EURASIP Journal on Applied Signal Processing 2004:2, 280–289

Effective Quality-of-Service Renegotiating Schemes for Streaming Video

Hwangjun Song
School of Electrical Engineering, Hongik University, 72-1 Sangsu-dong, Mapo-gu, Seoul 121-791, Korea
Email: hwangjun@wow.hongik.ac.kr

Dai-Boong Lee
School of Electrical Engineering, Hongik University, 72-1 Sangsu-dong, Mapo-gu, Seoul 121-791, Korea
Email: neferian@hotmail.com

Received 13 November 2002; Revised 25 September 2003

Effective quality-of-service renegotiating schemes for streaming video is presented. The conventional network supporting quality of service generally allows a negotiation at a call setup. However, it is not efficient for the video application since the compressed video traffic is statistically nonstationary. Thus, we consider the network supporting quality-of-service renegotiations during the data transmission and study effective quality-of-service renegotiating schemes for streaming video. The token bucket model, whose parameters are token filling rate and token bucket size, is adopted for the video traffic model. The renegotiating time instants and the parameters are determined by analyzing the statistical information of compressed video traffic. In this paper, two renegotiating approaches, that is, fixed renegotiating interval case and variable renegotiating interval case, are examined. Finally, the experimental results are provided to show the performance of the proposed schemes.

Keywords and phrases: streaming video, quality-of-service, token bucket, renegotiation.

1. INTRODUCTION

In recent years, the demands and interests in networked video have been growing very fast. Various video applications are already available over the network, and the video data is expected to be one of the most significant components among the traffics over the network in the near future. However, it is not a simple problem to transmit video traffics efficiently through the network because the video requires a large amount of data compared to other multimedia. To reduce the amount of data, it is indispensable to employ effective video compression algorithms. So far, digital video coding techniques have advanced rapidly. International standards such as MPEG-1, MPEG-2 [1], MPEG-4 [2], H.261 [3], H.263/+/++ [4], H.26L, and H.264 have been established or are under development to accommodate different needs by ISO/IEC and ITU-T, respectively. The compressed video data is generally of variable bit rate due to the generic characteristics of entropy coder and scene change inconsistent motion change of the underlying video. Furthermore, video data is time constrained. These facts make the problem more challenging. By the way, constant bit rate video traffic can be generated by controlling the quantization parameters and it is much easier to handle over the network, but the quality of the decoded video may be seriously degraded.

In general, suitable communications between the network and the sender end can increase the network utilization and enhance video quality at the receiver end simultaneously [5]. Generally speaking, the variability of compressed video traffics consists of two components: short-term variability (or high-frequency variability) and long-term variability (or low-frequency variability). Buffering is only effective in reducing losses caused by variability in the high-frequency domain, and is not effective in handling variability in the low-frequency domain [6]. Recently, some QoS (quality-of-service) renegotiating approaches have been proposed to handle the nonstationary video traffics efficiently over the network [7, 8, 9, 10, 11, 12], while the conventional QoS providing network negotiates QoS parameters only once at a call setup. For example, RCBR (renegotiated constant bit rate) [7, 8] is a simple but quite effective approach to support the QoS renegotiations. RCBR network allows the sender to renegotiate the bandwidth during the data transmission. Actually, the bandwidth renegotiations can be interpreted as a compromise of ABR (available bit rate) and VBR (variable bit rate). Over network supporting bandwidth renegotiations, how to determine the renegotiation instants and the required bandwidth is studied in [9, 10, 11, 12, 13]. In [11], Zhang and Knightly proposed the RED-VBR (renegotiated deterministic variable bit rate) service model to support VBR video that uses a traffic model called D-BIND (deterministic bounding interval-length dependent). Salehi et al. proposed the shortest path algorithm to reduce the number of renegotiations and the bandwidth fluctuation in [12]. In our previous work [10], we studied adaptive rate-control algorithms to pursue an effective trade-off between temporal and spatial qualities for streaming video and interactive video applications over RCBR network.

However, only bandwidth renegotiation is sometimes not sufficient to efficiently support the nonstationary video traffics and improve the network utilization. (The higher network utilization means that the better services are provided to users and/or more users are supported with the same network resources.) Generally speaking, more network resources are required for the media delivery as its traffic becomes more burst although the long-term average bandwidth is the same. Thus, we need more flexible QoS renegotiating approaches for streaming videos to improve network utilization and enhance video quality at the receivers end. In this paper, we consider not only channel bandwidth but also the burstiness of the traffic. To handle the problem, token bucket is adopted for the traffic model, and its parameters are estimated based on the statistical characteristics of compressed video traffic during the data transmission. This paper is organized as follows: a brief review of traffic models is introduced in Section 2; effective QoS renegotiating schemes are proposed in Section 3; experimental results are provided in Section 4 to show the superior performance of the proposed schemes; and finally, concluding remarks are presented in Section 5.

2. TRAFFIC MODEL

So far, various traffic models have been proposed for efficient network resource management such as policing, resource reservation, rate shaping, and so forth. For example, leaky bucket model [14], double leaky bucket model [15], token bucket model [16, 17], and so forth. As mentioned earlier, the token bucket model is adopted in this paper, which is one of the most popular traffic models and widely employed for IntServ protocol [18]. In the token bucket model, each packet can be transmitted through the network with one token only when tokens are available at the token buffer. The tokens are generally provided by network with a fixed rate. When the token buffer is empty, the packet must wait for a token in the smoothing buffer. On the other hand, the new arriving tokens are dropped when the token bucket is full. It means the waste of network resource. The token bucket model can be characterized by two parameters: token filling rate and token bucket size. The token filling rate and the token bucket size are related to the average channel bandwidth and the burstiness of the underlying video traffic, respectively. In general, more burst traffic needs a larger token bucket size, and complex token model has one more parameter than simple bucket model, that is, it can be characterized by the token filling rate, token bucket size, and peak rate. Their performance comparison can be found in [19].

