QoS Weighted Scheduling: Real-time Streaming of Multi-resolution Video

***Abstract*—**

In this work, we propose novel packet scheduling algorithm for real-time streaming of multi-resolution video. Our scheduling algorithm targets towards the situation where there exists relatively large fluctuation in bandwidth availability and the short queue depth in the receiver's end. The proposed algorithm determines *which packets to send* and *when to send them*. We develop *QoS significance* metric to represent the importance of a packet. QoS significance incorporates the composite transitive dependency in multi-resolution video. We develop *QoS weighted traffic smoothing* to determine the transmission schedule. We found that when the receiver has short queue length, e.g. mobile terminal, packet loss is very sensitive to the packet interval distribution. In determining the packet transmission schedule, we consider not only the bandwidth process of the packet traffic but also the importance of the individual packets. We introduce more slack for more important packet. We found that QoS weighted traffic smoothing makes the resulting bandwidth process burstier and entails more packet losses. However, it greatly enhances the user perceivable QoS. The simulation results show that the our scheme not only maximizes user perceivable QoS but also minimizes resource requirements.

Key Words: Multi-Resolution, Video Streaming, QoS, Packet Scheduling, Traffic Smoothing

I. Introduction

A. Motivation

Layered encoding has been proposed to cope with bandwidth variation in small and large time scale. It partitions the original content along the temporal, spatial and PSNR axis so that the video source, e.g. streaming server, can adaptively adjust the playback(or transmission) rate subject to the resource availability. Client reconstructs the original imagery from the partial information. The quality of reconstructed information is subject to the amount of information transmitted. The term resource here can be network bandwidth, CPU computing power, main memory buffer, disk bandwidth and etc. State of the art video compression technique exploits information redundancy in the original scene. Compressed video is a sequence of discrete scenes(frame). There are three types of frame: I, P and B. I frame is intra-coded frame. P frame carries difference between its preceding and succeeding I or P frames. B frame carries interpolated information between adjacent I or P frames. Dependent upon the *type* of the lost information(I, P or B).

There are two important issues in transporting real-time multimedia stream: *which packets to send* and *when to send them*. When the available bandwidth is much smaller than the playback rate of the file, we need to select proper subset of packets to maximize user perceivable QoS. If all packets are transmitted, it not only causes the subnet congestion but also user perceivable QoS significantly degrades due to random losses of the packets. The key ingredient here is to assign *importance* metric to each packet and to select packets based upon the metric. In addition to packet selection, determining the packet transmission time is the other important axis in multi-resolution video streaming. Video traffic smoothing was proposed to reduce the burstiness of the traffic [14], [15]. It adjusts either size of the packets or the transmission interval so that the bandwidth process(byte count) of the video stream becomes smooth. In our empirical study [16], we found that when client's queue capacity is small(as in mobile terminal), packet loss probability is very sensitive to the interval to its preceding packet.

In this work, we propose novel packet scheduling scheme for real-time streaming of multi-resolution video: *QoS Weighted Packet Scheduling*. We develop *QoS significance* metric to determine the packet importance. We use this to select the packets for transmission. Packet transmission interval is determined based upon the size as well as the importance of a packet.

B. Related works

There have been a number of efforts on devising a packet transmission schedule to maximize QoS as well as to minimize packet loss and jitter. The first efforts on this category are to minimize the burstiness on the underlying traffic [3], [8], [14], [15]. While these works try to minimize the burstiness on the traffic via properly controlling packet transmission interval, each of them has different criteria about "burstiness", e.g. minimize bandwidth requirements [3], number of changes [8], bandwidth variability [14], prefetching delay [15], and etc. These works focus on reducing the variability to minimize packet loss. There is an important assumption in these works. First, they assume perfect network, which means that the interval between two packets are preserved from the sender to the destination. However, best effort modern Internet infrastructure does not guarantee the interval. Second, they did not consider the "importance" of each packet. It is true that packet loss adversely affect the QoS. However, the significance of packet loss varies widely depending upon which frame type it belongs to. According to our experimental result, user perceivable QoS actually improves via increasing the packet loss.

