Available online at www.sciencedirect.com
ScienceDirect
Signal Processing: *Image Communication* 22 (2007) 809–832
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

# Adaptive bitstream switching of scalable video ☆

Osama A. Lotfallah[a,1], Geert Van der Auwera[b], Martin Reisslein[b,*]

[a]*Johnson Controls Inc., Milwaukee, WI 53202, USA*
[b]*Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287, USA*

Received 30 October 2006; received in revised form 8 June 2007; accepted 11 June 2007

**Abstract**

With scalable video coding that provides fine-granular quality degradation, such as fine granularity scalability (FGS) and progressive FGS (PFGS), or H.264 scalable video coding's (SVC) adaptive reference FGS (AR-FGS) coding, video can flexibly be streamed to receivers of heterogeneous bandwidths. However, the transmitted video is only efficiently encoded when the transmission bit rate is in the vicinity of the encoding bit rate. In this paper, we develop and evaluate a comprehensive suite of network-aware adaptive bitstream switching policies for point-to-point and point-to-multipoint streaming of fine granular scalable coded video to address this coding efficiency issue. Our approach stores a small number of encodings (versions) with different encoding bit rates for each video sequence and estimates the reconstructed quality using the motion activity levels of the underlying visual content (or, in general, any content descriptor(s) that highly correlate with the reconstructed quality). For unicast streaming, we then: (i) adaptively switch between the different encodings at the server, to improve the reconstructed video quality and (ii) adaptively drop packets during network congestion to ensure fairness between multiple unicast streams. For multicast streaming, we also adaptively switch between the different encodings to maximize the average video quality. Our adaptive bitstream switching policies consider the visual content descriptors as well as the network channel variability, while requiring only sample points from the rate-distortion curve of the video stream. From our extensive simulations with PFGS coding, we find that our adaptive unicast bitstream switching policy achieves on average a 0.8 dB improvement over the optimal non-adaptive streaming for a diverse 200-shot sequence from *Star Wars IV*. We have also verified our key findings with the latest scalable video coding standard, H.264 SVC.


*Keywords:* Adaptive streaming; Congestion control; Motion activity; Multicast; PFGS; Simulcast; SVC AR-FGS

## 1. Introduction

A key challenge of video delivery is to regulate the video transmission rate according to the network's and receiver's capabilities, i.e., to adapt the video transmission in a network- and client-aware manner. A potential approach for network- and client-aware video adaptation is the so-called simulcast or bitstream switching technique, which encodes

☆ A preliminary presentation of parts of this work appears in the Proceedings of the ACM Workshop on Advances in Peer-to-peer Multimedia Streaming 2005 [1].

*Corresponding author. Tel.: +1 480 965 8593; fax: +1 480 965 8325.
*E-mail addresses:* Osama.Lotfallah@jci.com (O.A. Lotfallah), Geert.Vanderauwera@asu.edu (G. Van der Auwera), reisslein@asu.edu (M. Reisslein).

[1] This work was conducted while O. Lotfallah was with Arizona State University, Tempe.

0923-5965/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2007.06.002

a given video at many different rates, i.e., into different versions, and then transmits the version with the highest rate that fits into the available network bandwidth. These techniques, while simple, have a number of significant drawbacks, such as the encoding of an impractically large number of versions (on the order of several tens of versions are required to adapt a Mbps video at the granularity of 100 Kbps). Also, there is no flexibility to scale down the bit rate/video quality of a stream during network transport, unless typically computationally demanding transcoding is performed at intermediate network nodes. Bitstream switching can also be applied over versions of different coding schemes, but an increase in the computational complexity of video decoding is inevitable [2–4]. Scalable (layered) video coding overcomes these drawbacks by encoding a video into a base layer and several enhancement layers. The base layer represents the basic video quality, which is typically transmitted with higher protection (often achieved with unequal error protection), and the enhancement layers, which are transmitted with lower protection, gradually improve the video quality [5–8]. A key limitation of layered video coding is that the video bit rate can only be adapted at the granularity of complete enhancement layers, whereby the number of layers is typically limited to a small number (at most 4–5 in practical encoders) resulting in rather coarse rate adaptation. Fine granularity layered coding techniques overcome this shortcoming by encoding the video into one base layer and one enhancement layer, whereby the enhancement layer bit rate can be finely adapted [6,9]. This flexibility in bit rate adaptation comes at the expense of relatively low compression efficiency. Progressive fine granularity scalability (PFGS) coding (also called two-loop fine granularity scalability (FGS) coding) overcomes this disadvantage and provides generally good compression efficiency as well as high flexibility in adapting the enhancement layer bit rate [10,11]. We verify our key PFGS findings with the H.264 adaptive reference FGS (AR-FGS) encoder [12] to assert that the same switching principles apply to state-of-the-art FGS schemes.

FGS coding makes it possible to encode the enhancement layer of the video with one (high, say 2 Mbps) bit rate and then to flexibly transmit the enhancement layer at any lower bit rate. However, there are compression inefficiencies: *for transmission at a low bit rate (say around 1 Mbps) the video could be coded (with the PFGS codec) much more efficiently by employing an enhancement layer encoding bit rate in the vicinity of the transmission bit rate. We quantify this efficiency gain achieved by selecting the encoding bit rate reasonably close to the actual transmission bit rate in Section 3.2 and demonstrate that it can reach on the order of 4 dB for some visual contents.* We also verify in Section 3.2, that for AR-FGS the quality gain can reach up to 1.8 dB for the same content. In summary, PFGS encoding—and, in general, any SNR scalable coding concept known today—is only optimized for the rate that is used during the encoding process [1,13]. Essentially all existing studies on PFGS video streaming have ignored the important aspect of the encoding bit rate selection and focus primarily on modeling the rate-distortion (R-D) characteristics of a given encoding or the optimal selection of the transmission bit rate based on the R-D model of one encoding [14,15].

Our main contribution in this paper is to develop and evaluate a comprehensive framework of network-aware adaptive bitstream switching policies for the streaming of scalable (layered) coded video. Our adaptive bitstream switching policies exploit video content features and cover common unicast and multicast streaming scenarios. While our framework applies generally to a wide range of scalable (layered) video coding schemes, to fix ideas, we focus on PFGS coding in our specific problem formulations. We optimally select the values of the encoding parameters, depending on visual content and network (channel) conditions so as to maximize the reconstructed video quality. Hence, we propose network-aware techniques for multimedia delivery using in-network processing application that utilizes the visual content descriptors, which can be stored at the video server for multimedia indexing as well as delivery adaptation. Specifically, in the context of PFGS streaming, we optimally adapt the enhancement layer encoding bit rate. To the best of our knowledge, this important encoding parameter setting problem has to date only been observed in [13] but no solution has been proposed, as detailed in Section 1.1. Our main approach is to pre-encode a given video with fine granularity scalable coding (such as PFGS coding) into a small number of versions with different encoding parameter values (e.g., different enhancement layer coding bit rates in the case of PFGS coding). In a sense, our approach combines simulcast with scalable coding. In contrast to simulcast, which requires many versions to adapt

the video bit rate to the available transmission rate, our approach requires only a small number of versions (about five versions are sufficient to achieve reasonable gains). The storage overhead of these encoding versions is negligible due to minimal hardware cost of current storage devices. Our approach achieves the adaptation to the available transmission bit rate by first selecting the encoding version (with the appropriate value of the encoding parameter) according to the current network (channel) condition and the current visual content descriptor, and then relying on the enhancement layer adaptation for fine tuning the transmission rate.

Key distinctions of our work are: (i) that we exploit the video content descriptors that show a high correlation with the reconstructed quality, (ii) that we develop and evaluate a suite of adaptive bitstream switching policies that augment existing unicast streaming, link rate regulation, and multicast schemes and are therefore of particular relevance to the application layer in network-aware multimedia services, (iii) that our adaptive policies do not require any modification to existing layer decoders; the minor added complexity is only in the video server, and (iv) the proposed adaptive bitstream switching policies are flexible enough to be applied for many transmission scenarios that use layered coding schemes (with any encoding parameters that control the enhancement layer rate and quality) and many visual content descriptors that can be used to estimate the reconstructed quality. Our previous study in [16,17] demonstrates that a number of visual content descriptors can be used to predict the quality degradation due to lossy packet transmission.

The paper is organized as follows. Section 2 presents overviews of the PFGS video encoder, and the AR-FGS encoder of the H.264 scalable video coding (SVC) standard which are used for the illustration of our adaptive bitstream switching framework. Section 3 discusses the mathematical problem formulation of video transmission from the perspective of the video server, when multiple layered encoded video versions of different visual content descriptors are used for transmission. Section 4 explains a comprehensive suite of adaptive bitstream switching policies for unicast, link optimization, and multicasting streaming schemes and their impact on the reconstructed qualities for each receiver. Section 5 presents simulation results for the proposed adaptive bitstream switching policies. We summarize the main conclusions in Section 6.

### 1.1. Related work

The important problem of adjusting the enhancement layer encoding rate for streaming PFGS video has been noted in [13] where a transcoding approach is developed that lowers the enhancement layer encoding rate of an encoding with a higher encoding rate to make the encoding more efficient for transmission at lower bit rates. In [18], a technique for frame-level bitstream switching using PFGS coding was proposed. Frame types similar to the SI and SP frames of H.264 coding were proposed, which reduce the compression efficiency. In addition, this technique adds computational resources to the existing PFGS encoders and decoders.

The visual content typically varies between video scenes (or video shots) of different actions and genres, especially for movie contents. The study of the reconstructed qualities for these visual contents has received relatively little attention in efforts to improve the application layer quality of service (QoS) of video streaming [2,15]. Video streaming can effectively exploit the available network resources by adapting to the visual content variability as well as the variability of the available network bandwidth. Conventional techniques for network-aware video streaming optimize utility metrics that are based on the rates of the receivers [14,19–22]. Some other techniques employ low-level visual content features, such as the frame-type of the video stream [23]. The frame-type is extracted from the syntax of the video stream and shows only low correlation with actual visual content descriptors (expressed by MPEG-7 descriptors [24–27]), which we consider in our study. The study [17] presents and evaluates adaptive streaming mechanisms, which are based on the visual content features, for non-scalable (single-layer) encoded video, whereby the adaptation is achieved by selectively dropping B-frames. The present study is complementary to [1,17] in that we consider scalable (layered) encoded video and develop content-based network-aware streaming mechanisms for layered encoded video.

Bandwidth adaptation has been identified as an essential requirement for video multicasting over the Internet [28,29]. In layered multicasting schemes, each receiver subscribes to layers based on its capabilities, see e.g. [30]. One of the key issues for layered multicast schemes is the intersession

fairness [31,32]. Hybrid adaptation layered multicast (HALM) is a recent protocol that outperforms other layer multicasting protocols by allowing adaptation at the video server, in addition to adaptation at the receivers [21,29,33]. HALM is most suitable for video streams coded using FGS or PFGS because of their simple real-time rate adaptation. Initially, the receivers predict their own available bandwidth (using a mathematical model, e.g., [34,35]) and use this prediction to join the appropriate multicasting group. The video server then receives these predictions in the form of real-time control protocol (RTCP) reports, and uses these predictions to assign transmission rates to each layer so as to maximize the average bit rate fairness index. In this study, we propose an extension to the HALM protocol by optimizing the transmission through bitstream switching using the reconstructed qualities.

## 2. Overview of PFGS coding and H.264 SVC's AR-FGS

PFGS improves bandwidth efficiency over the FGS scheme through motion compensation from reconstructed enhancement layer frames [10,11]. Fig. 1 illustrates an example of the PFGS codec where the base layer stream is coded using an H.264/MPEG-4 AVC codec. There are two predictions at the enhancement layer: a low quality prediction, which is reconstructed from the underlying base layer, and a high-quality prediction, which is reconstructed from the previous enhancement layer frame. The macroblocks of the current enhancement layer frame can be coded with reference to the high or low-quality prediction [11]. In Fig. 1, two switches ($s_1$ and $s_2$) select the reference for motion compensation and reconstruction. The residue after prediction is discrete cosine transformed (DCT), followed by bit plane coding similar to the FGS scheme [9]. Only the first $\alpha(t)$ bits (whereby $\alpha(t)$ represents the bit budget of the enhancement layer frame) are used to reconstruct the enhancement reference for the next frame. If the enhancement layer bitstream is truncated due to channel bandwidth fluctuations, the decoder reconstructs a degraded image, compared to the transmitted image. This results in a drifting error at the decoder until receiving an I-frame. In other words, a drifting error occurs if the enhancement layer transmission rate $r$ is less than the encoding rate $\alpha$, which is referred to as $\alpha(t)$ in Fig. 1. The enhancement layer-coding rate represents the encoding parameter $\alpha$ of the PFGS codec. In the case of other fine granularity layer coding techniques, $\alpha$ denotes the encoding parameter(s) that controls the

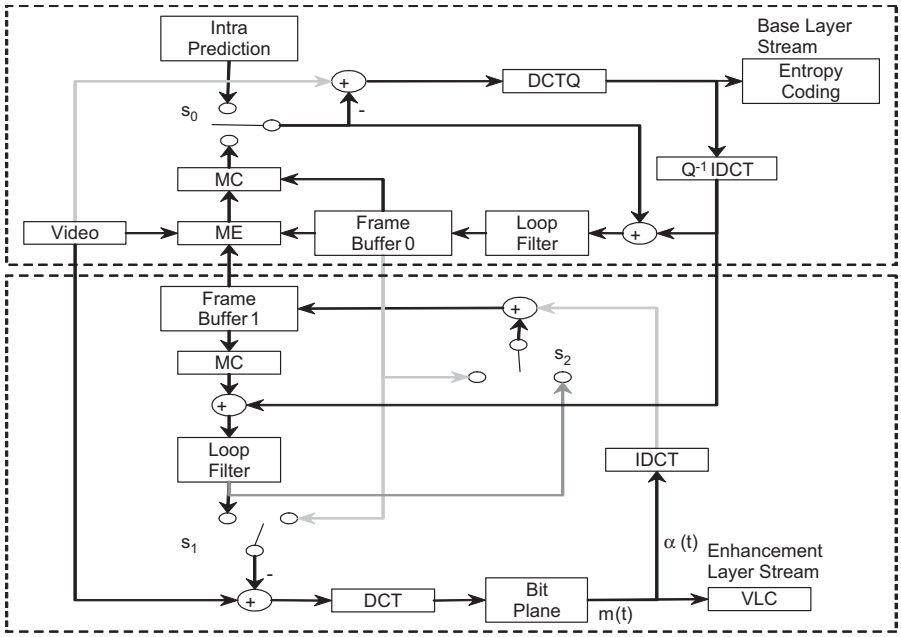


Fig. 1. Encoder structure of H.264/MPEG-4 AVC-PFGS (we used an under development version of the H.264 codec, which was originally named H.26L).

generation of different versions of the enhancement layer bitstream.

