SAPS: Significance-Aware Packet Scheduling for Real-time Streaming of Layer Encoded Video

Sungwoo Hong Youjip Won
Division of Electronics and Computer Engineering
Hanyang University Seoul, Korea zip: 133-791
{toggiya | yjwon}@ece.hanyang.ac.kr

Abstract

In this paper, we propose a novel packet scheduling algorithm for real-time streaming of layer encoded video. In real-time streaming of layer encoded video, application needs to select subset of compressed information so that it does not overflow the available network bandwidth nor limited client queue depth. We use greedy approach based upon "packet significance" in selecting packets. We develop a notion of packet significance which elaborately captures the importance of a packet over user perceivable QoS. Packet significance incorporates video semantics of a packet, i.e. I, P or B frame type, inter-layer and inter-frame dependency structure of layer encoded video. We apply packet significance weighted scheduling in determining the transmission schedule of selected packets. We perform experiment using publicly available MPEG-4 video traces. Our scheduling algorithm brings significant improvement on user perceivable QoS.

1 Introduction

1.1 Motivation

Due to the advancement of network, computer, and video compression technology, we can now enjoy bi-directional interactive multimedia service in ubiquitous fashion. With advancement in network, it brought not only the abundance in bandwidth but also the "variety" of bandwidth choices.

Real-time video streaming bears unique performance requirement: bandwidth guarantee and rate variability. The variable video frame size is realized as "bursty" network traffic. Numerous efforts have been proposed to reduce the burstiness of the real-time video traffic so that it can reduce the packet loss. This set of efforts is called traffic smoothing (or traffic shaping). To maximize user perceivable QoS, the sender needs to make right choice for two fundamental issues: "when to send?" and "what to send?". The first issue is about removing the burstiness of the traffic. The second issue is to select the subset of compressed information.

We develop a packet scheduling framework, Significance-Aware Packet Scheduling (SAPS). The contribution of our work is three folds. First, we develop a notion of packet significance which captures the QoS importance of a packet. Our scheduling framework elaborately harbors the inter-frame dependency as well as inter-layer dependency of a frame. Second, we successfully develop a unified framework for determining "when to send" and "what to send". Traffic smoothing algorithm and layer encoding scheme have been dealt with in a separate context. To properly exploit the underlying network resource and maximize user perceivable QoS, it is mandatory that these two issues are properly addressed in a single unified framework. Third, our scheduling framework incorporates not only the network's aspect of a packet but also the operating system's aspect of a packet. From network's point of view, bandwidth process is a prime concern. From operating system's point of view, however, packet count process (packets/sec) is more important, since network queue is represented by the array of packet pointers where the size of individual packet does not matter.

1.2 Related Works

Modern compression technology such as MPEG [5] achieves high compression efficiency by exploiting spatial, temporal and Signal-to-noise ratio (SNR) information redundancy. In addition, layered (scalable) encoding scheme has been suggested [2] to adapt to the varying network bandwidth availability.

To transport video packet over error-prone and resource constraint network, many packet scheduling methods have been suggested. To reduce traffic burstiness, J. Wu et al.[11]

[0]This work was supported by HY-SDR Research Center at Hanyang University, Seoul, Korea, Under the ITRC program of Ministry of Knowledge Economy, Korea

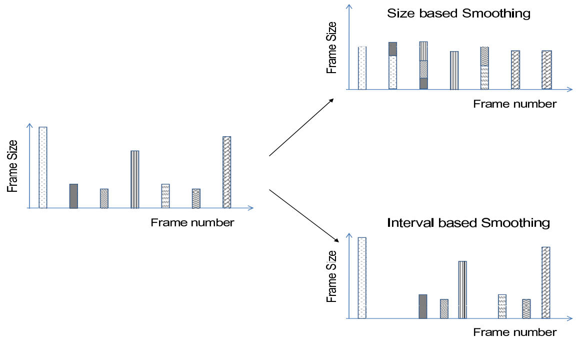


Figure 1. Size/Interval based Smoothing

suggested reducing playback quality variation by using rate smoothing and Dubois et al. [3] tried minimizing the burstiness of the network traffic with variable bit rate (VBR). In priority based packet scheduling, P. Frossard et al. [1] have proposed considering frame type and size to gauge the packet importance while L. Politis et al. [8] used MSE distortion value caused by the number of reference frames.