Multi-resolution video has been proposed to dynamically adjust the playback bandwidth dependent upon the bandwidth availability. MPEG supports many multi-resolution encoding schemes, such as temporal, spatial, and SNR methods [1].

In Fine Granularity Scalability(FGS) encoding scheme, the enhancement layer can consist of any number of bits [9]. It was adopted by MPEG-4 standard and supports fully adaptable network bandwidth availability using multi-resolution encoding methods. Nelakuditi et. al. suggested to reduce playback quality variation by using run length [13]. Cuetos et. al. also suggested multi-resolution video streaming algorithm maximizing bandwidth efficiency and minimizing rate variation by dynamic programming approach [5]. Kim et. al. suggested minimum quality variability based on aggregate layers [7]. Jiang et. al. suggest packet selection algorithm based on success probability [6]. Liu et. al. proposed the strategies of dynamic layering functions [10]. Chou et. al. proposed rate-distortion sense packet scheduling algorithm [4]. They suggested packet level dependency graphs of multi-resolution encoded media. Miao et. al. suggested packet scheduling algorithms based on expected run-time distortion value [12]. Their algorithm give us improvement of playback quality by layer ordering. Their algorithm has lacks of packet loss effects according to packet scheduling.

Boyce [2] analyzed Internet packet loss behavior as frame type. Loguinov et. al. [11] measured packet loss rate, round-trip time, and packet reordering over dial-up networks.

The existing efforts on real-time retrieval of video focused on one of the following issues: smoothing, error concealment, and network bit rate allocation. In practice, all these factors are tightly coupled with each other and should be dealt with in a single context. In this work, we focus our effort on developing packet scheduling algorithm which exploits the available network bandwidth, importance of individual packets, and queuing behavior at the client's end. We address the problem of selecting subset of layers for transmission and assignment of inter-packet interval. The main contribution of our work is to propose a simple and unified scheduling algorithms supporting both minimum distortion and proper packet transmission interval.

The remainder of the paper is organized as follows. Section II introduces basics of packet scheduling and problem formulation. Section III describes the details of the layer selection and packet scheduling algorithm. Section IV carries the result of the performance experiment. We conclude our work in section V.

II. Problem Formulation

A. Myths and Realities in Packet Scheduling

In our previous work, we examined the packet loss and QoS behavior of several traffic smoothing algorithms under various settings [16]. As a result of smoothing, we observe significant improvement on user perceivable QoS. After being smoothen, the packet intervals become proportional to the size of the packets. Fig. 1 illustrates the effect of traffic smoothing. In traffic smoothing, packet interval is adjusted so that the resulting *Bandwidth process* becomes less burtier than the original one. *Packet count process* of the traffic, on the other hand, becomes burstier as a result of smoothing. Fig. 2 illustrates the packet loss with and without traffic smoothing scheme in physical streaming service. *Lost packets* denotes the fraction of packets lost. Traffic smoothing actually increases the number of lost packets. *Sum of the packet sizes* denotes the fraction of data lost. However, actually the amount of lost data decreases significantly as a result of traffic smoothing. QoS weighted smoothing further improves the data loss.

