# Incorporating packet semantics in scheduling of real-time multimedia streaming

**Sungwoo Hong · Youjip Won**

Published online: 14 October 2009

**Abstract** In this work, we develop a novel packet scheduling algorithm that properly incorporates the semantics of a packet. We find that improvement in overall packet loss does not necessarily coincide with improvement in user perceivable QoS. The objective of this work is to develop a packet scheduling mechanism which can improve the user perceivable QoS. We do not focus on improving packet loss, delay, or burstiness. We develop a metric called, "Packet Significance," that effectively quantifies the importance of a packet that properly incorporates the semantics of a packet from the perspective of compression. Packet significance elaborately incorporates inter-frame, intra-frame information dependency, and the transitive information dependency characteristics of modern compression schemes. We apply packet significance in scheduling the packet. In our context, packet scheduling consists of two technical ingredients: packet selection and interval selection. Under limited network bandwidth availability, it is desirable to transmit the subset of the packets rather than transmitting the entire set of packets. We use a greedy approach in selecting packets for transmission and use packet significance as the selection criteria. In determining the transmission interval of a packet, we incorporate the packet significance. Simulation based experiments with eight video clips were performed. We embed the decoding engine in our simulation software and examine the user perceivable QoS (PSNR). We compare the performance of the proposed algorithm

This research was in part supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government(MEST) (No. R0A-2009-0083128) and the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency (NIPA-2009-(C1090-0902-0033)).

S. Hong (✉) · Y. Won
Division of Electrical and Computer Engineering,
Hanyang University, Seoul, South Korea
e-mail: toggiya@ece.hanyang.ac.kr, toggiya@empal.com

Y. Won
e-mail: yjwon@ece.hanyang.ac.kr

with best effort scheduling scheme and one with simple QoS metric based scheduling scheme. Our Significance-Aware Scheduling scheme (SAPS) effectively incorporates the semantics of a packet and delivers best user perceivable QoS. SAPS can result in more packet loss or burstier traffic. Despite these limitations, SAPS successfully improves the overall user perceivable QoS.

**Keywords** Real-time multimedia streaming · Scalable encoding · Packet significance · Traffic smoothing · Packet scheduling

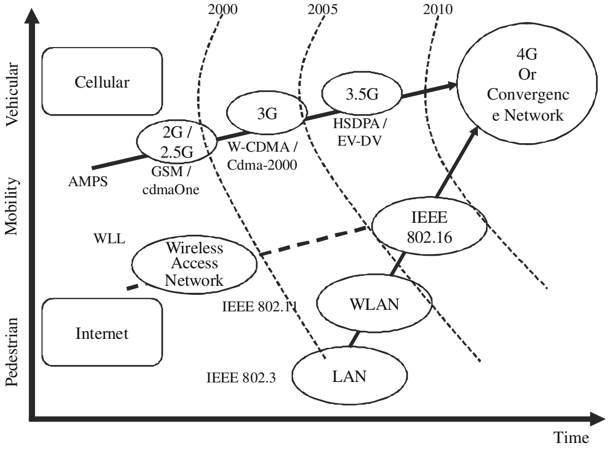
## 1 Introduction

### 1.1 Motivation

Due to the advancement of computer and communication technology, we can now enjoy bi-directional interactive multimedia services in a ubiquitous fashion. Rapid advancements in network technology have increased the amount of bandwidth to Gbyte/s in residential units (Fibre-To-The-Home) [17]. Wireless networks are also developing rapidly to meet different needs. Wi-Fi (IEEE 802.11 a/b/g/n) is designed for fixed client terminals with a 50 + Mbits/s bandwidth range [5]. Wireless Broadband Networks (IEEE 802.16) can deliver a few Mbits/s bandwidth for mobile clients with reasonable moving speeds such as car or subway. Communication technology of 3.5G (e.g. HSDPA) or 4G can be used by a client terminal with fast moving speed with much lower bandwidth than IEEE 802.16 [5]. It can be used in high speed trains where speeds reach 300 km/h (200 miles/h), e.g. ICE, Shinkansen, and KTX [12]. Advancements in network technology have brought us not only an abundance in bandwidth but also a “variety” of bandwidth choices. Recently emerging mobile devices can now exploit the diversity of wireless technologies. They can dynamically choose different MAC sub-layer technologies based upon availability, signal condition, client moving speed, and other parameters. For example, a certain cell phone can switch between 802.16 or 3G CDMA as an air-link interface for voice communication [35]. However, to exploit the underlying network resources and maximize the QoS of multimedia streaming services, it is necessary for all these factors to be properly addressed in a single scheme. This requirement must be satisfied in order to improve the QoS of Internet based TV service, IPTV [14], and wireless TV broadcasting services for mobile devices [13] (Fig. 1).

Real-time video streaming has a unique performance requirement that distinguishes itself from text-based best effort data services: bandwidth guarantee and rate variability. Since real-time video streaming requires that each compressed information needs to arrive at destination before its predefined deadline, a certain fraction of the bandwidth needs to be guaranteed for that connection either deterministically or stochastically. Modern compression technology exploits information redundancy in underlying video. The redundancy exists along the spatial, temporal, and Signal to Noise ratio (SNR) axis. Compressed video consists of sequence of frames where information redundancies are eliminated. In compressed images, the size of each frame varies widely depending on the information it carries. The frame can be self-sufficient, contain only the difference from its preceding frame, or can contain only interpolated information between its neighboring frame. As a side effect of

**Fig. 1** Evolution of network technology



redundancy elimination, the size of each frame can differ by an order of magnitude. The variability of the frame size is realized as "bursty" network traffic in real-time video streaming services. Numerous efforts have been proposed to reduce the burstiness of real-time video traffic so that the packet loss can be reduced and/or user perceivable QoS can be maximized [10, 24, 31, 33]. This set of efforts is called traffic smoothing (or traffic shaping).

Recent advancement in network technology makes real-time video streaming more complicated. From the content provider's point of view, the diversity of the client's access bandwidth increases the dimension of complexity for content management. As a result, the content provider may need to prepare several QoS versions of the content for each network speed. To overcome this problem, layered encoding known as the "scalable encoding technique" has been proposed [20]. In layer encoded content, the original imagery (or video) can be reconstructed from the subset of the compressed information. With a larger subset, we can reconstruct a better quality image (or video).

To maximize the user perceivable QoS, the sender, e.g. streaming server software, needs to make a right choice for two fundamental issues: "what to send?" and "when to send?" The first issue is the selection of the subset of compressed information for real-time streaming services. The second issue is the removal of the burstiness in the underlying traffic. A good streaming server or packet scheduler needs to properly incorporate the bandwidth availability of the underlying network and the characteristics of the compression scheme in making a choice. The modern traffic smoothing algorithm focuses on devising a packet transmission schedule without considering the importance of a packet from the perspective of QoS. To maximize the user perceivable QoS, it is mandatory that the sender properly incorporate the "importance" of a packet in devising a transmission schedule. Streaming of layer encoded content adds another dimension of complexity in multimedia streaming. It raises an issue of "what to send" that did not exist before. When the sender transmits all information without considering the available bandwidth, it not only congests the underlying subnet but also results in significant QoS degradation due to random packet loss. A streaming server needs to properly incorporate the underlying network bandwidth and importance of a packet in selecting packets for transmission. There is another important concern in packet scheduling. Most packet scheduling works deal with the bandwidth aspect of the underlying network. There are two aspects of the network traffic: the bandwidth process (bytes/s) and the packet count process (packets/s). These two axes are orthogonal to each other. From the perspective of network medium, bandwidth process is of more importance. However, from the perspective of network queue, the packet count process can be more important since the queue is an array of packet pointers rather than an array of packets. Therefore, removing the burstiness of the bandwidth process does not necessarily imply improvement in packet loss. More importantly, we can achieve better QoS via losing more packets [31, 33].

In this work, we develop a unified packet scheduling framework that properly addresses the issues raised above: "what to send" and "when to send." The contribution of our work is as follows: First, we develop a notion of "Packet Significance" that properly captures the QoS importance of a given packet. Our scheduling framework elaborately harbors the inter-frame dependency as well as the inter-layer dependency of a frame. When determining (selecting) the packets for transmission and determining the selected packet's transmission interval, packet significance plays a key role. Second, we successfully develop a unified framework for determining "what to send" and "when to send." A traffic smoothing (or shaping) algorithm and a layer encoding scheme have been dealt with in a separate context. However, to properly exploit the underlying network resource and maximize the 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 aspect of a packet but also the Operating System's aspect of a packet. From the perspective of the network, the bandwidth process is of prime concern. From the perspective of the Operating System, however, the packet count process (packets/s) is more important since the network queue is represented by the array of packet pointers where the size of the individual packet does not matter.

### 1.2 Related works

Real-time communications can be classified into two categories with respect to QoS guarantee: *deterministic* and *statistical* [1]. In order to satisfy their absolute QoS requirements in deterministic real-time communication, connections must reserve resources based on worst-case source traffic behavior, which may result in severe underutilization of network resources, especially when source traffic is bursty. By contrast, to make more efficient use of network resources, statistical real-time communication specifies QoS requirements in probabilistic (instead of deterministic) terms.

Statistical multiplexing is one type of communication link sharing scheme. Statistical multiplexing is an on-demand service rather than a deterministic one that pre-allocates resources for the data transmission. Data streams are transmitted over the divided communication channel to improve link utilization. In statistical multiplexing schemes, for example, the amount of bandwidth allocated in a network to a VBR source is less than its peak rate, but greater than its average rate. Then, the sum of the peak rates of connections multiplexed onto a link can be greater than the link bandwidth if the sum of their statistical bandwidths is less than or equal to the provisioned link bandwidth; this may result in a certain percentage of packet losses and/or deadline misses. Statistical real-time communication is especially useful for applications with burst traffic. The statistical multiplexing gain is known to be substantial, especially in variable-bit-rate (VBR) applications such as MPEG-coded video. Giordano et al. [11] presented the statistical multiplexing gain over the deterministic scheme by considering the fair distribution of the bandwidth between the homogenous and the heterogeneous sources (with respect to their Hurst parameters).

Statistical multiplexing does not make any promises about the end-to-end delay for an individual packet. Nor does the service make any promises about the variation of packet delay within a packet stream. Available bandwidth dynamically changes according to the subnet congestion situation. This can be due to the TCP congestion control algorithm or a session's start and/or termination. Because resources over a network are not guaranteed in a statistical multiplexing scheme, there have been many approaches that have tried to improve the QoS of streaming with limited resources.

A layered (Scalable) encoding scheme has been suggested to adapt to varying network bandwidth availability [20]. The MPEG-4 standard adopted Fine Granularity Scalability (FGS) encoding scheme to support fully adaptable network bandwidth availability using layered encoding methods [20]. In such coders, both the bandwidth of a layer and the number of layers can be dynamically manipulated with only minor quality degradation. However, the MPEG standard does not specify how the video packets should be delivered over the Internet to have a maximum user perceivable QoS. To transport video packets over error-prone and resource constrained networks, many packet scheduling methods have been proposed.