An overview of simple bucket model is shown in Figure 1.

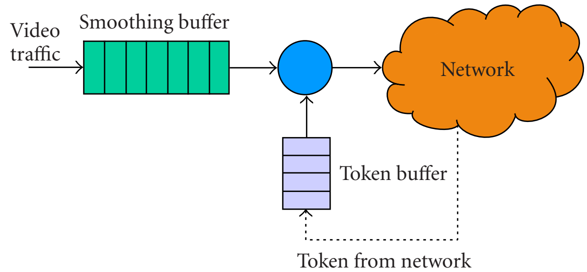


Figure 1: Overview of the simple token bucket model.

The token bucket is thought to be located in either the user side or the network side. The network needs the token bucket to policy the incoming traffics while the user requires the token bucket to generate the video traffic according to the predetermined specification. Smoothing buffer is also an important factor to determine the video traffic characteristics, which relates to packet loss rate and time delay. Since the smoothing buffer size is practically finite, buffer management algorithm is needed to minimize the degradation of video quality caused by buffer overflow. In this paper, the following buffer management is employed: B-, P-, and I-frames are discarded in sequence when smoothing buffer overflows. It is determined by how much the quality of the decoded video may be degraded when a frame is lost. When the I-frame is dropped, the whole GOPs (group of pictures) cannot be decoded since the I-frame is referenced for the following P-frames and B-frames. When the P-frame is dropped, the following frames in the GOPs disappear. However, only one frame is missing when the B-frame is dropped since the other frames do not reference it. To more improve the video quality, network needs to classify the incoming packets and consider the error corruption in the whole sequences caused by a specific packet loss [20, 21]. However, it is a big burden to network because of a large amount of computation. In this paper, we consider the renegotiations of token bucket parameters during data transmission as a solution to improve network utilization and enhance video quality at the receiver end.

3. PROPOSED TOKEN BUCKET PARAMETER ESTIMATING SCHEMES

Over the network supporting QoS renegotiations, the sender has to determine when QoS renegotiation is required and what QoS is needed for. Note that, in general, more renegotiations can increase the network utilization; however, they may cause larger signaling overhead. We assume that the compressed data for each frame is divided into fixed size packets, and thus the number of packets (N_i) for the ith frame is calculated by

$$N_i = \left\lfloor \frac{B_i}{P_{\max}} \right\rfloor, \tag{1}$$

where $\lfloor x \rfloor$ indicates the smallest integer that is greater than x, B_i is the amount of bits for the compressed ith frame, and $P_{\max}$ is the packet size. Under the assumption that the video stream is accepted by call admission control, we focus on only the QoS renegotiating process in this paper. In many cases, the compressed data may not be divided into the fixed size packets for the robust transmission. However, the above assumption is still reasonable if packets are assumed to consume the different number of tokens according to their size.

We examine two approaches for the QoS renegotiation: fixed renegotiating interval approach and variable renegotiating interval approach. Renegotiations are tried periodically in the fixed renegotiating interval case while they are tried only when required in the variable renegotiating interval case. It is expected that variable renegotiating interval approach can avoid unnecessary renegotiations and unsuitable renegotiating instants with higher computational complexity. In each renegotiating interval, we estimate the required token bucket parameters based on the statistical information of video traffic. That is, token filling rate and token bucket size are determined by the mean and the standard deviation of number of packets, respectively.

3.1. *Fixed renegotiating interval case*

First of all, the statistical information, mean and standard deviation of the underlying video traffic, is calculated in the reference window, and then the token bucket model parameters, token filling rate, and token bucket size are estimated to keep the packet loss rate in the tolerable range. Then, the whole time interval of the underlying video are divided into time intervals with the same size, and the mean and the standard deviation are calculated in each interval. Based on the information, the required token bucket model parameters in the arbitrary renegotiating interval are determined. The above processes can be summarized as follows: renegotiations are tried at every interval with these parameters:

$$R_i = \left(1 + \alpha \frac{m_i - M_{\text{ref}}}{M_{\text{ref}}}\right) \cdot R_{\text{ref}}, \tag{2}$$

$$Q_i = \left(1 + \beta \frac{\sigma_i - \sigma_{\text{ref}}}{\sigma_{\text{ref}}}\right) \cdot Q_{\text{ref}}, \tag{3}$$

where M_{ref} and m_i are the mean values of numbers of packets for each frame in the reference window; the ith renegotiating interval, respectively, σ_{ref} and σ_i are the standard deviations of numbers of packets for each frame in the reference window; the ith renegotiating interval, respectively, α and β are the weighting factors; R_i and Q_i are the token filling rate and the token bucket size in the ith renegotiating interval, respectively; and R_{ref} and Q_{ref} are the token filling rate and the token bucket size in the reference window, respectively. We assume that the number of packets for a frame in the reference window is Gaussian distributed for the simplicity, and then R_{ref} and Q_{ref} are determined by

$$\begin{aligned} R_{\text{ref}} &= \frac{\sum_{i=1}^{F_{\text{ref}}} N_i}{F_{\text{ref}}}, \\ Q_{\text{ref}} &= \sigma_{\text{ref}} \cdot I + M_{\text{ref}}, \end{aligned} \tag{4}$$

where F_{ref} is the number of frames in the reference window and I satisfies the following equation:

$$\Pr(X > I) \leq p, \tag{5}$$

where X is a Gaussian random variable with zero mean and unit standard deviation, and p is the tolerable packet loss probability.

