MPEG-4 Video Transfer with TCP-friendly Rate Control

Naoki Wakamiya, Masaki Miyabayashi, Masayuki Murata, and Hideo Miyahara

Graduate School of Engineering Science, Osaka University

1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan

{wakamiya, miyabayashi, murata, miyahara}@ics.es.osaka-u.ac.jp

Abstract. It is widely known that network bandwidth is easily monopolized by distributed multimedia applications due to their greedy UDP traffic. In this paper, we propose TCP-friendly MPEG-4 video transfer methods which enable real-time video applications to fairly share the bandwidth with conventional TCP data applications. We consider how video applications should regulate video quality to adjust video rate to the desired sending rate which is determined by TCP-friendly rate control algorithm. Carelessly applying TCP-friendly rate variation to the video application would seriously degrade the application-level QoS. For example, the control interval should be long enough to avoid the fluctuation of video quality caused by too frequent rate control. However, popular TCP-friendly rate control algorithms recommend that a non-TCP session regulates its sending rate more than once a RTT. Through simulation experiments, it is shown that high-quality and stable video transfer can be accomplished by our proposed methods.

1 Introduction

Since the current Internet does not provide QoS (Quality of Service) guarantee mechanisms, each application chooses the preferable transport protocol to achieve required performance. For example, traditional data applications such as `http`, `ftp`, `telnet` employ TCP which accomplishes loss-free data transfer by means of window-based flow control and retransmission mechanisms. On the other hand, loss-tolerant real-time multimedia applications such as video conferencing or video streaming prefer UDP to avoid unacceptable delay introduced by packet retransmissions. UDP is considered selfish and ill-behaving because TCP throttles its transmission rate against the network congestion whereas UDP does not have such control mechanisms. As the use of real-time multimedia applications increases, 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 devoted into investigation on the “TCP-friendly” rate control [1,2,3,4,5,6,7,8,9,10]. “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” [5]. A TCP-friendly system regulates its data sending rate according to the network condition, typically expressed in terms of the round-trip-time (RTT) and the packet loss probability, to achieve the same throughput that a TCP connection would acquire on the same path. In particular, TCP-Friendly Rate Control (TFRC) proposed in [9,10] has the feature of adjusting a

transmission rate so smoothly while coping with network congestion. Therefore, TFRC has been receiving attention as the effective rate control mechanism for realizing multimedia communications fairly sharing the network bandwidth with TCP data sessions. It is meaningful to consider TCP-friendly video transfer because many researchers are engaged in investigations on packet scheduling algorithms on a router with which ill-behaving, that is, non-TCP flows are penalized aiming to provide QoS-guaranteed service in the Internet.

In our previous works [7,8], we have been devoted into investigation of the applicability of TCP-friendly rate control to real-time MPEG-2 video communications. We proposed effective control mechanisms, called MPEG-TFRC, with consideration of several factors which affect the efficiency of rate control, such as length of control interval, algorithms to adjust video rate to the target rate. For example, although it is recommended that a TFRC system regulates sending rate more than once a RTT, it is unrealistic to control video quality so frequently, in some cases, at the rate higher than video frame rate. We verified the effectiveness of our proposed mechanisms through simulation experiments on video trace data. Further, we implemented the mechanisms on a actual video communication system and performed several experiments to investigate the applicability and practicality of our mechanisms and showed that MPEG-2 video application could fairly share link bandwidth with TCP connections without introducing serious video quality degradation and fluctuation. However, our mechanisms cannot be applied to wide area networks which consist of variety of networks such as PSTN, ISDN, wireless networks, optical fiber networks, because we only consider MPEG-2 video streams ranging from 1.5 Mbps to 24 Mbps.