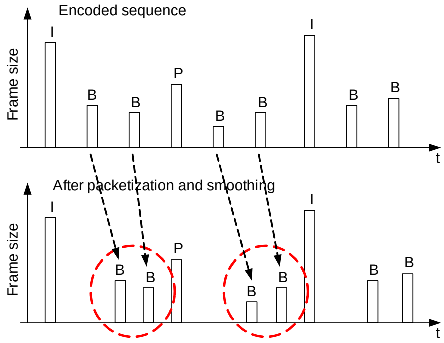


Fig. 1. Traffic Smoothing

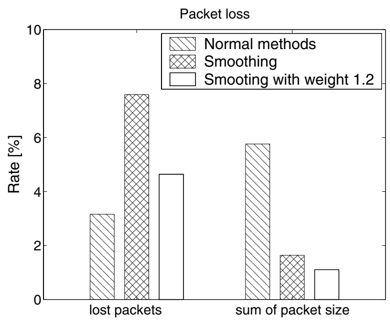


Fig. 2. Packet Loss and Data Loss with and without traffic smoothing

B. Byte count vs. Packet Count Process

There are two ways to view the time series of the network traffic: byte count and packet count. Byte count process is also often termed as bandwidth process. Byte count is the amount of bytes flown during a certain time interval. Packet count is the number of packets transmitted during a certain time interval. These two are different manifestations of the same phenomenon. Among these two metrics, traffic smoothing algorithms deal with byte count process. They aim at removing the burstiness of the bandwidth process. The practical situation is slightly different from what is expected in the literatures. Operating system maintains finite queue for UDP packets in its kernel address. This queue is specified by the *number* of packets. In modern compression technique, each type of packets has different *importance* and packet loss contributes to the user perceivable QoS in different way with respect to its importance. Loss of I frame packet will cause more damage to QoS than the loss of P or B type packets. Particularly in mobile wireless streaming, the playback rate is relatively low and the size of B Frame is much smaller than Ethernet MTU(Maximum Transfer Unit) size. Since traffic smoothing aims at minimizing rate variability of the bandwidth process,

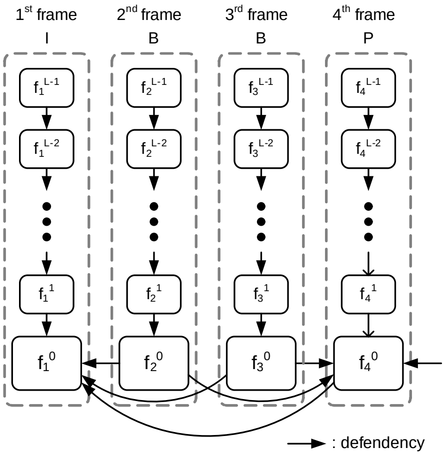


Fig. 3. Dependency of Multi-resolution video

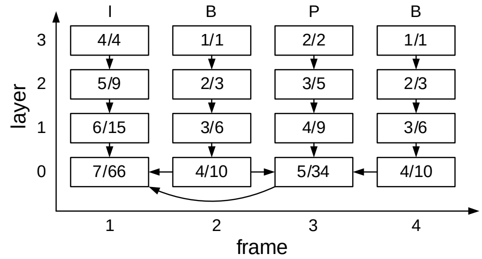


Fig. 4. Allocating Packet Significance

the transmission interval between B-frame packets become smaller as a result of smoothing. On the other hand, I-frame packet is order of magnitude larger than P or B frames. As a result of traffic smoothing, the interval between I type packet and its successor(or predecessor) becomes longer. Therefore, network traffic smoothing potentially makes the packet count process burstier. This again means that the packet loss can actually increase as a result of smoothing. Fig. 1 illustrates this situation.

C. Packet Significance

In this work, we use the notion of *QoS Significance* to represent the importance of a packet. Let us introduce some variables. G_i denotes the i^{th} GOP(Group Of Pictures) in a video content. Let f_j^i denote the j^{th} frame of i^{th} GOP. Each frame is layer encoded. $f_{j,k}^i$ denote the k^{th} layer information for frame f_j^i. Then, our packet scheduling algorithm is to find a transmission time $\mathcal{F}(f_{j,k}^i)$ for each $f_{j,k}^i \in G_i$ to maximize the user perceivable QoS. Significance of individual packet loss varies with respect to the frame type it belongs to and its position within a GOP. Unless noted otherwise, position of a frame means position within GOP. Loss of I frame packet affects all subsequent P or B frame packets within the same GOP. Loss of P frame packet becomes more significant as it is located at the earlier position within GOP. We aim at incorporating the importance of each packet in determining the packet transmission schedule.