3.2. *Variable renegotiating interval case*

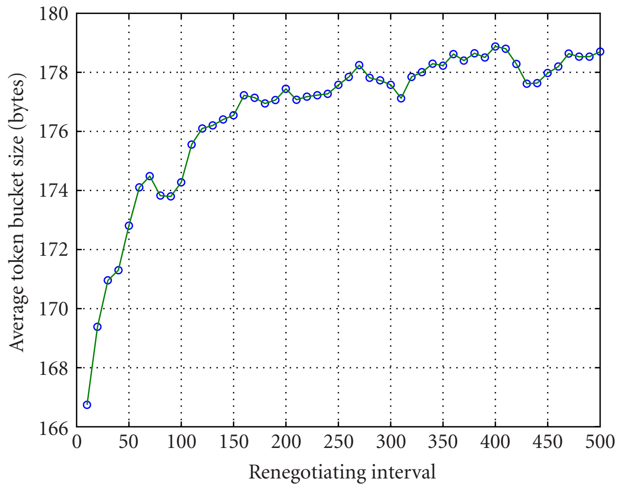
When the fixed renegotiating interval approach is tested, undesirable phenomena are sometimes observed. That is, the average token bucket size, token drop rate, and packet loss rate locally fluctuate as shown in Figures 2 and 3 even though their general trends globally decrease as the average renegotiating interval becomes small. One of the reasons is that the fixed renegotiating interval can make the inappropriate interval segmentation. To solve this problem, we consider a variable renegotiating interval approach. Now, we define the basic renegotiating interval unit consisting of several GOPs and address how to determine the renegotiating instants by using the basic unit. As shown in Figures 2 and 3 (the fixed renegotiating interval case), the graphs of average token bucket size, token drop rate, and packet loss rate look very similar. Thus, one of them can be used as a measure for the determination of renegotiating instants. In this paper, packet loss rate is employed. First, we calculate the packet loss rate in the current window, that is, the time interval since the latest renegotiation, and compute the new packet loss rate when the next basic renegotiating interval is included in the window. Second, we determine whether the next basic renegotiating interval is included or not in the window based on the difference between the two packet loss rates. It can be summarized as follows. If

$$\frac{\text{PLR}_{\text{next}}}{\text{PLR}_{\text{cur}}} > 1 + T(\mu, n), \tag{6}$$

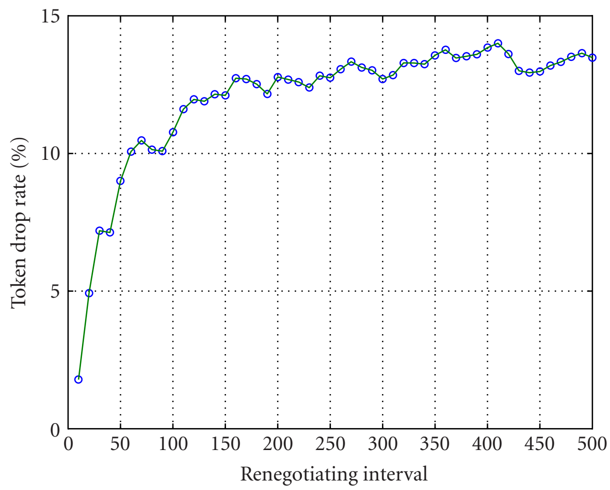
then the next basic interval is not included in the window. Otherwise, the next basic interval is included in the window. Where PLR_{cur} is the packet loss rate in the current window, PLR_{next} is the packet loss rate when the next basic renegotiating interval is included in the current window, n is the number of the minimum renegotiating intervals in the current window, μ is a variable determining the number of renegotiations, and $T(\mu, n)$ is a threshold function which must take into account the fact that the effect of the next basic renegotiating interval on PLP_{next} decreases as the window size increases. In this paper, $T(\mu, n)$ is simply defined by

$$T(\mu, n) = \frac{\mu}{100 \cdot n}. \tag{7}$$

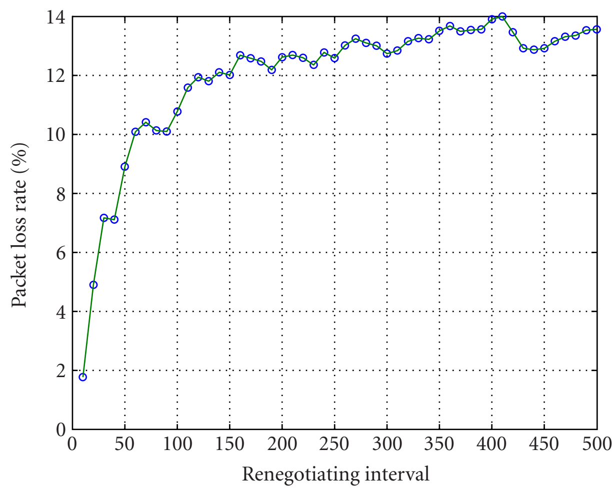
If the renegotiating instant is determined by the above process, the token bucket model parameters for the current interval are estimated by the same method ((2) and (3)) of the fixed renegotiating interval case. Basically, the length of the basic renegotiating interval unit is related to the network utilization and the computational complexity. As the length becomes smaller, network utilization can be improved while the required computational complexity increases.



(a)

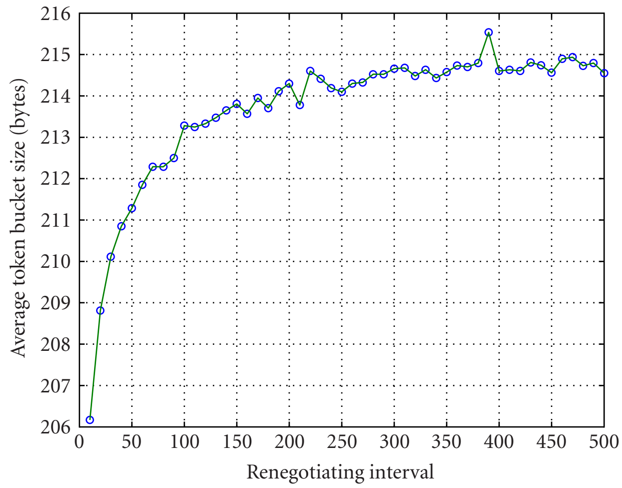


(b)