The latest H.264 SVC standard's AR-FGS coding mode [12], is a fine granularity video coder for low-delay applications. More specifically, temporal prediction is employed in the FGS layer of closed-loop P frames. The weighted prediction is formed adaptively from the enhancement layer reference and the base layer reference. In this work, we use AR-FGS with even weighted prediction in the JSVM 7.13 implementation to verify that our PFGS findings are also generally valid for state-of-the-art FGS encoding. We note that new flexible scalability concepts are emerging, for instance, in the context of the new H.264 SVC standard, such as medium grain scalability (MGS), which may provide sufficient flexibility for the use in the proposed adaptive bitstream switching of scalable video. Contrary to FGS, which supports network abstraction layer unit (NALU) truncation with bit granularity, MGS will offer a restricted number of truncation points for NALUs. Since bitstreams contain many NALUs, this medium truncation flexibility will still offer enough bit rate adaptation flexibility for streaming applications while being less complex than FGS. Both FGS and MGS are based on the same underlying motion compensation prediction structure, which limits decoder drift due to truncation while at the same time offering high rate-distortion efficiency. The detailed evaluation of adaptive bitstream switching of such scalability concepts that are still under development is left for future work.

To illustrate the performance of the PFGS codec as a function of the enhancement layer transmission rate $r$, we present encoding results for the first 15 min of the *Star Wars IV* movie, which contain a reasonable amount of content diversity. We consider the reconstructed qualities for video shots of different visual contents. Video shots are the minimal logical video sequence of the underlying movie. Automatic shot detection techniques have been extensively studied and simple shot detection algorithms are available [36]. We identified 200 video shots in the *Star Wars IV* excerpt. While we focus on motion-related content descriptors in this paper, the proposed techniques generally apply to any content descriptors that have high correlation with reconstructed visual qualities. The intensity of the motion activity in a video shot is represented by a scalar value, which ranges from 1 for a low level of motion to 5 for a high level of motion, and correlates well with the human perception of the level of motion in the video shot [25,27]. For each video shot, 10 human subjects estimated the perceived motion activity levels according to the guidelines presented in [26]. Alternatively, automatic extraction mechanisms, e.g., [17,26,27], could be used. The QCIF ($176 \times 144$) video format was used, with a frame rate of 30 fps, and an I-frame coded every 25 P-frames. The video shots were coded using a quantization scale of 28 (which is selected to obtain a base layer rate of about 100 Kbps).

Fig. 2 shows the average reconstructed qualities $Q_\beta(r,\alpha,\lambda)$, where $r$ represents the enhancement layer transmission rate, $\alpha$ represents $\alpha(t)$ in Fig. 1, i.e., the enhancement layer encoding rate, $\beta$ represents the base layer rate ($=100$ Kbps), and $\lambda$ represents the motion activity level of the underlying video shot. To generate these curves, we extracted lower bit rate streams by truncating each encoded video frame (picture) at the bit value corresponding to the transmission rate $r$. More specifically, with a transmission rate $r$ bit/s and a frame rate of 30 frames/s, each frame was truncated to $r/30$ bit. In the case of $r=0$, the reconstructed quality is obtained by decoding the base layer stream and this base layer quality can be of any PSNR value depending primarily on the color complexity of the video shot. (In particular, the base layer quality is responsible for activity level 1 achieving a higher reconstructed quality than activity level 5 in Fig. 2. In Fig. 5, we remove this base layer dependency.) Importantly, we observe that the relationship between $Q_\beta(r,\alpha,\lambda)$ and $r$ can be approximated by a spline function in the range of $r \in [0,1800]$. The non-linearity in the range of $r \in [1800,2000]$ is due to the significance of the least bit planes in the reconstructed quality. This effect is more apparent in shots of low motion activity levels, where many frame blocks are predicted from previously reconstructed enhancement layer frames. The results of Fig. 2 show the significance of the drifting error for two different encoding parameter values $\alpha$ and shots of different motion activity level $\lambda$. These results motivate us to use multiple encoding versions to improve the reconstructed qualities.

## 3. Definitions and foundations

In this section, we present the basic definitions and the problem formulation for our framework of adaptive bitstream switching policies for scalable encoded video. The video-streaming server has

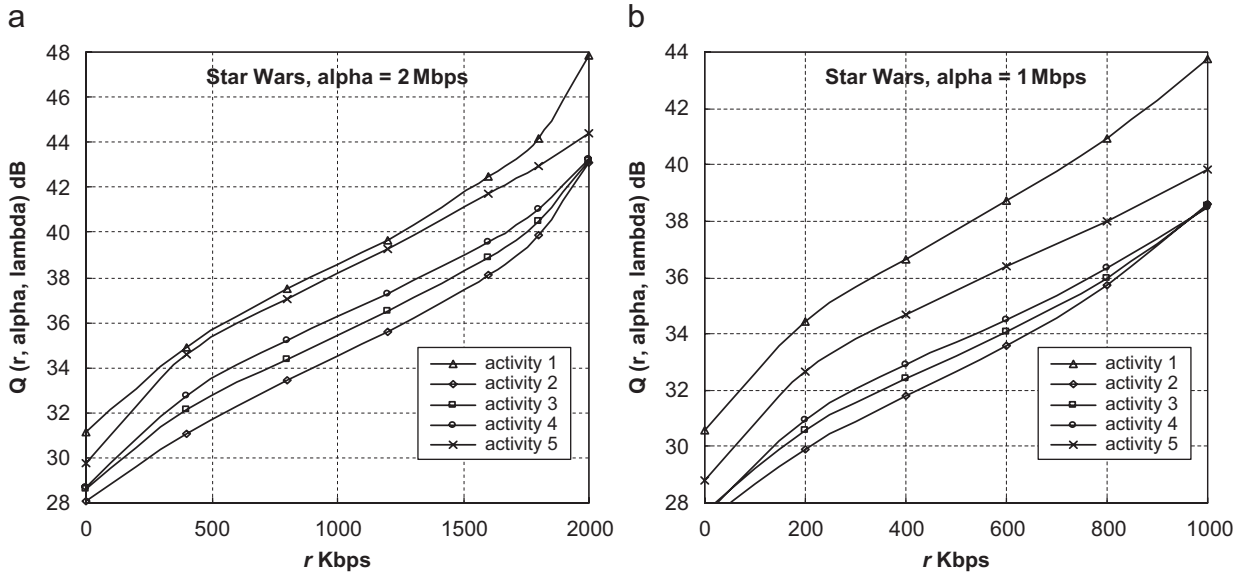


Fig. 2. Reconstructed video quality $Q$ as a function of enhancement layer transmission rate $r$ for different motion activity levels $\lambda$, enhancement layer encoding rate $\alpha = 1$ and 2 Mbps, fixed.

a limited capacity $R$, which may represent the capacity of the link connecting the video server to the network. This limited capacity $R$ can be shared among multiple video streams that are transmitted using unicast and multicast scenarios.

### 3.1. Basic terminology and problem formulation

As summarized in Table 1, we let $\alpha$ denote the value of the encoding parameter that controls the bit rate of the enhancement layer, and let $\lambda$ denote the visual content descriptor of the transmitted video sequence. The base layer rate is referred to as $\beta$, the enhancement layer transmission rate as $r$, and the aggregate rate of both base layer and enhancement layer as $c$. We let $Q_\beta(r,\alpha,\lambda)$ denote the average reconstructed quality, and denote the quality degradation $\Delta Q_\beta(r,\alpha,\lambda) = (Q_\beta(\alpha,\alpha,\lambda) - Q_\beta(r,\alpha,\lambda))/Q_\beta(\alpha,\alpha,\lambda)$. $\bar{Q}_\beta(\alpha,\lambda)$ denotes the average reconstructed quality for video multicasting (using $M$ enhancement layers), and $\bar{Q}(R)$ denotes the average reconstructed quality for the video streams that share the limited video server capacity $R$. We note that $\alpha$ and $\lambda$ could be more general and denote any general set of encoding parameters and content descriptors in our framework. In the presented formulation, we restrict ourselves to one encoding parameter and one content descriptor, and yet more specifically, focus on the PFGS enhancement layer coding rate and motion activity descriptors.

The streaming of $J$ scalable video streams at the video server with adaptation of the encoding parameters to maximize the average reconstructed quality can in general be formulated as the optimization problem:

$$(\boldsymbol{\alpha},\boldsymbol{\beta})^* = \arg\max_{(\beta_j,\alpha_j)} \left\{ \sum_{j=1}^{J} Q_{\beta_j}(r_j,\alpha_j,\lambda_j) \right\}, \quad (1)$$

subject to:

$$R \geqslant \sum_{j=1}^{J} c_j \quad \text{and} \quad r_j = \min\{\alpha_j, c_j - \beta_j\},$$

whereby

$$(\boldsymbol{\alpha},\boldsymbol{\beta})^* = ((\alpha_1,\beta_1)^*, (\alpha_2,\beta_2)^*, \ldots, (\alpha_J,\beta_J)^*)$$
$$\text{with} \quad \alpha_j \in \{\alpha_j^n : n = 1,2,\ldots,N\}$$
$$\text{and} \quad \beta_j \in \{\beta_j^k : k = 1,2,\ldots,K\}.$$

This problem formulation can determine the coding parameter settings that maximize the sum of the average reconstructed stream qualities. This is a natural maximization objective that is widely considered a basic optimization goal for streaming multiple video streams. A further refined, although significantly more complex optimization goal would be to maximize the sum of the average reconstructed qualities while assuring a fair allocation of network resources to the individual streams. Such a refined optimization is left for future work. The complexity

Table 1
Summary of basic notations

| Variable | Definition |
|---|---|
| $\alpha$ | Value of the encoding parameter. For PFGS stream, this refers to enhancement layer encoding rate; $\alpha(t)$ in Fig. 1 (bps) |
| $\lambda$ | Value of the content descriptor (such as motion activity level) of the underlying video shot |
| $\beta$ | Base layer encoding rate (bps) |
| $Q_\beta(r,\alpha,\lambda)$ | Average reconstructed quality for base layer of rate $\beta$ (Kbps), enhancement layer transmission rate $r$, enhancement layer encoding parameter value $\alpha$, and content descriptor $\lambda$ of the underlying video shot. The average reconstructed quality is measured as the average Y-PSNR of the frames in the considered video. (dB) |
| $K$ | Number of different base layer encodings |
| $N$ | Number of different enhancement layer encodings |
| $J$ | Number of video streams sharing the communication link |
| $M$ | Number of multicast layers |
| $R$ | Rate constraint of the shared link resources (i.e., the capacity of the video server) (bps) |
| $\Delta Q_\beta(r,\alpha,\lambda)$ | Quality degradation at rate $r$, calculated as $(Q_\beta(\alpha,\alpha,\lambda) - Q_\beta(r,\alpha,\lambda))/Q_\beta(\alpha,\alpha,\lambda)$ |
| $\Delta r$ | Amount of rate regulation, calculated as $(\alpha-r)/\alpha$ |
| $\alpha^*$ | Enhancement layer encoding rate that produces the maximum average reconstructed quality (bps) |
| $\beta^*$ | Base layer encoding rate that produces the maximum average reconstructed quality (bps) |
| $\rho_j$ | Optimal transmission rate of video stream $j$, for PFGS video streams sharing a communication resource (bps) |
| $c$ | Expected receiver bandwidth (bps) |
| $c_m$ | The aggregate bit rate of the first $m$ multicast layers (bps) |
| $p_m$ | The probability that the expected bandwidth ($r$) of a receiver is between $c_m$ and $c_{m+1}$ |
| $\bar{Q}_\beta(\alpha,\lambda)$ | Average reconstructed quality for base layer rate $\beta$, enhancement layer encoding rate $\alpha$, and $\lambda$ for the content descriptor of the underlying video shot. (dB) |
| $\bar{Q}(R)$ | Average reconstructed quality for rate constraint $R$ (dB) |

and storage requirement of Problem (1) are already quite large. The computational complexity using exhaustive search techniques is on the order of $O(JKN)$. The storage requirement of the video server in order to provide the video encodings with the $K$ different base layer and $N$ different enhancement layer rates is also on the order of $O(JKN)$. In the following sections, we consider special cases of this general streaming problem with focus on reducing the computational/storage requirements.

In this study, we analyze the visual content using the motion activity level as defined in the MPEG-7 standard [25]. Also, we have chosen the PFGS scheme for video coding due to its fine granularity feature and efficiency. The above streaming server problem provides a general framework that can accommodate many content descriptors and many video coding schemes. Our objective is to design adaptive streaming mechanisms that place the overhead on the video server and work with the existing decoders.

### 3.2. Foundations for point-to-point (unicast) streaming

In this section, we focus on an individual unicast video stream and remove the subscript $j$ of $\alpha_j^n$, $\beta_j^k$, $r_j$, and $\lambda_j$. Multiple unicast streams sharing a common resource are considered in Section 3.3.