QoS significance is a degree of importance of a packet from QoS point of view. We define a set of *parent* frames and a set of child frames for $f_{j,k}^i$. A set of parent frames for $f_{j,k}^i$, $\mathcal{P}(f_{j,k}^i)$, is a set of layers in f_j^i, i.e., $\mathcal{P}(f_{j,k}^i) = \{f_{j,l}^i \| l = 0, \ldots, j-1\}$. A set of *child* frames of $f_{j,k}^i$ is a set of layers which has $f_{j,k}^i$ as its parent, i.e. $\mathcal{C}(f_{j,k}^i) = \{f_{n,l}^m \| f_{n,l}^m \in P(f_{j,k}^i)\}$. Fig. 3 shows the dependency among the frames and layers. The notion of a set of child frames, $\mathcal{C}(f_{j,k}^i)$, provides an important ground in determining the significance of a layer, $f_{j,k}^i$. Loss of $f_{j,k}^i$ will cause the inappropriate decoding of not only $f_{j,k}^i$ itself but also all packets in $\mathcal{C}(f_{j,k}^i)$. $f_{j,k}^i(x,y)$ is a pixel value at (x,y) position of an image when a layer $f_{j,k}^i$ is properly decoded. $\widehat{f_{j,k}^i}(x,y)$ is a pixel value at (x,y) when $f_{j,k}^i$ is not properly decoded. We define *contribution* $\mathcal{D}(f_{j,k}^i)$ of $f_{j,k}^i$ in a frame f_j^i using PSNR as in Eq.1. Fig. 4 illustrates how significance value is assigned to individual layers.

$$\mathcal{D}(f_v^l) = \frac{255^2}{\frac{1}{W \times H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} |\hat{f}_v^l(x,y) - f_v^l(x,y)|^2} \quad (1)$$

where, H and W is the screen height and width. $\mathcal{D}\left(f_{j,k}^i\right)$ gives a quality metric of the layer loss. Significance of a layer $f_{j,k}^i$ is sum of all PSNR degradation which can occur due to the loss of $f_{j,k}^i$. Significance of $f_{j,k}^i$ is defined as in Eq. 2.

$$\mathcal{Q}(f_{j,k}^i) = \sum_{f_{n,m}^l \in \mathcal{C}(f_{j,k}^i)} \mathcal{D}(f_{n,m}^l) \quad (2)$$

III. QoS Weighted Packet Scheduling

A. Packet Selection

Packet selection is a process of determining the subset of packets to transmit satisfying a certain resource constraints, e.g. network bandwidth constraints. Let $\mathcal{F}(f_{j,k}^i)$ be the transmission timing of $f_{j,k}^i$. Transmission timing is the interval from the immediately preceding packet. $\mathcal{F}(f_{j,k}^i) = \infty$ when it is not selected for transmission. Let $\mathcal{S}(f_{j,k}^i)$ be the size of $f_{j,k}^i$. We assume that we can make reasonably accurate short term prediction on bandwidth availability [14]. Short term in this context is one or two GOP's worth of period. It usually corresponds to 2 sec (15frames/sec, GOP(15,3)) or 1 sec (30 frames/sec, GOP(15,3)). We define QoS of selected packets as in Eq. 3.

$$\xi(F, G_i) = \underbrace{\sum_{\mathcal{F}(f_{j,k}^i)<\infty f_{j,k}^i \in G_i} \mathcal{D}(f_{j,k}^i)}_{A} - \underbrace{\sum_{\mathcal{F}(f_{j,k}^i)<\infty f_{j,k}^i \in G_i} \mathcal{Q}(f_{j,k}^i)}_{B} \quad (3)$$

The term A denotes the user perceivable QoS which can be obtained by F. The term B denotes the QoS degradation caused by packet loss. Packet scheduling algorithm corresponds to maximizing A and minimizing B. Let w be the scheduling window. In packet selection phase, the algorithm selects the packets to transmit satisfying a bandwidth envelop $\rho(t)$. The packet selection problem is equivalent to knapsack problem where the size and significance of $f^i_{j,k}$ corresponds to the weight and value of an item in knapsack problem, respectively. The capacity constraint of a knapsack problem is determined by the bandwidth envelop $\rho(t)$ as $\mathcal{U} = \int_{t_0}^{t_0+\omega} \rho(t)dt$. We take the greedy approach in selecting the packets to transmit. The selection criteria, $\epsilon(f^i_{j,k})$ is the weight to value ratio as in Eq. 4.