To reduce traffic burstiness, Wu et al. suggested reducing playback quality variation using rate smoothing [34]. They used an arithmetic averaging filter to smooth out the rate during single-pass video encoding so that the targeted average video quality can be achieved. However, the proposed rate-smoothed scheme can only reduce rate fluctuations among the same type of video frames. Different types of frames need to be considered as well to maximize user perceivable QoS. Dubois tried to minimize the burstiness of the network traffic [10] with a variable bit rate (VBR). He showed a significant reduction in burstiness through the effect of statistical multiplexing in an ATM environment. The proposed scheme treated each frame as having the same importance; however, each frame has a very different effect on user perceivable QoS. In addition, the proposed scheme has not proven to work well in error-prone environments such as the Internet. Argyriou suggested a cross-layer error control scheme for wireless/wireline packet networks. He modeled the end-to-end delay and packet loss rate as a function of the automatic repeat request (ARQ) and forwarded the error correction (FEC) control mechanisms that are employed at the application and wireless link layers [2]. He noted that multimedia streaming is delay-sensitive. His efficient packet retransmission scheme was able to improve user perceivable QoS. However, he did not consider the difference in each packet's importance. If the packet's importance was calculated in the application layer, it could have been used in the link layer retransmission scheme. In this case, the user perceivable QoS can be improved significantly even with the same loss rate or delay constraint. Delgado et al. designed a Cross-Layer Optimizer module to apply several optimization strategies to different network layers so that challenges to Mobile

Ad hoc Networks (MANETs), such as frequent changes in network topology and node conditions, can be overcome [9]. Using redundancy information and a cross layer approach, their module provided unequal priority over different frame types, i.e. high priority to I frames, medium priority to P frames, and low priority to B frames. However, the importance of the same frame can differ significantly depending upon its size and its position within the group of pictures (GoP). Mansour et al. proposed joint rate and protection allocation for multi-user scalable video streaming using application layer error protection [21]. They noted that rate and distortion modeling is a keystone in model-based video rate and transmission algorithms. The proposed method of estimating the packet distortion algorithm considers the frame type and PSNR distortion over the decoded picture. The proposed method omits considering the correlation among the frames; hence, it does not reflect each packet's distortion information exactly. Miao et al. proposed optimized rate allocation algorithms that consider each packet's importance in terms of rate-distortion information [7]. Their experimental results showed a significant improvement in QoS: gains of 2.6 dB or more over systems that are not rate-distortion optimized. Frossard et al. suggested an optimization framework in which multiple senders can coordinate their packet transmission schedules, such that the overall quality over the video clients is maximized over the shared communication resources [4]. In [4, 7] packet importance was formulated as a summation of the packet size and mean square error (MSE) of the lost packets. They do not properly model the information dependency among the frames. According to this model, packet importance is proportional to packet size, which does not always hold. We illustrate this in Section 6.2. To quantify the importance of each packet, Politis et al. [24] suggested incorporating the number of reference frames including packet size and MSE. With this approach, they tried to capture the inter-frame dependency of the modern compression algorithm. By incorporating the reference count, their QoS model yields more accurate estimation on user perceivable QoS. However, they did not mention how video data should be delivered from the source node to the destination node. User perceivable QoS in the destination node can be different by orders of magnitude, even if the available bandwidth is the same. In this work, we deal with the packet QoS model and the Packet scheduling algorithm in a single framework. We determine the subset of packets and the inter-packet interval based upon the packet significance model which is developed as a part of this work.

To exploit the underlying network resource and to maximize QoS, it is mandatory that all these factors are properly addressed in a single framework. This work aims at devising a packet scheduling framework that properly incorporates the QoS importance of each packet so that user perceivable QoS is maximized. To effectively address this issue, we first define the metric "QoS Significance" of a packet. This elaborately quantifies the importance of a packet from the perspective of user perceivable QoS. Using this metric, we develop a greedy approach based packet scheduling algorithm that exploits the underlying network bandwidth and maximizes user perceivable QoS.

The remainder of the paper is organized as follows. Section 2 introduces the scalable encoding and packetizing bit-stream. Section 3 explains packet smoothing. Section 4 introduces the notion of packet significance. The significance aware packet scheduling algorithm is explained in Section 5. Section 6 carries the result of the performance evaluation. We conclude our work in Section 7.

## 2 Synopsis: scalable encoding and packetizing bit-stream

### 2.1 Scalable encoding

In compressing video, an encoder exploits the spatial and temporal redundancy in the original information. The encoder also adjusts the signal to noise ratio (SNR) to meet the data rate constraints. Scalable encoding has been proposed to selectively decode or transport the subset of the original information to cope with the resource variability in various physical components such as CPU clocks and network bandwidth. Scalability is achieved via partitioning the compressed information into a number of disjoint sets. Each set is called a *layer*. In MPEG-2 and H.263+, partitioning of the information can be done along one of the following axis: temporal scalability, spatial scalability, and Signal-to-Noise Ratio (SNR) scalability [26]. SNR scalability is a technique to code a video sequence into more than one layer at the same frame rate and the same spatial resolution, but with different quantization accuracy. The base-layer bitstream is first decoded by the base layer variable-length decoder (VLD). The inverse quantizer in the base layer produces the reconstructed DCT coefficients. The enhancement bitstream is decoded by the VLD in the enhancement layer and the enhancement residues of the DCT coefficients are produced by the inverse quantizer in the enhancement layer. A higher accuracy DCT coefficient is obtained by adding the base-layer reconstructed DCT coefficient and the enhancement-layer DCT residue. The DCT coefficients with a higher accuracy are given to the inverse DCT (IDCT) unit to produce the reconstructed image domain residues that are to be added to the motion-compensated block from the previous frame [3, 15, 20].

Figure 2a describes an example of spatial layering. Spatial scalability supports displays with different spatial resolutions. By decoding the base layer, the user can display a preview version of the decoded image at a lower resolution. Decoding the second layer results in a larger reconstructed image. Furthermore, by progressively decoding the additional layers, the viewer can increase the spatial resolution of the image up to the full resolution of the original image.

Figure 2b illustrates an example of partitioning a video along the temporal axis. The entire video clip is partitioned into four layers: base layer, enhancement layer $E_1$, enhancement layer $E_2$, and enhancement layer $E_3$. As shown in the figure, the base layer consists of I frames only. $E_1$ consists of P frames only. $E_2$ and $E_3$ consist of $B_{3i-1}$ and $B_{3i}$ frames, respectively, where $i = 1, 2, \ldots$. Subject to resource

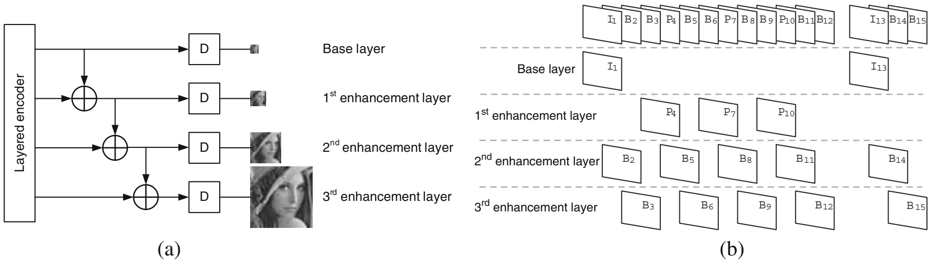


**Fig. 2** Scalable encoding. **a** Spatial scalability. **b** Temporal scalability

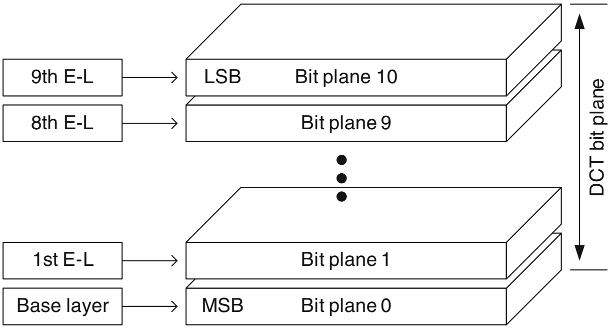


**Fig. 3** Conceptual relationship between DCT coefficients and layers

availability, the application selects the subset of the layers. A dependency among layers exists. A higher layer requires all of its lower layer data blocks for decoding. This is because the upper layer only carries the difference information with its lower layer to minimize the information redundancy.

Another scalable coding method is Fine-Grained Scalable (FGS) coding [20]. The basic idea of FGS is to code a video sequence into a base layer and an enhancement layer. The base layer uses non-scalable coding to reach the lower bound of the bit-rate range. The enhancement layer codes the difference between the original picture and the reconstructed picture using bit-plane coding [15] of the DCT coefficients. The bitstream of the FGS enhancement layer can be truncated into any number of bits per picture. The decoder reconstructs an enhancement video from the base layer and the truncated enhancement-layer bitstreams. The enhancement-layer video quality is proportional to the number of bits decoded by the decoder for each picture. Figure 3 illustrates the relationship among the layers and DCT coefficients.

### 2.2 Packetizing bit-stream

There are two fundamental usages of MPEG bitstreams: recording and transmission. Our major interest is how to send the packet, and thus we will focus on the transmission aspect of MPEG bitstreams. Transmission systems prefer discrete blocks of data, so elementary streams are packetized to form a packetized elementary stream (PES) [30]. Packet structure is shown in Fig. 4. It begins with a header containing an unique packet start code and a code that identifies the type of data stream. Optionally, the packet header may contain one or more time stamps that are used for synchronizing

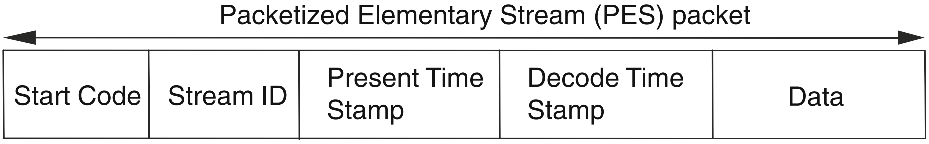


**Fig. 4** Packetized elementary stream (PES) structure is used to break up the continuous elementary stream

the video decoder to real-time and for obtaining lip-sync [16]. There are two types of time stamps: the presentation time stamp (PTS) and the decode time stamp (DTS). A presentation time stamp determines when the associated picture should be displayed on the screen, whereas a decode time stamp determines when it should be decoded. A presentation time stamp and a decode time stamp need not appear in every PES packet. Video frames always start on a packet boundary; hence, the last packet of each frame is smaller than the others.

## 3 Scheduling packets and traffic smoothing

### 3.1 Scheduling packet transmission

In our context, the notion of "packet scheduling" consists of two ingredients: (1) "what to transmit" and (2) "when to transmit." In a legacy sense, packet scheduling corresponds to "when to transmit" where packet scheduling determines the transmission time for outgoing packets. The layered encoding scheme brings another dimension of complexity to the packet scheduling problem: "What to transmit." A packet scheduler is required in order to select a certain fraction of the compressed information so that it does not overflow to the underlying subnet. In selecting a subset of frames, it is important to select the subset of layers so that we can maximize user perceivable QoS.

One of the key issues in packet scheduling is to reduce the burstiness of network traffic. This is called "traffic smoothing." Numerous efforts have been proposed in traffic smoothing research in order to minimize the variance of the traffic: the number of rate changes, pre-fetching delay, and maximum bandwidth requirement [8, 10, 18]. The traffic smoothing technique aims at removing the traffic burstiness so that it can make an indirect contribution to QoS via minimizing the packet losses. In the course of this reasoning, there are a number of fundamental issues that need to be verified from an engineering perspective.