#### 3.2.1. Selecting the enhancement layer encoding rate

We first illustrate the significance of selecting an appropriate value for the enhancement layer encoding rate $\alpha^n$ for the reconstructed quality, and subsequently formulate the corresponding special case of the video streaming problem (Eq. 1). (The superscript $k$ of $\beta$ is removed since a single base layer is used for this special case). We present the tradeoffs in selecting the enhancement layer encoding rate $\alpha$ for the video shots in the first 15 min of *Star Wars IV* in Fig. 3A. We have coded these shots using a fixed base layer rate (about 100 Kbps) and different enhancement layer encoding rates $\alpha^n$, $n = 1, \ldots, N$. We compare the average reconstructed qualities of PFGS encodings with different $\alpha^n$, by plotting quality difference curves of two PFGS encodings $Q_\beta(r,\alpha^n,\lambda) - Q_\beta(r,\alpha^{n+1},\lambda)$. $Q_\beta(r,\alpha^n,\lambda)$ represents the average reconstructed quality for all video shots of motion activity level $\lambda$. If the enhancement layer transmission rate $r$ is slightly higher than $\alpha^n = 1$ Mbps (see, e.g., case (1M-2M) for $r = 1.2$ Mbps in Fig. 3A(a)), the reconstructed qualities for all motion activity levels

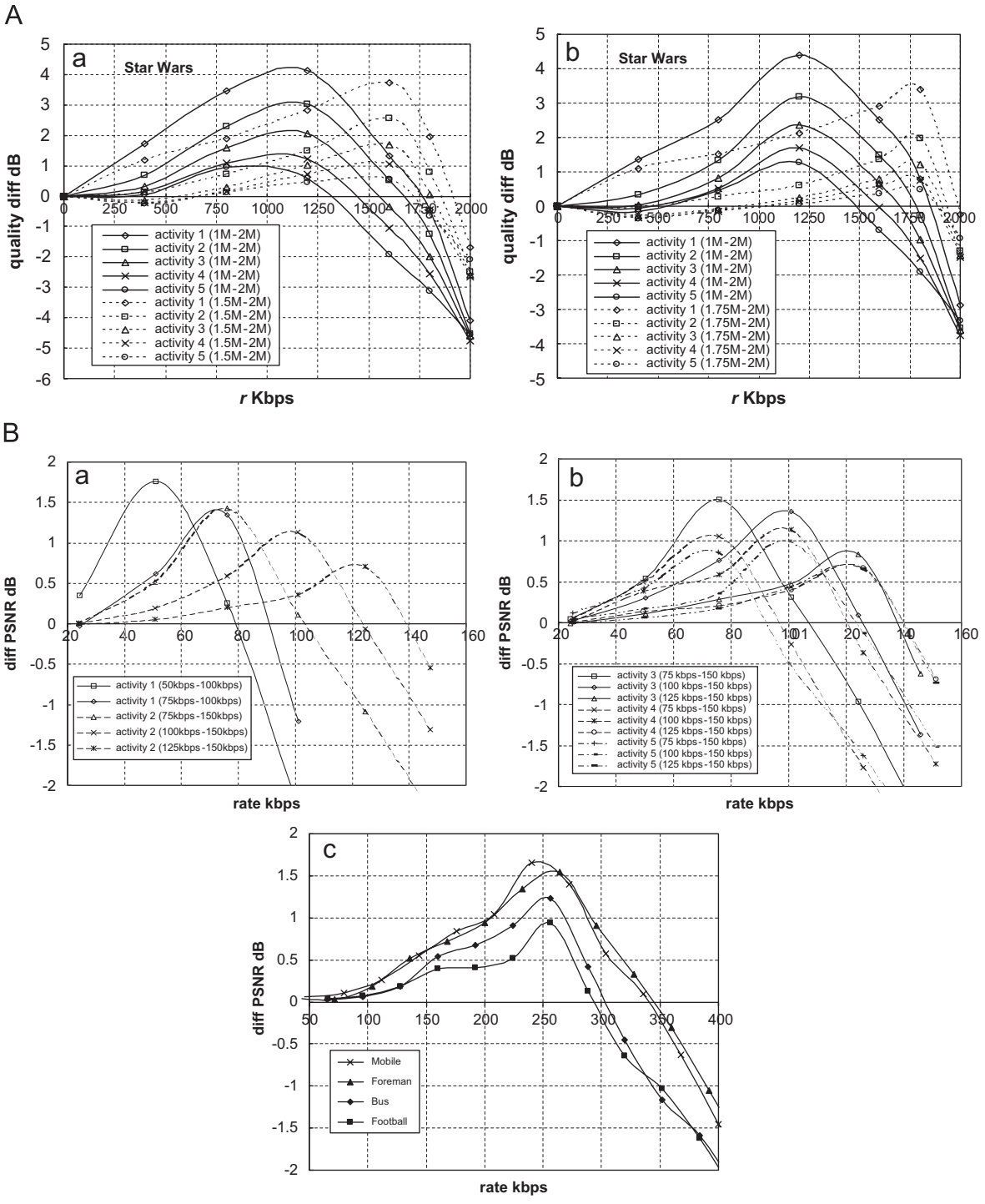


Fig. 3. (A) Quality difference $Q_\beta(r, \alpha^n, \lambda) - Q_\beta(r, \alpha^{n+1}, \lambda)$ as a function of enhancement layer transmission rate $r$ for PFGS encodings with different encoding rates $\alpha^n$ and different motion activity levels. (a) $\alpha^n = 1$ Mbps, $\alpha^{n+1} = 2$ Mbps and $\alpha^n = 1.5$ Mbps, $\alpha^{n+1} = 2$ Mbps, (b) $\alpha^n = 1.25$M bps, $\alpha^{n+1} = 2$ Mbps and $\alpha^n = 1.75$Mbps, $\alpha^{n+1} = 2$ Mbps. (B) (a, b) Quality difference as a function of transmission rate for AR-FGS encodings with different encoding rate combinations $\alpha^n$ and $\alpha^{n=1}$, grouped per motion activity level. (c) Quality difference for bus, foreman, football, and mobile sequences (256–512 kbps). (a) activity $\lambda = 1$, and 2 (b) activity $\lambda = 3$, 4, and 5 (c) bus, football, foreman, mobile.

are higher for the PFGS encoding with encoding rate $\alpha^n = 1$ Mbps, as indicated by the positive quality difference. In other words, for such a case (i.e., the available enhancement transmission layer rate is 1.2 Mbps) the video server is better off transmitting the video stream with a lower encoding rate (i.e., 1 Mbps) than the channel can offer (1.2 Mbps) in order to achieve higher reconstructed quality. If the enhancement layer transmission rate is much higher than 1 Mbps, significant quality degradation is incurred using the PFGS encoding with $\alpha^n = 1$ Mbps, as indicated by the negative quality difference, i.e., a higher reconstructed quality is achieved with the $\alpha^{n+1} = 2$ Mbps encoding. We observe from Fig. 3A that for video shots of low motion (activity 1) coding inefficiencies up to 4 dB are incurred by transmitting an encoding with $\alpha^n = 2$ Mbps at rate of 1.2 Mbps over streaming an encoding with $\alpha^n = 1$ Mbps, underscoring the importance of appropriate encoding rate selection as a function of the visual content.

In Fig. 3A, we define the critical point as the rate at which adapting the video streaming from encoding rate $\alpha^n$ to encoding rate $\alpha^{n+1}$ can improve the reconstructed qualities. These critical points are located at the intersection between the plotted curves and the $x$-axis. Note that the critical point for high motion activity levels is located at a lower rate compared to the critical points of lower motion activity levels. This effect is related to the bit plane coding technique used in the PFGS codec. The reconstructed qualities of shots of high motion activities are monotonically increased as the number of bit planes is increased, which can be provided by PFGS streams with encoding rate $\alpha^{n+1} > \alpha^n$. Comparing the curves in Fig. 3A, we observe that as the difference between $\alpha^n$ and $\alpha^{n+1}$ becomes smaller, the critical points move closer to $\alpha^{n+1}$. In addition, the maximum quality difference slightly decreases as the value of $(\alpha^{n+1} - \alpha^n)$ becomes smaller. For small rates such as 500 Kbps, the video stream for two different enhancement layer encoding rates ($\alpha^n$ and $\alpha^{n+1}$) depends on the method of tuning the $s_1$ and $s_2$ switches in Fig. 1. This results in better reconstructed quality for encodings with $\alpha^{n+1} > \alpha^n$ at a small rate. As the rate increases, this effect diminishes and the reconstructed quality for encodings with $\alpha^n < \alpha^{n+1}$ improves; see the slight "dips" in Fig. 3A around a rate of 500 Kbps.

To verify that identical principles apply to the newest scalable video encoding standard H.264 SVC, we have encoded all 200 shots of each motion activity level in the AR-FGS mode [12]. This mode is designed to improve FGS rate distortion efficiency, in particular of P frames. We employ similar encoder configuration settings as for PFGS, i.e., one I frame is encoded every 25 P-frames but we use the quantization scale 48 for the base layer. The different encoding rates that we use are $\alpha^n = 50$, 75 kbps and $\alpha^{n=1} = 100$ kbps for motion activity level 1, and rates $\alpha^n = 75$, 100, and 125 and $\alpha^{n=1} = 150$ kbps for activity levels 2, 3, 4, and 5. Fig. 3B(a) and (b) depicts quality difference curves grouped per motion activity level. We group the quality difference curves in this manner to show how quality differences evolve within a motion activity level contrary to Fig. 3A where quality difference curves are bundled according to enhancement encoding rate differences. As in Fig. 3A, we observe that the maximum of quality differences decreases with the activity level. Furthermore, it is clear that for AR-FGS encodings with different encoding rates, there exist transmission bit rate ranges where streams with lower encoding rates than the transmission rate result in substantial positive quality differences. We observe this for each activity level. Therefore, in the case of AR-FGS, the average reconstructed quality of a video stream transmitted at a prescribed rate can be improved by switching to the video encoding with the appropriate encoding rate. The intersection points between the quality difference curves and the $x$-axis are the critical points at which switching increases the average video quality within a particular motion activity level.

We also verify the quality difference principle with AR-FGS for the standard sequences Bus, Football, Foreman, and Mobile. The quality difference curves obtained for two encodings of these sequences at 256 and 512 Kbps are depicted in Fig. 3B(c). The Foreman and Mobile sequences, which have low motion activity levels, result in the highest quality differences up to 1.65 dB, while the Bus and Football sequences, which have increasing motion activity levels, have maximum quality differences of 1.23 and 0.94 dB, respectively.

If a single base layer encoding (with base layer rate $\beta$) is stored at the video server and the focus is on determining the optimal enhancement layer rate, the original streaming problem, Eq. (1), can be simplified to

$$\alpha^* = \arg\max_{\alpha^n}\{Q_\beta(r, \alpha^n, \lambda)\}, \quad (2)$$

subject to:

$$r = \min\{\alpha^n, c - \beta\}, \quad \text{with} \quad n = 1, \ldots, N.$$

This problem formulation selects the best value for an encoding parameter (in particular, the enhancement layer encoding rate $\alpha$ for PFGS streaming) given the enhancement layer transmission rate $r$, the base layer rate $\beta$, and the visual content of the underlying video sequence $\lambda$. The computational complexity and the storage requirements of this problem are $O(N)$. In Section 4, we present a detailed explanation of a potential solution technique for this problem.

### 3.2.2. Selecting the base layer rate

To examine the impact of the base layer rate $\beta^k$ on adaptive streaming, we first present the reconstructed qualities of the video shots of the first 15 min of *Star Wars IV* movie, assuming a fixed enhancement layer encoding rate of $\alpha = 1$ Mbps, and proceed to present the corresponding special case of the general optimization problem. (Note that all subscripts for $\alpha$ are removed since a single enhancement layer encoding is used for this special case.) We coded these shots using different quantization scales $q$ to generate different bit rates at the base layer. Fig. 4 shows the reconstructed qualities for various base layer rates $\beta^k$. The depicted rates represent the total (or cumulative) rate of the base layer and the corresponding enhancement layer transmission rate, while the quality represents the average reconstructed quality for all video shots of motion activity level $\lambda$. The results show that the reconstructed quality is monotonically increasing as the base layer rate is increasing. For large video transmission rate (i.e., the aggregate transmission rate is around $\alpha+\beta^k$), most video encodings result in very comparable reconstructed qualities. Increasing the base layer rate has an adverse effect on the ability to regulate video streams during the journey from the video server to the destination. Hence, the selection of the appropriate video encodings with different base layer rates is limited by the expected rate regulation during the video transmission.

