TCP データ通信との公平性を考慮した MPEG-4 動画像通信のための品質調整機構

宮林 正樹　若宮 直紀　村田 正幸　宮原 秀夫

大阪大学 大学院基礎工学研究科 情報数理系専攻
〒 560–8531 大阪府豊中市待兼山町 1–3
Tel: 06–6850–6588　Fax: 06–6850–6589
E-mail: miyabays@ics.es.osaka–u.ac.jp

あらまし　通信品質保証のないインターネットにおける分散型マルチメディアアプリケーションの普及は様々な問題を引き起こす．特に，UDP を用いた大量のマルチメディアトラヒックが流入することにより，TCP によるデータ通信の性能が著しく劣化し，プロトコル間の性能格差が生じることが指摘されている．そこで，プロトコル間の公平性を考慮した，動画像通信などのマルチメディアアプリケーションのための輻輳適応型レート制御の検討が行われている．

本稿では，TCP データ通信と公平に帯域を共有しつつ，ネットワーク状態の変動に応じて動的に品質調整を行うことで高品質かつ安定した MPEG-4 動画像通信を実現するための手法について検討している．パケット棄却率に応じて FEC (Forward Error Correction) 冗長度を動的に調整する制御手法を提案し，実測データに基づくシミュレーションによりその有効性を示している．

キーワード　動画像通信，FEC，MPEG-4，Fine Granular Scalability，TFRC

Dynamic Quality Adaptation Mechanisms for TCP-friendly MPEG-4 Video Transfer

Masaki MIYABAYASHI　Naoki WAKAMIYA　Masayuki MURATA　Hideo MIYAHARA

Department of Informatics and Mathematical Science
Graduate School of Engineering Science, Osaka University
1–3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
Tel: 06–6850–6588　Fax: 06–6850–6589
E-mail: miyabays@ics.es.osaka–u.ac.jp

Abstract　When a considerable amount of UDP traffic is injected into the Internet by distributed multimedia applications, it is easily driven congested. Consequently, bandwidth available to TCP connections is oppressed and their performance significantly deteriorates. In order that both multimedia applications and TCP-based ones fairly co-exist in the Internet, it becomes increasingly important to consider the inter-protocol fairness.

In this paper, to take into account the quality degradation caused by packet loss in video quality adaptation, we propose the video quality adjustment mechanisms which accomplish the high-quality, stable, and TCP-friendly video transfer under a lossy environment in cooperation with the FEC (Forward Error Correction) technique. Our mechanism adjusts the video quality in accordance with the TFRC (TCP-Friendly Rate Control) rate, the packet loss probability, and the video quality. Through simulation experiments, we show that our proposed method can provide high-quality, stable and TCP-friendly video transfer even in the unstable and lossy Internet.

key words　video transfer, FEC, MPEG-4, Fine Granular Scalability, TFRC

1 Introduction

It has been pointed out that selfish UDP traffic injected by multimedia application easily dominates network bandwidth and drives the network into congestion. As a result, the 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 necessary to consider the fairness among sessions. 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]. 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 [5] 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 work [8], we have focused on MPEG-4 video systems which have highly efficient coding algorithms and error resilience capabilities for TCP-friendly video transmission over the Internet, which consists of variety of networks such as PSTN, ISDN, ADSL, CATV, FTTH, wireless networks, and optical fiber networks. In consideration of TCP-friendly MPEG-4 video transfer with a flexible rate control, we have employed the Fine Granular Scalability (FGS) [9-12] as a video coding algorithm to accomplish scalable rate adjustment, and the TCP-Friendly Rate Control (TFRC) [5] as an underlying rate control protocol.

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 TFRC rate. Thus, we have tackled some issues including appropriate control interval, video quality regulation, and video rate adaptation. We have proposed some rate control methods to accomplish TCP-friendly video transfer with consideration of the application-level performance. Through simulation experiments, it has shown that high-quality, stable and TCP-friendly video transfer can be accomplished by regulating video rate at an interval of group of pictures. However, we have not taken into account packet loss within the network so that we could investigate the ideal performance of the TCP-friendly MPEG-4 video transfer.

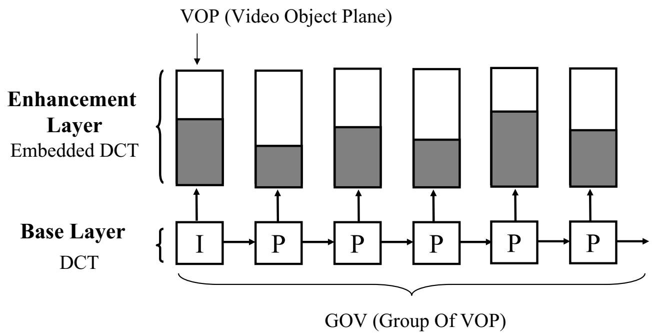

Figure 1: An example of FGS video structure

In this paper, to take into account the quality degradation caused by packet loss in video quality adaptation, we propose the video quality adjustment mechanisms which accomplish the high-quality and stable video transfer under a lossy environment in cooperation with the FEC (Forward Error Correction) technique. Our mechanism adjusts the video quality in accordance with the TFRC rate, the packet loss probability, and the resultant video quality. Through simulation experiments, we show that our proposed method can provide high-quality, stable and TCP-friendly video transfer even in the unstable and lossy Internet.

