Empirical Study of VBR Traffic Smoothing in Wireless Environment*

Youjip Won and Bowie Shim

Division of Electrical and Computer Engineering
Hanyang University, Seoul, Korea, 133-791
{yjwon,shim77}@ece.hanyang.ac.kr

Abstract. This work presents the result of the empirical study on the effect of VBR smoothing in broadband wireless network. Traffic smoothing of VBR stream has been the subjects of intense research during past several years. While preceding algorithms successfully remove burstiness in the underlying process, these works do not address how the respective smoothing algorithm can effectively improve the QoS in practical environment. We developed MPEG-4 streaming system and instrument the client terminal which is handheld mobile device. We examine the effect of smoothing over the packet loss behavior and empirical QoS under various different system settings. We use the rate variability as optimization criterion in generating the packet transmission schedule. We find that smoothing with small size buffer(10 Kbyte) brings a significant improvement on packet loss ratio and greatly enhances the QoS perceived by the end user. Via adopting smoothing technique in transporting multimedia traffic, we are able to increase the *acceptable quality* frame rate by 50%.

1 Introduction

1.1 Motivation

Due to the rapid advancement of CPU computing capability as well as the network transmission speed, we can enjoy the real-time remote playback of video stream without much difficulty these days. Further, the deployment of third generation wireless technology[4] makes it possible to access to streaming service without geographic limitation. Rapid proliferation of usage of mobile device accompanied by the availability of wireless network connection makes efficient support of video streaming service in mobile terminal emphasized more and more.

Unlike the general-purpose desk-top computer, which has abundant computing resources and storage capacity, mobile device in consumer electronics domain have stringent resource constraints due to its restriction on power consumption, pricing, reliability, etc. Thus, in this type of devices, special care needs to be taken in allocating resources to application and over-provisioning of resources should be strictly avoided.

* This work is funded by KOSEF through Statistical Research Center for Complex System at Seoul National University.

H. Unger, T. Böhme, and A. Mikler (Eds.): I^{2}CS 2002, LNCS 2346, pp. 193–204, 2002.

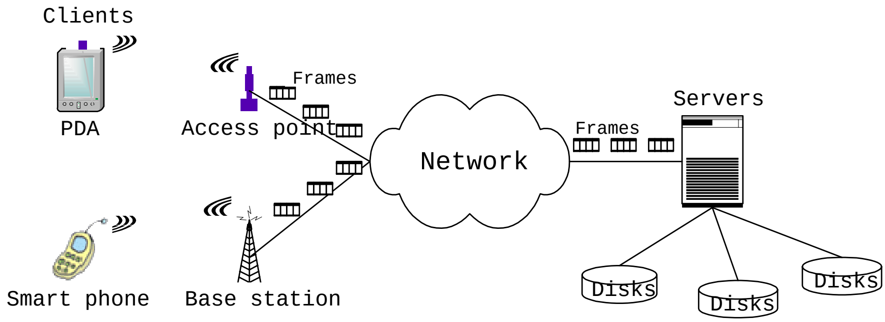


Fig. 1. Streaming Service in Wireless Internet Environment

Due to the inter frame coded nature of constant quality compression scheme, the resulting compressed video stream exhibits order of magnitude difference in successive frame sizes. This large variance in frame sizes raises burstiness in transmitting the compressed video for the real-time playback. Seamless delivery of multimedia data mandates that a certain amount of resources involved in transporting the VBR data from the server to the client needs to be dedicated. Although the mobile terminals can allocate resources, e.g. CPU cycles, memory buffers, based on peak-bit rate of the source stream, such over-provisioning is extremely wasteful and undermines the benefits of VBR encoding technique. Bandwidth smoothing techniques can reduce the burstiness of traffic and subsequently can facilitate more efficient resource usage.

The objective of *smoothing* is to provide better quality streaming service with minimum amount of resources by reducing the *burstiness of the traffic*. A number of elaborate smoothing algorithms have been published in various forums and literatures, each of which uses different performance metrics to compute the packet transmission schedule, e.g. rate variations, number of rate changes, client buffer utilization, to list a few. These techniques successfully removes the burstiness of the original packet traffic. However, we yet do not know how these smoothing techniques actually contribute to improving the *Quality of Service* perceived by the end user. Indeed, none of preceding works address how their smoothing algorithms can improve the *packet loss* and *jitter* behavior in actual system.