Remainder of the paper is organized as follows. Section 2 introduces scalable encoding and packet scheduling. Then, significance aware packet scheduling algorithm is explained at section 3. Section 4 carries the result of the performance evaluation. We conclude our work in section 5.

2 Scalable Encoding and Packet Scheduling

Scalable encoding has been proposed to selectively decode or transport a subset of original information to cope with resources variability in various physical components. Scalability is achieved via partitioning the compressed information into a number of disjoint sets. Each set is called a *layer*. Signal-to-noise ratio (SNR) scalability is a technique to code a video sequence into several layers (base layer and enhancement layers) at the same frame rate and spatial resolution, but different quantization accuracy. The bitstream of the Fine-Grained Scalable (FGS) enhancement layer can be truncated into any number of bits per picture.

In our context, the notion of "packet scheduling" consists of two ingredients: (*i*) when to transmit and (*ii*) what to transmit. Packet scheduler is required to select a certain fraction of compressed information so that it does not overflow the underlying subnet. In selecting, it is important to properly select the subset of layers so that we can maximize user perceivable QoS.

Traffic smoothing technique aims at removing the traffic burstiness so that it can make contributions on QoS via minimizing the packet losses. There are two main approaches in realizing traffic smoothing: (*i*) sized based and (*ii*) interval based smoothing. In size based smoothing, the packet scheduler controls the amount of information carried by a single packet so that size of each packet is similar. In interval based smoothing, the interval between the packets is determined based on the size of packet. Larger packet is allocated longer interval. Size based smoothing mandates that single packet can carry more than single frame. MPEG standard does not put any restriction on whether single packet contains more than one frame. In practice, however, most of the video streaming system does not allow that because loss of single packet may result in a loss of multiple frames. In addition, when single packet carries multiple frames, decoder needs to locate the boundary of individual frames for decoding. Locating the boundary of each frame can incur significant CPU overhead especially in a mobile hand held device which has a low-end CPU. Fig. 1 illustrates size and interval based smoothing. In this work, we focus on interval based smoothing approach.

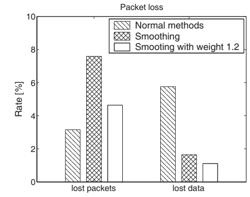


Figure 2. The Packet Loss and QoS [9]

There are two different aspects of the underlying network traffic: byte count (byte/sec) and packet count (packets/sec). Most of the existing works on traffic smoothing deal with bandwidth process. In operating systems, kernel maintains fixed length queue of packet pointers for UDP datagram. Packets reside in kernel address space and pointed by these pointers. From network queue's point of view, incoming traffic can become burstier as a result of interval based smoothing as illustrated in Fig. 1. Subsequently, packet loss can increase due to traffic smoothing [9]. However, reduction in packet loss does not necessarily imply the improvement on QoS nor improvement on PSNR. On the same token, increase in packet loss does not necessarily imply the QoS degradation. The impact of packet loss over user perceivable QoS varies dependent upon many factors such as frame type of lost packet, its position within GoP and its size. Packet scheduler needs to determine the transmission timing (or equivalently interval) so that more important packet becomes less vulnerable to packet loss. Fig. 2 illustrates the result of the physical experiment [9]. "Lost packet" denotes the ratio between the number of lost packets over total number of packets. "Lost data" is the ratio between the amount of lost data over total amount of data. Packet loss increased when interval based smoothing is applied. However, the total amount of lost data has decreased as a result of smoothing and user perceivable QoS has improved significantly. User perceivable QoS improved significantly not because packet loss decreased but because loss of "important" packets decreased [10]. Interval based traffic smoothing algorithms do not consider the QoS importance of a packet. However, interval based traffic smoothing algorithms successfully distinguish the packets based upon their respective QoS importance. It is found that this phenomenon is due to the inadvertent result of two technical characteristics. The first one is the way video frames are marshalled into packets. Since traffic smoothing aims at minimizing rate variability, the transmission interval between B type frame packets become smaller while the interval between I type packet and its successor (or predecessor) becomes longer. The second technical feature is the way operating system handles queue of packets. When a packet arrives, it is copied into main memory and operating system inserts the packet pointer into the queue of pointers. Interval based traffic smoothing algorithm controls the interval between outgoing packets to make the data rate smoother; hence, small size packets are more closely populated. From the operating systems's perspective in the receiving end, incoming traffic actually becomes burstier, and gets exposed to more packet loss. Since larger packet has relatively longer interval from the departure of the preceding packet, it is less likely that larger packet finds the queue full. Due to harmonious effort between packetization method and the kernel data structure of packet, traffic smoothing algorithm happens to incorporate packet importance.