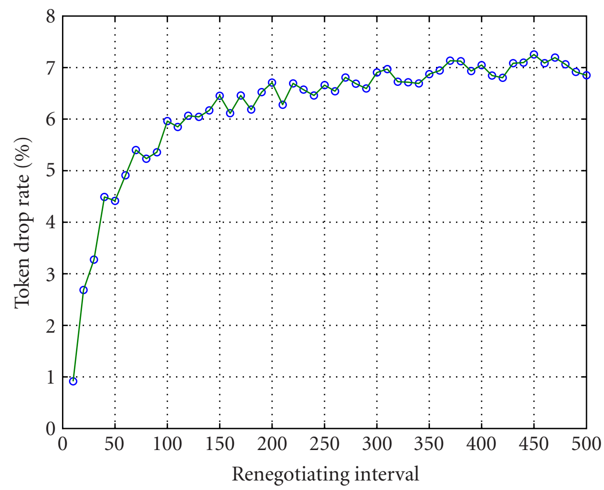


(c)

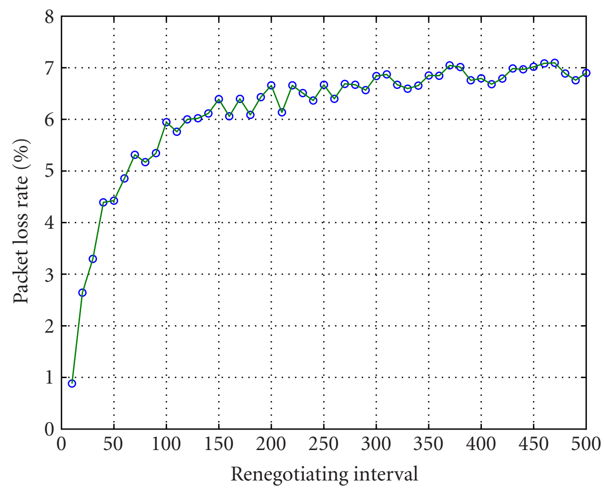
Figure 2: Performance comparison (the test trace file is Star Wars and the packet size is 100 bytes): (a) average token bucket size, (b) token drop rate, and (c) packet loss rate. The circles denote specific data at renegotiating intervals and the solid lines denote the interpolated values.



(a)



(b)



(c)

Figure 3: Performance comparison (the test trace file is Terminator 2 and the packet size is 100 bytes): (a) average token bucket size, (b) token drop rate, and (c) packet loss rate. The circles denote specific data at renegotiating intervals and the solid lines denote the interpolated values.

4. EXPERIMENTAL RESULTS

In the experiment, the test trace files are Star Wars ($240 * 352$ size) and Terminator 2 (QCIF size) encoded by MPEG-1 [22, 23, 24], whose lengths are 40 000 frames. The encoding structure is IBBPBBPBBPBB (i.e., 1GOP consists of 12 frames), and I-frames, P-frames, and B-frames are encoded with quantization parameters 10, 14, and 18, respectively. The encoding frame rate is 25 frames per second. As a result, the output traffics are VBR and their statistical properties are summarized in Table 1. The variables and threshold values of the proposed schemes are determined as follows.

(i) The tolerable maximum packet loss rate in (5) is set to 3%.
(ii) The smoothing buffer size is set to the average value of two GOPs (223516 bytes for Star Wars and 261714 bytes for Terminator 2).
(iii) The basic renegotiating interval is set to 10 GOPs.
(iv) The tested packet sizes are 100 bytes or 400 bytes.
(v) The reference window size is set to the whole frame number (40 000 frames).
(vi) The weighting factors α and β in (2) and (3) are set to 1.

To compare the performance of the proposed QoS renegotiating schemes, we use average token drop rate, average token bucket size, and token filling rate as the network utilization measure, and packet loss rate is employed as the video quality degradation measure.

4.1. Fixed renegotiating interval case

The performance comparison with respect to various fixed renegotiating intervals is shown in Tables 2, 3, 4, and 5, and Figures 2 and 3. It is observed that the average token bucket size is reduced by about 11% as the renegotiating interval decreases while the average token filling rate is almost the same for all renegotiating intervals (it can be understood since token bucket size is determined relatively by comparing the standard deviation in the reference window with that in the current renegotiating interval, see (2)). As a result, the network utilization can be improved. Furthermore, token drop rate is reduced by about 90% and packet loss rate is reduced by about 75% when the renegotiating interval is set to 10 GOPs. The same results are observed regardless of the packet size. It means that the waste of network resource caused by the dropped tokens and the video quality degradation caused by the lost packets can be significantly reduced. However, it is observed in Figures 2 and 3 that the average token bucket size and packet loss rate locally fluctuate even though the average renegotiating interval decreases. As mentioned earlier, one of the reasons is that inappropriate renegotiating instants may occur when the renegotiating interval is fixed.

4.2. Variable renegotiating interval case

In this section, variable renegotiating time interval case is examined. The experimental results are summarized in Tables 6, 7, 8 and 9, and Figure 4. It is observed in Tables 6 and 7 that the average token bucket size is almost the same, while token