In the context of smoothing of VBR stream, there are two important issues which deserve more attention. The first issue is to identify *relationship between smoothing criteria and packet loss*(and jitter) behavior. The second issue is to identify the *relationship between packet loss(and jitter) and Quality of Service* perceived by the user. Packet loss and jitter is two widely accepted metric for quality of service. However, these two metrics does not deliver sufficient clue about the quality of service perceived by the user. Same number of packet losses can affect the quality of service in many different ways. We carefully believe that even with the same number of packet losses, burstiness of packet losses and/or the type of frame which the lost packets belong to can significantly alter the way that the lost packet affects the QoS of the stream. In this work, we like to present the result of our empirical study on VBR traffic smoothing in broadband wireless Internet. Particularly, the client application runs on the mobile handheld device which may exhibit unique characteristics different from general purpose desk-top PC. We instrument the effect of smoothing on packet loss behavior in mobile terminal under various different system settings. Our experimental results reveal that smoothing enables the end system to effectively handle the incoming stream. We are able to increase the frame rate by 50% with the adoption of smoothing algorithm.

1.2 Related Works

A number of smoothing algorithms have been proposed, each of which uses different performance metrics and each of which generates different schedules. The performance metrics include the number of rate changes[6], the variability of the bandwidth requirement[17], the number of on-off segments in an on-off transmission model[21], the client buffer utilization[5], and general cost metrics through dynamic programming[14]. Feng and Rexford provide in depth survey of these techniques[7]. Boudec and Verscheure developed smoothing technique for guaranteed service network, e.g. RSVP[2]. Chang et al[3] proposes an window based smoothing algorithm which can be used for online smoothing. There have been a number of efforts to quantify the quality of service[1,9,18] in realtime multimedia delivery. Apteker et al[1] investigated the relationship between the number of streaming sessions and the quality of service of individual streams. Ghinea et al[9] investigated not only the degree of satisfaction but also the degree of understanding as a parameter to QoS.

The rest of the paper is organized as follows. Section 2 describes VBR traffic used in this experiment. Section 3 presents the smoothing technique used in this work. Section 4 presents the results of the experiments. Section 5 concludes the paper.

2 Empirical VBR Process

2.1 Characteristics of Compressed Stream

MPEG coding scheme exploits the temporal and spatial difference between successive frames. Stochastic characteristics of VBR bandwidth process may vary depending on the nature of the original video clip. It is worth noting that frame sequence we are dealing with is the one which is actually transmitted and is different from the order in which the frames are displayed. The B type frame has bidirectional dependency. It depends on the preceding I or P frame as well as the following I or P frame. To resolve the forward referencing problem, encoder reorders the frames such that the frame does not have to wait for the arrival of another frame for decompression.

From the original video scene, we generated 4 streams with frame rates 4, 5, 6, and 10 frames/sec, respectively. Table 1 summarizes the traffic characteristics.

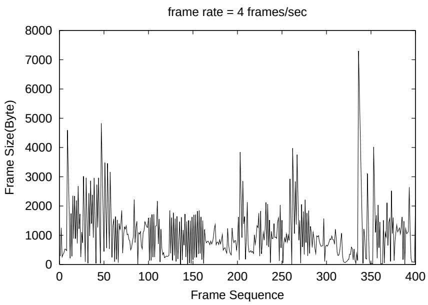
Each clip is encoded with MPEG 4(DIVX) codec. It is worth noting that DIVX codec does not have fixed GOP pattern. User only specifies the maximum distance between successive I frames and thus, the frame sequence yields irregular pattern.

Proper characterization of VBR traffic plays critical role in designing various components of the system: server, router, client, network transport, etc. and importance of which cannot be emphasized any further. There have been a number of efforts which rigorously examine the stochastic characteristics of empirical VBR process[12,16,8,20,13]. Most of these works focus their effort on properly identifying the inter GOP and intra GOP correlation structure of the frame size sequence. While the results of various traffic characterization studies have their own assumptions, the common findings are that the VBR traffic exhibits very bursty behavior.