$$\epsilon(f^i_{j,k}) = \frac{\mathcal{Q}(f^i_{j,k})}{\mathcal{S}(f^i_{j,k})} \quad (4)$$

The algorithm sorts all $f^i_{j,k} \in G_i$ with respect to the decreasing order of $\epsilon(f^i_{j,k})$. For transmission, it selects $f^i_{j,k} \in G_i$ one by one from the sorted list until the sum of the size of the selected information exceeds the capacity constraint $\mathcal{U}$. $\mathcal{F}(f^i_{j,k}) = \infty$ if $f^i_{j,k}$ is not selected. If the fraction of a single layer information can be arbitrarily selected for transmission, our problem of layer selection becomes *fractional* knapsack problem. In this case, the greedy approach based upon $\epsilon(f^i_{j,k})$ metric yields the optimal solution. MPEG-4 FGS(Fine Granularity Scalability) coding allows that the enhancement layer can be truncated at arbitrary size with graceful degradation of QoS. In case of MPEG-4 FGS encoded contents, our packet selection algorithm yields optimal QoS. Then, the schedule should satisfy the bandwidth constraints $\sum_{\mathcal{F}(f^i_{j,k})<\infty} S(f^i_{j,k}) \leq \mathcal{U}$.

B. QoS Weighted Smoothing

Our packet scheduling algorithm incorporates the packet size and significance(Eq. 4). Let ω and $\delta(f^k_{i,j})$ be the length of transmission interval and transmission timing of $f^k_{i,j}$. In legacy traffic smoothing, each packet is assigned a transmission interval with respect to its size. Let $\mathcal{S}(f^k_{i,j})$ be the size of a packet. Then, $\delta(f^k_{i,j}) = \omega\frac{\mathcal{S}(f^k_{i,j})}{\sum \mathcal{S}(f^k_{i,j})}$. If all packets are of the same size, each of them will be allocated $\frac{\omega}{|S|}$, where $|S|$ denotes the number of packets. We incorporate significance of a packet in determining its interval. They key idea is to assign larger interval to more important packets. The interval for $f^i_{j,k}$, $\delta(f^i_{j,k})$ is obtained as in Eq. 5. We introduce exponential coefficient e to adjust the effect of significance in the formation of transmission timing.

$$\delta(f^i_{j,k}) = \frac{\omega \times \mathcal{S}(f^i_{j,k}) \times \left\{\mathcal{Q}(f^i_{j,k})\right\}^e}{\sum_{F(f^i_{j,k})<\infty} \mathcal{S}(f^i_{j,k}) \times \left\{\mathcal{Q}(f^i_{j,k})\right\}^e} \quad (5)$$

Single layer information, $f^i_{j,k}$ can consist of multiple packets. We evenly distribute the packets for a same layer. The interval among the packets in a layer, $f^i_{j,k}$, is computed as $\delta(f^i_{j,k}) \times \frac{MTU}{\mathcal{S}(f^i_{j,k})}$.

IV. Performance Evaluation

A. Experiment setup

We perform simulation based experiment to examine the performance of QoS weighted smoothing algorithm. Fig. 5 illustrates the topology of our experiment network. There exist eight (source, destination) node pairs. There exists single streaming session which transmits real-time streaming data over RAP/UDP. There are four UDP and three TCP sessions. Total of eight sessions share the bottleneck link. The streaming contents are simulated following MPEG-4 FGS coded data with 500 kbits/sec and 15 frames/sec playback rate. It is coded with G(15,3) format. We compare the performance and efficiency of the streaming session under four different scheduling policy. First, the streaming server transmits the packets with respect to the frame playback schedule of the original contents. Therefore, when transmitting large size frame, the outgoing traffic becomes very bursty. The second, the third and the fourth case, the streaming server determines the schedule with respect to different exponential coefficients, i.e. $e = 0.0, -1.0, -2.0$.