The paper is organized as follows. In Section 2, we briefly introduce the FGS video coding algorithm, the TFRC mechanism, and our "G-G *smooth*" method proposed in [8]. In Section 3, we investigate the influence on the perceived video quality by packet loss and shortly explain FEC technique which enables high- and stable-quality video transmission by protecting the video data against packet loss. Then, we propose a dynamic quality adaptation mechanism for a lossy Internet environment. We evaluate our mechanism in Section 4. Finally, we summarize our paper and outline our future work in Section 5.

2 FGS Video Transfer with TCP-friendly Rate Control

In this section, we briefly introduce (1) the FGS (Fine Granular Scalability) algorithm [9-12] which is excellent in adaptation to the bandwidth variation among MPEG-4 video-coding standards, (2) TFRC (TCP-Friendly Rate Control) [5], which accomplishes fair-share of bandwidth among TCP and non-TCP connections, and (3) our FGS video rate control method proposed in [8].

2.1 FGS Video Coding Algorithm

To cope with the TCP-friendly rate control mechanism, video applications should adjust video traffic rate to the desired value 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.

In this paper, expecting higher flexible and scalable rate adjustment capability, we employ a Fine Granular Scalability (FGS) video coding algorithm [9-12] that is a compression method suitable for video streaming applications and is being introduced into MPEG-4 standards. Figure 1 illustrates a basic structure of the FGS video stream. A FGS video stream is composed of a sequence of VOP (Video

Object Plane), which is the basic unit of image data and is equivalent to the frame or picture of MPEG-1 and MPEG-2. A sequence of VOPs beginning from an I-VOP is called GOV (Group Of VOP) and defined by two parameters; the number of P-VOPs between two I-VOPs and the number of B-VOPs between two I or P-VOPs. FGS is also categorized into a layered coding algorithm. 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 it provides minimum picture 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 (quantization scale, etc.) and the amount of supplemental EL data added. Even if only little EL data is used in decoding a VOP, the perceived video quality is improved. Thus, it is effective to send as much EL data as possible in addition to the BL data as far as the video sending rate is higher than BL rate. Losses of the BL data have a significant influence on perceived video quality because BL is indispensable for decoding VOP. Compression efficiency of the EL data is not high since EL is coded without motion compensation technique. However, the EL data have the outstanding error tolerance because of the scalable coding algorithm and the locality of error propagation because loss of EL data only affects its VOP.

2.2 TFRC: TCP-Friendly Rate Control

TFRC [5] is a mechanism to have a non-TCP connection behave similarly to, but more stable than a TCP connection which traverses the same path. For this purpose, a TFRC sender estimates the network condition by exchanging control packets between the sender and receiver to collect feedback informations. The sender transmits one or more control packets in one RTT. On receiving the control packet the receiver returns a feedback information required for calculating RTT and estimating the loss event rate p_{tfrc}. 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:

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

where T_0 stands for retransmission timeout [2]. Finally, an application on TFRC adjusts its data 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".

2.3 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

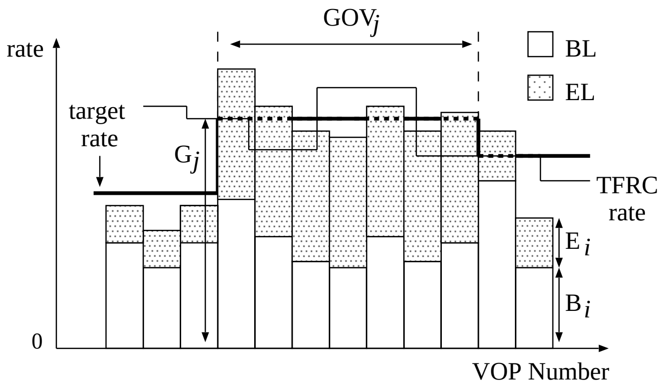

Figure 2: "G-G *smooth*" video rate adjustment method

the TCP-friendly rate control on the application-level performance/quality. 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 have considered the several issues such as control interval, video rate adjustment, and BL rate violation, in [8]. Then, we have proposed "G-G *smooth*" video rate adjustment method whose mechanism is described below.