For unicast communication, assuming that the video server stores a single enhancement layer encoding (with rate $\alpha$) for each base layer encoding (with base layer rate $\beta^k$), the streaming problem, Eq. (1), simplifies to

$$\beta^* = \arg\max_{\beta^k}\{Q_{\beta^k}(r, \alpha, \lambda)\}, \tag{3}$$

subject to:

$$r = \min\{\alpha, c - \beta^k\}, \quad \text{with} \quad k = 1, \ldots, K.$$

The best base layer rate $\beta^*$ can be specified given the enhancement layer transmission rate $r$, the encoding parameter (enhancement layer encoding

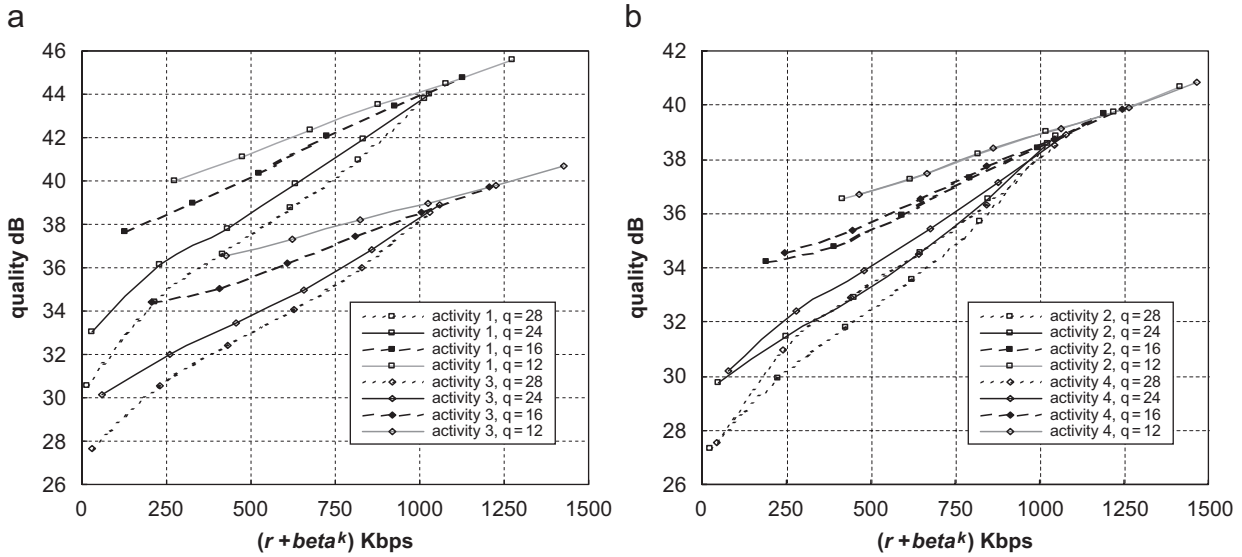


Fig. 4. Reconstructed qualities $Q$ as a function of the total (cumulative) rate ($r+\beta^k$) of the base layer and the corresponding enhancement layer transmission rate for different motion activity levels $\lambda$ and base layer of various qualities (i.e., various quantization scales $q$). The enhancement layer encoding rate is fixed at $\alpha = 1$ Mbps. (a) Activity levels $\lambda = 1$ and 3, (b) activity levels $\lambda = 2$ and 4.

rate $\alpha$ for PFGS stream), and the visual content descriptor of the underlying video sequence $\lambda$. The computational complexity and the storage requirement of this problem are $O(K)$. The storage requirement equals $2K$ ($K$ base layer encodings and the corresponding $K$ enhancement layer encodings, whereby each enhancement layer has the fixed encoding rate $\alpha$ that corresponds to one of the $K$ base layers), while the storage requirement of solving the streaming problem Eq. (2) is $N+1$ (one base layer encodings and $N$ enhancement layer encodings).

The problem of specifying the best combination of enhancement layer and base layer encoding rates for a single unicast stream can be formulated as

$$(\alpha^*, \beta^*) = \arg\max_{(\alpha^n, \beta^k)} \{Q_{\beta^k}(r, \alpha^n, \lambda)\} \quad (4)$$

subject to:

$$r = \min\{\alpha^n, c - \beta^k\}, \quad \text{with} \quad n = 1, 2, \ldots, N \quad \text{and} \quad k = 1, 2, \ldots, K.$$

This problem requires the enhancement layer transmission rate $r$ and the visual content descriptor of the underlying video sequence $\lambda$. The computational complexity and storage requirement of this problem are $O(NK)$.

### 3.3. Foundations for multiplexing unicast streams: content-dependent quality degradation due to packet drops

In this section, we address the video streaming optimization for multiple unicast streams sharing a common networking resource. The basic underlying observation is that enhancement layer rate regulation at the video server has different visual impacts if video packets carry visual content of different motion activity levels. We illustrate this effect by transforming the results in Fig. 2 into rate regulation and corresponding visual quality degradation in Fig. 5. The rate regulation represents the reduction in the enhancement layer transmission rate due to the packet drops at the bottleneck (or, equivalently, the packet loss ratio of the enhancement layer) normalized by the enhancement layer encoding rate $\alpha_j$, i.e., a rate regulation value of 1 represents the complete dropping of the enhancement layer. The quality degradation represents the ratio of the reduction in the reconstructed quality (due to the rate reduction) to the quality of the enhancement layer with encoding rate $\alpha$, i.e., we define quality degradation at rate $r_j$ as $(Q_{\beta_j}(\alpha_j, \alpha_j, \lambda_j) - Q_{\beta_j}(r_j, \alpha_j, \lambda_j))/Q_{\beta_j}(\alpha_j, \alpha_j, \lambda_j)$, whereby the amount of rate regulation is $(\alpha_j - r_j)/\alpha_j$. Note that the maximum quality degradation (achieved for a rate regulation value of (1) corresponds to the

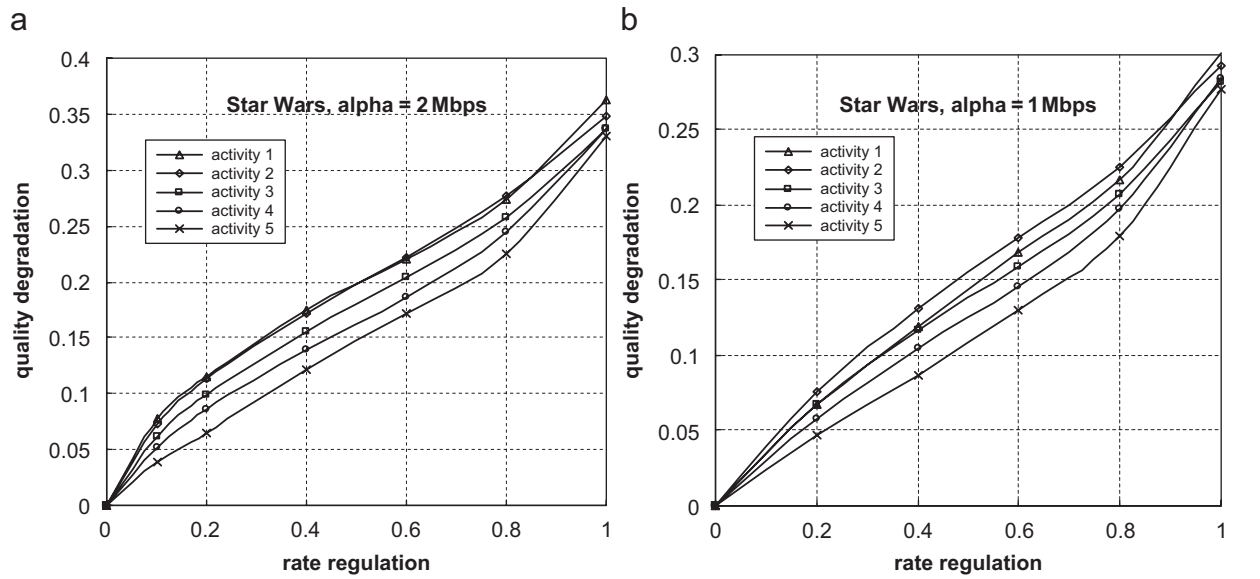


Fig. 5. Quality degradation (reduction in reconstructed quality normalized by the quality of enhancement layer with encoding rate $\alpha_j$) as a function of rate regulation (fraction of dropped enhancement layer packets) for different motion activity levels.

reception of only the base layer stream. Fig. 5 shows the quality degradation as a function of the amount of rate regulation for shots of different motion activity levels. The rate regulation of shots of high motion activity levels reduces the reconstructed quality relatively less compared to the rate regulations of shots of lower motion activity levels. This difference in quality degradation is due to the difference in the motion estimation loops that are used for shots of different motion activity levels. In a shot with higher motion activity, a larger fraction of the enhancement layer blocks is motion estimated with reference to the base layer frame. Hence, the quality degradation due to enhancement layer rate regulation is relatively smaller for shots of high motion activity levels. Conversely, in shots with a lower level of motion, a larger fraction of the enhancement layer blocks are motion estimated with reference to the preceding enhancement layer frame, resulting in more severe quality degradation due to rate regulation.

The problem of transmitting multiple video streams at the video server or intermediate nodes can be formulated as a link optimization over the shared video streams to determine the regulated enhancement layer transmission rates $\rho_j$, whereby the appropriate encoding parameter values ($\boldsymbol{\alpha}^*$,$\boldsymbol{\beta}^*$) would typically have been previously determined at the video server using the optimization problems expressed in Eqs. (2), (3), or (4). The following equations represent such link optimization problem:

$$\rho^* = \arg\max_{\rho_j}\left\{\sum_{j=1}^{J} Q_{\beta_j}(\rho_j, \alpha_j, \lambda_j)\right\}, \tag{5}$$

subject to:

$$R \geqslant \sum_{j=1}^{J} \rho_j, 0 \leqslant \rho_j \leqslant r_j, \quad r_j = \min\{\alpha_j, c_j - \beta_j\}.$$

The above optimization problem can be considered as the link optimization part of the original streaming problem, Eq. (1). This link optimization can be used by the video streaming server as well as intermediate nodes that represent proxy servers (which might be used for rate regulation between wired and wireless links). The solution to this optimization can be used to determine the appropriate enhancement layer rates $\rho_j$ that result in the maximum average reconstructed quality for a given link resource $R$. In order to accurately optimize the link resources, the proposed solution needs to have access to an approximation of $Q_{\beta_j}(r_j, \alpha_j, \lambda_j)$ for each shared video stream. This approximation is already stored in the video server to be used for determining the optimal encoding parameter values, and can be transmitted along with RTCP packets to intermediate nodes. We recommend to send this approximation during the early stages of video streaming (i.e., with the first few packets), which can be intercepted by some intermediate nodes that understand the RTP/RTCP transmission protocol and subsequently use this approximation to improve the average reconstructed qualities for videos sharing resources at these routers. The computational complexity of solving this optimization is of order $O(J)$.

We evaluate the performance of our rate regulation policies using the average reconstructed qualities of the streams and the quality fairness indices for the individual streams. The average reconstructed quality $\bar{Q}(R)$ can be expressed as

$$\bar{Q}(R) = \frac{1}{J}\sum_{j=0}^{J-1} Q_{\beta_j}(\rho_j, \alpha_j, \lambda_j),$$

$$R \geqslant \sum_{j=0}^{J-1} \rho_j, 0 \leqslant \rho_j \leqslant r_j, \tag{6}$$

where $R$ represents the rate constraint on the communication link (which is also denoted as TCP_AVR), and $\rho_j$ represents the enhancement layer rate for each individual PFGS video stream after applying the rate regulation. We calculate the quality fairness index for video stream $j$ based on the reconstructed quality ratios as

$$\text{Fairness\_Index}(j) = \frac{Q_{\beta_j}(\rho_j, \alpha_j, \lambda_j)}{Q_{\beta_j}(r_j, \alpha_j, \lambda_j)}, \tag{7}$$

where $Q_{\beta_j}(r_j, \alpha_j, \lambda_j)$ represents the maximum reconstructed quality for video stream $j$, which can be obtained with the maximum enhancement layer transmission rate of $r_j$ Kbps. This fairness index considers the QoS from the application layer respective. It is therefore better suited for video streaming than the existing fairness indices that evaluate the bandwidth fairness, which does not correlate linearly with the application layer QoS.

### 3.4. Foundations for multicasting: normalized rate regulation and normalized quality degradation

In this section, we present the foundations for our adaptive multicast streaming mechanism of multiple encodings of a video sequence, each encoded with a

different enhancement layer encoding rate $\alpha^n$. We remove the subscript $j$ of $\alpha_j^n$, $\beta_j^k$, and $\lambda_j$ for the considered case of multicasting a single video. We remove the superscript $k$ of $\beta_j^k$ since a single base layer is used for this multicasting. During the video transmission, the effective bandwidth varies dynamically. Multicasting protocols, such as HALM [21,33], dynamically adjust the enhancement layer transmission rates for each layer of the multicast tree. The challenging issue is to take into account the reconstructed visual qualities while adapting the encoding rate $\alpha^n$ so that the average reconstructed qualities are improved. We consider the visual content descriptors, expressed as $\lambda$, of the video sequence for the adaptation. We denote the quality of the base layer (which is constant) by $Q_\beta(0,\alpha^n,\lambda)$, the number of multicast layers by $M$, and the aggregate bit rate of the first $m$ multicast layers by $c_m$. In addition, $p_m$ represents the probability that the expected bandwidth $r_i$ of a receiver is between $c_m$ and $c_{m+1}$, i.e., $p_m = \mathrm{P}\{c_m \leqslant r_i < c_{m+1}\}$ and $p_M = \mathrm{P}\{c_M \leqslant r_i\}$. If the aggregate bit rate of the first $m$ multicast layers is below the encoding rate, i.e., $c_m \leqslant \alpha^n$, then the quality of multicast layer $m$ is obtained using $Q_\beta$ $(c_m,\alpha^n,\lambda)$, whereas multicast layers with $c_m \geqslant \alpha^n$ achieve the same quality as $Q_\beta$ $(\alpha^n,\alpha^n,\lambda)$. Formally, we let $l$ denote the highest indexed multicast layer with an aggregate rate $c_m$ less than or equal the encoding rate $\alpha^n$, i.e., $l = \max\{m{:}c_m \leqslant \alpha^n\}$. With this definition, the average reconstructed quality $\bar{Q}_\beta(\alpha^n,\lambda)$ for an encoding rate $\alpha^n$ can be expressed as

$$\bar{Q}_\beta(\alpha^n,\lambda) = p_0 Q_\beta(0,\alpha^n,\lambda) + \sum_{m=1}^{l} p_m Q_\beta(c_m,\alpha^n,\lambda) + Q_\beta(\alpha^n,\alpha^n,\lambda)\sum_{m=l+1}^{M} p_m. \tag{8}$$

Hence, the encoding rate that results in the maximum quality can be expressed as the optimization problem:

$$\alpha^* = \arg\max_{\alpha^n}\{\bar{Q}_\beta(\alpha^n,\lambda)\} \quad \text{Subject to}: \sum_{m=0}^{M} p_m = 1. \tag{9}$$