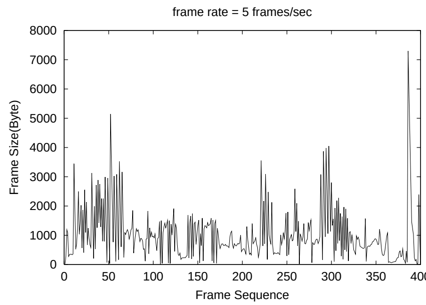
Figures in Fig. 2 illustrates the frame size sequence of compressed frames. It plots the size sequence of the first 500 frames. Simply from eyeball test, we can observe that the traffic exhibits very bursty characteristics. It is worth noting that our VBR bandwidth process is not linearly proportional to frame rate. For example, increasing the frame rate by 50% from 4 frames/sec to 6 frames/sec entails the increase in the playback rate by 18% from 4.0 Kbyte/sec to 4.7 Kbyte/sec. This is because the encoder exploits the inter frame dependency in compressing the original scene. The relationship between the frame rate and playback rate and their respective impact of QoS is subject to further investigation.

2.2 Creation of Stream File

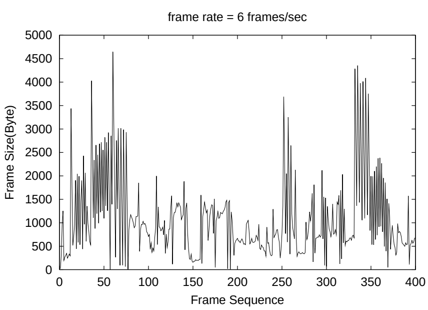
From technical point of view, streaming of video file should be distinguished from the playback of multimedia data from local storage. In local playback, the unit of data transfer is disk block which is usually 4 KByte(or multiples thereof). The components which constitutes the route from the local storage to the display in multimedia playback, e.g. disk interface, I/O bus, memory, system bus, decoder, CPU, video card, etc. has sufficient data rate and capacity. Thus, violation of timing constraints for a single frame can easily be identified and can be recovered. Further this process occur in single address domain. However, transporting the multimedia data over the network requires more elaborate mechanism. I/O unit size is much smaller. Maximum packet size is determined by MAC layer and usually is 1500 Byte(Ethernet MAC). This value includes the size of header and trailer and data payload. Further, the end systems(server and client) does have any control on jitter, delay, loss which occurs somewhere in the middle of transportation. In an effort to partly compensate this uncertainty, each packet is enhanced with information about the packet sequence number, arrival deadline, type of information, size of data payload, etc[15]. Adding this information to individual packets on the fly requires excessive CPU cycles. The MPEG-4 standard file format(`*.mp4`) suggests that the file contains the array of elements called *hint track* where each element contains the size of packet, deadline, decoding deadline, etc.



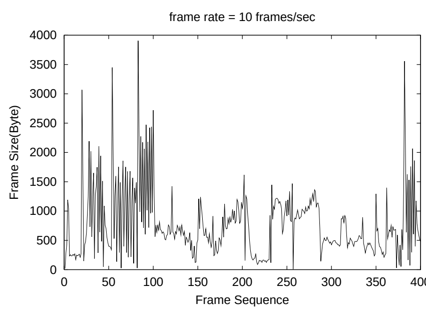
(a) 4 frames/sec, average playback rate = 4.0 KByte/sec, Peak Rate = 5.79 KByte/sec, $\sigma^2 = 2534$



(b) 5 frames/sec, average playback rate = 4.3 KByte/sec, Peak Rate = 7.25 KByte/sec$\sigma^2 = 4215$



(c) 6 frames/sec, average playback rate = 4.7 KByte/sec, Peak Rate = 7.21 KByte/sec, $\sigma^2 = 3853$



(d) 10 frames/sec, average playback rate = 6.1 KByte/sec, Peak Rate = 14.4 Kbyte/sec, $\sigma^2 = 1182$