1. **control interval**

   The interval that TFRC notifies the upper application of a new sending rate does not match the point that the application can change its data structure or amount. Considering the FGS video structure shown in Fig. 1, the target rate G_j of GOV$_j$ is defined as the TFRC rate at the beginning of GOV$_j$ as illustrated in Fig. 2.

2. **video rate adjustment**

   Adjustment of the FGS video rate to the target rate is performed by adding a portion of the EL data to the BL data. In the G-G *smooth* method, the video rate averaged over GOV$_j$ satisfies the target rate G_j. The rate E_i of the EL data added to each VOP$_i$ in the GOV is given as $E_i = (NG_i - \sum_{VOP_k \in GOV_j} B_k)/N$ where N and B_k stands for the number of VOPs in a GOV and the BL data rate, respectively. It is intended to achieve the smooth variation of video quality by equalizing the amount of supplemental EL data among VOPs, while the video rate may instantaneously exceed the target rate.

3. **BL rate violation**

   In the case that the available bandwidth becomes smaller by the occurrence of network congestion, the BL rate occasionally exceeds the target rate. Since the BL data are crucial for video decoding, they are always sent out and an excess is managed by reducing the EL rate of the following VOPs or GOVs. 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.

3 TCP-friendly MPEG-4 Video Transfer under a Lossy Environment

In [8], we have demonstrated that G-G *smooth* method proposed for the video rate control provides a high- and stable-

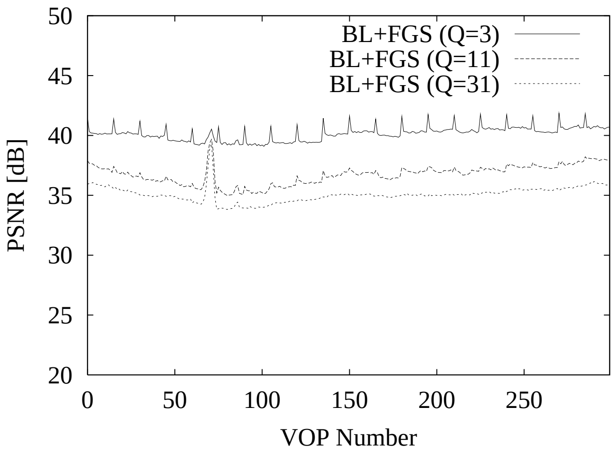

Figure 3: FGS video quality ($p = 10^{-4}$, rate = 1 Mbps)

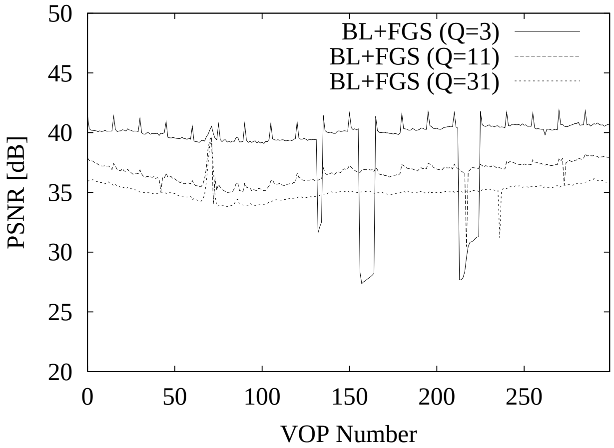

Figure 4: FGS video quality ($p = 10^{-3}$, rate = 1 Mbps)

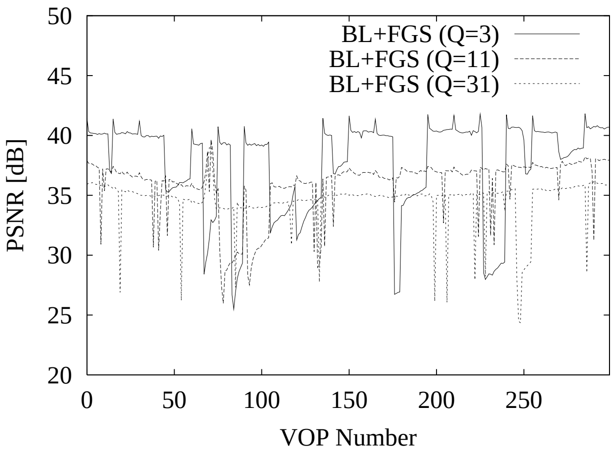

Figure 5: FGS video quality ($p = 10^{-2}$, rate = 1 Mbps)