Solving this optimization problem requires an approximation of $Q_\beta(c,\alpha^n,\lambda)$. Instead of storing sample points of $Q_\beta$ $(c,\alpha^n,\lambda)$ for every possible value of $c$ and $\alpha^n$, we propose to normalize both the rate and quality:

$$\text{rate\_norm} = \frac{c}{\alpha^n},$$

$$\text{PSNR\_norm} = \frac{Q_\beta(c,\alpha^n,\lambda) - Q_\beta(0,\alpha^n,\lambda)}{Q_\beta(\alpha^n,\alpha^n,\lambda) - Q_\beta(0,\alpha^n,\lambda)}. \tag{10}$$

Fig. 6 shows sample points of the normalized $Q_\beta(c,\alpha^n,\lambda)$, i.e., of PSNR_norm as a function of rate_norm, for five enhancement layer coding rates

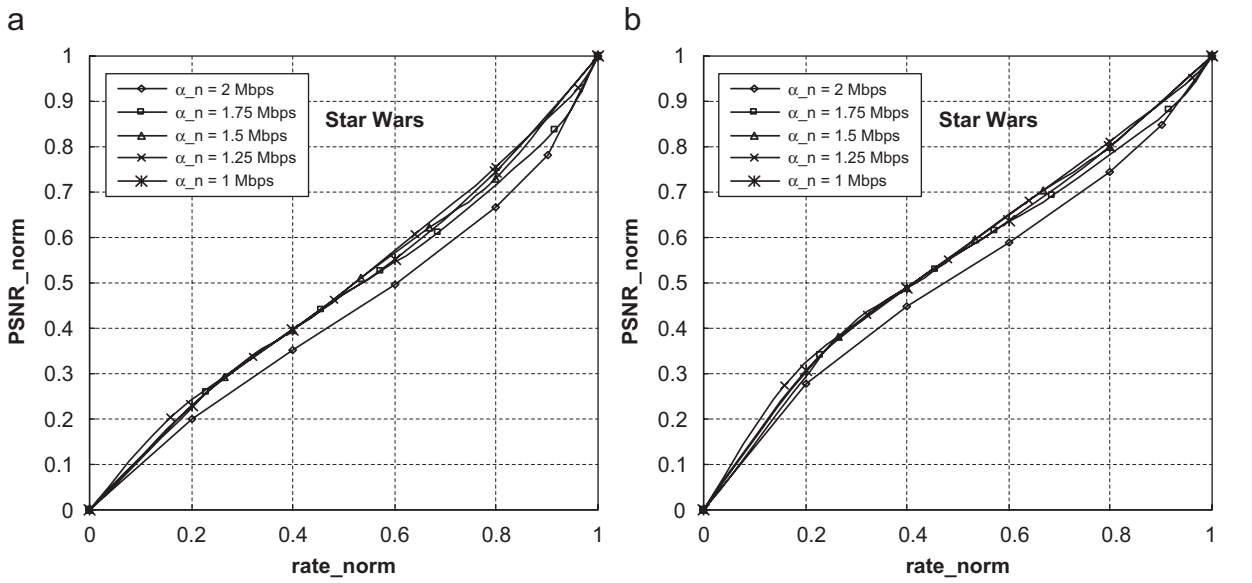


Fig. 6. Normalized quality as a function of normalized rate for different enhancement layer coding rates $\alpha^n$ for video shots of various motion activity levels $\lambda$ of *Star Wars IV* movie. (a) Activity $\lambda = 2$, (b) activity $\lambda = 4$.

$\alpha^n$ (1, 1.25, 1.5, 1.75, and 2 Mbps) for shots of different motion activity levels from *Star Wars IV*. For a given motion activity level, the plotted values of different $\alpha^n$ values can be reasonably approximated using a single polynomial function. Only for the case $\alpha^n = 2$ Mbps for low motion activity is a noticeable modeling error incurred. For low motion activity shots, i.e., $\lambda \leqslant 2$, the PFGS codec encodes many macroblocks with motion estimation from the reconstructed enhancement layer of the previous frame. Hence, any reduction in the enhancement layer transmission rate results in more quality degradation compared to the same rate reduction for video shots with higher motion activity levels. This effect results in a reduced normalized quality improvement (for the encoding with $\alpha^n = 2$ Mbps compared to other encodings with $\alpha^{n+1} < \alpha^n$) for a normalized additional rate; see Fig. 6(a). This effect diminishes as the motion activity level of the video shot increases, where more activity in the video shots results in more macroblocks encoded with motion estimation from the reconstructed base layer. The proposed normalization suggests that the values of $Q_\beta(c,\alpha^n,\lambda)$ for all $c$ and $\alpha^n$ values, for a given motion activity level $\lambda$ can be reduced into a *single* polynomial. This implies that for a given video sequence, any reduction in the enhancement layer rate by a specific ratio results in a similar quality degradation, independent of the original encoding rate $\alpha^n$.

## 4. Proposed scalable simulcasting mechanisms

In this section, we outline the proposed adaptive video streaming mechanisms for the different video streaming scenarios, namely unicast streaming, multiplexing of several unicast streams sharing limited network resources, and multicast. The performance of these proposed streaming mechanisms is evaluated in Section 5.

### 4.1. Unicast (point-to-point) streaming

We optimally select the encoding rate $\alpha^*$ of the PFGS enhancement layer from a set of pre-encoded encoding rates $\alpha^n$, $n = 1, \ldots, N$, according to the motion activity level of the underlying video shot. For the optimization, sample points of $Q_\beta(r,\alpha^n,\lambda)$ can be stored in the video server and be used for linear piece-wise approximation for curve fitting. This avoids the large storage overhead of the difference curves (shown in Fig. 3), which require sample points for every pair of video encodings. In addition, the motion activity level descriptors $\lambda$ for each video shot need to be accessed by the video server so that the optimizer can consider the visual content in adaptation. This amount of additional storage of sample points $Q_\beta(r,\alpha^n,\lambda)$ for $n = 1, \ldots, N$, and for the different values of the content descriptors $\lambda$ is negligible compared to the actual storage of the pre-encoded video streams. More specifically in the case of the motion activity as content descriptor, where $\lambda$ can take on the values $\lambda = 1, 2, 3, 4,$ or 5, we store sample points of $Q_\beta(r,\alpha^n,\lambda)$ for $n = 1, \ldots, N$ and $\lambda = 1, \ldots, 5$. We do not store $Q_\beta(r,\alpha^n,\lambda)$ for each video shot. Instead, a given video shot is characterized by its motion activity level $\lambda$ and the sample points corresponding to that $\lambda$ value.

In the case of adaptation of the enhancement layer rate $\alpha^n$, the streaming optimizer estimates $N$ quality values $Q_\beta(r,\alpha^n,\lambda)$, $n = 1, \ldots, N$, corresponding to the number of different enhancement layer encodings, and then selects the enhancement layer encoding with the largest reconstruction quality. This optimization detects the critical points of Fig. 3 and switches the video transmission to the encodings providing better visual quality. Similarly, in the case of adaptation of the base layer rate $\beta^k$, the optimizer estimates $K$ quality values corresponding to the number of different base layer encodings and selects the encodings with the largest reconstruction quality. The case of adaptation of both $\alpha^n$ and $\beta^k$ parameters (expressed in Eq. (4)) requires $NK$ quality estimations.

### 4.2. Proposed packet drop policies for multiplexing unicast streams

We explain content-dependent packet drop policies for pre-encoded PFGS unicast streams that share a common networking resource, e.g., shared bottleneck link, in this section. The considered streaming context is based on video streams with a prescribed enhancement layer encoding rate $\alpha_j$, a prescribed transmission rate into the network $r_j$, and a network bottleneck $R$ that requires the dropping of some packets downstream. The basic idea of the packet drop policies is that when multiple video streams share a bottleneck communication link, it is likely that the link buffer contains packets carrying shots of diverse motion activity levels. As observed in Figs. 5 and 6, the visual quality degradation depends on the motion activity level. In the case of rate regulation, the number of dropped packets

from each video sequence can be specified using different policies that exploit these content dependencies. We introduce four rate regulation policies:

- *Policy* 1: random dropping of packets from the video streams to meet the rate constraint $R$.
- *Policy* 2: equal packet loss ratios (or equal rate regulation ratios) for the video streams.
- *Policy* 3: the packet loss ratios are governed by the underlying shot activity level. The objective is to achieve equal quality degradations for the video streams. This can be implemented with a simple algorithm that has access to the sample points of $Q_{\beta_j}(r_j, \alpha_j, \lambda_j)$, and linearly approximating between these sample points.
- *Policy* 4: the bit budget is distributed among the video streams such that the average reconstructed qualities are maximized, which can be achieved with a greedy computational method. The method evaluates every possible rate combination using the approximation of $Q_{\beta_j}(r_j, \alpha_j, \lambda_j)$. The global maximum point is located and used to determine the optimal transmission rates $\rho_j$ for each PFGS video stream.

Existing buffer management schemes employ either policies 1 or 2, which do not need access to the sample points of $Q_{\beta_j}(r_j, \alpha_j, \lambda_j)$. On the other hand, policies 3 and 4 require access to these points, which can easily be exchanged to intermediate proxy servers during connection setup.

### 4.3. Proposed quality-adaptive encoding rate selection for multicasting

In this section, we propose an application layer multicasting scheme for optimally selecting the enhancement layer encoding rate $\alpha^n$ to maximize the average reconstructed visual qualities at a group of multicast receivers. Our optimization scheme takes the rate constraints (number of subscribed multicast channels) of the individual receivers into consideration in the optimization; see Eq. (9). The proposed quality-adaptation can be combined with any of the previously reported multicast methods, e.g., HALM [21,33], to improve the average reconstructed qualities of pre-encoded PFGS streams by considering the motion activity level of the video sequence. The basic idea of the proposed encoding rate selection is to periodically evaluate the multicast layer rates $c_m$ with an existing multicasting scheme, such as HALM, and then to find the encoding rate $\alpha^*$ that maximizes the average reconstructed qualities at the receivers. More specifically, each receiver estimates its expected bandwidth $r_j$, which guarantees TCP-friendly behavior of the video multicasting scheme. The video server then receives these bandwidth estimates every control period, and specifies the multicast layer rates $c_m$ using a dynamic programming technique [21]. These multicast layer rates are specified based on the distribution of the receiver rate estimates $p_m$. A receiver joins the multicasting group with a rate that best matches its own. For each video sequence, the video server has several pre-encoded encodings at different encoding rates $\alpha^n$, $n = 1, \ldots, N$. Eq. (10) is applied to determine the normalized rate for each multicast layer and encoding rate. By approximating the curves of Fig. 6 (using polynomial coefficients), the normalized quality degradation can be obtained. The actual reconstruction quality for each multicast layer is determined by storing sample points of $Q(\alpha^n,\alpha^n)$. For each available encoding rate, the average reconstructed qualities are calculated using Eq. (8). This method transmits the encoding with encoding rate $\alpha^*$ that produces the maximum average reconstructed quality at the group of multicast receivers.

## 5. Performance evaluations

In this section, we conduct comprehensive simulation experiments to evaluate the performance of the proposed video streaming optimizers. We particularly focus on presenting the average reconstructed qualities for various motion activity levels. The video sequences used in these simulations are similar to those used in Sections 2 and 3, i.e., we use 200 shots with a wide range of visual content extracted from the first 15 min of Star Wars IV.

### 5.1. Simulcasting for point-to-point streaming

#### 5.1.1. Comparison of optimal non-adaptive streaming and adaptive bitstream switching

Two simulation experiments using PFGS streams have been conducted with average TCP throughputs (AVR_TCP) of 1 and 1.5 Mbps, computed using an equation based model as [34]

$$\mathrm{AVR\_TCP} = 1.22\frac{\mathrm{MTU}}{\mathrm{RTT}\sqrt{\mathrm{Loss}}}, \tag{11}$$

where MTU denotes the packet size, RTT denotes the round-trip-time, and Loss denotes the steady-state drop rate. In each RTT, the video server

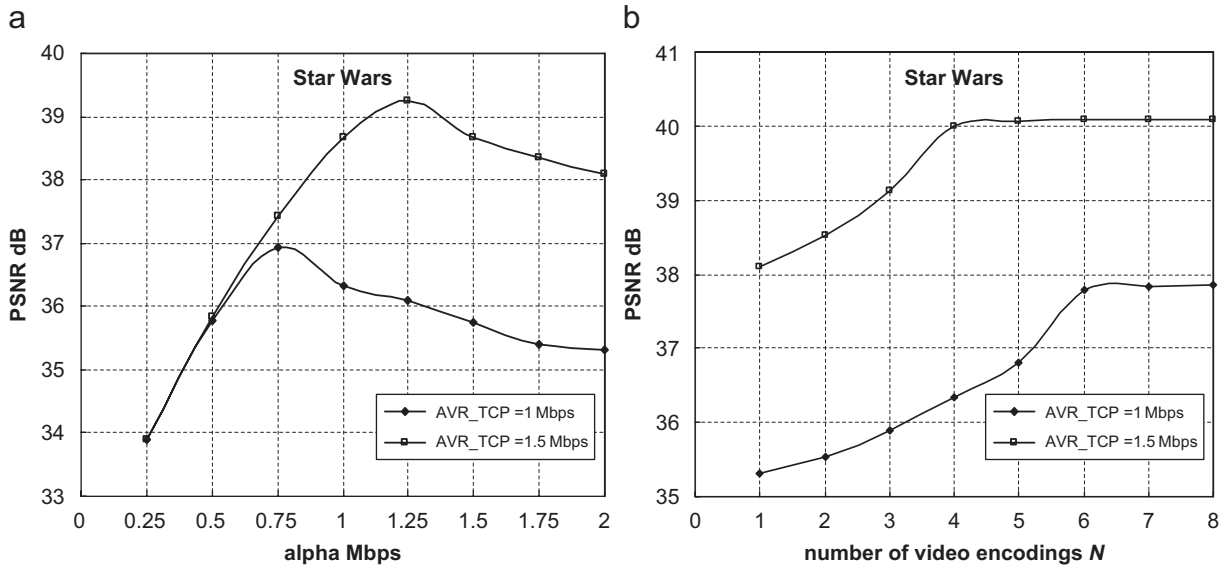


Fig. 7. Performance of adaptive scheme: approximately 0.8 dB quality improvement with adaptive scheme over best-performing non-adaptive scheme. (a) Reconstructed quality $Q$ with non-adaptive transmission as a function of enhancement layer encoding rate $\alpha$. (b) Reconstructed quality $Q$ with proposed adaptive transmission as a function of number of encodings $N$ (with different enhancement layer encoding rates $\alpha^n$, $n = 1, \ldots, N$).

computes the transmission rate using Eq. (11) and determines the PFGS video stream with the enhancement layer encoding rate $\alpha^*$ that maximizes the reconstructed quality. The packet loss ratios are generated using a two-state Markov model [35]. The video server receives the receiver report every RTT, set to 500 ms. Simulations using real Internet traces or using the ns-2 simulator are a topic for future research. The video server stores $N$ different PFGS encodings with the enhancement layer encoding rates $\alpha^n$:

$$\alpha^N = 2\,\text{Mbps} \quad \text{and} \quad \alpha^n = \alpha^{n+1} - 0.25, \tag{12}$$

where $n = 1, \ldots, N-1$, and the maximum value of $N$ is 8. For instance, if $N = 3$, the video server has three PFGS video streams with $\alpha^1 = 1.5$ Mbps, $\alpha^2 = 1.75$ Mbps, and $\alpha^3 = 2$ Mbps. In general, for a given $N$, the setting $\alpha^n = \alpha^N\, n/N$ for $n = 1, \ldots, N$ results in an equal spread of the enhancement layer coding rates. The adaptive bitstream switching is done at I-frames, whereby an I frame is coded every 25 frames. Note that bitstream switching is only required when there is a change in the transmission rate (computed every round trip time RTT) or a change in the motion activity (computed or accessed from the storage at the beginning of the video shot). We believe that this type of bitstream switching achieves a compromise between switching complexity and improving the reconstructed quality.