There are two main approaches in realizing traffic smoothing: (1) sized based and (2) interval based smoothing. Original video stream information is marshalled into transmission units as known as "packets" and transmitted through networks. To remove the burstiness in the underlying traffic, we can either adjust the size or transmission interval of each transmission unit. In size based smoothing, the packet scheduler controls the amount of information carried by a single packet so that the size of each packet is similar (if not the same). In interval based smoothing, the interval between the packets is determined based on the size of the packet. Larger packets are allocated longer intervals. When we determine the approach for smoothing, interval based or size based, we need to carefully incorporate the characteristics of the underlying streaming software. Compressed video consists of I, P, and B type frames whose sizes differ by an order of magnitude. Each frame is expected to be displayed in a fixed interval subject to its frame rate, e.g. 30 frames/s.

To remove the burstiness of the traffic, adjusting the amount of information carried by each packet may seem more natural. Most preceding works in traffic smoothing use a size based smoothing approach. However, size based smoothing is not feasible for various reasons. Size based smoothing mandates that a single packet carries two or more frames. The MPEG standard does not put any restrictions on

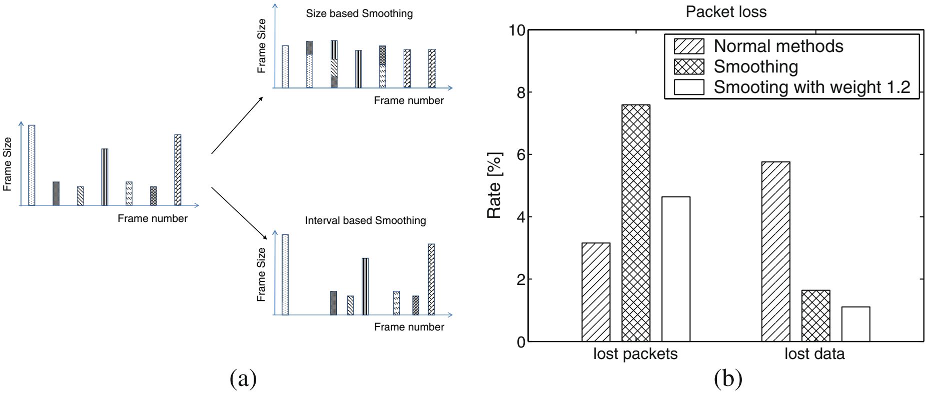


**Fig. 5** Traffic smoothing and its effect. **a** Size based smoothing vs. interval based smoothing. **b** Packet count vs. byte count [32]

whether a single packet contains more than one frame or not. In practice, however, most video streaming systems do not allow a single packet to carry more than one frame. If we allow multiple frames in a single packet, loss of a single packet may result in a loss of multiple frames. CPU overhead is another important concern that makes the size based smoothing infeasible. When single packets carry multiple frames, the decoder needs to locate the boundary of the individual frames for decoding. Locating the boundary of each frame can incur significant CPU overhead, especially in mobile hand held devices that have low-end CPU. Similar issues exist on the server side. For real-time streaming, each frame is padded with information required for remote playback, e.g. presentation time stamp, decode time stamp, and size of frame. To reduce the padding overhead, multimedia contents for streaming are preprocessed when stored. The content consists of a sequence of packets, rather than a sequence of frames that have decoding and transmission timing requirements. Figure 5a illustrates implementation approaches of size based smoothing and interval based smoothing. Packet schedulers do not adjust the packet size. Instead, they adjust the packet transmission interval to remove the burstiness and to improve the user perceivable QoS. In this work, we focus on an interval based smoothing approach.

### 3.2 Byte count and packet count

There are two different aspects to the underlying network traffic. The first one is the byte count (byte/s) aspect of the traffic. It denotes the amount of data transferred during a given time interval. The second one is the packet count (packets/s) aspect of the underlying traffic. The packet count denotes the number of packets transferred. Most existing work on traffic shaping and smoothing deal with the bandwidth process, i.e. byte count, aspect of the underlying traffic. Different component of the system deal with different aspects of the network traffic. A typical example is the way the Operating System handles packets. In the Operating System, the kernel maintains a fixed length queue for the UDP datagram as a queue of packet pointers. When packets arrive at the kernel address space, they are pointed by this queue of packet pointers. From the perspective of the network queue, incoming traffic can become burstier as a result of traffic smoothing. Subsequently, packet loss can increase due to traffic smoothing [33]. Figure 5a illustrates this situation. However, there is an important concern at this point. Reduction in packet loss does not necessarily imply reduction in byte loss or improvement in QoS. By the same token, the increase in packet loss does not necessarily imply an increase in byte loss or QoS degradation. The impact of packet loss over user perceivable QoS varies dependent upon the frame type of the lost packet it belongs to and its position within the group of pictures (GoP). For example, loss of the I frame packet affects all subsequent P or B frame packets in the same GoP. The effect of P frame loss varies subject to its position within the GoP. User perceivable QoS is majorly governed by the "type" of lost packets rather than the total number of lost packets. The packet scheduler needs to determine the transmission or interval timing so that the more important packet becomes less vulnerable to packet loss.

Won et al. performed a physical experiment for video streaming in an IEEE 802.11b environment [32]. It was found that traffic smoothing [19, 27] greatly improved user perceivable QoS. More interesting, however, is that the number of lost packets actually increased as a result of traffic smoothing. Figure 5b illustrates the packet loss and data loss ratio of the physical experiment [32]. "Lost packet" denotes the ratio between the number of lost packets compared to the total number of packets, while "Lost data" is the ratio between the amount of lost data (bytes) compared to the total amount of data (bytes).

Since the unit of the job from the perspective of the client device is the packet, the traffic can become burstier after applying traffic smoothing. Particularly in mobile wireless streaming environments, playback rate is relatively low and the size of the B Frame is much smaller than the maximum transmission unit (MTU) of the ethernet. Since traffic smoothing aims at minimizing the rate variability of the underlying traffic, the B frame packets become more densely populated, as shown in Fig. 5a. On the other hand, I frame packets carry a full data payload and therefore become more sparsely scheduled. As a result, packet loss increases when traffic smoothing is applied, as illustrated in Fig. 5b. However, the total amount of lost data decreased as a result of smoothing, and user perceivable QoS improved significantly because interval based smoothing successfully protects the more important packets from packet loss. Traffic smoothing improved the QoS by reducing the loss ratio of the I frame data. In addition, it is worth noting that depending on the type of the frame and its position within the GoP (if it is of the same frame type), the respective packet loss affects the user perceivable QoS in a different fashion. The existing smoothing algorithm does not incorporate this fact.

### 3.3 Packet loss and QoS behavior

Traffic smoothing greatly improves user perceivable QoS [19, 27]. However, smoothing the network traffic does not necessarily improve the packet loss. User perceivable QoS improves significantly, not because packet loss is decreased but because loss of "important" packets is decreased [32]. Interval based traffic smoothing algorithms do not consider the QoS importance of a packet. However, the interval based traffic smoothing algorithm successfully distinguishes 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 in which video frames are marshalled into packets. As mentioned in Section 3.1, a single packet does not carry more than one frame. In mobile wireless video streaming, the playback rate is relatively low and the size of the B frame is much smaller than the ethernet maximum transfer unit (MTU) size. I frame is an order of magnitude larger than the B frame. Since traffic smoothing aims at minimizing the rate variability of the bandwidth process, B frame packets are allocated a shorter transmission interval while the I frame packet is allocated a much longer transmission interval. The second technical feature is the way the Operating System handles the queue of packets. When a packet arrives, it is copied into the main memory, and the Operating System inserts the packet pointer into the queue of pointers. Each pointer represents the memory location of the respective packet. The way in which the video frame is marshalled and the way in which the Operating System handles incoming packets yield very interesting results when they are combined together. The interval based traffic smoothing algorithm controls the interval between the outgoing packets to make the data rate smoother. From the perspective of Operating Systems in the receiving end, incoming traffic actually becomes burstier as a result of smoothing, and the traffic subsequently gets exposed to more packet loss. As a result of smoothing, the traffic becomes burstier when transmitting the B frame packets and less burstier when transmitting the I frame packets. Traffic smoothing indirectly favors I frame packets over the frame type packet and significantly improves the user perceivable QoS.

There are important issues at this point which requires further elaboration: packet loss and packet importance. Packets do get lost on the Internet, which is unavoidable. Therefore, we need to select a subset of the frames and layers properly to minimize the packet loss. We need to incorporate the importance of a given packet in selecting packets and in determining its transmission schedule. This work is dedicated to developing a packet scheduling technique which elaborately encompasses QoS importance in selecting layers and frames for transmission and in determining a transmission schedule (Table 1).

**Table 1** Notation

| Symbol | Explanation |
|---|---|
| $f^i_{j,k}$ | $k_{th}$ layer of $j_{th}$ frame at $i_{th}$ GoP |
| $\mathcal{P}(f^i_{j,k})$ | Set of packets required to decode the packet in $f^i_{j,k}$ |
| $\mathcal{C}(f^i_{j,k})$ | Set of packets which has $f^i_{j,k}$ as its parent |
| $\mathcal{D}(f^i_{j,k})$ | PSNR degradation value with loss of $f^i_{j,k}$ |
| $\mathcal{Q}(f^i_{j,k})$ | PSNR degradation value with loss of $\mathcal{C}(f^i_{j,k})$ |
| $\xi(f^i)$ | Expected PSNR value in Client |
| $\mathcal{S}(f^i_{j,k})$ | Size of $f^i_{j,k}$ |
| $\delta(f^i_{j,k})$ | Transmission interval of $f^i_{j,k}$ |
| $t_0$ | Start time of a time window |
| $\omega$ | Window length |
| $\epsilon(f^i_{j,k})$ | Ratio of QoS significance and its size, $\dfrac{\mathcal{Q}(f^i_{j,k})}{\mathcal{S}(f^i_{j,k})}$ |
| $\rho(t)$ | Bandwidth constraints at time $t$ |
| $\mathcal{U}$ | Bandwidth envelope, $\int_{t_0}^{t_0+\omega} \rho(t)dt$ |

## 4 Packet significance

The effect of packet loss on QoS depends on the its frame type and its position within the group of pictures (GoP). Without I frame, there is no use in receiving the following B or P frames in the same GoP. In real-time video streaming in mobile wireless environments, the I frame appears much less frequently than in video streaming of high speed wired network environments or local playback of High-Definition quality content. For example, the default I-to-I distance in MPEG-4 compression is set at 250 frames. This is to increase the compression ratio [23]. We define the notion of *packet significance* to represent the importance of a frame in a packet.[1] A number of preceding works [7, 22, 24] attempted to quantify the importance of a given packet and to determine a transmission schedule based on its importance. To determine the importance of a packet, these works used packet size, frame type, number of reference frames, MSE distortion value in the decoded video and other parameters. However, they should be considered in a single framework. In this work, we not only consider the terms mentioned above but also calculate the loss effect of each packet in terms of PSNR for the higher user perceivable QoS.