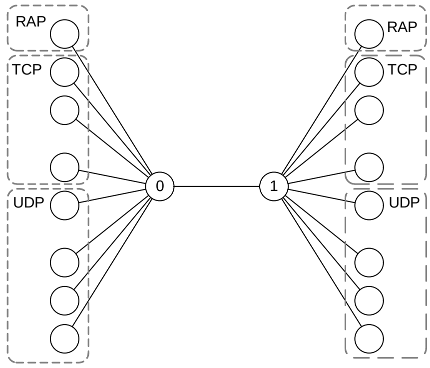
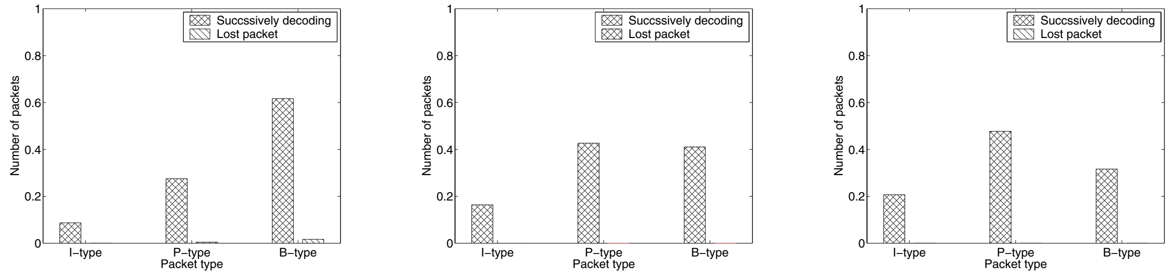


Fig. 5. Simulation parameters used in experiments

B. Results

In this experiment, we examine the percentage of each frame type in client's end. Fig. 6 illustrates the results. It illustrates the packet transmission results under three different scheduling scheme: normal smoothing, QoS weighted smoothing with $e = 0.0$, and QoS weighted smoothing with $e = -1.0$. When we do not incorporate the QoS significance, the ratio between the I, P and B type packets are approximately 1:4:9 in the receiver's end. The ratio between frames types in the original contents is preserved. As we incorporate the QoS significance in packet scheduling, we can see that the fraction of I type and P type frame increased. This is because scheduler selects the packets to send based upon its importance. Even though it is minor, normal streaming yields a few number of packet losses(Fig. 6(a). However, there is very few number of packet losses when we incorporate QoS significance in determining the interval of packets(Fig. 6(b) and Fig. 6(c)). We can also find that the ratio among frame type varies subject to *e*. With larger e, the number of large significance value packet (i.e., I-type packet) increases. On the other hand, if we choose the small number of e, the number of large significance value packet (i.e., I-type packet) decreases.

There are a number factors which govern the user perceivable QoS: packet loss, jitter and etc. The streaming server should carefully select the layer and packet transmission



(a) normal streaming (b) QoS Weighted Smoothing(e = -1.0) (c) QoS Weighted Smoothing(e = 0.0)

Fig. 6. Packet Delivery Statistics

schedule to maximize the QoS of the presentation.

Fig. 7 illustrates the correlation between the QoS and result. QoS expectation is $Q(f_v^l)$ values of transmitted layers and QoS result is $Q(f_v^l)$ values of the successfully decoded layers. Fig. 8 illustrates the bandwidth variability of a stream. As can be seen, QoS Weighted Smoothing algorithm reduces the bandwidth requirement of a stream via properly incorporating the bandwidth availability information.