The average reconstructed quality as a function of the number of enhancement layer rates $N$ is shown in Fig. 7(b). As the number $N$ of stored encodings increases, the average reconstructed quality increases. The performance of the adaptive transmission reaches a peak value at a relatively small number of 4–6 stored encodings. Storing more encodings (with lower $\alpha^n$) brings only negligible performance improvement. In addition, Fig. 7(a) shows the performance of the non-adaptive video transmission scheme. We observe that by encoding the video with enhancement layer coding rate $\alpha^n = 2$ Mbps and then streaming this encoding with an average transmission rate of 1 Mbps, coding inefficiencies of up to 1.8 dB are incurred over selecting the optimal encoding rate of $\alpha^n = 0.75$ Mbps. Generally, the optimal $\alpha^*$ for the non-adaptive transmission scheme depends on the average throughput of the transmission channel, which is impossible to predict for an entire movie delivered over time-varying channels typical for wireless networks and the Internet. Comparing the peak values in Fig. 7(a) and (b) (keeping in mind the different PSNR ranges of the two plots), we observe that the proposed adaptive scheme achieves about 0.8 dB improvement over the best non-adaptive

scheme. We thus conclude that for time-varying channels, it is beneficial to have multiple PFGS encodings with different enhancement layer coding rates $\alpha^n$.

*5.1.2. Comparison of adaptive switching of PFGS streams with simulcasting of non-scalable streams*

We compare conventional simulcasting that uses single layer codings of different quality levels with our proposed simulcasting using scalable layer coding, such as the PFGS scheme. We have selected the quantization scale values $q = 4$, 10, and 24 in MPEG-4 Part 2 coding which give average PSNRs of about 36, 32, and 28 dB, referred to in the following as high quality (HQ), medium quality (MQ), and low quality (LQ) encodings, to generate single layer streams of different quality. We regulate the transmission rate of the single layer video streams using frame dropping. Two B-frames are coded between I-frame and/or P-frame, which allows for about 50% regulation of the originally coded bitstream. Fig. 8A shows the RD curves for these single layer encodings in addition to the curves for the PFGS encoding with enhancement layer encoding rates $\alpha^n = 0.5$ and 1 Mbps. In our comparison of MPEG-4 simulcast and PFGS, we generate the MPEG-4 base (I and P frame) streams with higher bit rate in order to compensate for the higher compression efficiency in the base layer of PFGS. The observations drawn for the thus obtained results hold in general and are not codec dependent, as demonstrated by our subsequent verification experiments with SVC.

The results in Fig. 8A indicate that the visual content, such as the motion activity level, plays a key role in determining the best simulcast coding, which provides improvement for the reconstruction quality. We observe from Fig. 8A(a) that for low motion activity levels (such as activity level $\lambda = 2$), simulcasting using single layer coding is more efficient for HQ, MQ, and LQ encodings. This is mainly because B-frame dropping, used for rate regulation of single layer coding, can be effectively concealed by an elementary copying scheme, which we employ as it provides low-complexity error concealment at the receiver. (Sophisticated error concealment schemes, such as [37,38], can at the expense of increased receiver complexity improve the reconstructed video quality and shift the curves for the non-scalable coding in Fig. 8A somewhat up.) However, for medium to high motion activity levels, the superiority of simulcasting using single layer coding is not guaranteed for many quality levels and many video transmission rates, as illustrated in Fig. 8A(b). There is a transmission rate, where switching between simulcasting using single layer and PFGS coding is beneficial, see the rate of the intersection point between the solid and dotted lines in Fig. 8A. The design of a system that can switch the video transmission from non-scalable coding to FGS scalability is presented in [4] by using neural network technology. A potential extension of this neural network-switching scheme to other single layer and scalable layer coding can be implemented as future work.

*5.1.3. Verification experiments for SVC*

The new SVC encoder has significantly higher coding efficiency for layered scalable and FGS encodings. Therefore, we verify our simulcast principles with this new encoder. We select high R-D efficiency encoding settings for the single-layer encodings. SVC employs hierarchical B-frames for single-layer encoding, which result in temporal scalability. For our experiments, we use 15 B-frames in between the key pictures (I- or P-frames) with only one I-frame at the beginning of each video shot. This number of hierarchical B-frames results in five temporal layers, which can be dropped to adapt the bit rate and are replaced by frame duplication. The quality of each duplicated frame is computed using the Y-PSNR value based on the duplicated frame and the original frame. This simple quality measure reflects to an extent the subjective impression of human observers when frames are dropped for rate adaptation. For low motion scenes, successive frames are similar and therefore the duplicated frames and the original frames are similar, resulting in a high PSNR value. On the other hand, if there is high motion activity then the duplicated frames and the original frames are quite different, and the computed PSNR value is low. The averaging of all PSNR values of a reconstructed stream with dropped temporal layers, or equivalently with duplicated frames, reflects the quality loss incurred by frame duplication.

All video shots in each motion activity class are encoded and the single-layer RD curves that include rate adaptation points are averaged over all shots. We depict these curves in Fig. 8B for quantizer parameters (QP) 32, 40, and 48 for motion activity classes 2 and 4. The RD point corresponding to the single-layer encoding without temporal layer dropping is the top right point of each curve.

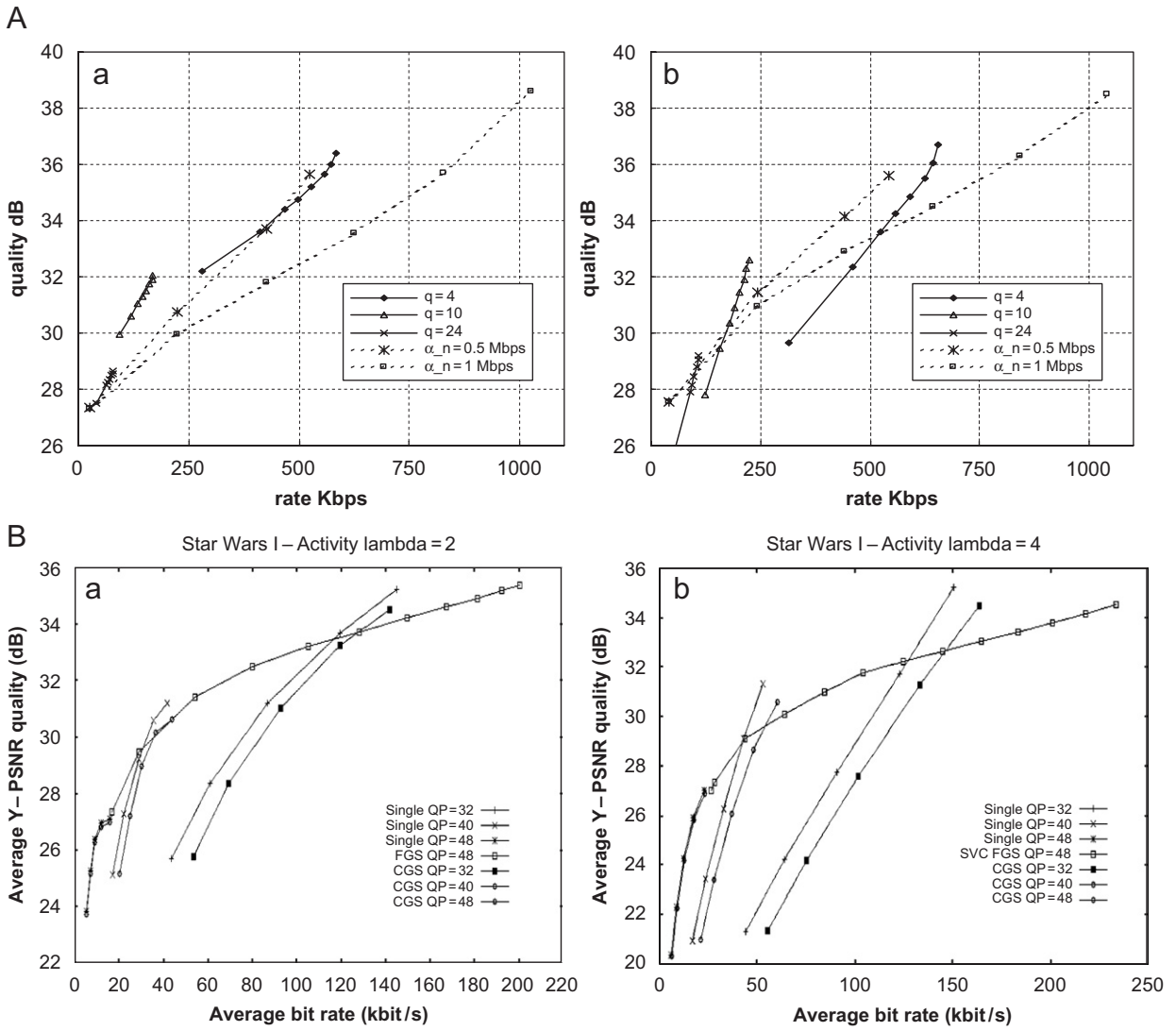


Fig. 8. (A) The reconstructed qualities as function of the overall rate $c$ for non-scalable coding (using different visual quality controlled by the quantization scale $q$) and PFGS coding (using different enhancement layer encoding rates $\alpha^n$) for video of various motion activity levels. For non-scalable coding, B-frame dropping is used to regulate the video transmission. (B) The reconstructed qualities as function of the overall rate $c$ for H.264 SVC single-layer (Single) coding, coarse grain scalable coding (CGS), and SVC FGS coding for video of various motion activity levels. For single-layer and CGS coding, B-frame dropping is used to regulate the video transmission. (a) Activity $\lambda = 2$, (b) activity $\lambda = 4$.

When temporal layers are dropped, the average bit rate is reduced as well as the average quality. We observe this behavior when traversing the curves from top right to bottom left, with the last point representing the RD point of the temporal base layer that only consists of I and P key pictures. RD points in between two RD points on these curves can be obtained by partially dropping B-frames belonging to a particular temporal layer instead of dropping the entire temporal layer. This allows for finer adjustments of the bit rate. We observe that the slope of these curves increases with the motion activity. The reason for this is that for low motion, frames are similar and frame duplication results in a lower average quality drop compared to the high motion situation where frame duplication results in more noticeable differences. At the same time, the bit rate significantly decreases with each dropped temporal layer, resulting in the observed slope behavior.

Simulcasting three single-layer streams is less efficient because redundancies exist among the three streams. SVC supports coarse grain scalability (CGS) through interlayer prediction tools to reduce these redundancies, resulting in a layered scalable bitstream. We encode each shot with CGS employing QPs 48, 40, and 32. Averaging the bit rates and qualities over each shot in each motion activity class results in the CGS curves depicted in Fig. 8B alongside the single-layer curves. Again, we drop temporal layers to obtain the RD points of each curve in the same manner as for the single-layer encodings. There is, however, a noticeable drop in RD efficiency compared to single-layer encoding when each single-layer curve is individually compared to the corresponding CGS quality layer, e.g., we compare the single-layer encoding with QP = 32 with the CGS quality layer obtained for QP = 32, which is standard in the SVC coding efficiency literature. However, the bit rate for CGS layer QP = 32 also supports the lower quality layers and hence the functionality is not comparable to single-layer coding. A fair comparison is to simulcast the three single-layer streams and to consider the aggregated bit rate when comparing to CGS. This would reveal the bit rate advantage of CGS compared to single-layer encoding for the same supported quality layer scalability. We do not provide curves for this comparison in Fig. 8B, because our goal is to switch between individual single-layer streams and not the simulcasting of all single-layer streams. Therefore, given the lower RD efficiency of CGS for this particular application, we do not further consider CGS in the discussion of this experiment.

In the next experiment, we employ SVC FGS with the same hierarchical B-frame structure as for the single-layer encodings and three FGS layers. The FGS base layer has QP = 48. Each video shot is encoded this way and subsequently bitstreams are extracted. The bit rates and average qualities of all video shots of a motion activity class are averaged and result in the FGS curves in Fig. 8B. We observe from Fig. 8B that the FGS curves intersect the single-layer curves. This means that switching between single-layer streams and FGS streams would result in a better average quality for a given bit rate, as for PFGS in Fig. 8A. (For motion activity 1, not shown due to space constraints, the FGS and single layer curves do not intersect; hence switching is limited to the single-layer streams. However, the optimal design of such a switching application could involve more FGS streams with FGS base layers that are optimally chosen with switching in mind.)