3 Significance-Aware Packet Scheduling

3.1 Packet Significance

We define the notion of *significance* to represent the importance of a frame or layer in a packet[1]. $f^i_{j,k}$ denotes k_{th} layer information for j_{th} frame of i_{th} GoP. A set of parent packets, $\mathcal{P}(f^i_{j,k})$, denotes a set of packets which are required to decode packet $f^i_{j,k}$. A set of *child* packets of $f^i_{j,k}$ is a set of packets which has $f^i_{j,k}$ as its parent, i.e. $\mathcal{C}(f^i_{j,k}) = \{f^m_{n,l} \mid f^i_{j,k} \in P(f^m_{n,l})\}$. Loss of $f^i_{j,k}$ causes the inappropriate decoding of not only $f^i_{j,k}$ itself but also all packets in its child packet, $\mathcal{C}(f^i_{j,k})$. Let $f^i_{j,k}(x,y)$ is a pixel value (RGB) at (x,y) position of an image when $f^i_{j,k}$ is properly decoded and $\widehat{f}^i_{j,k}(x,y)$ is a pixel value when $f^i_{j,k}$ is not properly decoded. We define *contribution* $\mathcal{D}(f^i_{j,k})$ as in Eq. 1.

$$10\log\frac{W \times H \times 255^2}{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}|\hat{f}^i_{j,k}(x,y) - f^i_{j,k}(x,y)|^2} \quad (1)$$

[1] Frame or layer is transported in the form of packets; hence, we use the term *packet information* as a layer or frame information contained in the packet, and vice versa

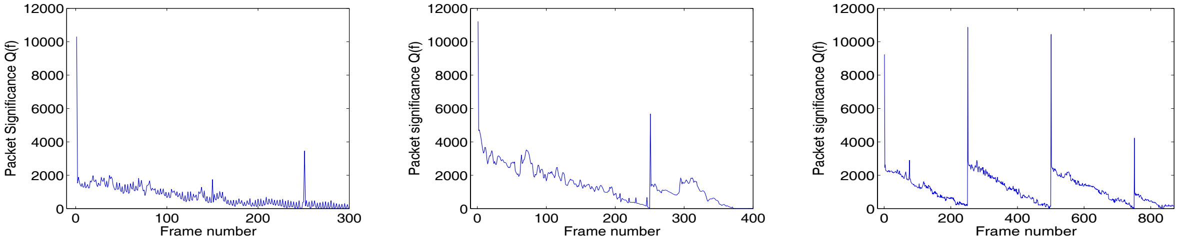
where H and W is the screen height and width. $\mathcal{D}(f^i_{j,k})$ gives a quality metric of $f^i_{j,k}$ loss. Significance of a packet $f^i_{j,k}$ is sum of all subsequent PSNR degradation which can occur due to the loss of $f^i_{j,k}$. Significance of $f^i_{j,k}$ is defined as $\mathcal{Q}(f^i_{j,k}) = \sum_{f^l_{n,m} \in \mathcal{C}(f^i_{j,k})} \mathcal{D}(f^l_{n,m})$. It is worth noting that packet significance $\mathcal{Q}(f^i_{j,k})$ effectively captures the information dependency among frames or between layers.