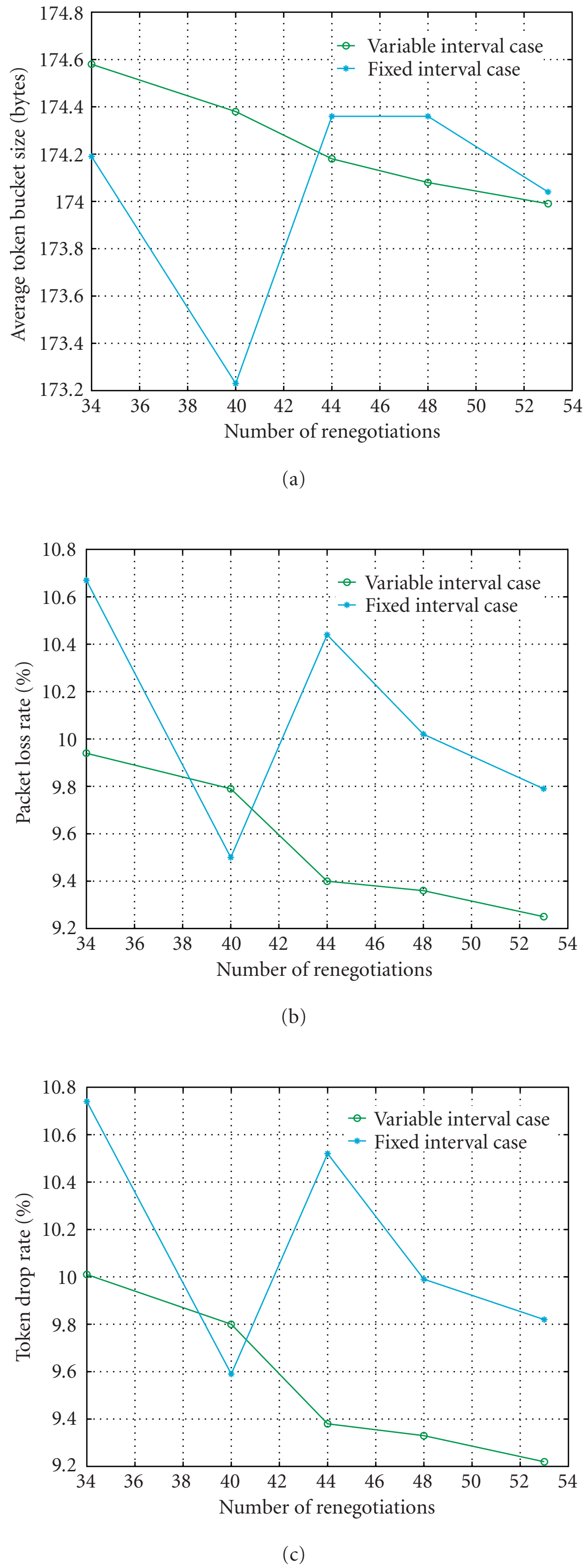


Figure 4: Performance comparison between variable renegotiating interval scheme and fixed renegotiating interval scheme (the test trace file is Star Wars and the maximum packet size is 100 bytes): (a) average token bucket size, (b) packet loss rate, and (c) token drop rate.

Table 1: Statistical properties of test MPEG trace files.

Trace files	Minimum value (bytes)	Maximum value (bytes)	Average (bytes)	Standard deviation (bytes)
Star Wars	275	124816	9313.2	12902.725
Terminator 2	312	79560	10904.75	10158.031

Table 2: Performance comparison of the fixed renegotiating interval case when the packet size is 100 bytes and the test trace file is Star Wars encoded by MPEG-1.

Fixed renegotiating interval	With renegotiations							Without renegotiation
Interval (GOPs)	10	20	50	90	130	200	300	3330
Avg. token filling rate	93.59	93.60	93.67	93.59	93.67	93.76	93.54	94
Avg. token bucket size (bytes)	166.74	169.34	172.81	173.80	176.20	177.44	177.58	185.06
Token drop rate (%)	1.78	4.92	9.00	10.08	11.89	12.78	12.70	17.26
Packet loss rate (%)	1.77	4.90	8.91	10.10	11.81	12.62	12.74	16.90

Table 3: Performance comparison of the fixed renegotiating interval case when the packet size is 400 bytes and the test trace file is Star Wars encoded by MPEG-1.

Fixed renegotiating interval	With renegotiations							Without renegotiation
Interval (GOPs)	10	20	50	90	130	200	300	3330
Avg. token filling rate	23.75	23.78	23.76	23.82	23.73	23.74	23.70	24
Avg. token bucket size (bytes)	42.35	43.02	43.90	44.21	44.76	45.22	45.08	47.02
Token drop rate (%)	1.68	4.81	8.70	9.97	11.50	12.38	12.36	17.21
Packet loss rate (%)	1.75	4.73	8.71	9.79	11.62	12.49	12.60	16.39

Table 4: Performance comparison of the fixed renegotiating interval case when the packet size is 100 bytes and the test trace file is Terminator 2 encoded by MPEG-1.

Fixed renegotiating interval	With renegotiations							Without renegotiation
Interval (GOPs)	10	20	50	90	130	200	300	3330
Avg. token filling rate	109.53	109.54	109.47	109.49	109.49	109.52	109.56	110
Avg. token bucket size (bytes)	206.17	208.81	211.29	212.50	213.47	214.30	214.66	215
Token drop rate (%)	0.91	2.69	4.42	5.36	6.04	6.71	6.90	8.37
Packet loss rate (%)	0.88	2.64	4.43	5.34	6.02	6.66	6.84	8.25

Table 5: Performance comparison of the fixed renegotiating interval case when the packet size is 400 bytes and the test trace file is Terminator 2 encoded by MPEG-1.

Fixed renegotiating interval	With renegotiations							Without renegotiation
Interval (GOPs)	10	20	50	90	130	200	300	3330
Avg. token filling rate	27.74	27.76	27.80	27.77	27.77	27.88	27.80	28
Avg. token bucket size (bytes)	52.47	53.18	53.77	54.15	54.28	54.64	54.62	55
Token drop rate (%)	0.86	2.64	4.41	5.26	5.94	6.78	6.86	8.33
Packet loss rate (%)	0.88	2.57	4.20	5.17	5.82	6.30	6.66	7.48

Table 6: Performance comparison between variable renegotiating interval case and fixed renegotiating interval case when the test trace file is Star Wars encoded by MPEG-1 and the maximum packet size is 100 bytes.