The simulation results with SVC reported in Fig. 8B demonstrate that our proposed adaptive bitstream switching can be implemented with SVC (JSVM 7.13) by generating multiple encodings with various enhancement layer encoding rates. This can be achieved by applying two-pass encoding. In the first pass, regular FGS (or AR-FGS) coding is used to generate a frame based analysis of the enhancement layer rate distribution. In the second pass, a specific encoding with a particular enhancement layer rate (which is represented by $\alpha$ in our notation) is optimized using the analysis data file of the first phase.

#### 5.1.4. Comparison of PFGS and FGS

Different bit plane coding schemes, such as FGS and PFGS, use different control methods in motion prediction and therefore have different impacts on the reconstruction quality, which we examine averaged for each motion activity level. To avoid the dependency of the reconstruction quality on the base layer coding, the performance evaluation is based on two difference metrics extracted from the transmission rate and the reconstruction quality: (i) The difference between the transmission rate and the base layer rate and (ii) the difference between the reconstructed quality of the enhancement layer at the transmission rate and the reconstruction quality of the base layer. Fig. 9 shows a comparison between FGS and PFGS streams, coded with $\alpha^n = 0.5$, 1, and 2 Mbps.

We observe from Fig. 9 that, in general, PFGS streams with a transmission rate of $\alpha^n$ give better reconstruction quality than FGS streams for any motion activity level due to using two prediction loops at the PFGS codec; compare the upper right end points of the dotted lines to the corresponding points on the solid lines in Fig. 9. The PFGS curves are generated by uniformly truncating the enhancement layer transmission rate between 0 and $\alpha^n$. For $\alpha^n = 0.5$ and 1 Mbps, the PFGS reconstructed qualities are better than the corresponding FGS reconstructed qualities for any enhancement layer transmission rate in the range from 0 to $\alpha^n$. However, for $\alpha^n = 2$ Mbps, the reconstructed quality of FGS is better than PFGS for any enhancement layer transmission rate in the range from 0 to the intersection point between the FGS curve and PFGS curve with $\alpha^n = 2$ Mbps in Fig. 9. We notice that for lower motion activity video sequences, the intersection point is closer to $\alpha^n = 2$. The similarity between

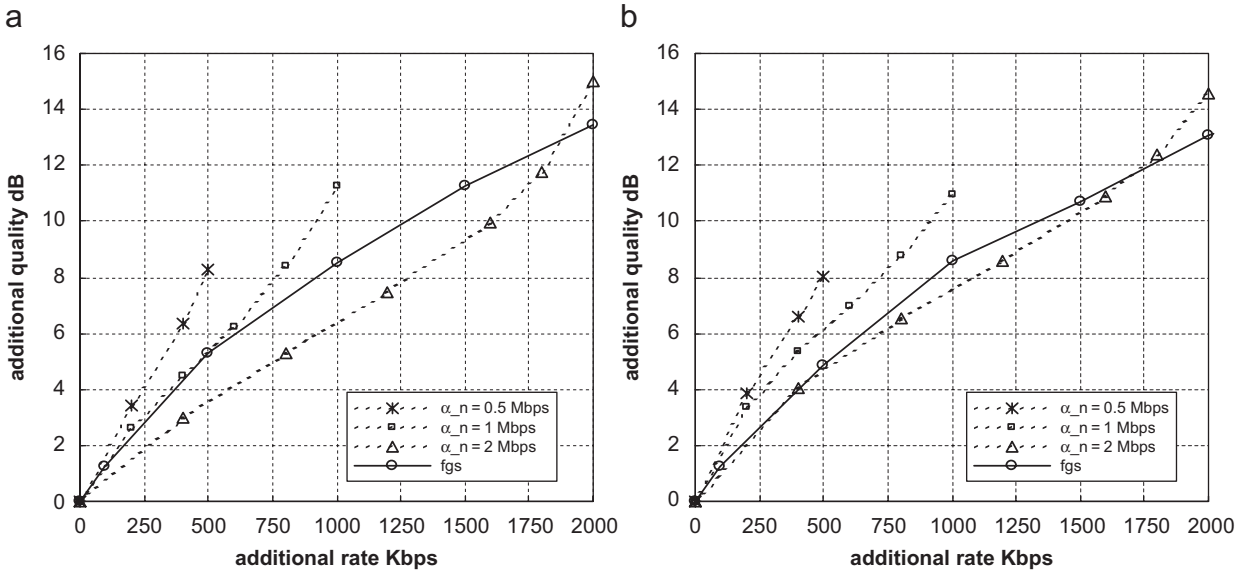


Fig. 9. The reconstructed qualities as function of the overall rate $c$ for FGS and PFGS streams: (a) activity $\lambda = 2$ and (b) activity $\lambda = 4$.

FGS and PFGS for video sequences containing high motion activity level cannot be demonstrated since we are truncating the PFGS stream in order to meet the required transmission rate. The design of a video server switching scheme to select between FGS to PFGS streams to improve the reconstruction quality can be implemented using a neural network similar to the technique proposed in [17].

### 5.2. Packet drop policies for multiplexing unicast streams

This section evaluates the four dropping policies outlined in Section 4.2 through simulation experiments with average TCP throughputs of 2, 3, 4, and 5 Mbps. For simplicity, four PFGS video streams share the communication resources, which we derived from the video shots of *Star Wars IV* as follows. An encoding with enhancement layer encoding rate of $\alpha^n = 2$ Mbps forms the basis for the streams. Four streams with enhancement layer transmission rates (at the server) of $r_j = 1.75$, 1.5, 1.25, and 1 Mbps are extracted from this $\alpha^n = 2$ Mbps encoding. We adapt this streaming with the different transmission rates from the same encoding and do not employ the adaptive encoding parameter selection of Section 4.1 (i.e., do not select the optimal $\alpha^*$ and $\beta^*$) to study in isolation the effects of the dropping policies; the adaptive rate selection of encoding rate parameters and the adaptive dropping examined here could be combined for improved efficiency. In order to create randomness similar to user demands, the start time of transmitting the video streams differs by 1 min: the video stream with 1.75 Mbps starts at the initial simulation time, the stream with 1.5 Mbps at 1 min, and so on. The rate regulation policy takes place every RTT ($= 500$ ms). The following results represent the average over the 15 min of *Star Wars IV*.

We found that the first three policies achieve very close average reconstructed qualities PSNR_AVR. This result can be attributed to the fact that these drop policies are simulated over a bottleneck link of the same rate constraint $R$, and to the relatively little diversity in shot activity levels in the link buffer. It is expected that increasing the number of video streams sharing the bottleneck link can introduce sufficient diversity in shot activity levels, such that rate regulation policies 3 and 4, which consider shot activity levels, achieve better average reconstructed qualities. Policy 4 only achieved 0.15 dB improvement over the first three policies, which is a negligible gain compared to the computational complexity of maximizing the average reconstructed qualities. We conclude that the different policies achieve approximately the same average reconstructed quality.

As illustrated in Fig. 10, policies 1 and 4 are random in nature, so that the quality fairness index for each PFGS video stream depends on the rate constraints of the link expressed by the AVR_TCP; see Eq. (11). On the other hand, policy 2 rewards low bit rate videos with higher fairness indices

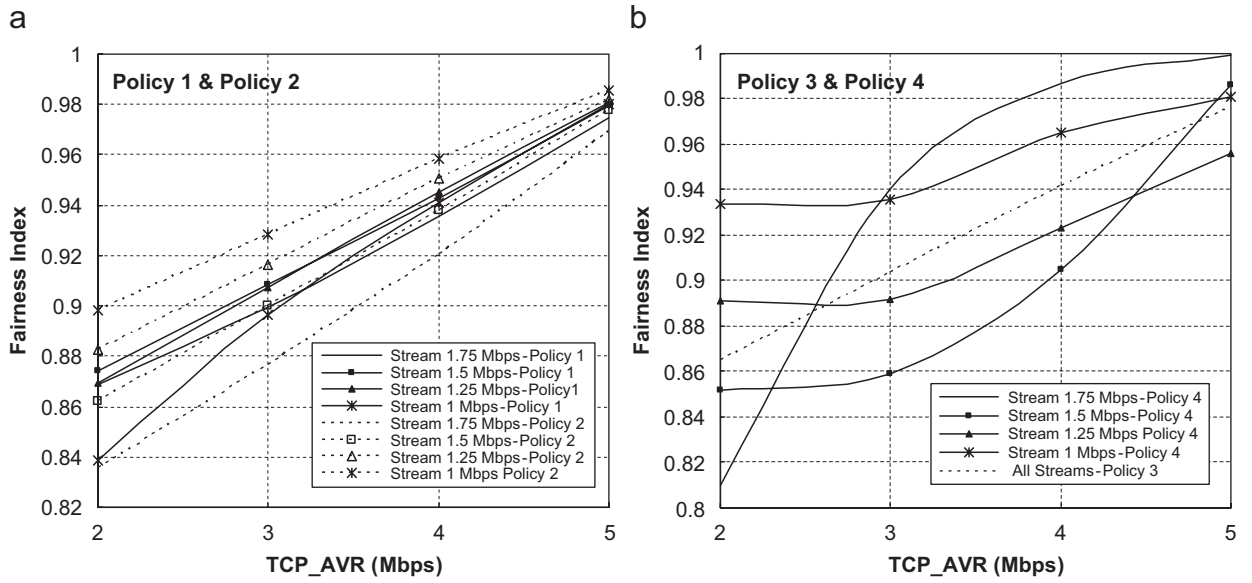


Fig. 10. Fairness index of various dropping policies.

compared to higher bit rate videos. In other words, the quality degradation for low bit rate video is reduced, while the quality degradation for high bit rate video is increased. As observed from Fig. 10(b), policy 3 achieves equality in the fairness indices among all PFGS video streams, i.e., the reconstructed qualities of each video stream are degraded fairly (by the same quality degradation ratios).

### 5.3. Encoding rate selection for multicast PFGS stream

Three separate simulation experiments were conducted to comprehensively evaluate the multicasting scenario. In these simulations, we focus on the performance comparison between FGS, PFGS with a single encoding, and adaptive PFGS with multiple encodings, as explained in Section 4.3. We consider the average additional quality improvements to the base layer quality. The receiver bit rates are randomly selected between 0.5 and 1.75 Mbps using a uniform distribution and the performance evaluations are averaged over 1000 runs of these receiver distributions. The multicasting layer rates are distributed according to two different policies: (a) uniformly distributed in the range of receiver rates (denoted as ra1) and (b) receiver rates are sorted and the multicast layer rates are selected to evenly split this ordered list (denoted as ra2). Policy ra2 is very similar to the dynamic technique proposed in

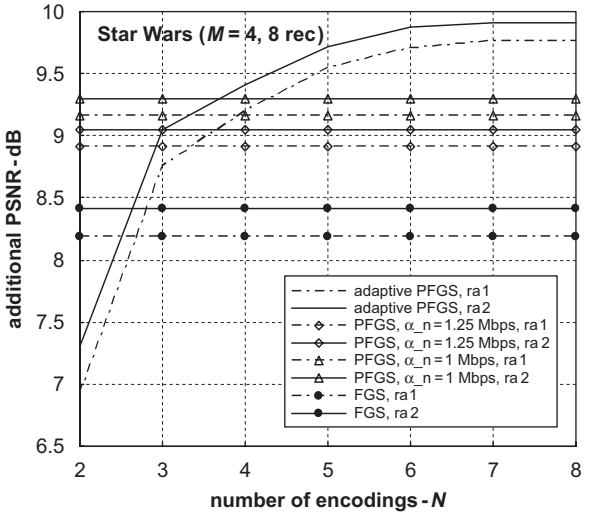


Fig. 11. The additional quality gain for PFGS and FGS multicasting as function of the number of encodings ($N$) in the case of $M = 4$ multicast layers and 8 receivers.

[21,33]. As shown in the following results, the problem of determining the multicast layer rates is orthogonal to our approach of selecting the optimal encoding for transmission. The enhancement layer encoding rates for the adaptive PFGS multicasting are set according to Eq. (12).

The first simulations target the impact of the number of encodings $N$ on the performance of the adaptive PFGS multicasting. Fig. 11 is for the

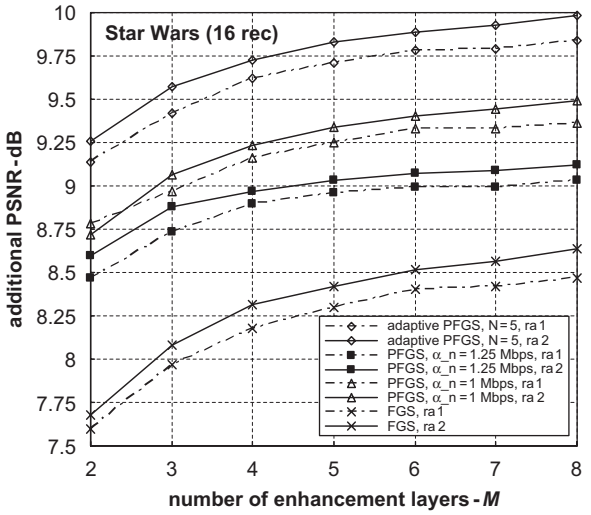


Fig. 12. The additional quality gain for a PFGS and FGS multicasting as function of the number of multicast layers ($M$) in the case of 16 receivers.