3.2 Packet Selection

Packet selection is a process of determining the subset of packets for transmission satisfying resource constraints. Let $\mathcal{F}(f^i_{j,k})$ be the transmission interval of $f^i_{j,k}$, which is the interval from its immediately preceding packet. Let $\mathcal{S}(f^i_{j,k})$ be the size of $f^i_{j,k}$. Current bandwidth availability is assumed to be informed to the streaming server or content delivery network (CDN) by the system [1]. We define total QoS of selected packets in f^i as in Eq. 2.

$$\xi(f^i) = \underbrace{\sum_{\mathcal{F}(f^i_{j,k})<\infty} \mathcal{D}(f^i_{j,k})}_{A} - \underbrace{\sum_{f^i_{j,k}\ is\ lost} \mathcal{Q}(f^i_{j,k})}_{B} \quad (2)$$

Condition $\mathcal{F}(f^i_{j,k}) < \infty$ denotes the set of *"selected"* packets. Term A corresponds to PSNR values resulting from transmitting selected packets and B denotes QoS degradation caused by a packet loss. Our objective is to maximize $\xi(f^i)$ via properly selecting subset of packets and determining transmission schedule. Our process consists of two phases: packet selection and packet transmission. 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 envelope as $\mathcal{U} = \int_{t_0}^{t_0+\omega} \rho(t)dt$, where t_0, ω and $\rho(t)$ denote start time of the window, one GoP time length and the available bandwidth at t, respectively. We take greedy approach. Let $\epsilon(f^i_{j,k})$ be the ratio between QoS significance and its size, i.e. $\epsilon(f^i_{j,k}) = \mathcal{Q}(f^i_{j,k})/\mathcal{S}(f^i_{j,k})$. The algorithm sorts all packets in each GoP with respect to the decreasing order of $\epsilon(f^i_{j,k})$ and selects one by one until sum of the selected information exceeds the capacity constraint. It is worth noting that through the bandwidth adaptation and packet selection process, SAPS does not require higher bandwidth. Hence, it won't impact other packets of different media or other applications in terms of bandwidth requirement. Once we determine the set of packets to transmit, we need to determine packet transmission schedule of selected packets. Here, determining a transmission schedule is equivalent to determining an interval between packet departure to avoid packet loss. The key idea is to assign larger interval to more



(a) Packet Significance (Akiyo) (b) Packet Significance (Foreman) (c) Packet Significance (Grandma)

Figure 3. Packet Significance Value Distribution

important packet. Let $\delta(f^i_{j,k})$ denote the time interval between $f^i_{j,k}$ and its immediate predecessor and it is defined as in Eq. 3.

$$\delta(f^i_{j,k}) = \frac{\omega \times \mathcal{S}(f^i_{j,k}) \times \mathcal{Q}(f^i_{j,k})}{\sum_{F(f^i_{j,k})\in f(^i)} \mathcal{S}(f^i_{j,k}) \times \mathcal{Q}(f^i_{j,k})} \qquad (3)$$

If $f^i_{j,k}$ consists of multiple packets, the interval is computed as $\delta(f^i_{j,k})/\left\lceil \frac{\mathcal{S}(f^i_{j,k})}{MTU} \right\rceil$.

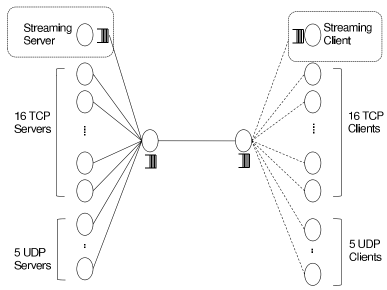


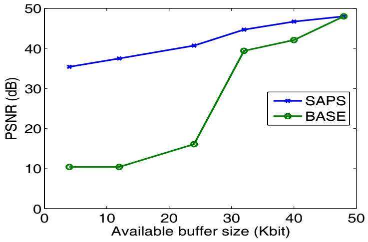
Figure 6. Topology of the experiment setup