In this paper, we focus on MPEG-4 video systems which have highly efficient coding algorithms and error resilient capabilities, and investigate TCP-friendly video rate control mechanisms. Specifically, we employ Fine Granular Scalability (FGS) [11,12,13] as a video coding algorithm to accomplish highly efficient and scalable rate control. The rate adjustment based on regulating the level of quantization in our previous work on MPEG-TFRC is also applicable to MPEG-4 video systems. However, to successfully adapt to the TCP-friendly rate, we should know the relationship among the quantizer scale and the resultant video rate. In addition, the rate control is not flexible enough because the quantizer scale is a discrete value. First we describe the basic characteristics of FGS algorithm, then we consider mechanisms for adjusting quality and rate of FGS video streams according to the TFRC mechanism. Through simulation experiments, we show that our proposed methods can provide high-quality, stable and TCP-friendly video transfer.

The paper is organized as follows. In Section 2, we briefly introduce MPEG-4 video coding technique and FGS algorithm, then evaluate the basic characteristics of FGS. In Section 3, we propose several methods of FGS video transfer which accomplish TCP-friendly and high quality real-time video communication. Finally, we summarize our paper and outline our future work in Section 4.

Fig. 1. An example of MPEG-4 video structure

2 Fine Granular Scalability Coding Algorithm

In this paper, we consider real-time video applications in narrow bandwidth networks and employ compressed video streams coded by MPEG-4, specifically, FGS algorithm, which is excellent in adaptation to the bandwidth variation, compression efficiency and error tolerance among MPEG-4 video-coding standards.

2.1 MPEG-4 Video Coding Technique

An MPEG-4 video stream (VOS: Visual Object Sequence) consists of one or more visual objects (VO). Each VO sequence consists of several layer streams (VOL: Video Object Layer). A layer video stream is composed from a sequence of VOP (Video Object Plane), each of which corresponds to a frame. MPEG-4 accomplishes high compression ratio by applying optimum coding algorithm suited to each VO. A VO corresponds to whole of a rectangle frame as in MPEG-1 and MPEG-2, a specific region of a frame, or an natural object such as a human, an animal, a building and so on. MPEG-4 can handle both rectangular and arbitrary shape VO, but all VOs are first divided into macroblocks of 16×16 pixels, which consists of four 8×8 blocks of luminance and two 8×8 blocks of chrominance (called 4:2:0 format). Coding operation is performed on a macroblock basis. In this paper, we only consider traditional rectangle VOs as in MPEG-1 and MPEG-2. However, our method is applicable to arbitrary shape VO only if there exist techniques to regulate video rate.

An MPEG-4 video stream consists of three types of VOPs as shown in Fig 1. VOP is the basic unit of image data and is equivalent to the frame or picture of MPEG-1 and MPEG-2. I-VOP is a self-contained intra-coded picture and coded using information only from itself. P-VOP is predictively coded using motion compensation algorithm referring to the previously coded VOP. B-VOP is a bidirectionally predictive-coded VOP using the differences between both the previous and next VOPs. An I-VOP is directly and indirectly referred to by all following P and B-VOPs until another frame is coded as an I-VOP. It is recommended to have I-VOPs regularly in a MPEG-4 video stream since motion compensation-based compression efficiency decreases as the distance from a referring frame to a referred picture becomes longer. In addition, by inserting I-VOPs as “refreshing points”, we can achieve error resilience in the video transfer. Since an entire frame can be completely reconstructed from an successfully received I-VOP, error

propagation can be interrupted when video quality is degraded in the preceding VOPs due to accidental packet losses. A sequence of VOPs beginning from an I-VOP is called GOV (Group Of VOP) and defined by two parameters, number of P-VOPs between two I-VOPs and number of B-VOPs between two P-VOPs. In an example of Fig. 1, they are 3 and 2, respectively. Using GOV structure is highly recommended for regularity of video traffic, error resilience and accessibility to video contents.