V. Conclusion

In this work, we propose novel packet scheduling algorithm, QoS Weighted Smoothing algorithm. QoS Weighted Smoothing algorithm framework consists of majorly two components: packet selection and packet scheduling. Our work is motivated by the emprical study we performed in prior study. Video traffic smoothing does not necessarily improve the packet loss nor does it improve the burstiness of the traffic. However, still it greatly improves the user perceivable QoS. The reason is that the video smoothing provides better protection for more important packets. Incidentally, this is because important information, I-frame, is much larger than P or B frames. In the same token, we incorporate the notion of QoS significance in selecting the packets and in determining the packet transmission schedule. We develop QoS significance of a packet which represents the degree of transitive dependency of a packet and select the packets based upon the respective QoS significance. We apply QoS weighted traffic smoothing in determining the transmission schedule. We find that packet loss probability is very sensitive to the interval between its predecessor. This observation is particularly true in mobile streaming environment where the client has very small receiver queue depth. Via incorporating the significance of a packet, QoS Weighted Smoothing algorithm allocates more interval for more important packet. We confirm through simulation that QoS Weighted Smoothing algorithm improves user perceivable QoS, significantly.

References

[1] Information technology – coding of audio-visual objects – part 2: Visual, December 2001.

[2] Jill M. Boyce. Packet loss resilient transmission of MPEG video over the internet. *Signal Processing: Image Communication*, 15:7–24, September 1999.

[3] Ray-I Chang, Meng-Chang Chen, Jan-Ming Ho, and Ming-Tat Ko. An effective and efficient traffic smoothing scheme for delivery of online VBR media streams. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, March 1999.

[4] Philip A. Chou and Zhourong Miao. Rate-distortion optimized streaming of packetized media. Research Technical Report MSR-TR-2001-35, Microsoft Research Center, February 2001.

[5] Philippe de Cuetos and Keith W. Ross. Adaptive rate control for streaming stored Fine–Grained Scalable Video. In *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, pages 3–12. ACM Press, 2002.

[6] Zhimei Jiang and Leonard Kleinrock. A packet selection algorithm for adaptive transmission of smoothed video over a wireless channel. *J. Parallel Distrib. Comput.*, 60(4):494–509, 2000.

[7] Taehyun Kim and Mostafa H. Ammar. Optimal quality adaptation for MPEG-4 fine-grained scalable video. In *Proceedings of IEEE INFOCOM 2003*, April 2003.

[8] Simon S. Lam, SImon Chow, and David K. Y. Yau. An algorithm for lossless smoothing of MPEG video. *SIGCOMM Comput. Commun. Rev.*, 24(4):281–293, 1994.

[9] Weiping Li. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):301–317, March 2001.

[10] Jiangchuan Liu, Bo Li, Y. Thomas hou, and Imrich Chlamtac. Dynamic layering and bandwidth allocation for multi-session video broadcasting with general utiity functions. In *Proceedings of the IEEE INFOCOM 2003*, April 2003.

[11] Dmitri Loguinov and Hayder Radha. End-to-end Internet Video traffic dynamics: Statistical study and analysis. In *Proceedings of IEEE INFOCOM 2002*, July 2002.

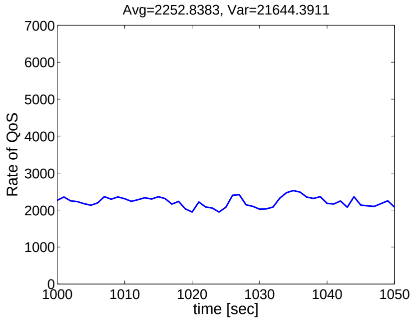
[12] Zhourong Miao and Antonio Ortega. Optimal scheduling for streaming of scalable media. In *Proceedings of 34th Asilomar Conference on Signals, Systems, and Computers*, November 2000.

[13] Srihari Nelakuditi, Raja Harinath, Ewa Kusmierek, and Zhi-Li Zhang. Providing smoother quality layered video stream. In *Proceedings of NOSSDAV 2000, The 10th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, June 2000.

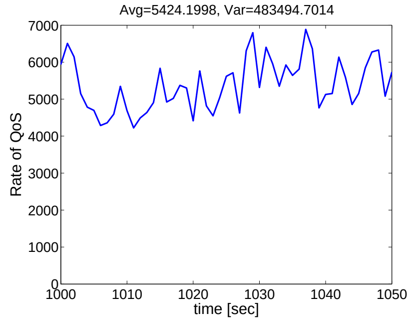
[14] James D. Salehi, Zhi-Li Zhang, Jim Kurose, and Don Towsley. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. *IEEE Transactions on Networking*, 6(4):397–410, August 1998.