Without loss of generality, we assume that video is layer encoded. $f^i_{j,k}$ denotes the $k_{th}$ layer information for the $j_{th}$ frame of the $i_{th}$ GoP. We define a set of *parent* packets and *child* packets for $f^i_{j,k}$. 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 $f^i_{j,k}$,$\mathcal{C}(f^i_{j,k})$ is a set of packets that has $f^i_{j,k}$ as its parent, i.e. $\mathcal{C}(f^i_{j,k}) = \left\{ f^m_{n,l} \,|\, f^i_{j,k} \in P\left(f^m_{n,l}\right)\right\}$. Loss of $f^i_{j,k}$ causes inappropriate decoding of not only $f^i_{j,k}$ itself but also all packets in its child packet, $\mathcal{C}(f^i_{j,k})$. Figure 6 schematically illustrates the dependency among the frames and layers. Dependency among the frames and layers is represented using arrows. A $-$ $>$ B denotes that A depends on B.

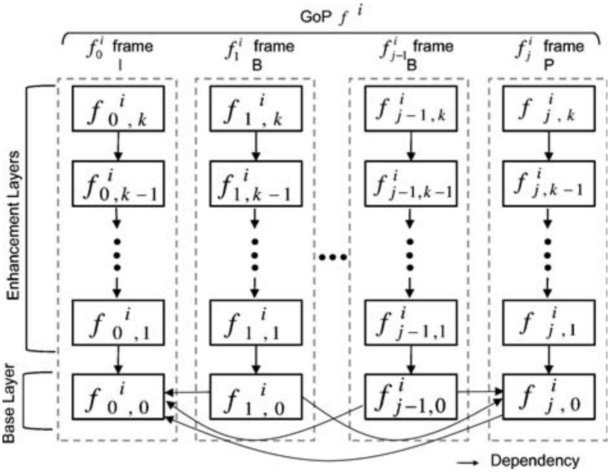
We developed a model to represent the quality of an image. An image consists of a set of pixels that are arranged in a two dimensional matrix. The number of pixels in a screen is called the resolution, e.g. HD: 1024*768, VGA: 640*480 and QCIF: 320*240. Each pixel is usually represented by 24 bits. Let $f^i_{j,k}(x, y)$ and $\widehat{f^i}_{j,k}(x, y)$ be the pixel value at the (x,y) position when a packet $f^i_{j,k}$ is properly decoded and when $f^i_{j,k}(x, y)$ is not properly decoded, respectively. We define the "Contribution" of $f^i_{j,k}$, $\mathcal{D}(f^i_{j,k})$ as in (1).

$$\mathcal{D}(f^i_{j,k}) = 10 \cdot \log_{10} \frac{W \times H \times 255^2}{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} \left| \hat{f}^i_{j,k}(x, y) - f^i_{j,k}(x, y) \right|^2} \tag{1}$$

where H and W are the screen height and width, respectively. The contribution of $f^i_{j,k}$ is based upon the notion of PSNR. $\mathcal{D}(f^i_{j,k})$ gives the quality metric of the $f^i_{j,k}$ loss. The contribution $\mathcal{D}(f^i_{j,k})$ is a metric for quality degradation on the respective

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

**Fig. 6** Dependency of MPEG-4 FGS video



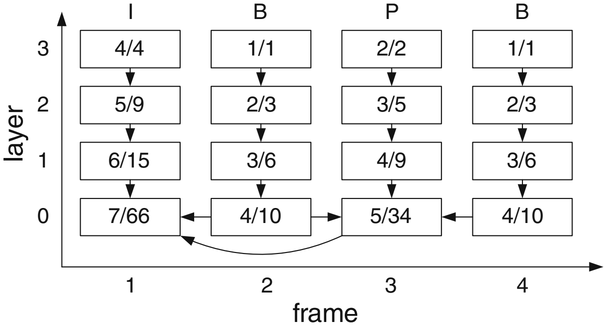
frame when packet $f^i_{j,k}$ is lost. Now, we define the notion of *Packet Significance*. The significance of packet $f^i_{j,k}$ is the sum of all PSNR degradations that can occur due to the loss of $f^i_{j,k}$. When $f^i_{j,k}$ is lost, it causes quality degradation not only in the frame it belongs to but also in its child frames. The significance of $f^i_{j,k}$, $\mathcal{Q}(f^i_{j,k})$ is defined as in (2).

$$\mathcal{Q}(f^i_{j,k}) = \sum_{f^l_{n,m} \in \mathcal{C}\left(f^i_{j,k}\right)} \mathcal{D}\left(f^l_{n,m}\right) + \mathcal{D}(f^i_{j,k}) \tag{2}$$

Figure 7 illustrates an example of how the significance value is assigned to individual layers.

As can be seen, the packet significance $\mathcal{Q}(f^i_{j,k})$ effectively captures the information dependency among the frames or between the layers.

**Fig. 7** Allocation of packet significance for each layer



## 5 Significance-aware packet scheduling

### 5.1 Packet selection: what to transmit

*Packet selection* is a process of determining the subset of packets for transmission satisfying certain resource constraints, e.g. network bandwidth availability or queue depth. Let $\delta\left(f^i_{j,k}\right)$ be the transmission interval of $f^i_{j,k}$, which is the interval from its immediately preceding packet. Let $\mathcal{S}\left(f^i_{j,k}\right)$ 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 [7, 25]. Short term in this context is one or two GoP's worth of period which corresponds to 2 s (15 frames/s, GoP (15,3)) or 1 s (30 frames/s, GoP (15,3)). The objective of packet scheduling is to maximize user perceivable QoS. We define user perceivable QoS as the sum of the QoS contribution of the transmitted packets subtracted by the QoS significance of the lost packets. Let $\mathcal{I}\left(f^i\right)$ be the set of selected packets for transmission. Then, user perceivable QoS of sending packets $\mathcal{I}\left(f^i\right)$ can be formulated as in (3).

$$\xi\left(f^i\right) = \underbrace{\sum_{f^i_{j,k}\in\mathcal{I}(f^i)} \mathcal{D}\left(f^i_{j,k}\right)}_{A} - \underbrace{\sum_{f^i_{j,k}\ is\ lost,\ f^i_{j,k}\in\mathcal{I}(f^i)} \mathcal{Q}\left(f^i_{j,k}\right)}_{B} \tag{3}$$

The term A corresponds to the sum of the PSNR values resulting from the transmit selected packets. Term B denotes the QoS degradation caused by the packet loss.

Our objective is to maximize $\xi\left(f^i\right)$ via properly selecting a subset of the packets and via properly determining the transmission schedule. Our process consists of two phases: packet selection and packet transmission. In the packet selection phase, we choose a subset of the packets that does not exceed a given bandwidth envelope. The packet selection problem is equivalent to the knapsack problem where the size and significance of $f^i_{j,k}$ corresponds to the weight and value of an item in the knapsack problem, respectively. The capacity constraint of the knapsack problem is determined by the bandwidth envelope as $\mathcal{U} = \int_{t_0}^{t_0+\omega}\rho(t)dt$, where $t_0$, $\omega$ and $\rho(t)$ denote the start time of the window, its length and the available bandwidth at $t$, respectively. We take a greedy approach in selecting the packets to transmit. The selection criteria, $\epsilon\left(f^i_{j,k}\right)$ is the ratio between the packet significance and its size, i.e. $\epsilon\left(f^i_{j,k}\right) = \frac{\mathcal{Q}\left(f^i_{j,k}\right)}{\mathcal{S}\left(f^i_{j,k}\right)}$. The algorithm sorts $f^i_{j,k} \in f^i$ with respect to the decreasing order of $\epsilon\left(f^i_{j,k}\right)$ and selects $f^i_{j,k} \in f^i$ one by one from the sorted list until the sum of the size of the selected information exceeds the capacity constraint $\mathcal{U}$. The schedule should satisfy the bandwidth constraints, i.e. $\sum_{f^i_{j,k}\in\mathcal{I}(f^i)} \mathcal{S}\left(f^i_{j,k}\right) \leq \mathcal{U}$.

### 5.2 Significance-based packet interval allocation: when to transmit

Once we determine the set of packets to transmit, we need to determine the packet transmission schedule of the selected packets. Determining a transmission schedule is equivalent to determining an interval between the packet departures. In the packet selection phase, we properly select a subset of packets that does not exceed the bandwidth availability. However, the possibility of packet loss still exists. The network queue depth at the intermediate node or at the client may not be sufficient to accommodate all selected packets when they arrive in a bursty manner. The sender needs to control the transmission interval to avoid packet loss. We incorporate the packet significance in determining its interval. The key idea is to assign a larger interval to more important packets. Let $\delta\left(f^{i}_{j,k}\right)$ denote the time interval between $f^{i}_{j,k}$

```
SAPS($e$)
$k \leftarrow 1$
for $i \leftarrow 1$ to N
    do $F \leftarrow GoP[i]$
        for $j \leftarrow 1$ to $F.layers$
            do $A[k] \leftarrow F[j]$
                $k \leftarrow k + 1$
// Every data within each GoP is stored in the the array A

SORT-QOS($A$)
// Data in array A is sorted
// with respect to its packet significance

$sum \leftarrow 0$
for $i \leftarrow 1$ to $GoP.layers$
    do $sum \leftarrow sum + A[i].size$
        if $c < sum$
          then break
          else $B[i] \leftarrow A[i]$
// Selecting the later to transmit within available bandwidth c

REORDER($B$)

// Reordering selected packets in the decoding order
$total \leftarrow 0$
for $i \leftarrow 1$ to $B.layers$
    do $total \leftarrow total + \left((B[i].\,size)\,(B[i].\,QIF)^{e}\right)$

$k \leftarrow 1$
for $i \leftarrow 1$ to $B.layers$
    do $d \leftarrow \lceil B[i].\,size\,/\mathrm{MTU} \rceil$
        for $j \leftarrow 1$ to $d$
            do $P[k] \leftarrow$ SLICE($B[i], j$)
                $P[k].time \leftarrow \left((B[i].\,size)\,(B[i].\,QIF)^{e}\right) / (d \times total \times \mathrm{T} \times GoP.frames)$
                $k \leftarrow k + 1$

return $P$
// Packetization process and Packet interval allocation
// process based on packet size and its significance
```

**Fig. 8** Pseudo code for significance aware packet scheduling

and its immediate predecessor. If the value of $\delta\left(f^i_{j,k}\right)$ is 30 ms, then it would wait for 30 ms after $f^i_{j,k-1}$ has been transmitted. $\delta\left(f^i_{j,k}\right)$ is defined as in (4).

$$\delta\left(f^i_{j,k}\right) = \frac{\mathcal{S}\left(f^i_{j,k}\right) \times \mathcal{Q}\left(f^i_{j,k}\right)}{\sum_{F\left(f^i_{j,k}\right)\in\mathcal{I}(f^i)} \mathcal{S}\left(f^i_{j,k}\right) \times \mathcal{Q}\left(f^i_{j,k}\right)} \times \text{Time duration of one GoP length} \quad (4)$$

The typical time duration of one GoP length is 1 s. In this experiment, we varied the time duration of one GoP length from 1 to 8 s. The size of $f^i_{j,k}$ can be greater than the maximum transfer unit size and it can span multiple packets. When $f^i_{j,k}$ consists of multiple packets, we evenly distribute these packets at allocated intervals. The interval among the packets in $f^i_{j,k}$ is computed as $\delta\left(f^i_{j,k}\right)/\left\lceil \frac{\mathcal{S}\left(f^i_{j,k}\right)}{MTU} \right\rceil$.