2.2 FGS Video Coding Algorithm

To cope with TCP-friendly rate control mechanisms, video applications should adjust video traffic rate to the desired rate by controlling the amount of video data. Since the amount directly corresponds to the video quality, rate control can be accomplished by regulating video quality. One way of video rate regulation is changing coding parameters such as frame rate, frame size and degree of quantization. Those are related to temporal, spatial and SNR resolution of compressed video data, respectively. In our previous works on TCP-friendly MPEG-2 video transfer [7,8], we regulate the MPEG-2 video rate by choosing appropriate quantizer scale, i.e., SNR resolution, according to the desired target rate determined by a TCP-friendly mechanism. Through experiments, it is shown that high-quality and TCP-friendly MPEG-2 video transfer can be performed with our proposed method. We take into account the relationship among quantizer scale, video rate and perceived video quality obtained in our research work on QoS mapping method for MPEG-2 video [14]. This is a good starting point of determining the control parameter appropriate for video rate adjustment with consideration of the perceived video quality. However, we found that the parameter changing is somewhat a coarse control and generated video traffic does not necessarily fit to the desired TCP-friendly rate. This is because we have only stepwise variation of possible video rate due to discrete set of parameters. Furthermore, the relationship among coding parameters and resultant video rate differs among video streams.

One might think of scalable or layered video coding algorithms standardized in MPEG-2 and MPEG-4, i.e., temporal, spatial and SNR scalability. The layered coding is certainly another way of video rate adjustment, but not powerful enough. Even if we combine two or more scalabilities, the number of achievable rate is at most several tens [15].

In this paper, expecting higher flexible and scalable rate adjustment capability, we employ Fine Granular Scalability (FGS) video coding algorithm [11,12,13] considered as a compression method suitable for video streaming applications and being introduced into MPEG-4 standards. Figure 2 illustrates the basic structure of FGS video stream. FGS is also categorized into layered coding algorithm and an FGS video stream consists of two layers, Base Layer (BL) and Enhancement Layer (EL). The BL is generated using motion compensation and DCT (discrete cosine transform)-based conventional MPEG-4 coding algorithm and provides minimum video quality. The EL is generated from the BL data and the original frame. The embedded DCT method is employed for coding EL to obtain fine-granular scalable compression. By combining BL and EL, one can enjoy higher quality video presentation.

The video quality depends on both the encoding parameters (quantizer scale, etc.) and the amount of supplemental EL data added. Even if only little EL data is used in

Fig. 2. An example of FGS video structure

Fig. 3. Relationship among quantizer scale and average rate

decoding a VOP, the perceived video quality becomes higher. Thus, it seems effective to send as much EL data as possible in addition to the BL data, according to the target rate which is generally higher than BL rate. Losses of the BL data have a significant influence on perceived video quality because BL is indispensable for decoding VOP. On the other hand, although compression efficiency is not high since EL is coded without motion compensation technique, the EL data have the outstanding error tolerance because of the scalable coding algorithm and the locality of error propagation where loss of EL data only affects the VOP.

2.3 Basic Characteristics of FGS

In this section, we evaluate the basic characteristics of FGS, in terms of variations of rate and quality. We use two MPEG-4 test sequences, “coastguard” and “akiyo”. They are QCIF large (176×144 pixels) and consist of 300 frames. They are coded at 30 frames-per-second (30 fps) with GOV structure of one I-VOP and 14 P-VOPs. No B-VOP is used avoiding inherent coding delay. We choose the quantizer scale from 1 to 31 to investigate the effect of coding parameter on the coded video. We employed both sequences in all experiments. However, due to the space limitation, we only show results of “coastguard”, but the results with “akiyo” are consistent with those shown in this paper.

In Fig. 3, we depict the relationship among quantizer scale and average video rate of BL, EL and BL+EL. The video quality variation in terms of SNR (Signal to Noise Ratio) against the quantizer scale is shown in Fig. 4. It is shown that the BL rate decreases as the quantizer scale increases, and the video quality also decreases. The quality degradation is supplemented by the EL data whose total amount grows as the minimum video quality attained by the BL data deteriorates. In a case of the smallest quantizer scale, i.e., 1, no EL data is generated and the video quality is the highest because no DCT coefficient is quantized.