Fig. 2. Basic Statistics of Compressed Streams

Creating the *streamable* MPEG-4 file involves a series of conversion. This is not elaborate nor state of art technology at all. However, we like to present the process briefly mainly to help the understandings. Original video clip contains audio and video information. Each information is compressed by the respective encoder, i.e. audio codec and video codec. In case of video, original YUV signals are compressed with divx codec and `*.cmp` file is created as a result. In case of audio file, it is compressed using audio codec, e.g. G.723 and the respective `*.cmp` file is created. `*.cmp` files for video and audio are then multiplexed into single file `*.mp4` file. MP4 file format is rooted at Quicktime file format[10]. MP4 file format introduces additional data structures(atom) which further faciliates the manipulation of MPEG-4 compressed data. They include the atom for copyright information, object descriptor, track information, etc. In the last step, `mp4` file is enhanced with packetization information as known as *hint track* and is converted into `mov` format file.

3 Smoothing of Empirical Process

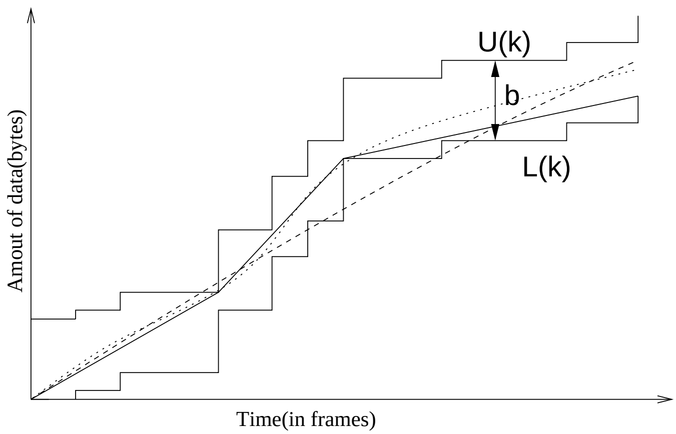


Fig. 3. Smoothing: Transmission schedule should lie between $L(k)$ and $U(k)$ while minimizing the burstiness of the traffic

A compressed video stream consists of n frames, where frame i requires f_i byte of storage. To avoid the underflow of the data in the client buffer, the server alway transmit enough data by k^{th} frame, $L(k) = \sum_{i=1}^{k} f_i$. However, since the client buffer size is b, the client should not receive more data than $U(k) = L(k) + b$ by frame k. Let $c_i, i = 1, \ldots, N$ be the transmission rate during frame slot i of the smoothed video stream. Then, any valid transmission plan should satisfy that $L(k) \leq \sum_{i=1}^{k} c_i \leq U(k)$. The objective of smoothing is to find c_i which minimizes burstiness of the traffic while satisfying the continuity requirement. Fig. 3 illustrates the various schedule and upper and lower bound of the schedule. Any valid transmission plan should lie between $L(k)$ and $U(k)$.

The smoothing algorithm generates different schedule depending on the optimization criteria and thus appropriate smoothing algorithm needs to be carefully chosen depending on the characteristics of the system. For example, when the client has small size buffer, the optimization should focus on minimizing the client buffer utilization. In the current version of our streaming system, we find that the packet loss mostly occurs at the client's end. In our system, the typical situation of packet loss is that buffer space for UDP socket is full and decoding process is not fast enough to decode the frames and to make the room for newly arriving packets in the decoding buffer. This situation is more likely to happen when the packets arrive at bursty manner. In an effort to minimize the packet losses, we generate the schedule which minimizes the variability of the packet arrival rate, $\sum_{i=1}^{N}(c_i - \overline{c})^2$, $\overline{c} = \frac{\sum_{i=1}^{N} c_i}{N}$, with the given buffer size. This algorithm is originally developed by Salehi et al[17]. The objective of our work is to perform empirical study on the effectiveness of VBR smoothing in real testbed and thus we like to omit the detailed description of the algorithm itself.