Figure 8 illustrates the algorithm. The algorithm consists of three parts. From line 1 to line 7, it stores all the packets within the GoP into an array A. Then, the data in the array is sorted with respect to its packet significance value. From line 10 to 17, it selects the layers to transmit within the available bandwidth $c$. The layers in array A that will be transmitted are sorted with the value of $\mathcal{Q}\left(f^i_{j,k}\right)$,but that sorted order is not appropriate for the client to decode them sequentially. Hence, layers that are to be transmitted are sorted again in the decoding order. This process is shown in line 17. From line 19 to the end, the scheduler determines the transmission interval based upon the packet significance value.

## 6 Performance evaluation

### 6.1 Experiment setup

We compare three packet scheduling algorithms: Significance Aware Packet Scheduling (SAPS), Size Based Packet Scheduling (SBPS), and Bit-rate based best effort packet scheduling (Best-Effort). These algorithms have different criteria in selecting packets and setting up a transmission schedule. The packet scheduling algorithm can be divided into three processes: the layer selection process, the packet selection process, and the interval allocation process. The layer selection phase is identical in SAPS, SBPS, and Best-Effort (BE). The SAPS, SBPS, and BE algorithms determine the highest layer to transmit multi-layer encoded video based on the given available bandwidth. In this layer selection phase, the packet scheduler determines the smallest number of layers that exceed the bandwidth availability. Let us assume that we require 250 KByte/s and 320 Kbytes/s to transmit $L_1$ $L_2$, and $L_1$, $L_2$ and $L_3$, respectively, and the current available bandwidth is 300 KByte/s. Then, the highest layer to be transmitted is determined to be $L_3$, and the packets belonging to $L_1$, $L_2$ and $L_3$ information are subject to the following packet selection process. In the case of streaming a single-layer encoded video, this process is omitted. After the layer selection process, the SAPS and SBPS algorithms select a subset of the packets from each GoP in order not to exceed the available bandwidth. SAPS selects the subset of packets based upon the packet significance. The SBPS algorithm selects the packets from the beginning of GoP until it reaches its limit. BE transmits all packets in the layers selected in the layer selection phase. SAPS and SBPS algorithms take efforts not to exceed the available bandwidth so that it can minimize the packet

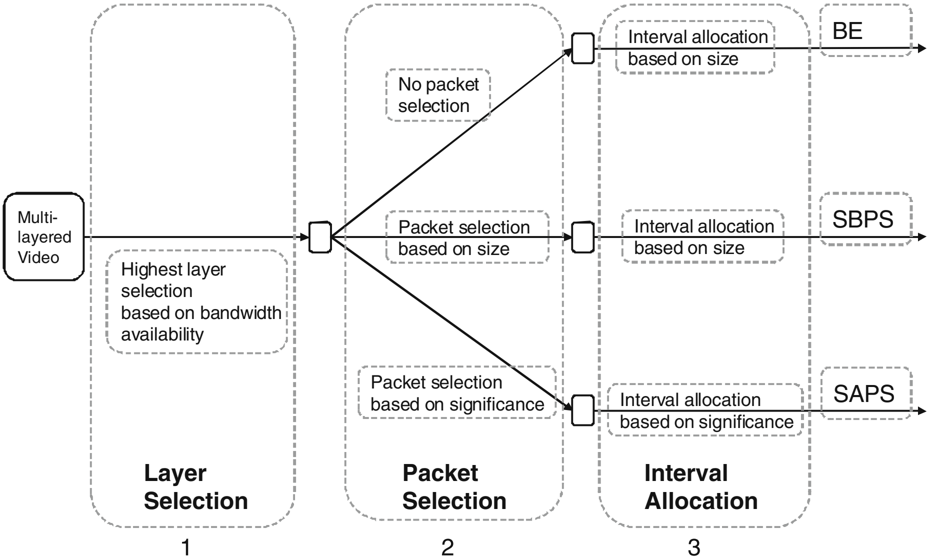

Multi-layered Video
Highest layer selection based on bandwidth availability
Layer Selection
1
No packet selection
Packet selection based on size
Packet selection based on significance
Packet Selection
2
Interval allocation based on size
Interval allocation based on size
Interval allocation based on significance
Interval Allocation
3
BE
SBPS
SAPS

**Fig. 9** Packet scheduling algorithm

loss. Packets sent by the BE algorithm are therefore exposed to random loss due to the bandwidth fluctuations. Once packets are selected, each algorithm determines the packet transmission interval. While the SAPS scheme allocates the transmission interval based on the packet significance, the SBPS and Best-Effort allocate the packet intervals based on the packet size. A larger sized packet is allocated a larger interval than the smaller sized packet. These three process are illustrated in Fig. 9.

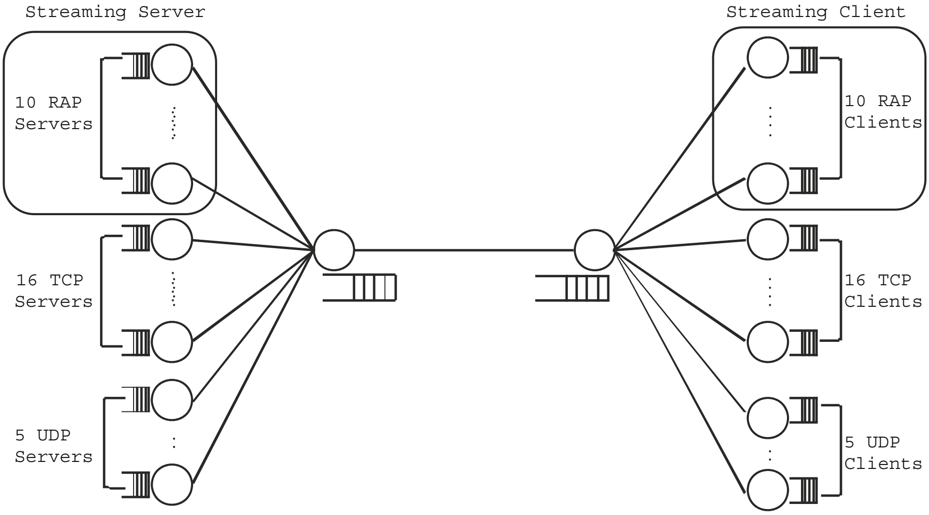


**Fig. 10** Topology of the experiment setup

**Table 2** Network simulator 2 (NS-2) parameters

| Parameter | Value |
|---|---|
| Bottleneck bandwidth | 6,000 kbits/s–50,000 kbits/s |
| Bottleneck delay | 100 ms |
| Bottleneck queue type | Drop tail |
| TCP Source type | TCP/reno |
| UDP Source type | UDP/CBR |

A simulation was performed using the Network Simulator 2 (NS-2) [28]. Our network topology has a dumbbell setting as illustrated in Fig. 10. There are 31 node pairs: 31 servers and 31 clients. Ten nodes were designed for the streaming server. Each of these nodes services one streaming client. The rate adaptive protocol (RAP) was used to service these streaming clients [25]. We generate background traffic, which is a mixture of TCP and UDP traffic. The reason to generate the background traffic is primarily to more closely model the real situation. A total of 16 TCP and 5 UDP node pairs share the link. The File transfer protocol (FTP) and the hypertext transport protocol (HTTP) application sessions run over TCP and the constant bit rate (CBR) video streaming application runs over UDP, respectively (Table 2).

We used actual video clips in our experiment (Table 3). These video clips and their respective packet traces are publicly available at [29]. We used a total of eight video clips, each of which has a different scene nature: color histogram, motion dynamics, and etc. We performed an experiment using an extensive set of video clips to examine how the proposed SAPS algorithm behaves under different video contents. Some experimental results are omitted where there is not much difference between the video clips. Among the eight video clips, two of them were layer encoded (one base layer and three advanced layers) and six of them were encoded with a single-layer. Table 4 illustrates the bandwidth allocation for each layer. Table 3 presents the basic information on video in the clips.

**Table 3** Frame Statistics, (Clip 1:*Akiyo*, Clip 2:*Foreman*, Clip 3:*Grandmother*, Clip 4:*Mother and Daughter*, Clip 5:*Salesman*, Clip 6:*Miss AM*, Clip 7:*Starcraft* and Clip 8:*Boa*)

| File | Clip1 | Clip2 | Clip3 | Clip4 | Clip5 | Clip6 | Clip7 | Clip8 |
|---|---|---|---|---|---|---|---|---|
| Compression type | MS MPEG-4(MP43) | | | H.264/MPEG-4 AVC | | | MPEG-4 FGS | |
| Frame rate ($frame/s$) | 30 | | | | | | | |
| Picture size ($pixel$) | 176 * 144 | | | | | | 720 * 480 | 720 * 520 |
| Number of Frames | 300 | 400 | 870 | 930 | 420 | 930 | 17,160 | 7,680 |
| Time ($s$) | 10 | 13.3 | 29 | 31 | 14 | 4 | 572 | 256 |
| Bit-rate ($kbit/s$) | 183 | 346 | 265 | 306 | 285 | 323 | 916 | 797 |
| Mean ($byte$) | 879 | 1741 | 1166 | 1,275 | 1,186 | 1,347 | 3,819 | 3,322 |
| Variance ($byte^2$) | 1071,- | 847,- | 481,- | 1,528,- | 1,845,- | 1,010,- | 12,510,- | 6,643,- |
| GoP structure | $I(P)^{249}$ | | | IBBPBBPBBP | | | | |
| Encoder | MEncoder [20, 23] | | | | | | | |

**Table 4** Bandwidth allocation for individual layers

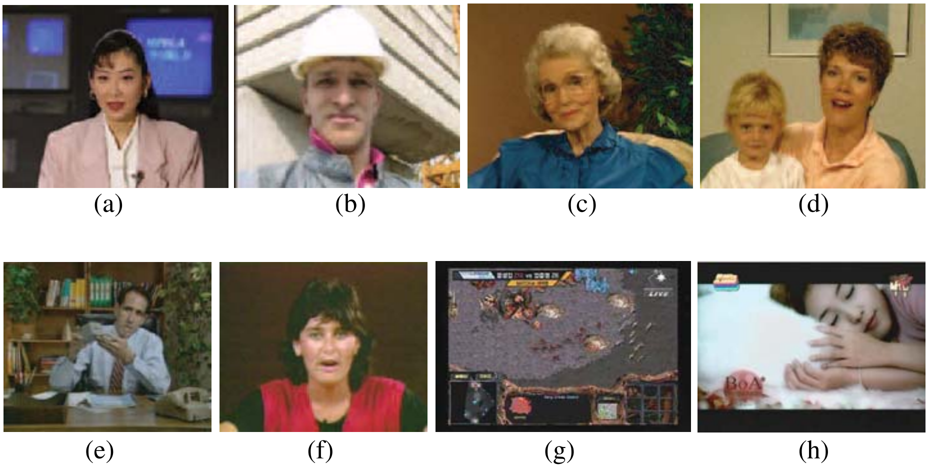
| Layer | Layer | Cumulative BDW (Kbits/s) | Subscriber line type |
|---|---|---|---|
| 1 | 76.8 | 76.8 | 128 Kbits/s Dual ISDN |
| 2 | 153.6 | 230.4 | 384 Kbits/s DSL or Cable Modem |
| 3 | 230.4 | 460.8 | 768 Kbits/s DSL or Cable Modem |
| 4 | 439.2 | 900 | 1.5 Mbits/s or over DSL |

Figure 11 shows a sample scene in each video clip.