[15] Olivier Verscheure, Pascal Frossard, and Jean-Yves Le Boudec. Optimal media streaming in a rate-distortion sense for guaranteed service networks. In *Fifteenth International Symposium on Mathematical Theory of Networks and Systems*, August 2002.

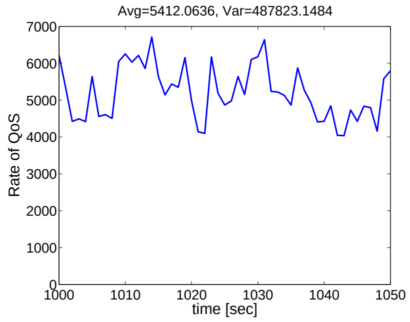
[16] Youjip Won and Bowie Shim. Empirical study of VBR traffic smoothing in wireless environment. In Herwig Unger, Thomas Böhme, and Armin R. Mikler, editors, *Innovative Internet Computing Systems : Second International Workshop*, volume 2346 of *Lecture Notes in Computer Science*, pages 193–204. Springer-Verlag Heidelberg, June 2002.



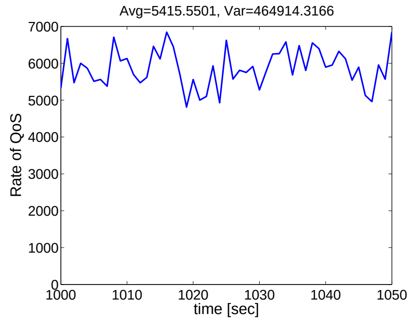
(a) normal streaming



(b) QoS Weighted Smoothing(e = -2.0)

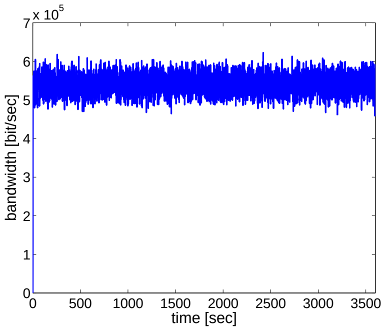


(c) QoS Weighted Smoothing(e = -1.0)

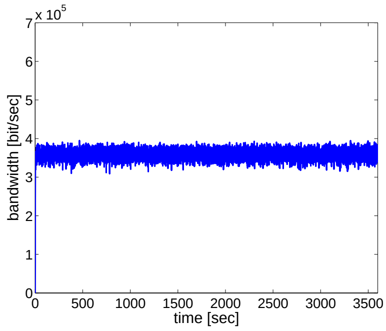


(d) QoS Weighted Smoothing(e = 0.0)

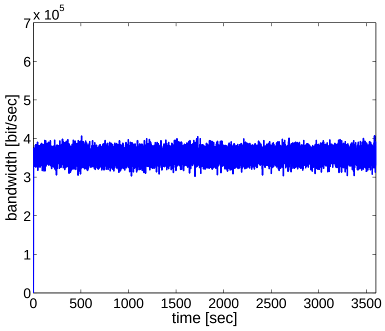
Fig. 7. $\mathcal{Q}\left(\mathrm{f}_v^l\right)$ on server and client side



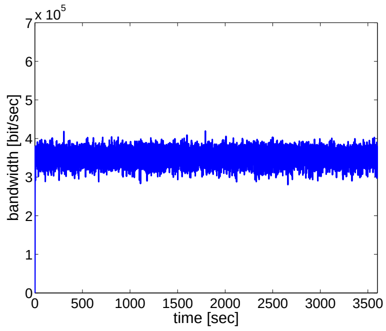
(a) normal streaming



(b) QoS Weighted Smoothing(e = -2.0)



(c) QoS Weighted Smoothing(e = -1.0)



(d) QoS Weighted Smoothing(e = 0.0)

Fig. 8. Bandwidth variability