4 Performance Evaluation

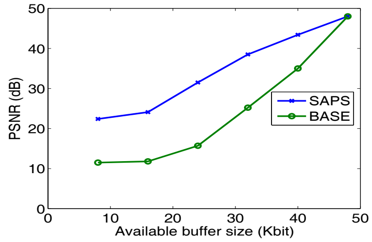
We examine the effectiveness of Semantics-Aware Packet Scheduling algorithm (SAPS). We compare SAPS with packet scheduling algorithm where transmission interval is linearly proportional to the size of a frame, which we call it as BASE for simplicity's sake. BASE does not consider the semantics of a packet in determining the transmission schedule. We simulate in NS-2 [4] over the network depicted in Fig. 6. We use three publicly available and widely used video clips [7]. Three video clips are compressed by MPEG-4 encoder [6]. All compressed video clips have 300 kbits/sec and 30 frame/sec, with 176*144. GoP structure of compressed video is $I(P)^{249}$. There exist 16 TCP and 5 UDP node pairs sharing the link. File transfer protocol (FTP) application is running over TCP. The maximum bandwidth from each TCP and UDP node pair corresponds to 1Mbyte/sec and 128 Kbyte/sec, respectively. Client starts displaying video 2 seconds after the transmission has begun. If packets arrive out of order sequence, then the respective packet stays at the queue until all of the required packets arrive before the play-out deadline or discarded. In addition, in case that at least single packet is dropped or corrupted during the transmission, and is not able to recover with recovery scheme such as FEC, then the entire packets consisting one frame will be discarded. If a frame is lost during the transmission or arrived later than the play-out deadline, the previous frame concealment scheme is used at the decoder. The average time taken for the calculation of each packet significance is approximately 0.5 seconds. For example, the total time taken to calculate 30 frames' packet significance in 30 FPS movie is $\frac{\frac{FPS}{FPS}+\frac{FPS-1}{FPS}+\cdots\frac{1}{FPS}}{2} = \frac{31}{2}$. Hence the average time taken per frame is $\frac{31}{2*30} \approx 0.5$. We assume that this information has been calculated at encoding time and transmitted to the streaming server or CDN with movie file. Packet significance can be computed offline and therefore does not interfere with the real-time video streaming session. Fig. 3 illustrates the packet significance distribution. As can be seen, packet significance varies subject to its frame type and the position within GoP. I frame, at every 250_{th}, shows very high significance value. P frames, immediately after I frame, are likely to have higher value of packet significance than those frames far from the I frame whether the frame size is big or not.

4.1 Effect of Client Queue Depth

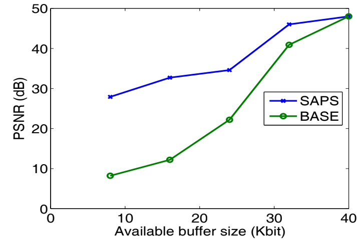
Fig. 4(a), 4(b) and 4(c) illustrate the performance of SAPS and BASE under different client queue depth. In both algorithms, PSNR increases with the increase in client queue depth. When client queue depth is small, we observe significant difference in PSNR values between SAPS and BASE algorithms. When 8 Kbit is allocated for the client queue, PSNR values are around 10_{dB} in BASE algorithm. Under same client queue depth, PSNR values



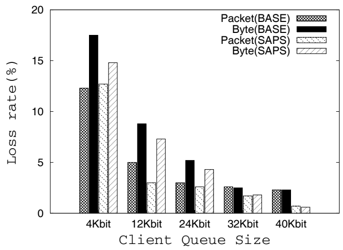
(a) PSNR vs. Queue Depth (Akiyo)



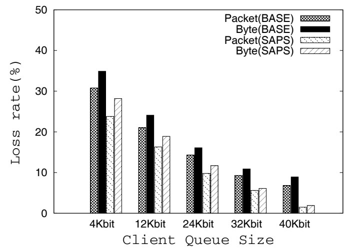
(b) PSNR vs. Queue Depth (Foreman)



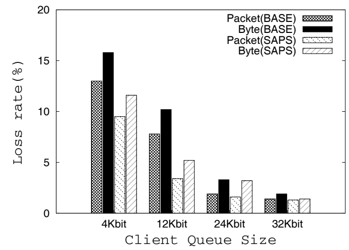
(c) PSNR vs. Queue Depth (Grandma)



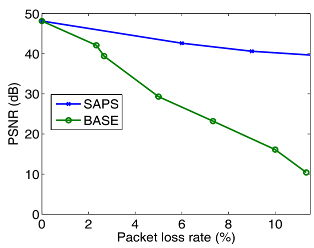
(d) Packet & Byte Loss Rate (Akiyo)