The objective of our study is to improve the *user perceivable QoS*. We do not intend to improve the packet loss or frame loss. Therefore, it is critical that our simulation environment properly reconstructs video scenes with incoming packets. We implemented a frame decoding engine in the client's node. The decoding engine assembles the incoming packets and reconstructs the image. We compute the PSNR value for each frame reconstructed at the client side. A streaming client starts displaying video 2 s after the beginning of transmission. If packets arrive out of the order sequence, the respective packet stays at the queue until all of the required packets arrive before the play-out deadline or are discarded. When one or more packets in a frame are not delivered and the client is not able to recover the original frame, all the packets consisting of one frame are discarded. If a frame is lost during the transmission or arrives later than the deadline, the previous frame concealment scheme is used at the decoder [6].

## 6.2 Frame size and packet significance distribution

Here, we examine the frame size and packet significance distribution.


(a) (b) (c) (d)

(e) (f) (g) (h)

**Fig. 11** Sample scene of each encoded video. **a** Akiyo. **b** Foreman. **c** Grandmother. **d** Mother and daughter. **e** Salesman. **f** Miss AM. **g** Starcraft. **h** Boa

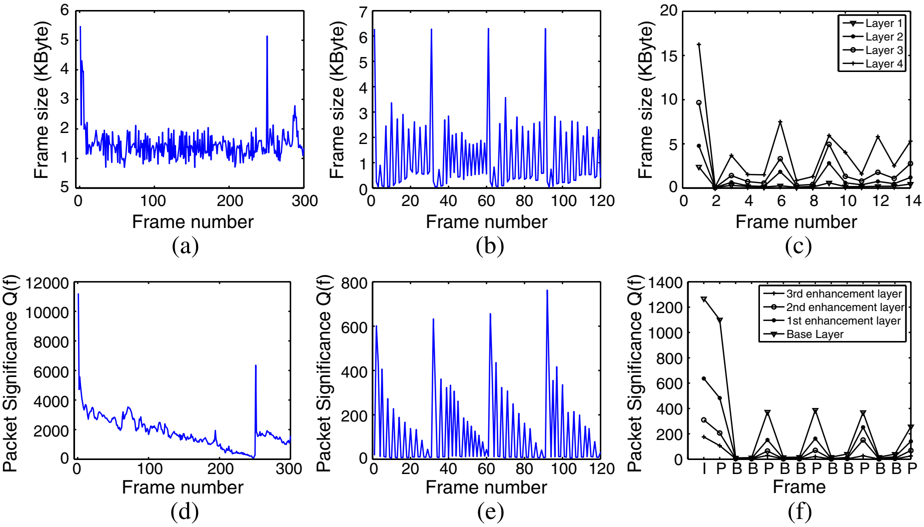


**Fig. 12** Frame size and packet significance. **a** Frame size (*Foreman*). **b** Frame size (*Salesman*). **c** Frame size distribution (*Starcraft*). **d** Packet significance (*Foreman*). **e** Packet significance (*Salesman*). **f** Packet significance (*Starcraft*)

We used different encoding scheme (Microsoft MPEG-4 v3, H.264, and MPEG-4 FGS [20, 23]) and GoP structures and examined the effect of these parameters on packet significance.

Figure 12 illustrates the frame size and the respective packet significance over the packet size of video traces of *Foreman*, *Salesman* and *Starcraft*. The GoP structure of each video trace is $IP^{249}$, $IBBPBBP$, and $IBBPBBP$ with the layer encoding scheme, respectively. In Fig. 12a and d, the I frame is located every $250^{th}$ frame. The packet significance of P frames immediately after I frame has a higher value than the P frames farther from the I frame, whether the corresponding frame size is large or not. The P frame that appears relatively early in the GoP has a larger packet significance value. This is because it has more child frames as it appears earlier within the GoP, and therefore the packet significance increases. Figure 12b and e illustrate the packet significance distribution over different traces, the GoP structure, the GoP size and the encoding scheme. The video trace of the GoP structure is IBBPBBPBBP with a size of 30 and encoded with H.264. As can be seen in the figures, the I frame, which is located at every $30^{th}$ frame, has a high packet significance value. We can find an interesting phenomenon in Fig. 12d and e. The packet significance values are an order of magnitude larger in Fig. 12d than in Fig. 12e. The main cause is the respective GoP size. The GoP size of *Foreman* and *Salesman* are 250 and 30, respectively. The *Foreman* content is generated primarily for mobile multimedia streaming application where interactive playback is not permitted and saving bandwidth is of prime concern. The single I frame covers 8.3 s of playback length in the *Foreman* video clip and therefore packets that appear in the front part of the GoP are highly significant. The packet significance value is smaller in the Salesman video clip because the GoP size is 30.

In both Fig. 12d and e, the P frame has a relatively high significance value when it is located near the I frame. In contrast, the B frames have a very low significance value compared to the I or P frames because they have much fewer dependent frames. Figure 12c and f illustrate the size and significance value of the scalable encoded video traces for *Starcraft*. The encoding rate is about 900 Kbits/s and there are four layers.

### 6.3 Bandwidth availability and user perceivable QoS

We examined the effectiveness of Significance-Aware Packet Scheduling under conditions of different bandwidth availability. The queue depth at the bottleneck link was set to 100,000. We compared three packet scheduling algorithms: (1) Significance-Aware Packet Scheduling (SAPS), (2) Size Based Packet Scheduling (SBPS) and (3) pre-defined bit-rate based best effort packet scheduling (Best-Effort). The latter two algorithms do not incorporate the QoS importance of a packet. We used four real video traces in performing our experiment [29]. Two of them were layer encoded (Clip 7 and Clip 8) and the rest were single-layer encoded contents (Clip 4 and Clip 5). The reason we test both layer encoded and single-layered video streams was to examine the effectiveness of the layered encoding scheme under limited bandwidth availability.

In this experiment, we examined the PSNR value and packet loss behavior of the three scheduling schemes under varying bottleneck bandwidths. Figures 13 and 14 illustrate the results. Figure 13 illustrates the user perceivable QoS of the three scheduling schemes for four video clips. As can be seen, under all circumstances, SAPS exhibits the best QoS. This is due to the fact that SAPS selected and successfully transmitted frames that have a higher impact on the decoded video. However, as available bandwidth becomes larger, the advantage of SAPS becomes less significant.

Figure 13a and b illustrate the PSNR value for multi-layer encoded video traces. Figure 13c and d illustrate the PSNR value for a single-layer encoded video. We varied the bottleneck bandwidth from 6 Mbits/s to 51.3 Mbits/s, which corresponds to the client bandwidth approximately from 128 Kbits/s to 990 Kbits/s. When the bottleneck link bandwidth is high, most of the packets successfully reach the client, and there is no significant difference in user perceivable QoS (PSNR) among SAPS, SBPS, and Best-Effort. When the bottleneck link bandwidth is smaller, the difference becomes more significant and the advantage of adapting packet significance becomes more significant (Fig. 13c and d).

Figure 14 illustrates the packet transmission behavior for three different packet scheduling schemes. We examine the fraction of successfully transmitted as well as lost packets under varying bandwidth. Figure 14a, b, c, and d illustrate the performance results in clip 7 (*Starcraft*). The X-axis denotes the bottleneck link bandwidth. For each bottleneck link bandwidth, there are three bars, each of which denotes SAPS, SBPS, and Best-Effort scheduling schemes. With 41.3 Mbits/s bottleneck bandwidth, SAPS and SBPS selects the same amount of packets for transmission (48%) after the layer selection and packet selection process. The Best-Effort scheduling scheme selects 68% of packets for transmission after the layer selection process. Let us look at the fraction of successfully delivered packet: $\frac{\text{Size of successfully delivered packets}}{\text{Total packet size}}$. Even though the Best-Effort

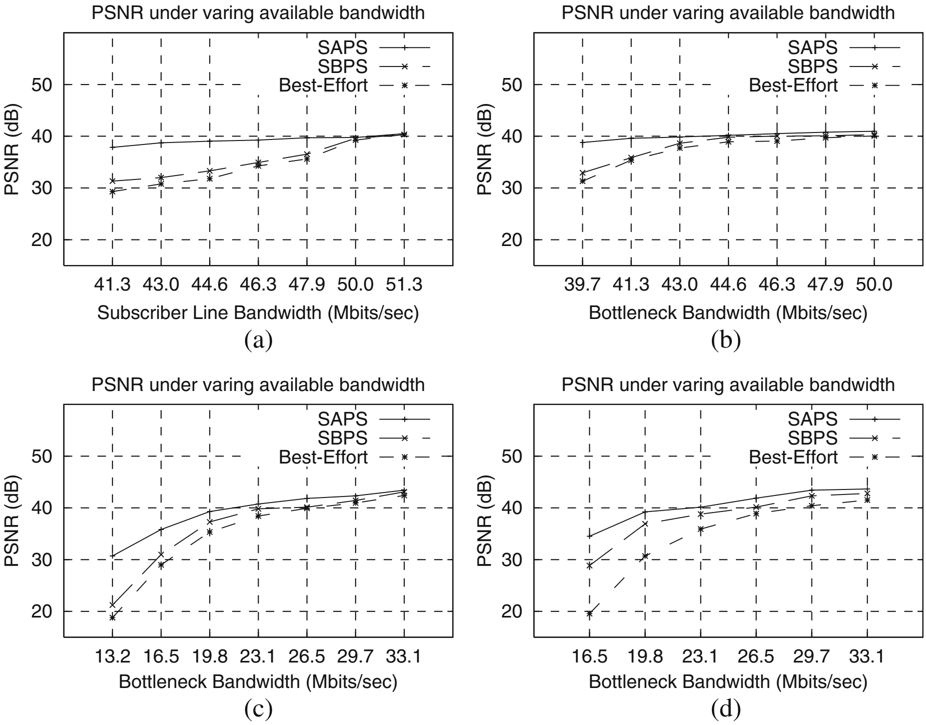
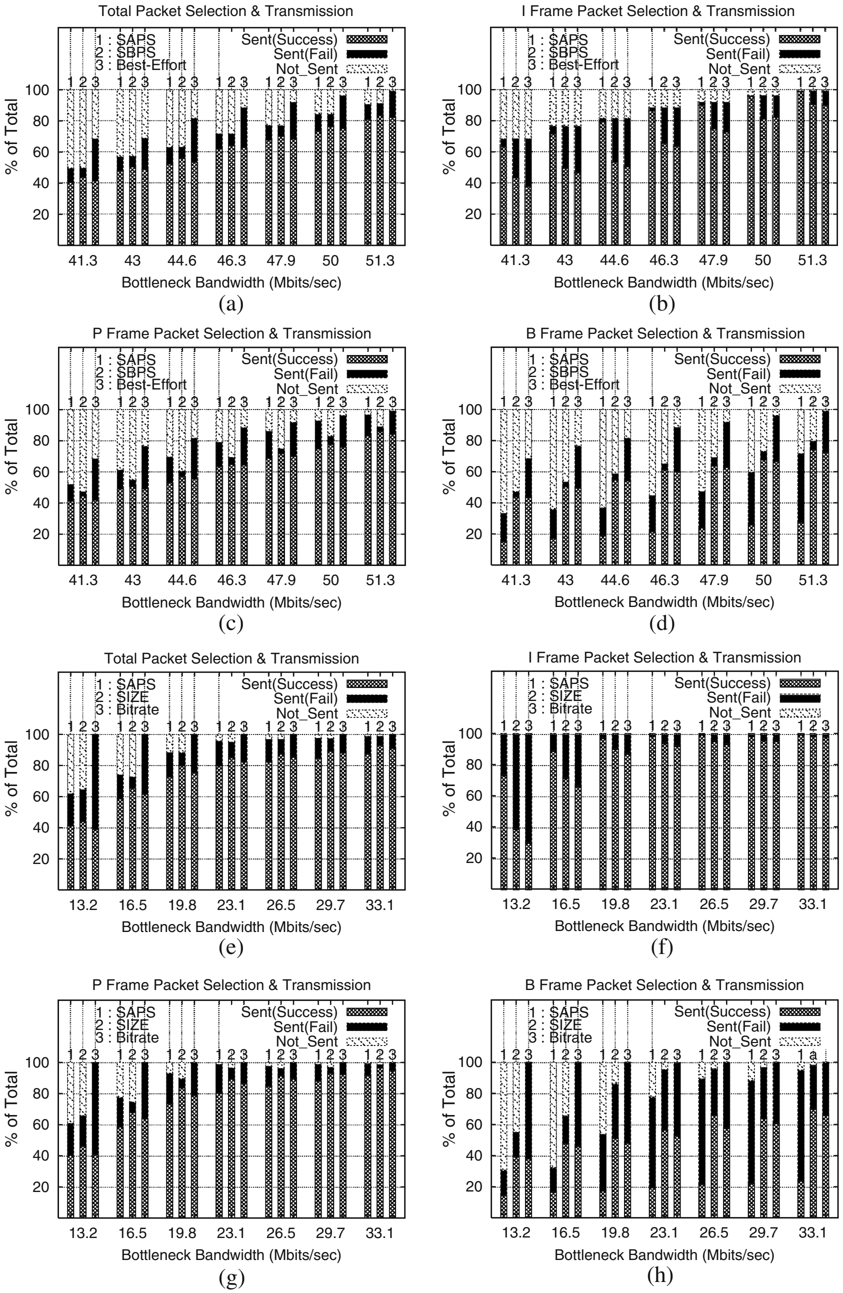


