a 16-RTT interval. When the control interval is

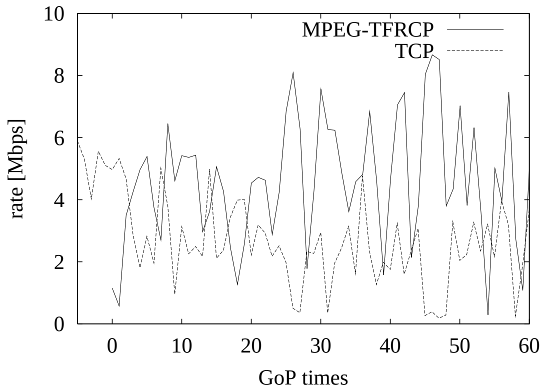

Figure 9: Rate variation (16-RTT)

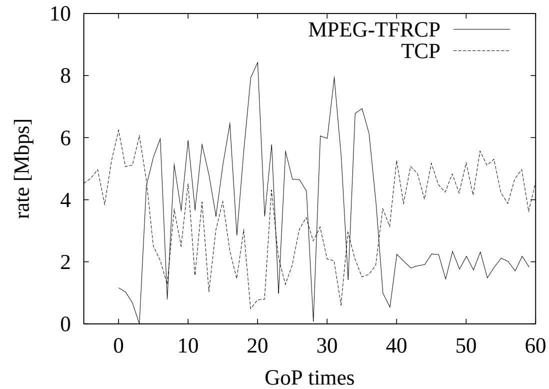

Figure 10: Rate variation (64-RTT)

shorter in comparison to 32-RTT (Figs. 7 and 8), the sending rate extremely changes. This is because the sender cannot obtain enough feedback information to accurately estimate the network condition.

In the case of a longer control interval 64-RTT (Fig. 10), the rate variation becomes relatively stable. However, the sender keeps to send the video data at the higher rate in contrast with TCP connection. The unfair condition tends to be hold longer than the control with shorter interval. The influence becomes heavier in the real system because RTT changes during the session and could suddenly increase as shown in Fig. 6.

Thus, we conclude that the 32-RTT control interval is preferable in our system in order that our TFRCP connection receives the almost same throughput as a TCP connection and the perceived video quality becomes high and stable.

4 Conclusion

In this paper, we have shown, through real system experiments, that it is appropriate for real-time video application to use a rate control which imitates the TCP window-based flow control. We can achieve stable video transfer when video rate is regulated at a control interval of 32 times as long as RTT. We have found that our MPEG-TFRCP attained high-quality and stable video transfer fairly with TCP.

However, there still remain several research works. First, in this paper we have experimented with the small-scale network. We must consider the large-scale network in which a large number of sessions co-exists. Second, though our system employ RTCP to obtain the feedback information required for the estimation of network condition, those RTCP packets may be lost when the network is heavily loaded. In that case, our MPEG-TFRCP cannot send video data at the proper rate because precision of the network condition estimation degrades. We should adopt some error recovery mechanism to avoid the inaccurate estimation. We are currently investigating those issues and we will achieve further improved MPEG-TFRCP in the near future.

Acknowledgements

This work was partly supported by Research for the Future Program of Japan Society for the Promotion of Science under the Project “Integrated Network Architecture for Advanced Multimedia Application Systems”, a Grant-in-Aid for Scientific Research (A) 11305030 and a Grant-in-Aid for Encouragement of Young Scientists 12750322 from The Ministry of Education, Science, Sports and Culture of Japan, Special Coordination Funds for promoting Science and Technology of the Science and Technology Agency of the Japanese Government, and Telecommunication Advancement Organization of Japan under the Project “Global Experimental Networks for Information Society Project.”

References

[1] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, “The macroscopic behavior of the TCP congestion avoidance algorithm,” *ACM SIGCOMM Computer Communication Review*, vol. 27, pp. 67–82, July 1997.

[2] J.-C. Bolot and T. Turletti, “Experience with control mechanisms for packet video in the Internet,” *ACM SIGCOMM Computer Communication Review*, vol. 28, pp. 4–15, January 1998.

[3] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP throughput: A simple model and its empirical validation,” *Proceedings of ACM SIGCOMM’98*, vol. 28, pp. 303–314, October 1998.

[4] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, “A model based TCP-friendly rate control protocol,” *UMASS-CMPSCI Technical Report*, October 1998.

[5] R. Rejaie, M. Handley, and D. Estrin, “RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet,” *Proceedings of IEEE INFOCOM’99*, March 1999.

[6] N. Wakamiya, M. Murata, and H. Miyahara, “On TCP-friendly video transfer with consideration on application-level QoS,” *to be presented at IEEE ICME 2000*, July 2000.

[7] N. Wakamiya, M. Murata, and H. Miyahara, “On TCP-friendly video transfer,” *submitted to SPIE Internet III*, 2000.

[8] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, “QoS mapping between user’s preference and bandwidth control for video transport,” *Proceedings of IFIP IWQoS’97*, pp. 291–302, May 1997.