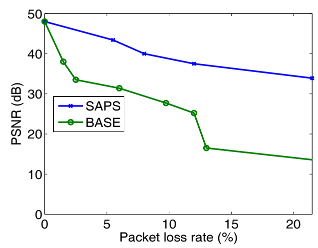
(e) Packet & Byte Loss Rate (Foreman)



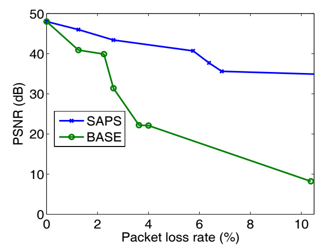
(f) Packet & Byte Loss Rate(Grandma)



(g) PSNR vs. Packet Loss (Akiyo)



(h) PSNR vs. Packet Loss (Foreman)



(i) PSNR vs. Packet Loss (Grandma)

Figure 4. Performance under Varying Client Queue Depth

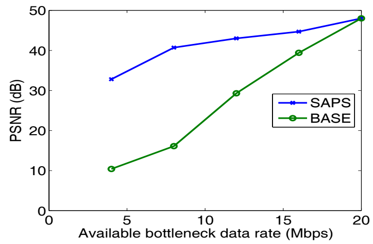
are 35_{dB}, 23_{dB} and 28_{dB} for Akiyo, Foreman and Grandmother video traces, respectively. PSNR values become two to three times larger when we apply SAPS algorithm in scheduling packets. When client queue is sufficiently large, there is no packet loss due to queue overflow, so PSNR values of the two algorithms become same. Fig. 4(d), 4(e) and 4(f) illustrate packet loss and byte loss rate with two schemes under varying client queue depth. As shown in the figures, packet loss and byte loss rate decreases with increase in client queue depth. In all cases, SAPS shows lower packet loss rate and byte loss rate except 4Kbit client queue depth are allocated in Akiyo. However, although packet loss rate shows higher in SAPS than BASE, BASE shows lower PSNR and higher byte loss rate in Fig. 4(a) and 4(d), respectively.

Fig. 4(g), 4(h) and 4(i) illustrate different manifestation of SAPS. In these figures, the relationship between packet loss and PSNR of both schemes are illustrated under decreasing client queue depth. As can be seen, SAPS exhibits much higher PSNR values than BASE algorithm under the same packet loss rate. This is due to the fact that SAPS successfully adapts to client queue availability so that more important packets become less vulnerable to packet loss with client queue overflow.

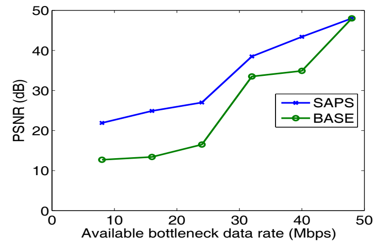
4.2 Effect of Bandwidth Availability

Fig. 5(a), 5(b) and 5(c) illustrate the PSNR of SAPS and BASE as a function of the available bottleneck bandwidth on the shared channel. In all figures, PSNR increases with the capacity of bottleneck link. When bottleneck link capacity reaches 40 Mbps, SAPS and BASE achieve same PSNR. However, when bottleneck link capacity becomes smaller, SAPS manifests its capability of handling significance. For example, when the available bottleneck bandwidth is 16 Mbps, the PSNR gain of SAPS over BASE is 5_{dB}, 9_{dB} and 8_{dB} for Akiyo, Foreman and Grandmother, respectively.