**Fig. 13** PSNR under varying bottleneck bandwidth. **a** PSNR (*Starcraft*). **b** PSNR (*Boa*). **c** PSNR (*Mother and daughter*). **d** PSNR (*Salesman*)

scheduling scheme sent the largest number of packets, the fraction of successfully delivered packets were approximately the same in all scheduling schemes. Let us take a detailed look at the packet transmission behavior. We observed that the SAPS yields the best QoS among the three scheduling schemes. Figure 14b, c, and d illustrate the results. Each of the three figures illustrates the packet transmission behavior for the I, P, and B frames, respectively. If all the I frames are sent, the value becomes 100% in Fig. 14b. When the bottleneck bandwidth is 41.3 Mbits/s, 63% of the I frames are sent in all three scheduling schemes. All the schemes select the same fraction of I frames. However, the largest fraction of these packets is successfully delivered in SAPS. This is because SAPS effectively protects the I frame by allocating sufficient intervals with preceding packets, and therefore the I frame packets are less vulnerable to loss. SBPS has a higher success rate than the Best-Effort scheme. Even though both schemes transmit the same amount of I frame packets, the Best-Effort scheduling scheme transmits a larger number of packets when considering all frame types (I, P, and B). I frame packets in Best-Effort scheduling schemes are mostly likely to get exposed to failure. In examining Fig. 14d we can get a clearer idea. This figure illustrates the B frame packet's behavior. With the 41.3 Mbits/s bottleneck link bandwidth, 32%, 47%, and 63 % of the B frame packets are transmitted in the SAPS, SBPS, and Best-Effort scheduling schemes, respectively, while 13%, 43%, and 42% from the total B frame packets are successfully transmitted. As we can see, SAPS has a higher packet loss rate in sending the B frames.



(a) (b) (c) (d) (e) (f) (g) (h)

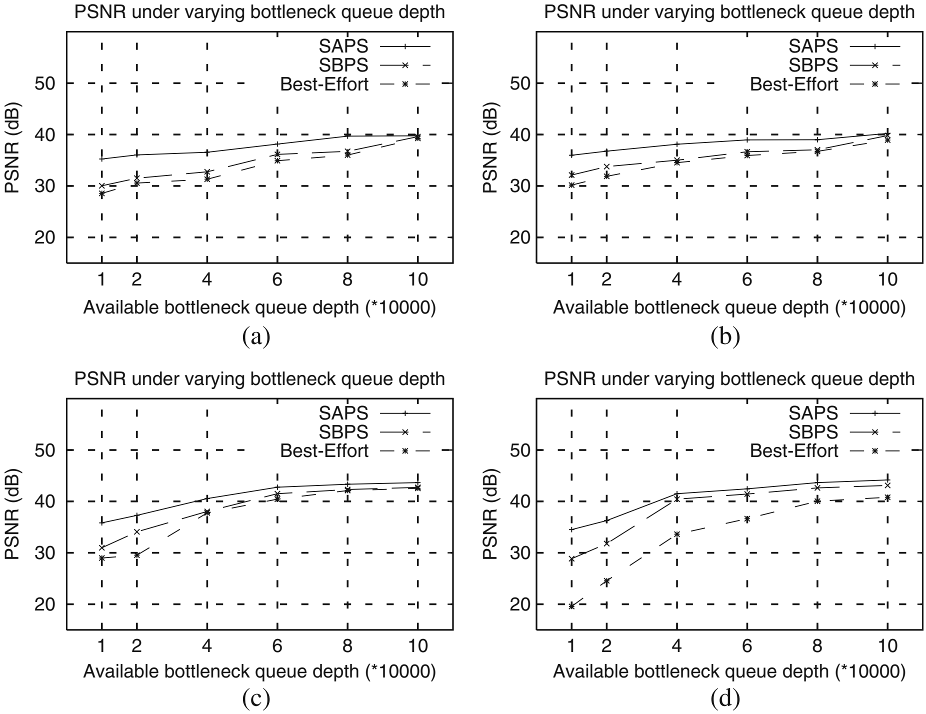
**Fig. 14** Packet loss. **a** Total frame loss (*Starcraft*). **b** I frame loss (*Starcraft*). **c** P frame loss (*Starcraft*). **d** B frame loss (*Starcraft*). **e** Total frame loss (*Mother and daughter*). **f** I frame loss (*Mother and daughter*). **g** P frame loss (*Mother and daughter*). **h** B frame loss (*Mother and daughter*)

In summary, SAPS effectively maximizes the user perceivable QoS by properly incorporating packet semantics in selecting and transmitting packets. Figure 14e to h denote the results for the single-layer encoded video *Mother and Daughter*. Similarly, SAPS yields the best I frame packet success rate and the worst B frame packet success rate. We performed experiments for all eight video clips. Since the results of the experiments were similar, we omitted the results obtained from the other video clips.

### 6.4 Queue depth of the bottleneck link and user perceivable QoS

In this section, we vary the bottleneck queue depth from 10,000 to 100,000 with fixed bottleneck bandwidth availability and examine the performance of each scheme for scalable video traces. We set the bottleneck bandwidth high enough so that we could accurately analyze the ability of adapting to the variable bottleneck queue depths.

Figure 15a and b illustrate the PSNR value variations of the three schemes under varying bottleneck queue depths. When the bottleneck queue is sufficiently large, the PSNR values of the three schemes are almost the same. However, as the bottleneck queue gets shorter, the difference in the three schemes becomes more significant. For example, when the bottleneck queue size is 40,000, the PSNR value of the two video traces for SAPS, SBPS, and Best-Effort scheme are (36, 32, 31) and (38, 35, 34), respectively.



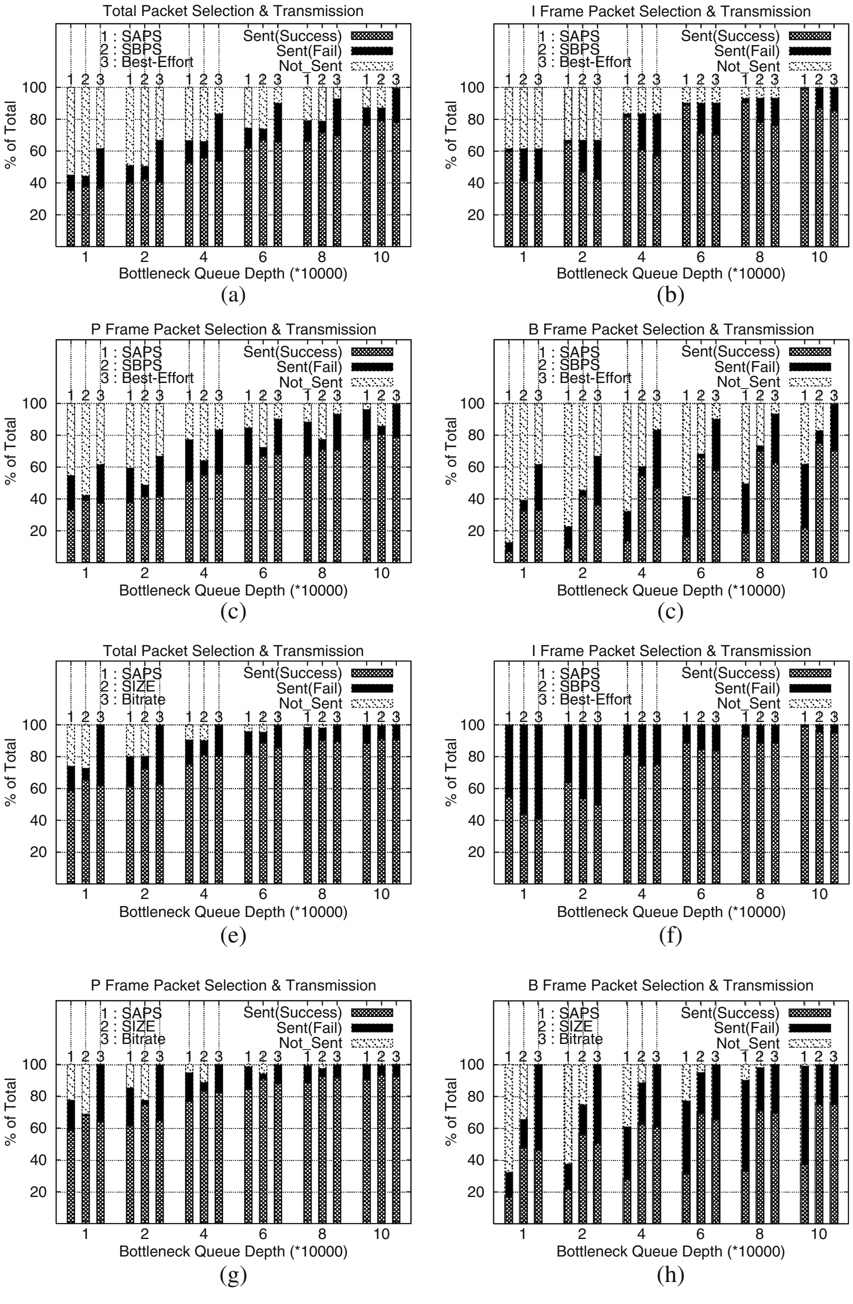
(a)