	Variable renegotiating approach				Fixed renegotiating approach			
μ	Number of renegotiation	Average token drop rate (%)	Average token bucket size (bytes)	Average packet loss rate (%)	Number of renegotiation	Average token drop rate (%)	Average token bucket size (bytes)	Average packet loss rate (%)
10	53	9.22	173.99	9.25	53	9.82	174.04	9.79
20	48	9.33	174.08	9.36	48	9.99	174.36	10.02
30	44	9.38	174.18	9.40	44	10.52	174.26	10.44
40	40	9.80	174.38	9.79	40	9.59	173.23	9.50
50	34	10.01	174.58	9.94	34	10.74	174.19	10.67

Table 7: Renegotiating time instants of variable renegotiating interval case and fixed renegotiating interval case when the test trace file is Star Wars encoded by MPEG-1 and the maximum packet size is 100 bytes.

Method	QoS renegotiating instants (frame number)
Variable interval	0, 600, 840, 2280, 2880, 3000, 3840, 3960, 4680, 5400, 5760, 7200, 7320, 7920, 8280, 9120, 10080, 10560, 11520, 15120, 15840, 17880, 19440, 20160, 20760, 21240, 21720, 21840, 22320, 22680, 23760, 24840, 24960, 25800, 26400, 27240, 28920, 29520, 29640, 29760, 30120, 30720, 31320, 33360, 33600, 33840, 35400, 35520, 36480, 37560, 37920, 38280, 38640
Fixed interval	0, 732, 1464, 2196, 2928, 3660, 4392, 5124, 5856, 6588, 7320, 8052, 8784, 9516, 10248, 10980, 11712, 12444, 13176, 13908, 14640, 15372, 16104, 16836, 17568, 18300, 19032, 19764, 20496, 21228, 21960, 22692, 23424, 24156, 24888, 25620, 26352, 27084, 27816, 28548, 29280, 30012, 30744, 31476, 32208, 32940, 33672, 34404, 35136, 35868, 36600, 37332, 38064

Table 8: Performance comparison between variable renegotiating interval case and fixed renegotiating interval case when the test trace file is Terminator 2 encoded by MPEG-1 and the maximum packet size is 100 bytes.

	Variable renegotiating approach				Fixed renegotiating approach			
μ	Number of renegotiation	Average token drop rate (%)	Average token bucket size (bytes)	Average packet loss rate (%)	Number of renegotiation	Average token drop rate (%)	Average token bucket size (bytes)	Average packet loss rate (%)
10	46	5.16	212.25	5.13	46	5.19	212.08	5.14
20	44	5.54	212.73	5.47	44	5.70	212.47	5.64
30	43	5.60	213.83	5.51	43	5.76	212.66	5.73
40	43	5.60	212.79	5.51	43	5.76	212.6	5.73
50	43	5.60	212.79	5.51	43	5.76	212.6	5.73

Table 9: Renegotiating time instants of variable renegotiating interval case and fixed renegotiating interval case when the test trace file is Terminator 2 encoded by MPEG-1 and the maximum packet size is 100 bytes.

Method	QoS renegotiating instants (frame number)
Variable interval	0, 120, 480, 1080, 1800, 2400, 3720, 5040, 5520, 5880, 7920, 8160, 8880, 9960, 10680, 12000, 12480, 13440, 14760, 15240, 15960, 16680, 17880, 18720, 19560, 20400, 20880, 22080, 23280, 24120, 24600, 25560, 26760, 27000, 27600, 28920, 29040, 32160, 32760, 33120, 33840, 34800, 35160, 35760, 36840, 38040
Fixed interval	0, 852, 1704, 2556, 3408, 4260, 5112, 5964, 6816, 7668, 8520, 9372, 10224, 11076, 11928, 12780, 13632, 14484, 15336, 16188, 17040, 17892, 18744, 19596, 20448, 21300, 22152, 23004, 23856, 24708, 25560, 26412, 27264, 28116, 28968, 29820, 30672, 31524, 32376, 33228, 34080, 34932, 35784, 36636, 37488, 38340

Table 10: Performance comparison between the proposed algorithm and bandwidth renegotiating scheme (test trace file is Star wars).

Number of renegotiations	Proposed algorithm		Channel bandwidth renegotiating algorithm	
	Token drop rate (%)	Packet loss rate (%)	Token drop rate (%)	Packet loss rate (%)
53	9.22	9.25	9.79	9.89
48	9.33	9.36	9.89	10.04
44	9.38	9.40	9.94	10.22
40	9.80	9.79	10.56	10.42
34	10.0	9.94	10.62	10.62

Table 11: Performance comparison between the proposed algorithm and bandwidth renegotiating scheme (test trace file is Terminator 2).

Number of renegotiations	Proposed algorithm		Channel bandwidth renegotiating algorithm	
	Token drop rate (%)	Packet loss rate (%)	Token drop rate (%)	Packet loss rate (%)
46	5.16	5.13	5.59	5.46
44	5.54	5.47	5.99	5.83
43	5.60	5.51	6.04	5.88
43	5.60	5.51	6.04	5.88
43	5.60	5.51	6.04	5.88

drop rate and packet loss rate are reduced by 8.6% and 7.5%, respectively, when the number of renegotiations is changed from 43 to 46. Thus, the waste of network resource can be reduced and the video quality degradation caused by the lost packets can be decreased too. In addition, it is observed that average token drop rate, average token bucket size, and token filling rate monotonically decrease while those of fixed renegotiating approach locally fluctuate. We can see the obvious differences of the renegotiating time instants in Tables 7 and 8. It means that we can predict the traffic characteristics more accurately by the interpolation method when μ changes. Hence, we can conclude that variable renegotiating approach can determine the renegotiating instants more effectively than fixed renegotiating approach at the cost of the increased computational complexity.