Table 1. Characteristic of Smoothen Traffic

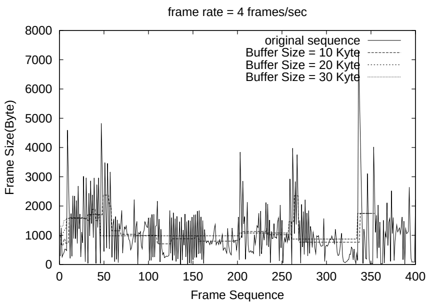
Frame Rate	r_{frame}	4 fps	5fps	6fps	10fps
Original Traffic	μ	4.0	4.3	4.7	6.1
	σ^2	2534	4215	3853	1182
	Peak	5.79	7.25	7.21	14.4
Buffer Size = 10KByte	σ^2	1178	1882	1791	688
	r_{peak}	5.2	7.0	6.8	12.2
Buffer Size = 20KByte	σ^2	842	989	661	412
	r_{peak}	4.6	6.8	5.8	10.1
Buffer Size = 30KByte	σ^2	718	524	459	342
	r_{peak}	4.2	4.9	4.9	7.2

Table 1 illustrates the statistical characteristics of the empirical processes. For each of the original video clips, we generate three different transmission schedules based on different client buffer sizes: 10KByte, 20KByte, and 30KByte, respectively. The buffer size used for smoothing is very small compared to the buffer size used in the preceding works which typically ranges from 64KByte to 32MByte. The smoothing buffer size bears direct relationship to service startup latency. In this work, we assume that longer than 5 sec's startup latency is not acceptable and we select the appropriate smoothing buffer size given this maximum startup latency constraint. To be described in detail in section 4, benefits of smoothing with these small size buffers are quite phenomenal. r_{frame} and r_{peak} in Table 1 denotes frame rate(frames/sec) and peak data rate(Byte/sec), respectively. As can be seen, adoption of smoothing algorithm reduces the variance(σ^2) of the VBR sequence. Variance decreases with respect to increase in the size of smoothing buffer. Fig. 4 visualizes the VBR traffic sequences under various smoothing conditions.

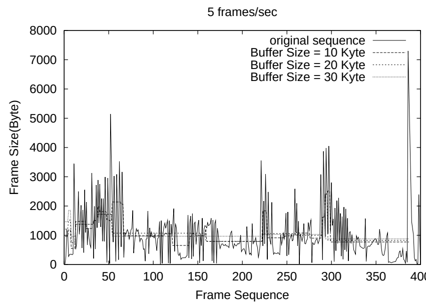
4 Experiment

4.1 Environment Setup

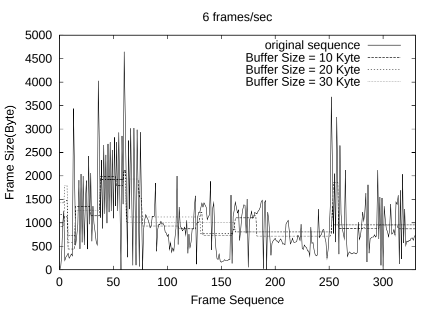
Development of comprehensive streaming system is itself rather challenging task, which took us two years of effort. Our MPEG-4 streaming system, *SMART* is developed on Linux environment and runs on Linux box with dual pentium II(550MHz) processors. The client application(MPEG-4 player) is developed on WinCE platform(iPAQ with 64 MByte of main memory)[11]. The client connects to the server via 10 Mbits/sec wireless LAN connection. Control informa-



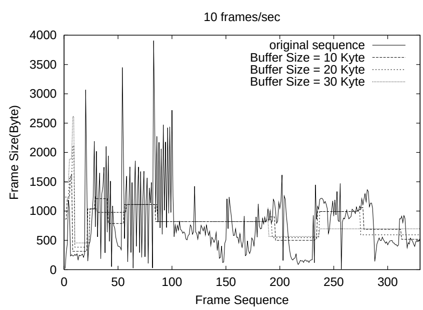
(a) 4 frames/sec, average playback rate = 4.0 KByte/sec, Peak Rate = 5.79 KByte/sec, $\sigma^2 = 2534$



(b) 5 frames/sec, average playback rate = 4.3 KByte/sec, Peak Rate = 7.25 KByte/sec$\sigma^2 = 4215$



(c) 6 frames/sec, average playback rate = 4.7 KByte/sec, Peak Rate = 7.21 KByte/sec, $\sigma^2 = 3853$



(d) 10 frames/sec, average playback rate = 6.1 KByte/sec, Peak Rate = 14.4 Kbyte/sec, $\sigma^2 = 1182$

Fig. 4. Original VBR sequence and the smoothen traffic

tions, e.g. `open`, `play`, `pause`, `stop`, `close`, are transferred over RTSP protocol. Streaming data is transported using RTP protocol implemented over UDP. Original video clip is 6 min long. Basic statistics on the underlying processes are shown in Table 1.