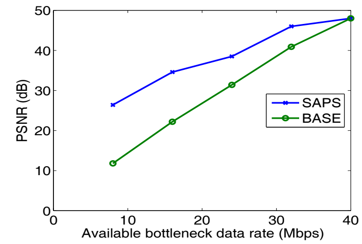
Fig. 5(d), 5(e) and 5(f) illustrate the packet loss and byte loss rate with two schemes under different bottleneck bandwidth. In all figures, packet loss and byte loss rate decreases as higher bottleneck bandwidth is allocated. SAPS scheme



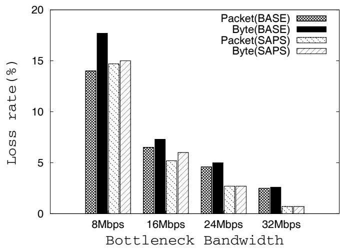
(a) PSNR vs. Bandwidth (Akiyo)



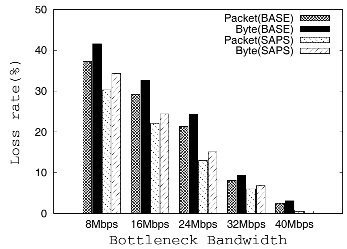
(b) PSNR vs. Bandwidth (Foreman)



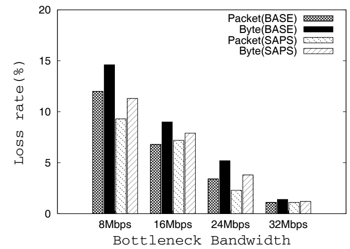
(c) PSNR vs. Bandwidth (Grandma)



(d) Packet & Byte Loss Rate (Akiyo)



(e) Packet & Byte Loss Rate (Foreman)



(f) Packet & Byte Loss Rate (Grandma)

Figure 5. Performance under Varying Bottleneck Bandwidth

shows lower packet loss and byte loss rate all the time except when the 8 Mbps bottleneck bandwidth is allocated in Akiyo. However, as shown in Fig. 5(a) and 5(d), SAPS shows higher PSNR value and lower byte loss rate although it shows higher packet loss rate.

5 Conclusion

In this work, we develop a novel packet scheduling framework which properly incorporates the significance of a packet. We analyze the data structure of the network packets in operating system and make packet scheduling frameworks effectively exploit the data structure. To achieve this objective, we develop elaborate metric to represent the importance of a packet from user QoS's point of view: Packet Significance. We develop video streaming framework, Significance Aware Packet Scheduling (SAPS), which consists of packet selection and packet scheduling phase taking account of packet significance. Through simulation based experiment, we find that via properly incorporating the packet significance, we can increase PSNR value by upto factor of three. SAPS manifests itself under resource stringent environment, e.g. real-time video streaming in mobile wireless network.

References

[1] P. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media. *Multimedia, IEEE Transactions*, 8(2):390–404, Apr. 2006.

[2] P. D. Cuetos, P. Guillotel, K. W. Ross, and D. Thoreau. Implementation of adaptive streaming of stored MPEG-4 FGS video. In *Proceedings of ICME*, pages 405–408, Lausanne, Switzerland, Aug. 2002.

[3] J. Dubois. Burstiness reduction of a doubly stochastic ar-modeled uniform activity vbr video. *Transactions on WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*, 23:454–458, Aug. 2007.

[4] Information Sciences Institute. The Network Simulator - ns-2. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/User_Information

[5] D. Marpe, T. Wiegand, and G. Sullivan. The h.264/mpeg4 advanced video coding standard and its applications. *Communications Magazine*, 44(8):134–143, Aug. 2006.

[6] MEncoder. Program for encoding video+audio. [Online]. Available: http://www.mplayerhq.hu/

[7] D. Park and S. Hong. Video trace file. [online]. Available: http://www.dmclab.hanyang.ac.kr/data/mpeg2data/Video_traces.htm.

[8] I. Politis, M. Tsagkaropoulos, T. Pliakas, and T. Dagiuklas. Distortion optimized packet scheduling and prioritization of multiple video streams over 802.11e networks. *Advances in Multimedia*, 2007-2(6), April 2007.

[9] S. Song, Y. Won, and I. Song. Empirical study of user perception behavior for mobile streaming. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 327–330, Juan-les-Pins, France, 2002. ACM NY, USA.

[10] Y. Won and B. Shim. Empirical study of VBR traffic smoothing in wireless environment. In *Proceedings of the second international workshop on Innovative Internet Computing Systems*, 2346:193–204. Springer, June 2002.

[11] J. Wu, J. Cai, and C. Chen. Single-pass rate-smoothed video encoding with quality constraint. *IEEE Transactions on Signal Processing Letters*, 14(10):715–718, Oct. 2007.