By observing variations of average rate and video quality of BL+EL data, we find that, as the quantizer scale becomes large, more bandwidth is required to obtain the same video quality as the lower quantizer scale. Figure 5 shows the relationship among the

Fig. 4. Relationship among quantizer scale and video quality

Fig. 5. Relationship between average rate and video quality

average video rate and the average video quality for several quantizer scales Q . Each line corresponds to a quantizer scale and the leftmost point of the line stands for the case of decoding only the BL data, while the rightmost point does the case of combining with all EL data. Thus, each line shows a range of possible video rate with the quantizer scale. When we plot some additional points on lines, they fluctuate a little but the relationship among lines still hold. If the desired data sending rate is below the minimum possible rate, the sender should buffer and smooth the data. On the other hand, the sender cannot satisfy the desired rate when it is beyond the maximum possible rate. The figure also shows that the smaller quantizer scale leads to the higher quality video presentation when the average rate is the same. Thus, we can expect the high-quality video transfer with the limited bandwidth when we appropriately choose the quantizer scale as small as possible.

However, a video stream coded with small quantizer scale intending to achieve high quality has limited capability of rate adjustment, which can be seen in Fig. 5 as a shorter line. For example, the video stream with quantizer scale $Q = 3$ only meets the bandwidth from 778 Kbps to 1,624 Kbps while the possible rate with $Q = 31$ ranges from 43 Kbps to 2,252 Kbps. If the coded video rate exceeds the target rate determined by TCP-friendly rate control mechanism, a video sender must regulate sending rate by buffering some part of video data making a sacrifice of smoothness of video presentation and interactivity of video application. Moreover, even if the target rate is in the range of possible rate, one must be prepared for the serious degradation of perceived video quality. Since the Internet is a best-effort network and no end-to-end QoS provisioning can be expected, packet losses cannot be avoided in video transfer especially when the video application employs UDP shunning delay introduced by retransmission. As shown in Fig. 3, a proportion of BL in entire video data increases as the quantizer scale decreases. Consequently, a possibility that BL data will be lost becomes higher and, as a result, the perceived quality of decoded video at a receiver considerably deteriorates.

Thus, an appropriate strategy might be to have a quantizer scale as large as possible, considering its error resilience and capability of rate adjustment. However, even if all

data are successfully transmitted and received by a receiver without error and loss, the perceived video quality is smaller than that of video data with smaller quantizer scale.

In the following sections, we investigate how we should employ the FGS coding algorithm and adjust video rate when TCP-friendly rate control mechanism is applied to video application. Some preliminary results useful for investigating appropriate quantizer scale selection algorithm are also shown.

3 FGS Video Transfer with TCP-friendly Rate Control

In this section, we first briefly introduce TFRC (TCP-Friendly Rate Control) [9,10], which accomplishes fair-share of bandwidth among TCP and non-TCP connections. Then, we propose several rate control methods to adjust FGS video rate to the desired sending rate determined by TFRC. Through simulations, we evaluate the effectiveness and practicality of our proposed methods.

3.1 TCP-friendly Rate Control

TFRC is the rate regulation algorithm to have an non-TCP connection behave similarly to, but more stable than a TCP connection which traverses the same path. It means that a TFRC connection reacts to network condition, typically congestion indicated by packet losses. For this purpose, a TFRC sender estimates the network condition by exchanging control packets between end systems to collect feedback informations. The sender transmits one or more control packets in 1 RTT. On receiving the control packet the receiver returns a feedback information required for calculating RTT and packet loss probability p . The sender then derives the estimated throughput of a TCP connection which competes for bandwidth on the path that the TFRC connection traverses. The estimated TCP throughput r_{TCP} is given as:

$$r_{TCP} \approx \frac{MTU}{RTT \sqrt{\frac{2p}{3}} + T_0 (3 \sqrt{\frac{3p}{8}}) p (1 + 32p^2)}$$

where T_0 stands for retransmission timeout [3]. Finally, the TFRC sender adjusts its data sending rate to the estimated TCP throughput r_{TCP} by means of, for example, video quality regulation. From now on, we call the estimated TCP throughput r_{TCP} , which determines the target rate of the application-level rate regulation, as “TFRC rate”.