When transmitting the MPEG-4 packet over RTP, we can specify the required packet transmission time in RTP packet header. The actual value in this field corresponds to the offset from the original frame display time. If the transmission time fields of all packets are 0, the packets belonging to the same frame are transmitted in bursty manner as if the frame is the unit of transmission. By properly adjusting this value, we can distribute the packet transmission over the time line and can make the resulting packet traffic *smoother*. Packetization information along with the respective packet transmission timing is recorded in the hint track of the file. File format and hint track structure are developed compliant with[19].

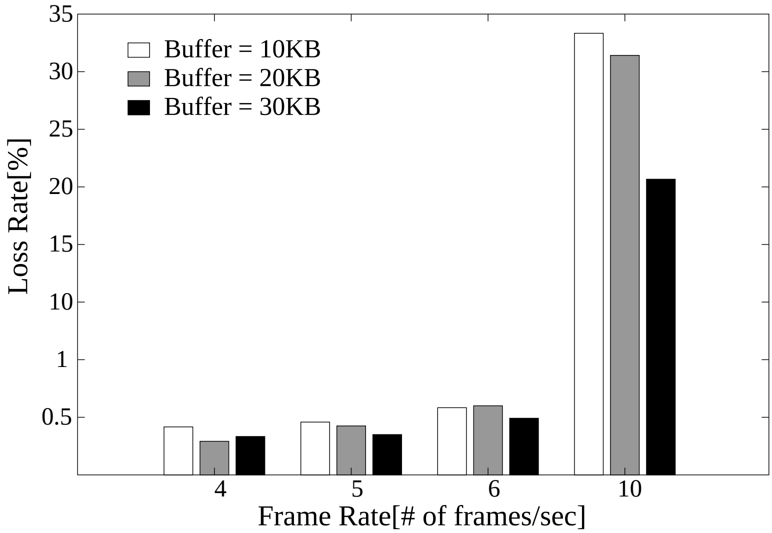


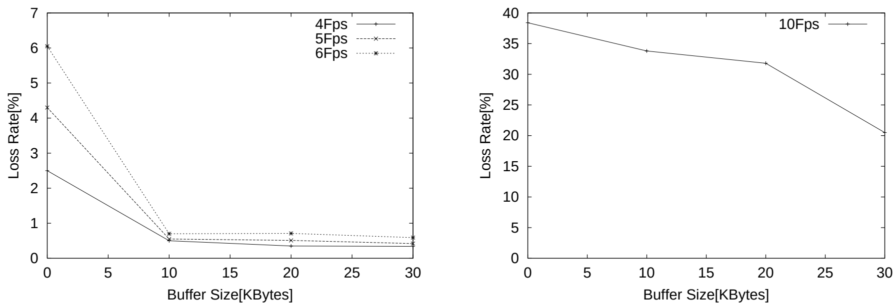
Fig. 5. Frame Rate vs. Packet Loss

Average playback of each file corresponds to 4.0KByte/s, 4.3KByte/s, 4.7KByte/s and 6.1KByte/s, respectively. RTP packetization and the packet transmission time information is attached to this file. For each frame rate, we generate three files each of which has different packet transmission schedule. We use three different buffer sizes: 10 KByte, 20 KByte, and 30 KByte for smoothing.