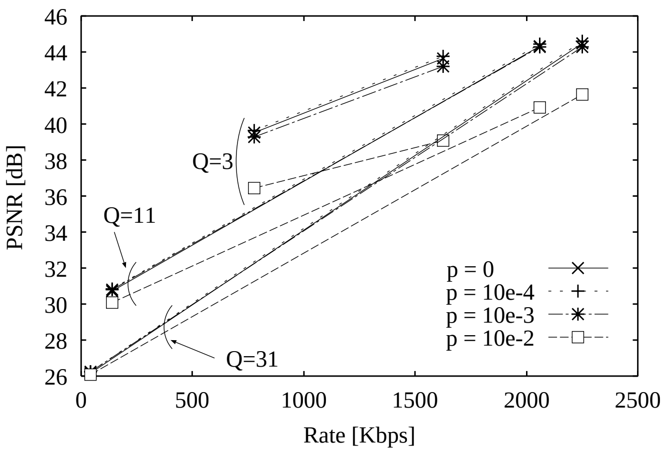

Figure 6: Relationship between average rate and video quality

quality video transfer on TFRC sessions. In those simulations, we have not taken into account packet loss within the network so that we could investigate the ideal performance of the TCP-friendly MPEG-4 video transfer. However, any kind of TCP-friendly rate control such as TFRC cannot avoid packet loss in nature.

In this section, we propose the video quality adjustment mechanisms which accomplish the high-quality video transfer under a lossy environment in cooperation with the FEC (Forward Error Correction). We first start by investigation of basic characteristics of our scheme described in the last section, i.e., G-G *smooth*, under a lossy environment. Then we propose new mechanisms and evaluate them.

3.1 Investigation of the Influence of Packet Loss on Video Quality

In Figs. 3 through 5, we show results of experiments for the case that a video sequence “coastguard” is coded at an average rate of 1 Mbps. “coastguard” is a QCIF-large video sequence and played back at 30 fps. These figures show the video quality variation when only successfully received video data are decoded. Each of these figures corresponds to the packet loss probability of 10^{-4}, 10^{-3}, and 10^{-2}, respectively. In each of these figures, results for three quantizer scales, i.e., 3, 11, and 31, are depicted. In the experiments, each packet of 1 Kbyte-length is examined and lost at the given packet loss probability.

For the packet loss probability of 10^{-4}, none of about 1,860 packets was lost and no quality degradation was observed. As the packet loss probability increases, a video stream with a smaller quantizer scale begins to be affected. Comparison among the figures indicates that the smaller the quantizer scale is, the higher the level of degradation is. The degradation of video quality is notable in the case of $Q = 3$ owing to the loss of BL data. In the cases of $Q = 11$ and 31, annoying spike-shaped degradations were observed. The subjective video qualities in terms of MOS were 3.95, 3.88, 2.63 for video streams with of $Q = 3$, 11, and 31, respectively, at 10^{-2} of packet loss probability. From the result of MOS (Mean Opinion Score) evaluation, we can see that users tend to feel uncomfortable by spike-shaped degradations, that are perceived as flickers. On the other hand, degradations caused by the loss of BL data, which last during a while, is rather acceptable. Therefore, we conclude that we need to protect both EL and BL data and it is better to employ as small a quantizer scale as possible.

We also conducted experiments for the other settings of the video rate. All results are summarized in Fig. 6 in terms of the average SNR, the average video rate, and the quantizer scale, for different level of packet loss. Despite tra-

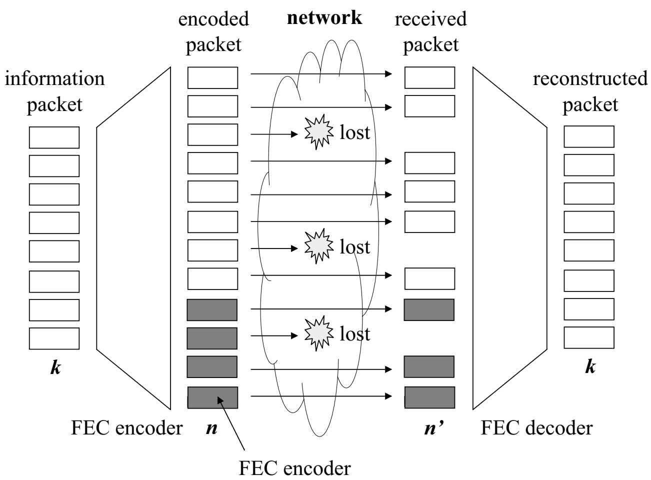

Figure 7: An example of FEC technique

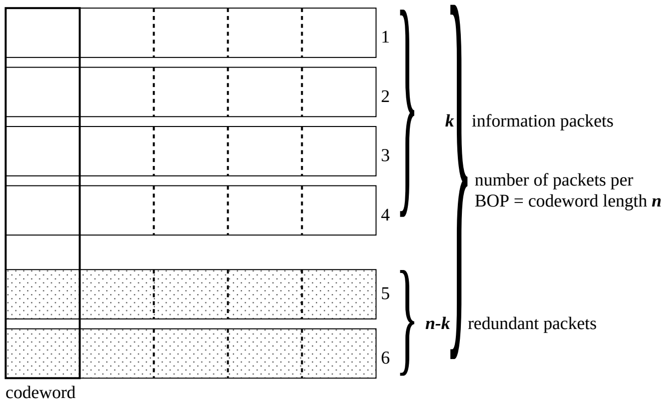

Figure 8: An example of BOP with $n = 6$, $k = 4$

jectories of quality variation depicted in the figures, average SNR is higher when we choose smaller quantizer scale regardless of the average video rate and the packet loss probability.

