광대역 무선 환경에서 VBR 트래픽 스무딩에 대한 연구

심보욱* 원유집 송승호
한양대학교
(shim77*, yjwon, seungho)@ece.hanyang.ac.kr

Empirical Study on Smoothing of VBR Stream for Broadband Wireless Network

Bo-Wook Shim* Yu-jip Won
Dept. of Electronic Engineering, Hanyang University

Abstract

In this work, we like to present the result of study obtained from actual experiment. We instrument the effect of smoothing on packet loss behavior in mobile terminal on broadband wireless network under various different system settings. This activity requires comprehensive streaming software suite including streaming server, CODEC, and streaming client.

1. 서 론

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. How to cope with the burstiness of the underlying VBR traffic plays critical role in determining the efficiency of the system including hardware and software. In an effort to mitigate the burstiness of the traffic, a number of bandwidth smoothing algorithms for the delivery of compressed prerecorded video have been proposed in various public forums and literatures.

Each of these algorithms has its own assumption and optimization criterion in minimizing the burstiness of the traffic. These techniques effectively remove the burstiness required for the playback of continuous video. Unfortunately, however, none of these works address how the smoothing algorithm can contribute to improve the Quality of Service in practical environment. In this work, we present the lessons learned from implementing the smoothing algorithm in our streaming solution suite, which consists of streaming server and the client. Our system is designated to run on broadband wireless Internet environment. The combination of broadband wireless Internet and mobile terminal yields rather unique combination different from the desk top PC wired with high speed Internet connection. It is found that processing capability of the client actually is the limiting factor of QoS in our environment. The contribution of this work lies in the fact that we present the experience obtained from the actual implementation of smoothing technique in real system. By incorporating smoothing technique in transporting VBR stream, we are able to increase the frame rate by 50₩%. Interestingly, we find that the improvement in QoS is much more dependent upon the frame rate (frames/sec) than upon playback

2. System Description

We adopt existing framework of *Darwin streaming Server[6]*. The client application, *Reako player[7]* is developed for local and remote online playback of MPEG-4 simple profile format data. Reako Player is developed on Windows CE 3.0 platform(iPAQ, 32MByte RAM). Streaming server runs on PIII 500 MHz PC with 128 MByte of memory. The server machine is loaded with Windows NT server 4.0. Streaming server and Reako Player communicates via 10Mbps wireless LAN connection.

3. VBR Traffic

From the original video clip, we generated 4 stream of different playback rates and frame rates.

Table 1 summarizes the traffic characteristics. Each clip is encoded with MPEG 4(DIVX) codec.

Frame Rate	4 fps	5fps	6fps	10fps
Playback Rate (Kbytes/sec)				
μ	4.0	4.3	4.7	6.1
σ^2	2534	4215	3853	1182
Peak	5.79	7.25	7.21	14.4

TABLE 1 Original Data Characteristic

Creating the streamable mpeg 4 files involves a series of conversion. This is neither elaborate nor state of art technology at all. In fact, it turns out to be quite labor-intensive process. However, we like to explain it briefly to help the understandings. Original video clip contains audio and video information. The respective encoder, i.e. audio codec and video codec, compresses each information. In case of video, original YUV signals are compressed with *divx* codec and **.cmp* file is created as a result. **.cmp* files for video and audio are then multiplexed into single file **.mp4* file. Finally, *mp4* file is enhanced with packetization information as known as hint track and is converted into *mov* format file.

4. Traffic Smoothing

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 always transmit enough data, $L(k)=\sum_{i=1}^{k} f_i$. 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 be the transmission rate during frame slot i of the smoothed video stream. Then, any valid transmission plan should satisfy that $L(k)\le\sum_{i=1}^{k} c_i \le U(k)$. There are a number of smoothing algorithms each of which uses different optimization criteria and each of which generates different schedule. Depending on the characteristics of the system, different smoothing algorithm needs to be used. For example, when the client has small size buffer, the optimization should focus on minimizing the client buffer utilization. We found that in our environment where mobile device is connected to server via broadband wireless network, we found that transport layer of end terminal is bottleneck point. This may not apply when available bandwidth of the underlying network is relatively low, as in IMT-2000,or CDMA 2000 environment, however. In classic queuing system principle, average queue length is proportional to variance of the inter-arrival time and job loss probability is obtained from the tail behavior of the queue length distribution. In this regard, we focus on minimizing the variability of packet arrival rate with the given buffer size. Table 2 illustrate the statistical characteristics of the smoothen data. For each of the original video clips, we generate three different transmission schedule based on different client buffer sizes: 10Byte, 20 KByte, and 30KByte, respectively. RTP packetization information along with the respective packet transmission timing is recorded in the hint track of the file. r_{peak} and r_{frame} in Table 2 denotes frame rate(frames/sec) and peak data rate(Byte/sec), respectively.