4.2 Result of Experiment

Fig. 5 illustrates the packet loss behavior of the streaming session. X-axis and Y-axis denote the frame rate and packet loss probability, respectively. When the frame rate is relatively low, i.e. upto 6 frames/sec, smoothing the traffic brings rather significant improvement on packet loss even though the smoothing buffer size is very small, e.g. 10 KByte. However, introducing larger size buffer beyond 10 KByte does not entail profitable improvement in the packet loss behavior. Meanwhile, in the stream with higher playback rate(10 frames/sec), using larger size buffer for smoothing continuously improves the packet behavior.

Fig. 6 illustrates the packet loss behavior under different buffer size. Smoothing the original stream with 10 KByte buffer size dramatically decreases the packet loss behavior. Especially for 6 frame/sec stream, the packet loss probability drops from 7% to 0.5%. This is phenomenal leap from practical point of view. When packet loss probability is 7%, the quality of the scene is *not acceptable* for service. However, when the packet loss is 0.5%, we are actually not able to recognize any frame corruption nor jitter in playback.



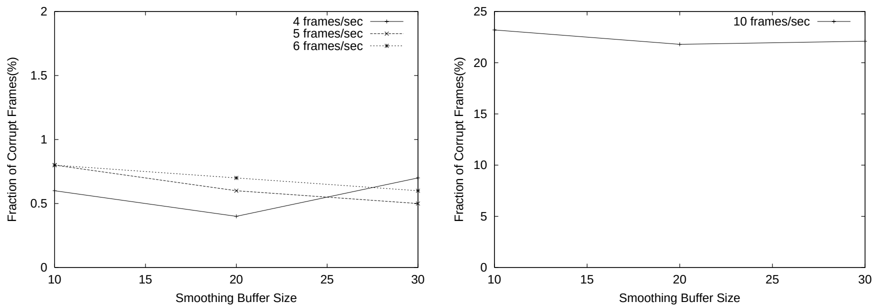
(a) 4,5,6 frames/sec (b) 10 frames/sec

Fig. 6. Smoothing Buffer Size vs. Packet Loss Probability

Fig. 6(b) illustrates the packet loss behavior under 10 frame/sec stream. Without smoothing, approximately 38% of the packets are lost. Using 30KByte buffer size, packet loss probability drops down to 21%. This improvement seems far greater than what we achieved in 6 frames/sec stream through smoothing: decrease of packet loss from 6% to 0.7%. Interestingly, however, we are not able to recognize any improvement on quality of stream in case of 10 frames/sec stream. With or without smoothing, the quality of the stream is far from what can be accepted with reasonable tolerance. We found that in 10 frames/sec stream, the number of corrupt frames, i.e. the frame one of whose constituents is lost, remains almost the same even with smoothing. Refer to the figures in Fig. 7. They illustrates the fraction of corruption frame with different smoothing buffer size, 10 KByte, 20 KByte, and 30 KByte, respectively. Frame is said to be corrupt if one or more of its packets are missing. With 4 to 6 frames/sec playback, less than 1 % of the frames are corrupt. However, in 10 frames/sec playback, more than 20% of the frames are corrupt even with the smoothing. This may suggest that human perception behavior is actually more vulnerable to the frame corruption than to the packet loss.

5 Conclusion

This work presents the result of our study on VBR smoothing in broadband wireless network. A number of elaborate smoothing techniques have been proposed in various public forums and literatures. Each of these techniques has different assumption and smoothing criteria. These algorithms successfully removed burstiness in the original empirical process. However, their works leave much to be desired to obtain practical implications of smoothing on the user perceivable QoS. There are two important points which need further attention in the area of traffic smoothing. The first one is the impact of smoothing on packet loss and jitter behavior. This work requires sophisticated modeling the



(a) Frame Rate = 4,5,6 frames/sec (b) Frame Rate = 10 frames/sec

Fig. 7. Frame Corruption Behavior in %

actual system. The second one is the relationship between the packet loss and jitter and QoS perceived by the end user. Even though we discard the fact the human perception is by nature subjective, same packet loss and jitter can affect the QoS in many different ways depending on its distribution as well as the frame type of the respective packet. Thus, it is by no means trivial task to investigate the benefit of smoothing from the perspective of user perceivable QoS. In this work, we like to address both of these issues. We develop MPEG-4 streaming suite and embed the smoothing algorithm in the transport layer. We examine the packet loss behavior with respect to different frame rates and different smoothing buffer sizes. We also examine the quality of each scene with different frame rates and different smoothing buffer size. We use the rate variability as the metric for optimization. With smoothing, we were able to increase the acceptable quality frame rate by 50%(10% in bandwidth). Also, the experimental results suggest that human perception *may* be more vulnerable to frame corruption behavior than the packet loss behavior. Novelty of our work lies in the fact that our work present the benefit of smoothing in actual broadband wireless Internet environment. This study cannot be possible without rigorous system modeling and comprehensive system development and implementation.