3.2 FGS Video Transfer on TFRC Connection

If an application successfully adjusts its sending rate to the TFRC rate, TCP-friendly data transfer can be accomplished. However, TFRC itself does not consider the influence of the TCP-friendly rate control on the application-level performance. For example, the TFRC sender changes its sending rate at least once a RTT. Such a frequent rate control obviously affects the perceived video quality when a video application regulates amount of coded video data by controlling video quality according to the target rate. Thus, to accomplish TCP-friendly video transfer with consideration of the application-level performance, i.e., video quality, we should consider the following issues.

Fig. 6. Variants of video rate adjustment

1. control interval

The FGS video rate can be regulated by discarding a portion of the EL data. In this paper, considering the FGS video structure shown in Fig. 2, we propose VOP-based method (V method) and GOV-based method (G method). In the case of the V method, the target rate V_i of a VOP_{*i*} is defined as the TFRC rate at the beginning of VOP_{*i*}. Analogously, the target rate G_j of a GOV_{*j*} is defined as the TFRC rate at the beginning of GOV_{*j*} in the G-method. Those are illustrated in Fig. 6 where (a) corresponds to the V method whereas (b) and (c) show the G method case. In Figs. 7 and 8, which correspond to the V and G method respectively, we also show variations of the target rate derived from trace data of simulated TFRC connections.

2. video rate adjustment

Adjustment of the FGS video rate to the target rate is performed by discarding a portion of the EL data. There are the alternatives of rate adjustment methods, VOP-based and GOV-based. For the VOP-based adjustment, we further propose two methods, i.e., V-V and G-V methods. In the case of the V-V method (Fig. 6(a)), the target rate V_i of the VOP_{*i*} is first determined by the V method from the TFRC rate, and the rate (or amount) of the additional EL data E_i is obtained by subtracting the BL data rate B_i from the target rate V_i . On the other hand, the G-V method (Fig. 6(b)) first determines the target rate G_j of a GOV_{*j*} by means of the G method, then applies the identical rate to all VOPs in the GOV ($V_i = G_j, \text{VOP}_i \in \text{GOV}_j$). Then, the video rate adjustment is performed VOP by VOP as $E_i = G_j - B_i$ for a VOP_{*i*} $\in$ GOV_{*j*} in the G-V method. Finally, in the GOV-based rate adjustment method, called as a G-G method (Fig. 6(c)), the video rate averaged over GOV_{*j*} satisfies the target rate G_j . The rate of the EL data added to each VOP in the GOV is given as $E_i = (NG_j - \sum_{\text{VOP}_k \in \text{GOV}_j} B_k)/N$ where N stands for the number of VOPs in a GOV and identical among all VOPs in the GOV. The G-G method is proposed to achieve the smooth variation of video quality by equalizing the amount of supplemental EL data among VOPs, but the instantaneous video rate may exceeds the target rate.

3. BL rate violation

Even if the quantizer scale is carefully determined considering the network condition, the BL rate occasionally exceeds the available bandwidth for the video application. Since the BL data are crucial for video decoding, they are always sent out but an excess is managed by reducing the EL rate of the following VOPs or GOVs. In

Table 1. FGS video rate control methods

	control interval	rate adjustment	excess canceler
V-V <i>early</i>	VOP-based	VOP-based	<i>early</i>
V-V <i>smooth</i>	VOP-based	VOP-based	<i>smooth</i>
G-V <i>early</i>	GOV-based	VOP-based	<i>early</i>
G-V <i>smooth</i>	GOV-based	VOP-based	<i>smooth</i>
G-G <i>smooth</i>	GOV-based	GOV-based	<i>smooth</i>

Fig. 7. VOP rate variation **Fig. 8.** GOV rate variation

the *smooth* method, the excess is divided and equally assigned to the rest of VOPs in the GOV, thus averaged rate over several VoPs matches the target rate. On the other hand, to cancel the excess as fast as possible, the *early* method assign much excess to a VOP right after the the mischievous VoP, thus only a few VOPs are affected.