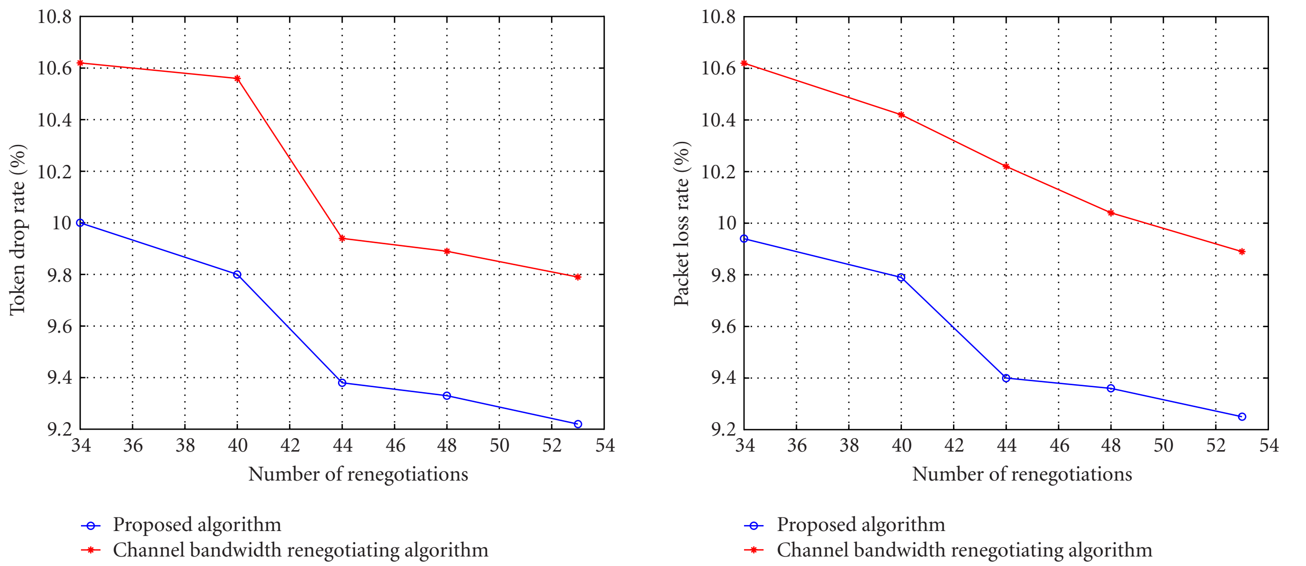
4.3. Performance comparison with bandwidth renegotiating schemes

In this section, we compare the proposed algorithm with bandwidth renegotiating algorithms. Actually, it is not easy to simply compare the performance with bandwidth renegotiating algorithms since they provide the deterministic services and consider the different network situations. Thus, we implemented the channel bandwidth renegotiating scheme by token bucket model with a piecewise constant token filling rate and a fixed token bucket size (it is set to the average value of the proposed algorithm) and then tested various renegotiating interval cases. The experimental results are summarized in Tables 10 and 11, and Figure 5. As shown in the tables and figure, we observe that the proposed algorithm can reduce both the packet loss rate and the token drop rate. The reason is that the proposed algorithm treats token bucket size as well as token filling rate as control variables while the bandwidth renegotiating schemes consider only token filling rate as a control variable.

We would like to give some remarks on the experimental results. We obtain Figure 6 when the histograms of video traffics are drawn. They look like Poisson distributed although we assume Gaussian distribution for simplicity. This mismatch can cause some errors, and the basic renegotiating interval may also be related to the errors. As the length of basic renegotiating interval becomes small, the performance may be improved at the expense of higher computational complexity.

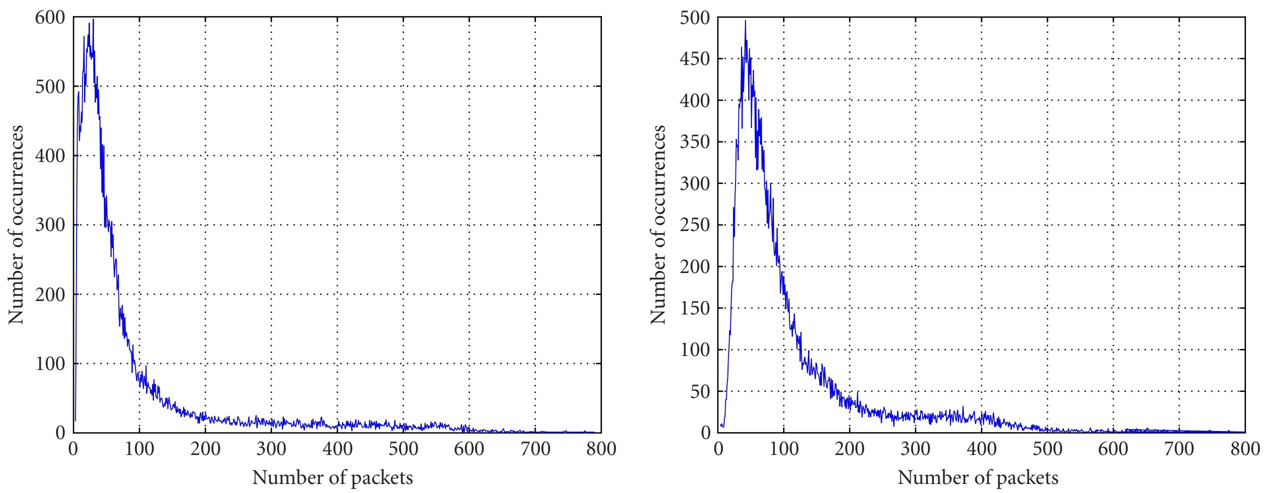
5. CONCLUSION AND FUTURE WORK

In this paper, we presented effective token bucket parameter renegotiating schemes for streaming video over network supporting QoS renegotiations. Two approaches, fixed renegotiating interval case and variable renegotiating interval case, are examined. The experimental results showed that the average token bucket size and the packet loss rate are significantly reduced as the number of renegotiations increases. Furthermore, variable renegotiating interval case avoids the inappropriate renegotiating instants of fixed renegotiating interval case at the cost of the increased computational complexity. Based on these observations, we can conclude that the proposed flexible QoS renegotiating approach can improve the network utilization compared to the bandwidth renegotiating approach and is a promising technique for the effective streaming video. On the other hand, if Tables 6 and 8 are stored as metadata in database, we can estimate the average token bucket model parameters of the new video on-demand request by linear interpolation method with a low computational complexity. Basically, the information may be very



(a) (b)

Figure 5: Performance comparison between the proposed algorithm and bandwidth renegotiating scheme (when the test trace file is Star Wars and packet size is 100 bytes): (a) token drop rate and (b) packet loss rate.