References

1. R.T. Apteker, J.A. Kisimov, and H. Neishlos. Video acceptability and frame rate. *IEEE Multimedia*, 2(3):32–40, Fall 1995.
2. Jean-Yves Boudec and Olivier Verscheure. Optimal smoothing for guaranteed service. *IEEE/ACM Trans. on Networking*, 8(6):689–696, 2000.
3. Ray-I Chang. Dynamic window-based traffic-smoothing for optimal delivery of online vbr media streams. pages 127–134, 2000.
4. M Dinis and J. Fernandes. Provision of sufficient transmission capacity for broadband mobile multimedia: A step toward 4g. *IEEE Communications Magazine*, 39(8):46–54, Aug. 2001.

5. W. Feng. Rate-constrained bandwidth smoothing for the delivery of stored video. In *In Proceedings of IS&T/SPIE Multimedia Networking and Computing*, pages 58–66, Feb. 1997.
6. W. Feng, F. Jahanian, and S. Sechrest. Optimal buffering for the delivery of compressed prerecorded video. *ACM Multimedia Systems Journal*, pages 297–309, 1997.
7. Wu-Chi Feng and Jennifer Rexford. Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video. *IEEE Trans. on Multimedia*, 1(3):302–313, September 1999.
8. M. Garrett and W. Willinger. Analysis, modeling and generation of self-similar vbr video traffic. In *Proceedings of SIGCOMM 94*, pages 269–280, 1994.
9. G. Ghinea and J.P. Thomas. Qos impact on user perceptoin and understanding of multimedia video clips. In *Proceedings of ACM Multimedia*, Bristol, U.K., 1998.
10. `http://www.quicktime.com`.
11. `http://www.reakosys.com`.
12. M. Krunz and S. Tripathi. On the characterization of vbr mpeg streams. In *Proceedings of SIGMETRICS '97*, pages 192–202, 1997.
13. P. Manzoni, P Cremonesi, and G. Serazzi. Workload models of vbr video traffic and their use in resource allocation policies. *IEEE Trans. on Networking*, 7(3):387–397, June 1999.
14. J. M. McManus and K. W. Ross. Video on demand over atm: Constant-rate transmission and transport. *Telecommunication Systems*, 9, 1998.
15. A framework for the delivery of mpeg-4 over ip-based protocols. Technical Report draft-singer-mpeg4-ip-03, Audio-Video Transport WG, Internet Draft, July 2001.
16. K. Ryu and A. Elwalid. "the importance of long-range dependence of vbr video traffic in atm traffic engineering: Myths and realities". *ACM Computer Communication Review*, 26:3–14, October 1996.
17. J. Salehi, Z. Zhang, J. Kurose, and D. Towsley. Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing. In *Proceedings of ACM SIGMETRICS*, Philadelphia, PA, May 1996.
18. R. Steinmetz. Human perception of jitter and media synchronization. *IEEE J. Selected Areas in Communication*, 14:61–72, Jan. 1996.
19. Y. Matsui T. Nomura and H. Kimata. RTP Payload Format for MPEG-4 Audio/Visual Streams. In *ISO/IEC*, November 2000.
20. Youjip Won and Jongwoo Jeon Soohan Ahn. Performance analysis of nonstationary model for empirical vbr process. In *Proceedings of IEEE Globecom '01*, St. Antonion, TX, USA, Nov. 2001.
21. J. Zhang and J. Hui. Traffic characteristics and smoothness criteria in vbr video transmission. In *In Proceedings of IEEE Int. Conf. Multimedia Computing and Systems*, June 1997.