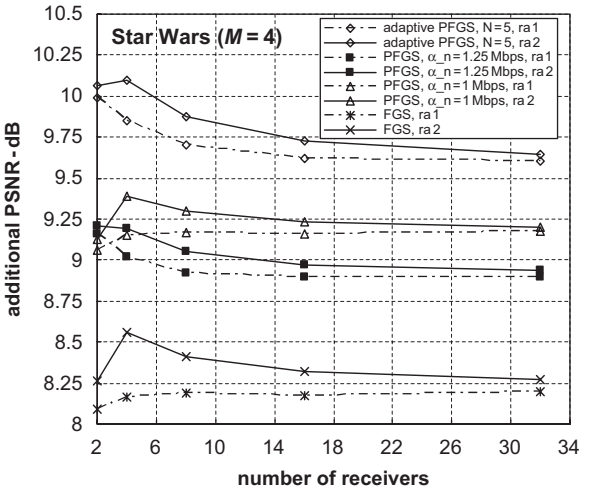


Fig. 13. The additional quality gain for PFGS and FGS multicasting as function of the number of receivers in the case of $M = 4$ multicast layers.

layered multicasting scenario using 8 receivers and $M = 4$ multicast layers for FGS streams, single PFGS streams (with $\alpha^n = 1$, 1.25, and 1.5 Mbps), and adaptive PFGS with different encodings and shows the quality improvement as a function of the number of encodings $N$ used in the adaptive PFGS multicasting. PFGS streaming using adaptive or non-adaptive multicasting techniques achieves better reconstruction quality than FGS streaming. Increasing the number $N$ of encodings beyond 6 versions has minor impact on the quality improvement of adaptive PFGS streaming. We observe in this experiment as well as subsequent experiments that policy ra2 for dynamically selecting the multicast layer rates gives an average quality improvement of 0.2 dB compared to a simple policy, such as ra1. Compared to the best single PFGS streaming (with $\alpha^n = 1$ Mbps) and FGS streaming, an additional quality of 0.54 and 1.55 dB, respectively, is achieved (either using policy ra1 or policy ra2), if $N = 6$ PFGS encodings are used.

The second simulations address the influence of increasing the number of multicast layers $M$ on the average reconstruction quality. In these simulations, the number of receivers is increased to 16 and the number of encodings for adaptive PFGS streaming is fixed at $N = 5$ encodings. Fig. 12 shows the additional quality improvement to the base layer quality for FGS streams, single PFGS streams (with $\alpha^n = 1$ and 1.25 Mbps), and adaptive PFGS streaming. The results demonstrate that the visual quality monotonically increases as the number of multicast layers $M$ increases regardless of the used policy to determine the multicast layer rates, which is consistent with the *Theorem* presented in [21]. Using more than $M = 6$ multicast layers in this scenario has a minor impact on the reconstruction quality. Adaptive PFGS streaming can maintain at least a 0.5 dB improvement to the optimal single PFGS stream and maintain at least a 1.3 dB improvement to FGS streams.

The last simulations are conducted to evaluate the impact of the number of receivers on the average reconstruction quality. Fig. 13 considers a multicasting scenario with $M = 4$ for FGS streams, single PFGS streams (with $\alpha^n = 1$ and 1.25 Mbps), and adaptive PFGS streaming (with $N = 5$). The average reconstruction quality exhibits inconsistent variations for a small number of 2 or 4 receivers (likely due to lacking accuracy for such a small number of receivers), but stabilizes for 8 or more receivers. Fig. 14 shows the reconstructed qualities of the proposed adaptive PFGS with respect to the number of encodings for multicasting scenarios with a variable number of receivers. We consistently observe that for these scenarios with 8 or more receivers, an increasing number of encodings provides improved quality up to about 6 encodings, where the quality improvement levels out.

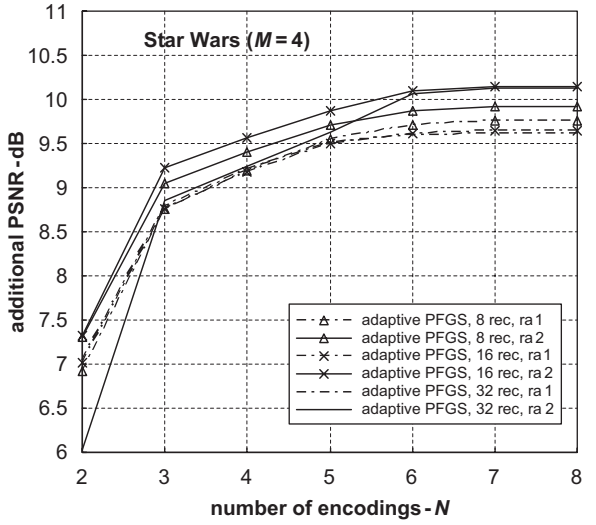


Fig. 14. The additional quality gain for adaptive PFGS as function of the number of encodings ($N$) in the case of $M = 4$ multicast layers and different numbers of receivers.

## 6. Conclusion

We have developed and evaluated a suite of network-aware adaptive bitstream switching policies for the streaming of pre-encoded fine granular scalable coded videos. Our suite encompasses bitstream switching policies for the adaptive streaming of an individual unicast stream, an adaptive packet dropping scheme for the fair multiplexing of multiple unicast streams over a shared bottleneck link, as well as for the adaptive multicasting of a video stream. The adaptive bitstream switching policies consider both the visual content variability and network bandwidth fluctuations. The adaptive bitstream switching policy for unicast streaming selects an appropriate pre-encoding from a number of stored coding versions of the video and improves the reconstructed video quality by 0.8 dB compared to the best non-adaptive scheme for a 200 scene shot sequence from *Star Wars IV*. Our performance analysis of content-dependent packet drop policies for unicast streams that share a bottleneck link considers policies utilizing the available bit budget by either randomly dropping video packets, equalizing the packet drop among shared video streams, equalizing the quality degradation among shared video streams, or optimizing the average reconstructed quality. The performance is evaluated using the average reconstructed quality and a quality-based fairness index. We have found that the policies give comparable average reconstructed quality, but only the policy equalizing the quality degradation achieves fair quality degradations. In addition, our adaptive multicast bitstream switching policy achieves improvements of an additional 0.54 dB for the 200 shot *Star Wars IV* sequence by storing multiple coding versions of a video and selecting the appropriate pre-encoding according to the network condition of the current multicasting tree and the underlying visual content. Overall, our suite of adaptive bitstream switching policies underscores that motion-related visual content shows a high correlation with the reconstructed qualities for video streaming over heterogeneous networks such as P2P overlay networks deployed over the Internet and wireless networks. Future work can extend this framework by investigating other visual content descriptors and the latest medium granularity scalability mode (MGS) of the SVC extension of H.264/AVC.

## Acknowledgments

We are grateful to Dr. Feng Wu from Microsoft China for providing the H.264/MPEG-4 AVC-PFGS codec. This work was supported in part by the National Science Foundation through Grant no. ANI-0136774. Any opinions, findings, and conclusions or recommendations expressed in this material are these of the authors and do not necessarily reflect the views of the National Science Foundation. We are indebted to the anonymous reviewers whose detailed and thoughtful feedback on earlier versions of this manuscript has helped to greatly improve this article.

## References

[1] O. Lotfallah, M. Reisslein, S. Panchanathan, Adaptive bitstream switching of pre-encoded PFGS video, in: Proceedings of the ACM Workshop on Advances in Peer-to-peer Multimedia Streaming, 2005, pp. 11–20.

[2] S.-F. Chang, A. Vetro, Video adaptation: concepts, technologies, and open issues, Proc. IEEE 93 (1) (January 2005) 148–158.

[3] W. Deng, Y.T. Hou, W. Zhu, Y.-Q. Zhang, J.M. Peha, Streaming video over the Internet: approaches and directions, IEEE Trans. Circuits Systems Video Technol. (CSVT) 11 (3) (2001) 282–300.

[4] O. Lotfallah, S. Panchanathan, Adaptive scheme for internet video transmission, Proc. IEEE ISCAS 2 (2003) 872–875.

[5] S.-R. Kang, D. Loguinov, Impact of FEC overhead on scalable video streaming, Proc. ACM NOSSDAV (2005) 123–128.

[6] J.-R. Ohm, Advances in scalable video coding, Proc. IEEE 93 (1) (2005) 42–56.
[7] F. Yang, Q. Zhang, W. Zhu, Y.-Q. Zhang, Bit allocation for scalable video streaming over mobile wireless Internet, Proc. IEEE INFOCOM 3 (2004) 2142–2151.
[8] Q. Zhang, G. Wang, W. Zhu, Y.-Q. Zhang, Robust scalable video streaming over Internet with network-adaptive congestion control and unequal loss protection, in: Proceedings of the Packet Video Workshop, Kyongju, Korea, 2001.
[9] W. Li, Overview of the fine granularity scalability in MPEG-4 video standard, IEEE Trans. CSVT 11 (3) (2001) 301–317.
[10] F. Wu, S. Li, Y.-Q. Zhang, A framework for efficient progressive fine granularity scalable video coding, IEEE Trans. CSVT 11 (3) (2001) 332–344.
[11] Y. He, F. Wu, S. Li, Y. Zhong, S. Yang, H.26L-based fine granularity scalable video coding, Proc. IEEE ISCAS 4 (2002) IV548–IV551.
[12] Y. Bao, M. Karczewicz, X. Wang, J. Ridge, FGS coding with adaptive reference for low-delay application, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), October 2006, pp. 185–188.
[13] J. Xu, F. Wu, S. Li, Transcoding for progressive fine granularity scalable vide coding, Proc. IEEE ISCAS 3 (2004) III-765-8.
[14] P. de Cuetos, P. Seeling, M. Reisslein, K.W. Ross, Comparing the streaming of FGS-encoded video at different aggregation levels: frame, GoP, scene, Int. J. Commun. Syst. 18 (5) (2005) 449–464.
[15] M. Dai, D. Loguinov, H. Radha, Rate-distortion modeling of scalable video coders, Proc. IEEE ICIP 2 (2004) 1093–1096.
[16] O. Lotfallah, M. Reisslein, S. Panchanathan, A framework for advanced video traces, EURASIP J. Appl. Sig. Process., vol. 2006, Article ID 42083, pp. 1–21.
[17] O. Lotfallah, M. Reisslein, S. Panchanathan, Adaptive video transmission schemes using MPEG-7 motion intensity descriptor, IEEE Trans. CSVT, 2006, vol. 16, no. 8, pp. 929–946.
[18] X. Sun, F. Wu, S. Li, W. Gao, Y.-Q. Zhang, Seamless switching of scalable video bitstreams for efficient streaming, IEEE Trans. Multimedia 6 (2) (April 2004) 291–303.
[19] S. Gorinsky, H. Vin, The utility of feedback in layered multicast congestion control, Proc. ACM NOSSDAV, 2001, pp. 93–102.
[20] M.R. Ito, V. Bai, A packet discard scheme for loss control in IP networks with MPEG video traffic, in: The Eighth International Conference on Communication Systems, 1, August 2002, pp. 497–503.
[21] J. Liu, B. Li, Y.-Q. Zhang, An end-to-end adaptation protocol for layered video multicast using optimal rate allocation, IEEE Trans. Multimedia 6 (1) (2004) 87–102.
[22] N. Wakamiya, M. Miyabayashi, M. Murata, MPEG-4 Video Transfer with TCP-friendly Rate Control, vol. 2216, Springer, Berlin, 2001, pp. 29–42.
[23] Z.-L. Zhang, S. Nelakuditi, R. Aggarwa, R.P. Tsang, Efficient server selective frame discard algorithms for stored video delivery over resource constrained networks, Proc. IEEE INFOCOM (1999) 472–479.
[24] A. Divakaran, A. Vetro, K. Asai, H. Nishikawa, Video browsing system based on compressed domain feature extraction, IEEE Trans. Consumer Electron. 46 (3) (2000) 637–644.
[25] S. Jeannin, A. Divakaran, MPEG-7 visual motion descriptors, IEEE Trans. CSVT 11 (6) (2001) 720–724.
[26] K.A. Peker, A. Divakaran, Framework for measurement of the intensity of motion activity of video segments, J. Vis. Commun. Image Represent. 15 (3) (2004) 265–284.
[27] X. Sun, B.S. Manjunath, A. Divakaran, Representation of motion activity in hierarchical levels for video indexing and filtering, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 1, September 2002, pp. I-149–I-152, vol.1.
[28] X. Li, M. Ammar, S. Paul, Video multicast over the Internet, IEEE Network Mag. 13 (2) (2003) 46–60.
[29] J. Liu, B. Li, Y.-Q. Zhang, Adaptive video multicast over the Internet, IEEE Multimedia 10 (1) (2003) 22–33.
[30] S. McCanne, V. Jacobson, M. Vetterli, Receiver-driven layered multicast, Proc. ACM Sigcomm (1996) 117–130.
[31] S. Cheung, M. Ammar, X. Li, On the use of destination set grouping to improve fairness in multicast video distribution, Proc. IEEE INFOCOM (1996) 553–560.
[32] T. Jiang, E. Zegura, M. Ammar, Inter-receiver fair multicast communication over the Internet, Proc. ACM NOSSDAV (1999) 103–114.
[33] J. Liu, B. Li, Y.-Q. Zhang, A hybrid adaptation protocol for TCP-friendly layered multicast and its optimal rate allocation, Proc. IEEE INFOCOM 3 (2002) 1520–1529.
[34] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, Proc. ACM SIGCOMM 28 (4) (1998) 303–314.
[35] S. Wenger, Error patterns for internet experiments, in ITU Telecommunications Standardization Sector (October 1999) Document Q15-I-16r1.
[36] J.S. Boreczky, L.A. Rowe, Comparison of video shot boundary detection techniques, SPIE Storage and Retrieval for Still Images and Video Databases IV 2664 (1996) 170–179.
[37] Y. Chen, K. Yu, J. Li, S. Li, An error concealment algorithm for entire frame loss in video transmission, in: Proceedings of the Picture Coding Symposium, December 2004.
[38] Y. Chen, K. Xie, F. Zhang, P. Pandit, J. Boyce, Frame loss error concealment for SVC, J. Zhejiang Univ. SCIENCE A 7 (5) (2006) 677–683.