In table 1, we summarize possible rate control methods obtained by combining above mentioned methods. We should note here that there is not the G-G *early* method because the amount of EL data added to each VOP in the GOV must be identical in the “G-G” method.

3.3 Simulation Results

In this section, we compare six control methods proposed in the preceding section through simulation experiments. In comparison, we also consider another method called MP4-TFRC. The MP4-TFRC control method adjusts video rate by changing the quantizer scale as in our previous works on MPEG-2 [7,8].

In Figs. 7 and 8, we show the variation of target rate V_i and G_j of five simultaneous video sessions with TFRC rate control. These figures are obtained by applying V and G methods (see item “control interval” in Sec. 3.2) to trace data of the TFRC connections generated by a network simulator ns-2 [16]. A simulated network consists of two nodes and one 10Mbps bottleneck link of 15 msec delay connecting them. Each node has

Fig. 9. Video rate variation (V-V vs. G-V)

Fig. 10. Video quality variation (V-V vs. G-V)

thirty end systems via 150Mbps access links of 5 msec delay. The end systems on one node behave as senders and the others are receivers. Ten TFRC connections, ten TCP connections and ten UDP connections compete for the bottleneck bandwidth. In the following experiments, frame rate of coded video is 30 fps and the number of pictures in a GOV is 30. Figures shown in this section correspond to one of five video sessions for the sake of readability.

In Figs. 9 and 10, we show simulation results of the V-V and G-V methods when the FGS video data are generated from the test sequence “coastguard” by employing the quantizer scale of 2. The quantizer scale is determined to keep the BL rate below the minimum target rate during the session (TFRC session 2 in Figs. 7 and 8) in order to see the ideal performance of the video rate adjustment. Thus, the excess canceler is irrelevant. In those figures, “BL” and “BL+EL” correspond to the result of transmitting and decoding the BL data only and the BL with entire EL data, respectively.

In both methods, rate controls are successful and the FGS video rate follows the target. Although the target rate of each VOP differs among methods, the video rates averaged over longer interval, e.g., the duration of the session, are almost the same and the TCP-friendly video transfer are performed. The V-V and G-V methods differ in the control interval. The former adjusts the FGS rate VOP by VOP and the latter does GOV by GOV. As a result, the video rate of the G-V method changes much from GOV to GOV whereas that of the V-V method gradually increases or decreases according to the TFRC rate variation. The affect of rate variation can be seen in the video quality in Fig. 10.

The video quality in terms of SNR differs among GOVs in the G-V method, but is more stable than the V-V method regarding the difference among VOPs which belong to the same GOV. The reason that we find wedge-shaped and periodical quality degradation in the figure comes from the VOP-based rate adjustment. In the VOP-based adjustment, the resultant FGS rate becomes equal to the target rate by adding EL data $E_i = V_i - B_i$ or $E_i = G_j - B_i$. This means that the amount of supplemental EL data differs among VOPs even if we employ the G method where the target rates of VOPs in a GOV are identical. As a result, I-VOPs whose BL size is larger have less EL data than the other types of VOPs and experience lower quality. The interval between two wedges in the video quality variation corresponds to the distance between two successive I-VOPs.

Fig. 11. Video rate (V-V *early* vs. *smooth*)

Fig. 12. Video quality (V-V *early* vs. *smooth*)

Fig. 13. Video rate (V-V *smooth* vs. G-G *smooth*)

Fig. 14. Video quality (V-V *smooth* vs. G-G *smooth*)

From these observations, we can conclude that the V-V method achieves more preferable results than the G-V method in regarding variations of the FGS video rate and the video quality.