Consequently, we should send the video stream of the smallest quantizer scale within the range of the possible rate if we want to attain a higher quality video transfer with the limited bandwidth. However, the perceived video quality with such a smaller quantizer scale suddenly fluctuates under a lossy environment. Therefore, in order to accomplish a stable-quality video transfer, we should employ a mechanism to minimize the influence by packet loss on video quality. In the following subsection, we consider the effective control mechanisms using a FEC technique to protect the video quality against packet losses.

3.2 FEC: Forward Error Correction

In this subsection, we briefly introduce the FEC (Forward Error Correction) technique employed in [13, 14], which enables high- and stable-quality video transmission by protecting the video data against packet loss.

The idea of the Forward Error Correction (FEC) in a packet oriented video transmission scheme is to generate redundant packets at the sender, which can be used at the receiver to recover lost video data packets as illustrated in Fig. 7. As far as the number of lost packets is below $n - k$, k of information packets can be reconstructed from successfully received packets. We use Reed-Solomon (RS) codes [15] for forward error correction in this paper. RS codes are perfectly suitable for error protection against packet loss, because they are maximum distance separable codes, i.e., there are no other codes that can reconstruct erased symbols from a smaller number of received code symbols [16].

An RS code defined by RS (n, k) can be easily employed for protecting packets against loss if the n length codewords are formed across k source information and $n-k$ redundant packets as shown in Fig. 8 where we use $n = 6$ and $k = 4$. FEC coding is performed against each BOP (Block of Packet). A BOP consists of n parts, or blocks of packets as illustrated in Fig. 8. n is called the codeword length. Note that the scheme depicted in Fig. 8 introduces no additional delay at the sender. The sender has only to store copies of the information packets until k packets have been sent. Then the $n - k$ redundancy packets are generated and transmitted just after the last information packet. RS (n, k) can generally correct $f = \lfloor \frac{n-k}{2} \rfloor$ symbol errors. With the knowledge of the error position, it can correct up to $f = n - k$ symbol errors, that is, the information packets can be reconstructed from any subset of k correctly received packets using erasure decoding [17]. It is applicable to our case because each packet contains its packet number, and therefore the exact positions of lost packets in BOP are known at the receiver. As soon as any k packets of a BOP have been received, all lost information packets can be reconstructed. Thus, our FEC scheme requires a receiver buffer which can at least hold k packets. However, the receiver does not need to defer play-back as far as there is no packet loss. Even when one or more packets are lost, they are recovered as soon as the receiver obtains k packets.

To evaluate the effectiveness of an RS (n, k) code under a lossy environment, we need to know the probability that more than $n - k$ packets are lost in a certain network condition. We can compute this probability if we know the probability $P(m, n)$ that m out of n packets are lost. $P(m, n)$ is called the block error density function. It is a simple binomial distribution in the case of a memoryless channel with given packet loss probability as follows,

$$P(m, n) = \binom{n}{m} P_{link}^{m} (1 - P_{link})^{n-m} \qquad (2)$$

where P_{link} is the actual packet loss probability observed in the network. We can derive the probability P_{video} that the video packet is lost by using the following equation.

$$\sum_{m=n-k+1}^{n} P(m, n) = 1 - (1 - P_{video})^{k}, \qquad (3)$$

that is,

$$P_{video} = 1 - \sqrt[k]{1 - \sum_{m=n-k+1}^{n} P(m, n)}. \qquad (4)$$

3.3 Dynamic Quality Adaptation Mechanism with Packet-Loss Protection for TCP-friendly FGS Video Transfer

In accordance with the considerations given in the previous subsection, we propose a dynamic quality adjustment mechanism which obtains the video transfer of stable quality.

Our mechanism explicitly reacts against packet loss and employs the FEC technique to protect video data from packet loss. To accomplish the efficient control, we have to consider how much redundant packets should be added to the video packets under a lossy environment. Specifically, introducing redundancy decreases the bandwidth that the server can use to transfer video data. When the system faces to a congestion, the target rate given by the TFRC algorithm is decreased. At the same time, the sender should increase the number of the redundant packets to achieve a higher level of protection against the higher loss probability. Consequently, the bandwidth left for the video data become suppressed very much and as a result, the video quality is expected to deteriorate.