(b)

(c)

(d)

**Fig. 15** PSNR under varying bottleneck queue depths. **a** PSNR (*Starcraft*). **b** PSNR (*Boa*). **c** PSNR (*Mother and daughter*). **d** PSNR (*Salesman*)



Total Packet Selection & Transmission

(a)

I Frame Packet Selection & Transmission

(b)

P Frame Packet Selection & Transmission

(c)

B Frame Packet Selection & Transmission

(c)

Total Packet Selection & Transmission

(e)

I Frame Packet Selection & Transmission

(f)

P Frame Packet Selection & Transmission

(g)

B Frame Packet Selection & Transmission

(h)

**Fig. 16** Packet loss under varying bottleneck queue depths. **a** Total frame loss (*Starcraft*). **b** I frame loss (*Starcraft*). **c** P frame loss (*Starcraft*). **d** B frame loss (*Starcraft*). **e** Total frame loss (*Mother and daughter*). **f** I frame loss (*Mother and daughter*). **g** P frame loss (*Mother and daughter*). **h** B frame loss (*Mother and daughter*)

Figure 16a illustrates the results of the packet transmission behavior of *starcraft* under varying bottleneck queue depths. The larger fraction of packets was selected as the bottleneck queue becomes larger in SAPS and SBPS. The Best-Effort scheme transmits all the packets, so it shows 0 percentage of Not_Sent. The percentage of total selected packets for the SAPS and SBPS schemes is substantially the same. The SBPS scheme yields a slightly higher total frame success ratio than the other two schemes. However, this did not result in a higher PSNR value. Figure 16b illustrates the fraction of selected I frames from all the I frames for the three schemes. The SAPS scheme selects almost all the I frames (more than 99%) and the value of Sent(Fail) is very low, even when the bottleneck queue is very small. SBPS and Best-Effort schemes, however, illustrate the different results in terms of success ratio. Although the SBPS and Best-Effort schemes selected all the I frames for transmission, a smaller fraction of the I frame packets were successfully delivered to the client. This is because the SAPS scheme allocates a longer interval based on its packet significance value, while the two other schemes allocate intervals based on the size of the frame. Figure 16e illustrates the total frame success rate over the variable bottleneck queue depth. As shown in the figure, the SBPS scheme illustrates the highest frame success rate. However, SAPS successfully transmitted more important packets such as the I frame compared with SBPS or Best-Effort schemes. Figure 16f illustrates this. The SAPS scheme intentionally makes traffic burstier for low significance packets such as B frame packets while making traffic less burstier for more important packets.

## 7 Conclusion

Providing better quality multimedia services under limited network and computational resource restrictions is an everlasting technical challenge. Various efforts have been dedicated to exploiting the underlying network resource: traffic shaping, minimizing packet loss, forward error correction, error resilient coding and scalable encoding. The ultimate goal of all these efforts is to maximize the user perceivable QoS. However, each of the above mentioned topics has been implemented in separate contexts and further these metrics are not directly proportional to each other. Traffic shaping and packet loss minimization have been handled as Transport Layer issues. Forward error correction, error resilient coding and scalable encoding have been in the picture coding domain. All of these works have their own optimization criteria, e.g. packet loss, rate variability, Signal-to-Noise Ratio, but to maximize effectiveness, it is mandatory that all of the above mentioned knobs need to be addressed in a unified framework to maximize the user perceivable QoS. In this work, we develop an elaborate model to present the QoS importance of a packet termed the “Packet Significance.” The significance of the packet incorporates the inter-frame dependency and inter-layer dependency of a given packet. We use packet significance in determining the packet transmission schedule. Using significance aware packet scheduling, we can improve the user perceivable QoS; however, the packet loss probability actually becomes worse as a result. The significance-aware packet scheduling algorithm manifests itself when underlying video frames are “not” layer encoded. Even though the contents are not layer encoded, it properly captures the QoS importance of the individual packets and properly selects the subsets of packets so that it efficiently exploits the available bandwidth.

## References

1. Aras CM, Kurose J, Reeves DS, Schulzrinne H (1994) Real-time communication in packet-switched networks. In: Proceedings of the IEEE, vol 82, pp 122–139
2. Argyriou A (2008) Cross-layer error control for multimedia streaming in wireless/wireline packet networks. IEEE Trans Multimedia 10(6):1121–1127
3. Cai H, Zeng B, Shen G, Xiong Z, Li S (2007) Error-resilient unequal error protection of fine granularity scalable video bitstreams. EURASIP J Appl Signal Process 2006(1):118–118
4. Chakareski J, Frossard P (2005) Distributed packet scheduling of multiple video streams over shared communication resources. In: Proceedings on multimedia signal processing, IEEE, Shanghai, China, pp 1–4
5. Chan HA (2007) Comparing wireless data network standards. In: Proceedings of AFRICON 2007, Windhoek, South Africa, pp 1–15
6. Chen Y, Hu Y, Au O, Li H, Chen C (2008) Video error concealment using spatio-temporal boundary matching and partial differential equation. IEEE Trans Multimedia 10(1): 2–15
7. Chou P, Miao Z (2006) Rate-distortion optimized streaming of packetized media. IEEE Trans Multimedia 8(2):390–404
8. D'Auria B, Resnick S (2006) Data network models of burstiness. Trans Adv Appl Probab 38(2):373–404
9. Delgado G, Frias V, Igartua M (2006) Video-streaming transmission with qos over cross-layered ad hoc networks. In: Proceedings of SoftCOM 2006, Dubrovnik, Croatian, pp 102–106
10. Dubois J (2007) Burstiness reduction of a doubly stochastic ar-modeled uniform activity vbr video. Transactions on World Academy of Science, Engineering and Technology 23:454–458
11. Giordano S, Pagano M, Pannocchia R, Russa F (1996) A new call admission control scheme based on the self similarnature of multimedia traffic. In: Proceedings of IEEE communications, conference record, converging technologies for tomorrow's applications, 1996, Dallas, TX, USA, pp (3)1612–1618
12. Givoni M (2006) Development and impact of the modern high-speed train: a review. Transactions on Transdisciplinary Journal, Transport Reviews 26(5):593–611(19)
13. Ha V, Choi S, Jeon J, Lee G, Shim W (2004) Portable receivers for digital multimedia broadcasting. IEEE Trans Consum Electron 50(2):666–673
14. Hei X, Liang C, Liang J, Liu Y, Ross K (2007) A measurement study of a large-scale p2p iptv system. IEEE Trans Multimedia 9(8):1672–1687
15. Huan Yu C, Hongshen C, Jenqneng H, Jianshung W (2005) Bitplane coding of DCT coefficients for image and video compression. In: Proceedings on circuits and systems, 2005. ISCAS 2005. IEEE International Symposium on, Kobe, JAPAN, vol 4, pp 3419–3422
16. Information technology (2002) Coding of audio-visual objects—part 2: visual
17. Kim G, Kim J (2007) Wavelength division multiplexing-passive optical network based ftth field trial test. J Opt Soc Korea 11(3):101–107
18. Kim T, Ammar MH (2003) Optimal quality adaptation for MPEG-4 fine-grained scalable video. In: Proceedings of IEEE INFOCOM 2003 San Francisco, CA, USA
19. Lam SS, Chow S, Yau DK (1994) An algorithm for lossless smoothing of mpeg video. Transactions on SIGCOMM Comput Commun Rev 24(4):281–293
20. Li W (2001) Overview of fine granularity scalability in MPEG-4 video standard. IEEE Trans Circuits Syst Video Technol 11(3):301–317
21. Mansour H, Nasiopoulos P, Krishnamurthy V (2008) Real-time joint rate and protection allocation for multi-user scalable video streaming. In: IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008, Juan-les-Pins, France. ACM, New York, pp 1–5
22. Mayer-Patel K, Le L, Carle G (2002) An mpeg performance model and its application to adaptive forward error correction. In: IMC '02: Proceedings of the tenth ACM international conference on multimedia, Juan-les-Pins, France. ACM, New York, pp 1–10
23. MEncoder (2009) Program for encoding video+audio [online]. http://www.mplayerhq.hu/

24. Politis I, Tsagkaropoulos M, Pliakas T, Dagiuklas T (2007) Distortion optimized packet scheduling and prioritization of multiple video streams over 802.11e networks. Transactions on Advances in Multimedia 2007(1):1–11. doi:10.1155/2007/76846
25. Rejaie R, Handley M, Estrin D (1999) RAP: an end-to-end rate-based congestion control mechanism for realtime streams in the internet. In: Proceedings of IEEE INFOCOM, New York, NY, USA, vol 3, pp 1337–1345
26. Richardson IE (2003) H.264 and MPEG-4 video compression: video coding for next-generation multimedia. Wiley, New York
27. Salehi JD, Zhang Z, Kurose J, Towsley D (1998) Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing. IEEE Trans Netw 6(4):397–410
28. The network simulator - ns-2 Information Sciences Institute [online]. http://nsnam.isi.edu/nsnam/index.php/user_information
29. Video traces file [online]. http://www.dmclab.hanyang.ac.kr/data/mpeg2data/video_ traces.htm
30. Watkinson J (2007) MPEG handbook. MIT, Cambridge
31. Won Y, Shim B (2002) Effect of VBR traffic smoothing on broadband wireless Internet. In: Proceedings of SPIE ITCOM 2002, Boston, MA, USA, vol 4865, pp 225–233
32. Won Y, Shim B (2002) Empirical study of VBR traffic smoothing in wireless environment. In: Proceedings of the second international workshop on Innovative Internet Computing Systems, vol 2346 of Lect Notes Comput Sci. Springer, New York, pp 193–204
33. Won Y, Jung J, Jun Y, Chang I, Hong S (2007) Qos weighted scheduling: real-time streaming of multi-resolution video. In: Proceedings of Graphics and Visualization in Engineering (GVE 2007), Clearwater, Florida, USA
34. Wu J, Cai J, Chen C (2007) Single-pass rate-smoothed video encoding with quality constraint. IEEE Trans Signal process Letters 14(10):715–718
35. Zipper J, Stoger C, Hueber G, Vazny R, Schelmbauer W, Adler B, Hagelauer R (2007) A single-chip dual-band cdma2000 transceiver in 0.13 m cmos. IEEE J Solid-State Circuits 42(12):2785–2794

**Sungwoo Hong** is a Ph.D candidate in Distributed Multimedia Computing lab of Hanyang University. He majored in network, operating system and multimedia. His current research includes multimedia networking or multimedia delivery and wireless resource contron mechanism. For that, he is focusing on the Cross-Layer Optimization (CLO) which aggressively exploits information among different layers to maximize user perceivable QoS.

**Youjip Won** received the B.S. and the M.S. degree in Computer Science from Department of Computer Science and Statistics, Seoul National University, Korea in 1990 and 1992, respectively and Ph.D in Computer Science from the University of Minnesota, Minneapolis in 1997. After graduation, he worked for Server Architecture Lab, Intel as Server Performance Analyst till 1999. Since 1999, he has been on board of faculty members in Dept. of Electrical and Computer Engineering, Hanyang University, Seoul, Korea, where he is now Associate Professor. His research interests include Operating System, Computer Networks, Multimedia, Performance Analysis.