Next, we compare the excess cancelers of the V-V method. In Figs. 11 and 12, simulation results of the V-V *early* and V-V *smooth* methods on the TFRC session 4 are shown. In these figures, we employ the quantizer scale of 3 to make the BL rate higher than the target rate on relatively low TFRC rate. As a result, the BL rate in I-VOPs (VOPs 210, 225, 240, 255, 270 and 285 in the example) exceeds the target rate V_i . In such cases, the V-V *early* method reduces the amount of EL data added to a few VOPs right after the I-VOP and faces serious but instantaneous quality degradation. On the other hand, the V-V *smooth* method which fairly share excess among the rest of VOPs in the GOV, the degree of quality degradation is almost the same among VOPs in the GOV. However, the FGS video rate stays lower than the target rate during the GOV in the V-V *smooth* method whereas the V-V *early* method soon recovers from the rate decline. In addition, the V-V *early* method achieves higher video quality than the V-V *smooth* method in most of the time.

Fig. 15. Video rate (G-G *smooth* vs. MP4-TFRC)

Fig. 16. Video quality (G-G *smooth* vs. MP4-TFRC)

Fig. 17. Video rate under lossy condition

Fig. 18. Video quality under lossy condition

In Figs. 13 and 14, we compare the V-V *smooth* and G-G *smooth* methods to investigate the effect of video rate adjustment on the TFRC session 5. The G-G *smooth* method equalizes the amount of EL data among VOPs in a GOV to achieve the stable video quality. The FGS video rate follows the target rate in the V-V *smooth* method where the amount of EL data to add is determined VOP by VOP. The step-wise variation of the VoP-based target rate is due to the TFRC rate variation. On the other hand, the variation of FGS video rate in the G-G *smooth* method resembles that of BL rate because the identical amount of EL data is added to each VOP. In addition, the variation of video quality in the G-G *smooth* method is more gradual than that of the V-V *smooth* method. However, the instantaneous video rate of the G-G *smooth* method is not necessarily TCP-friendly and may introduce the smoothing delay required to make the data sending rate TCP-friendly. Furthermore, without an appropriate estimator of BL rate variation, the G-G *smooth* method introduces one GOV-time delay because all of the BL rate B_i of VOPs in the GOV must be known in advance to determine the amount of EL data to add to each VOP. Thus, the G-G *smooth* method is preferable when the video application emphasizes video quality while the V-V *smooth* is faithful to the TFRC rate.

To see superiority of FGS algorithm to DCT-based MPEG-4, we compare the G-G *smooth* method with the quantizer scale-based rate control method, i.e., MP4-TFRC. The MP4-TFRC is similar to the G-G *smooth* method except that the video rate is regulated by choosing an appropriate quantizer scale to fit the video rate to the target rate G_i according to the relationship among quantizer scale and resultant BL rate (Fig. 3). Results on the TFRC session 1 are shown in Figs. 15 and 16. Figure 15 indicates that the capability of rate control of the MP4-TFRC is poor and TCP-friendly video transfer cannot be expected. This is because that the quantizer scale-based rate control is coarse and there is often no appropriate quantizer scale with which the resultant video rate matches the target rate. Even if the quantizer scale is appropriately chosen, the resultant video rate has a highly bursty nature because of the GOV-based rate adjustment. Moreover, the coarse control leads to the sudden and drastic quality variation as shown in Fig. 16 which is triggered by increasing or decreasing the quantizer scale.

So far, we do not take into account packet loss to see the ideal performance of the TCP-friendly MPEG-4 video transfer. As mentioned in Sec. 2, packet loss affects the video quality. Figures 17 and 18 compares the variation of video traffic sent from the sender on the TFRC session 5 and video quality affected by packet loss for various quantizer scales, 3 and 6. Video data are segmented into packets of 1 KBytes long and packets are randomly discarded at a 10^{-3} probability. As shown in those figures, the video rate does not differ much among the quantizer scale because we apply the rate control method G-G *smooth* to the video data. The video quality is higher when the quantizer scale is smaller as long as there is no packet loss. However, once one or more packets are lost, the video quality considerably deteriorate in the video with a smaller quantizer scale because the proportion of BL data to the entire video traffic is large. Thus, we should carefully determine the quantizer scale taking account of not only the available bandwidth but the packet loss probability.