Tables 1, 2, and 3 provides results of evaluations on effects of the FEC protection under lossy environments. The video stream "coastguard" is coded at the average rate of 1 Mbps. Each table corresponds to different quantizer scales, i.e., 3, 11, and 31. The setting of P_{ave}, the average of actual packet loss observed on the path, is shown in the leftmost columns with a corresponding average SNR which is obtained when no protection is applied. The other four columns list the average video quality in terms of SNR when FEC is employed to achieve the target probability P_{target} under a lossy condition of corresponding P_{ave}. For example, the top and leftmost value of Table 1 indicates that the video quality becomes 36.27 dB when the video stream is coded with a quantizer scale of 3. To improve the packet loss probability observed at the client from 10^{-1} to 10^{-2}, 112 Kbps out of 1 Mbps is devoted to the redundant packets and video traffic is adjusted to average rate of 888 Kbps. Here the rate dedicated to FEC redundant traffic, r_{FEC}, is given as $G_j \times f/n$ where G_j, f, and n are the TCP-friendly sending rate by Eq. (1) at the beginning of GOV$_j$, the number of redundant packets, and the number of sending packets within the GOV$_j$, The number of redundant packets f required to achieve the target packet loss probability P_{target} under a lossy condition of the packet loss probability P_{ave} is derived from the following equation, respectively.

$$f = \min_f \left\{ \sum_{i=f+1}^{n} \binom{n}{i} P_{ave}^{i} (1 - P_{ave})^{n-i} \le 1 - (1 - P_{target})^{n-f} \right\} \quad (5)$$

Observations on each row indicate that there is a tradeoff between the level of protection and the video quality. In the congested environment, i.e., $P_{ave} = 10^{-1}$ and 10^{-2}, introducing redundancy improves the video quality to some extent. However, beyond the target probability of 10^{-3} in most of the cases, the video quality deteriorates since overhead becomes evident. Especially when the network is lightly loaded, i.e., $P_{ave} = 10^{-4}$, redundancy only disturbs the video transfer.

Table 1: Averaged PSNR ($Q = 3$, Sending rate = 1 Mbps)

	P_{target}			
P_{ave}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
10^{-1} (25.46)	36.27	39.02	39.59	39.64
10^{-2} (36.74)		40.01	39.98	39.93
10^{-3} (39.81)			39.98	40.08
10^{-4} (40.22)				39.97

Table 2: Averaged PSNR ($Q = 11$, Sending rate = 1 Mbps)

	P_{target}			
P_{ave}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
10^{-1} (26.15)	34.53	35.20	35.13	34.91
10^{-2} (35.50)		36.32	36.28	36.26
10^{-3} (36.77)			36.67	36.53
10^{-4} (36.80)				36.68

Table 3: Averaged PSNR ($Q = 31$, Sending rate = 1 Mbps)

	P_{target}			
P_{ave}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
10^{-1} (26.83)	33.71	34.10	33.91	33.74
10^{-2} (33.45)		34.80	34.83	34.73
10^{-3} (34.81)			35.05	34.94
10^{-4} (35.14)				35.05

A new dynamic quality adaptation mechanism in cooperation with FEC becomes as follows.

1. Determine the TCP-friendly sending rate G_j by Eq. (1) at the beginning of GOV$_j$, based on the feedback information (RTT, p_{tfrc}) obtained by the TFRC algorithm.
2. Derive the probability of the actual packet loss, P_{j-1}, by dividing the number of lost packets by that of sent packets in the preceding GOV time. Information required for the derivation is obtained through a feedback mechanism of TFRC or a dedicated mechanism.
3. Calculate the smoothed packet loss probability P_{ave} by applying the exponential moving average employed in TFRC. That is,

$$P_{ave} = \frac{\sum_{i=1}^{8} w_i P_{j-i}}{\sum_{i=1}^{8} w_i}. \quad (6)$$

   Weighting parameter w_i is determined as the following equations

$$w_i = \begin{cases} 1, & 1 \le i \le 4 \\ \dfrac{9-i}{5}, & 4 < i \le 8 \end{cases}$$

4. Determine the redundant rate r_{FEC} to achieve the target loss probability P_{target} from P_{ave} and G_j.
5. Determine the sending rate R_{video} allocated to the video data by subtracting the FEC redundant rate r_{FEC} from the actual sending rate G_j.
6. Apply the G-G *smooth* mechanism to the determined video rate R_{video}.
7. Generate redundant packets at the redundant rate r_{FEC} and send them in addition to video packets.