(a) (b)

Figure 6: Histogram of test video traffics: (a) Star Wars and (b) Terminator 2.

helpful to design a simple but quite effective call admission control algorithm. For the complete solution, we need the rate shaping/adaptation algorithm to adjust the compressed video bitstream when the QoS requests are sometimes rejected which is under our current investigation.

ACKNOWLEDGMENT

This work is supported by the University Fundamental Research Program supported by the Ministry of Information & Communication of the Republic of Korea.

REFERENCES

[1] ISO/IEC 13818 (MPEG-2), "Generic coding of moving pictures and associated audio information," November 1994.
[2] ISO/IEC JTC 1/SC 29/WG 11 N4030, "Overview of the MPEG-4 standard," March 2001.
[3] ITU-T Recommendation H.261, "Video Codec for Audio Visual Services at $p * 64$ kbits/s," March 1993.
[4] ITU-T Recommendation H.263 version 2, "Video coding for low bitrate communication," January 1998.
[5] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: Tradeoffs and potential," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 952–973, 1998.

[6] Z.-L. Zhang, J. Kurose, J. D. Salehi, and D. Towsley, "Smoothing, statistical multiplexing, and call admission control for stored video," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 1148–1166, 1997.

[7] M. Grossglauser, S. Keshav, and D. C. Tse, "RCBR: A simple and efficient service for multiple time-scale traffic," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 741–755, 1997.

[8] A. Mohammad, "Using adaptive linear prediction to support real-time VBR video under RCBR network service model," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 635–644, 1998.

[9] T.-Y. Kim, B.-H. Roh, and J.-K. Kim, "Bandwidth renegotiation with traffic smoothing and joint rate control for VBR MPEG video over ATM," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 693–703, 2000.

[10] H. Song and K. M. Lee, "Adaptive rate control algorithms for low-bit-rate video under the networks supporting bandwidth renegotiation," *Signal Processing: Image Communication*, vol. 17, no. 10, pp. 759–779, 2002.

[11] H. Zhang and E. W. Knightly, "RED-VBR: A renegotiation-based approach to support delay-sensitive VBR video," *ACM Multimedia Systems Journal*, vol. 5, no. 3, pp. 164–176, 1997.

[12] J. D. Salehi, Z.-L. Zhang, J. Kurose, and D. Towsley, "Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 397–410, 1998.

[13] M. Wu, R. A. Joyce, H.-S. Wong, L. Guan, and S.-Y. Kung, "Dynamic resource allocation via video content and short-term traffic statistics," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 186–199, 2001.

[14] B. V. Patel and C. C. Bisdikian, "End-station performance under leaky bucket traffic shaping," *IEEE Network*, vol. 10, no. 5, pp. 40–47, 1996.

[15] D. Grossman, "Definition of VBR service," Contribution ATM Forum/94-0816, September 1994.

[16] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service," IETF RFC 2212, September 1997.

[17] S. Verma, R. K. Pankaj, and A. Leon-Garcia, "Call admission and resource reservation for guaranteed QoS services in Internet," *Computer Communication*, vol. 21, no. 4, pp. 362–374, 1998.

[18] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service," IETF RFC 2475, December 1998.

[19] J. Glasmann, M. Czermin, and A. Riedl, "Estimation of token bucket parameters for videoconferencing systems in cooperate networks," in *International Conference on Software, Telecommunications and Computer Networks*, Trieste, October 2000.

[20] N. Farber, K. Stuhlmuller, and B. Girod, "Analysis of error propagation in hybrid video coding with application to error resilience," in *Proceedings of IEEE International Conference on Image Processing*, Kobe, Japan, October 1999.

[21] J.-G. Kim, J. Kim, J. Shin, and C.-C. J. Kuo, "Coordinated packet level protection employing corruption model for robust video transmission," in *SPIE Proc. of Visual Communication and Image Processing*, San Jose, Calif, USA, January 2001.

[22] Berkeley Multimedia Research Center, ftp://mm-ftp.cs.berkeley.edu/pub/.

[23] Bellcore, ftp://ftp.telecordia.com/pub/vbr/video/trace/.

[24] KAIST, http://viscom.kaist.ac.kr/.

Hwangjun Song received his B.S. and M.S. degrees from the Department of Control and Instrumentation, School of Electrical Engineering, Seoul National University, Korea, in 1990 and 1992, respectively, and his Ph.D. degree in electrical engineering systems, University of Southern California, Los Angeles, Calif., USA, in 1999. He was a Research Engineer at LG Industrial Lab., Korea, in 1992. From 1995 to 1999, he was a Research Assistant in SIPI (Signal and Image Processing Institute) and IMSC (Integrated Media Systems Center), University of Southern California. Since 2000, he has been a faculty member with the School of Electronic and Electrical Engineering, Hongik University, Seoul, Korea. His research interests include multimedia signal processing and communication, image/video compression, digital signal processing, network protocols necessary to implement functional image/video applications, control system and fuzzy-neural system.

Dai-Boong Lee received his B.S. degree from Hongik University, Seoul, Korea, in 2002, where he is currently working toward his M.S. degree in Multimedia Communication System Lab., School of Radio Science and Communication Engineering. His research interests include packet scheduling, quality-of-service network, Int/Diffserv, network resource renegotiation algorithm, network management, and visual information processing.