4 Conclusion

In this paper, we proposed and evaluated several TCP-friendly FGS video transfer. Through simulation experiments, we showed that the G-G *smooth* method, which determines the target rate every GOV and adds the identical amount of EL data to each VOP in the GOV, is preferable for video rate control in order to achieve the high and stable video quality.

However, there still remains some research issues. When the video application employs TFRC as the transport protocol, the video data injected into the transport layer are smoothed to fit to the TFRC rate, but such a smoothing delay is not considered in this paper. We showed only preliminary results on evaluation of influence of packet loss on the video quality. As mentioned in the previous section, the quantizer scale should be determined based on the network condition. We are currently considering a quantizer scale selection algorithm with which the video sender dynamically changes the coding parameter.

Acknowledgments. This work was partly supported by Special Coordination Funds for promoting Science and Technology and a Grant-in-Aid for Encouragement of Young

Scientists 12750322 from the Ministry of Education, Culture, Sports, Science and Technology, Japan, Research for the Future Program of Japan Society for the Promotion of Science under the Project “Integrated Network Architecture for Advanced Multimedia Application Systems”, and Telecommunication Advancement Organization of Japan under the Project “Global Experimental Networks for Information Society Project,”

References

1. Mathis, M., Semke, J., Mahdavi, J., Ott, T.: The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM SIGCOMM Computer Communication Review* **27** (1997) 67–82
2. Bolot, J.C., Turletti, T.: Experience with control mechanisms for packet video in the Internet. *ACM SIGCOMM Computer Communication Review* **28** (1998) 4–15
3. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP throughput: A simple model and its empirical validation. In: *Proceedings of ACM SIGCOMM’98*. Volume 28. (1998) 303–314
4. Rejaie, R., Handley, M., Estrin, D.: RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet. In: *Proceedings of IEEE INFOCOM’99*. (1999)
5. Padhye, J., Kurose, J., Towsley, D., Koodli, R.: A model based TCP-friendly rate control protocol. In: *Proceedings of NOSSDAV’99*. (1999)
6. Bansal, D., Balakrishnan, H.: TCP-friendly congestion control for real-time streaming applications. MIT Technical Report MIT-LCS-TR-806 (2000)
7. Wakamiya, N., Murata, M., Miyahara, H.: On TCP-friendly video transfer with consideration on application-level QoS. In: *Proceedings of IEEE ICME 2000*. (2000)
8. Miyabayashi, M., Wakamiya, N., Murata, M., Miyahara, H.: MPEG-TFRCP: Video transfer with TCP-friendly rate control protocol. In: *Proceedings of IEEE ICC 2001*. (2001) 137–141
9. Widmer, J.: Equation-based congestion control. Diploma Thesis, University of Mannheim (2000)
10. Floyd, S., Handley, M., Padhye, J., Widmer, J.: Equation-based congestion control for unicast applications: the extended version. Technical Report TR-00-003, International Computer Science Institute (2000)
11. Radha, H., Chen, Y.: Fine-granular-scalable video for packet networks. In: *Proceedings of Packet Video’99*. (1999)
12. van der Schaar, M., Radha, H., Dufour, C.: Scalable MPEG-4 video coding with graceful packet-loss resilience over bandwidth-varying networks. In: *Proceedings of IEEE ICME 2000*. (2000)
13. Radha, H., van der Schaar, M., Chen, Y.: The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP. *IEEE Transactions on Multimedia* **3** (2001) 53–68
14. Fukuda, K., Wakamiya, N., Murata, M., Miyahara, H.: QoS mapping between user’s preference and bandwidth control for video transport. In: *Proceedings of IFIP IWQoS’97*. (1997) 291–302
15. Fukuda, K., Wakamiya, N., Murata, M., Miyahara, H.: Real-time video multicast with hybrid hierarchical video coding in heterogeneous network and client environments. *Proceedings of IFIP/IEEE MMNS’98* (1998)
16. The VINT Project: UCB/LBNL/VINT network simulator - ns (version 2) (1996) available at <http://www.isi.edu/nsnam/ns/>.