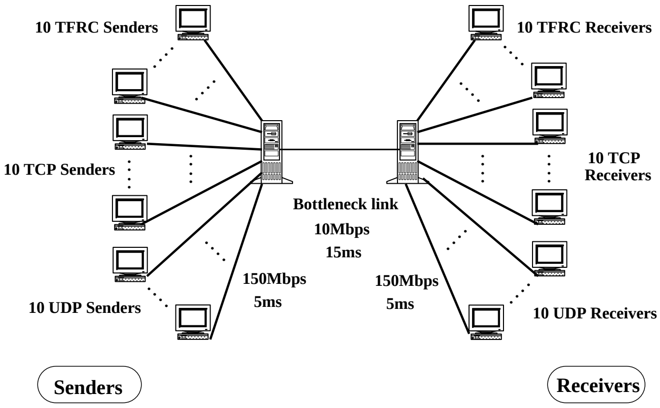

Figure 9: Simulation network model

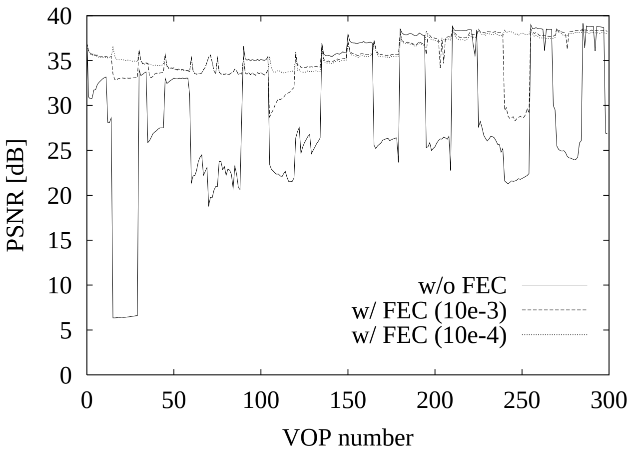

Figure 11: Video quality variation (w/ FEC vs. w/o FEC)

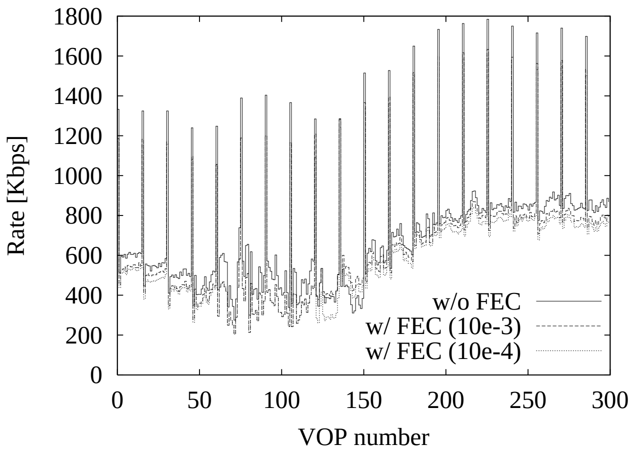

Figure 10: Video rate variation (w/ FEC vs. w/o FEC)

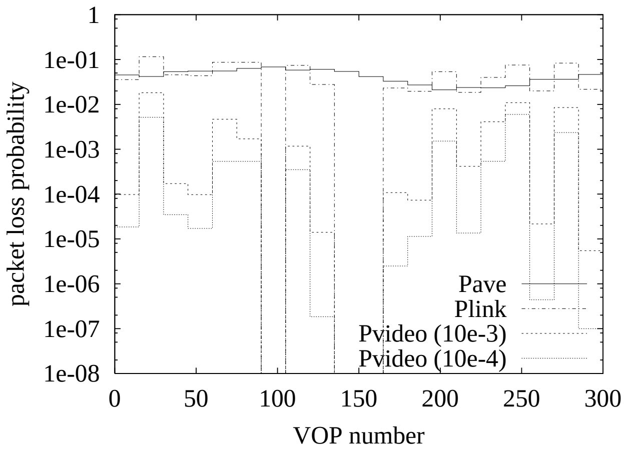

Figure 12: Loss rate variation (w/ FEC vs. w/o FEC)

4 Simulation Results

In this section, we evaluate the effectiveness of our dynamic quality adaptation mechanism proposed in the previous section through simulation experiments. Our mechanism is based on the "G-G *smooth*" method, which provided a high- and stable-quality video transfer on TFRC sessions as demonstrated in our previous work [8]. As shown in Fig. 9, a simulated network consists of two nodes and one 10 Mbps bottleneck link of 15 msec delay connecting them. Each node has thirty end systems via 150 Mbps 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, the frame rate of coded video is 30 fps and the number of pictures in GOV is 30.

In Figs. 10 through 12, we show simulation results of the method with the FEC technique and without it, respectively. In the case of the original G-G *smooth* method, the quantizer scale of 5 is used during a session. On the other hand, the quantizer scale of 6 is applied in the case of the method with the FEC since the actual video rate is suppressed by FEC packets. Figure 10 shows the rate variation of the bandwidth allocated to the video data, R_{video}. Trajectories of perceived video quality under the lossy environment are shown in Fig. 11. Figure 12 depicts the variance of packet loss probabilities P_{video}, P_{link}, and P_{ave}. Through these figures, the target packet loss probability P_{target} is set to 10^{-3} and 10^{-4}.