Buffer	r_{frame}	4fps	5fps	6fps	10fps
10Kbyte	σ^2	1178	1182	1791	688
	r_{peak}	5.2	7.0	6.8	12.2
20Kbyte	σ^2	842	989	661	412
	r_{peak}	4.6	6.8	5.8	10.1
30Kbyte	σ^2	718	524	459	342
	r_{peak}	4.2	4.9	4.9	7.2

TABLE 2 Characteristic of Smoothen Traffic

5. Experiment

Original video clip is 6 minute long. We generate four mp4 files with different frame rates, 4,5,6 and 10, respectively. 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. This packetization related information is called hint track. File format and hint track structure is developed to be compliant with *RFC3016[11]*.

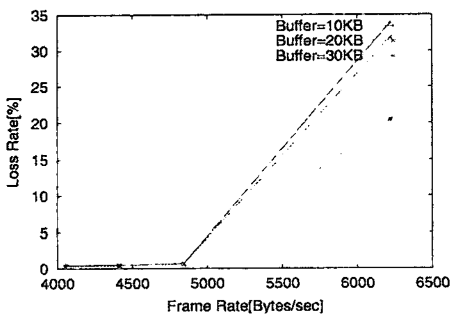

Fig 3 Frame Rate vs. Packet loss

For each frame rate, we generate 3 files each of which has different packet transmission schedule. We use three different buffer sizes: 10 KByte, 20KByte, and 30 KByte for smoothing. Fig.1 illustrates the packet loss behavior under different frame rates. X-axis and Y-axis denotes playback rate and packet loss probability. As can be seen for lower rate playbacks, i.e. 4.0 KByte/sec, 4.3 KByte/sec and the packet loss. On the other hand, for 6.1 KByte/sec playback(10frame/sec), increasing buffer size improves packet loss probability significantly.

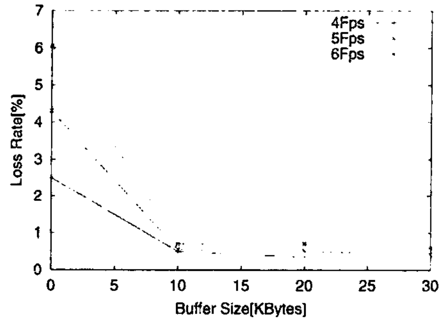

Fig 1 Buffer Size vs. Packet Loss

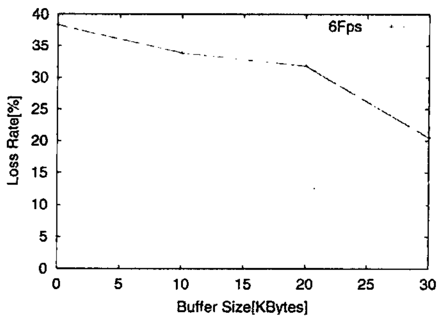

Fig 2 Buffer Size vs. Packet Loss

Fig.2 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 4.7 KByte/sec, increasing buffer size beyond 10 KByte does not bring significant improvement on 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. Fig.3 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 observed in 6 frames/sec stream: 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. This may suggest that human perception behavior is actually more vulnerable to the frame corruption than to the packet loss.

6. Conclusion

This work presents the result of our study on VBR smoothing in broadband wireless network. We implement the smoothing algorithm in our streaming software suite and examine the packet loss behavior under various different system settings. 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).

7.Reference

[1]Jean-Yves Boudec and Oliver Verscheure. Optimal smoothing for guaranteed service. /IEEE/ACM Trans. On Networking, 8(6):689-696, 2000

[2]Ray-I Chang. Dynamic window-based traffic-smoothing for optimal delivery of online vbr media streams. Pages 127-134, 2000

[3]W.Feng. Rate-constrained bandwidth smoothing for the delivery of stored video. In proceeding of IS&T/SPIE Multimedia networking and Computing, pages 127-134, 2000

[4]W.Feng, F.jahanian, and S.Sechrest. Optimal buffering for the delivery of compressed prerecorded video. ACM Multimedia Systems Journal, pages 297-309, 1997

[5]Wu-Chi Feng and JenniferRexford. Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video. IEEE Trans. on Multimedia, 1(3):302–313, September 1999.

[6]http://www.quicktime.com

[7]http://Reakosys.com

[8]Leonard Kleinrock. Queuing Systems. John Wiley & Sons, Inc., 1975.

[9]J.M. McManus and K.W. Ross. Video on demand over ATM: Constant-rate transmission and transport. Telecommunication Systems, 9, 1998.

[10]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.

[11] Y.~Matsui T.~Nomura and H.~Kimata. RTP Payload Format for MPEG-4 Audio/Visual Streams. In ISO/IEC, November 2000.