We can see that the FEC technique is effective for protecting the video quality from degradation caused by packet loss. Although packets are lost at probability 10^{-2} to 10^{-1} in the network, it is suppressed toward 10^{-6} to 10^{-2} with a help of FEC. As a result, video quality becomes stable even under the lossy network environment where an original method suffers from much degradation of video quality. However, quality degradation due to loss of BL data is noticeable when FEC is targeting at the probability of 10^{-3}. This implies that the protection targeted at 10^{-3} is not effective enough against instantaneous but serious packet loss.

When we apply the target loss probability of 10^{-4}, the bandwidth available to the video data becomes obviously lower than that in the case of $P_{target} = 10^{-3}$ because of the increased FEC redundancy. As a result, the video quality becomes lower for loss-free GOVs. However, trajectories of quality are kept stable owing to a higher level of protection targeted at the video packet loss probability of 10^{-4}. The perceived video quality becomes much better than that in the case of using the method without the FEC technique. Examples of images displayed on the monitor are shown in Figs. 13 and 14 for the method without FEC and with FEC, respectively. Thus, on the basis of the above observations, we conclude that choosing the target packet loss probability

Figure 13: Displayed VOP 70 (w/o FEC)

Figure 14: Displayed VOP 70 (w/ FEC and $P_{target} = 10^{-4}$)

of 10^{-4} is effective to protect the video packet from loss at the sacrifice of only a slight quality degradation at most 1 dB in our experiments.

5 Concluding Remarks

In this paper, we have proposed the video quality adjustment mechanisms which accomplish the high-quality video transfer under the lossy environment. Our mechanism employs the FEC to protect video packets from loss and dynamically regulate video rate in accordance with the network condition.

Although the effectiveness of our mechanism is verified through simulation experiments, there still remains some research issues. When the video application employs TFRC, the video data injected into the transport layer should be smoothed to fit to the TFRC rate, but such a smoothing delay is not considered in this paper. Furthermore, we should consider the improved mechanisms to change the quantizer scale dynamically in accordance with the packet loss probability, in order to attain the real-time video transfer with high and stable video quality under both loss-free and lossy conditions.

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. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of ACM SIGCOMM'98*, vol. 28, pp. 303–314, September 1998.

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

[4] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," in *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video 1999 (NOSSDAV'99)*, June 1999.

[5] M. Handley, J. Padhye, S. Floyd, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification." Internet-Draft draft-ietf-tsvwg-tfrc-03.txt, work in progress, July 2001.

[6] M. Miyabayashi, N. Wakamiya, M. Murata, and H. Miyahara, "MPEG-TFRCP: Video transfer with TCP-friendly rate control protocol," in *Proceedings of IEEE International Conference on Communications 2001 (ICC2001)*, vol. 1, pp. 137–141, June 2001.

[7] N. Wakamiya, M. Murata, and H. Miyahara, "On TCP-friendly video transfer," in *Proceedings of SPIE International Symposium on Information Technologies 2000*, November 2000.

[8] N. Wakamiya, M. Miyabayashi, M. Murata, and H. Miyahara, "MPEG-4 video transfer with TCP-friendly rate control," in *Proceedings of 4th IFIP/IEEE International Conference on Management of Multimedia Networks and Services 2001 (MMNS2001)*, vol. 1, pp. 29–42, October–November 2001.

[9] H. Radha and Y. Chen, "Fine-granular-scalable video for packet networks," in *Proceedings of Packet Video'99*, April 1999.

[10] M. van der Schaar, H. Radha, and C. Dufour, "Scalable MPEG-4 video coding with graceful packet-loss resilience over bandwidth-varying networks," in *Proceedings of IEEE International Conference of Multimedia & EXPO (ICME2000)*, July 2000.

[11] Text of ISO/IEC 14496-2, "MPEG-4 Video FGS v.4.0." Proposed Draft Amendment (PDAM), Noordwijkerhout, the Netherlands, March 2000.

[12] H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Transactions on Multimedia*, vol. 3, pp. 53–68, March 2001.

[13] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Image Communication, Special Issue on Real-time Video over the Internet*, vol. 15, pp. 77–94, September 1999.

[14] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource allocation for multimedia streaming over the Internet," *IEEE Transactions on Multimedia*, vol. 3, pp. 339–355, September 2001.

[15] R. E. Blahut, *Digital Transmission of Information*. Addison-Wesley, 1990.

[16] R. E. Blahut, *Theory and Practice of Error Control Codes*. Addison-Wesley, 1983.

[17] A. J. McAuley, "Reliable broadband communication using a burst erasure correcting code," in *Proceedings of ACM SIGCOMM'90*, pp. 297–306, September 1990.