

Scalable Video Transcaling for the Wireless Internet

Hayder Radha

*Department of Electrical and Computer Engineering, Michigan State University, MI 48824-1226, USA
Email: radha@msu.edu*

Mihaela van der Schaar

*Department of Electrical and Computer Engineering, University of California, Davis, CA 95616-5294, USA
Email: mvanderschaar@ece.ucdavis.edu*

Shirish Karande

*Department of Electrical and Computer Engineering, Michigan State University, MI 48824-1226, USA
Email: karandes@msu.edu*

Received 5 December 2002; Revised 28 July 2003

The rapid and unprecedented increase in the heterogeneity of multimedia networks and devices emphasizes the need for scalable and adaptive video solutions for both coding and transmission purposes. However, in general, there is an inherent trade-off between the level of scalability and the quality of scalable video streams. In other words, the higher the bandwidth variation, the lower the overall video quality of the scalable stream that is needed to support the desired bandwidth range. In this paper, we introduce the notion of wireless video transcaling (TS), which is a generalization of (non-scalable) transcoding. With TS, a scalable video stream, that covers a given bandwidth range, is mapped into one or more scalable video streams covering different bandwidth ranges. Our proposed TS framework exploits the fact that the level of heterogeneity changes at different points of the video distribution tree over wireless and mobile Internet networks. This provides the opportunity to improve the video quality by performing the appropriate TS process. We argue that an Internet/wireless network gateway represents a good candidate for performing TS. Moreover, we describe hierarchical TS (HTS), which provides a “transcaler” with the option of choosing among different levels of TS processes with different complexities. We illustrate the benefits of TS by considering the recently developed MPEG-4 fine granularity scalability (FGS) video coding. Extensive simulation results of video TS over bit rate ranges supported by emerging wireless LANs are presented.

Keywords and phrases: transcoding, FGS, scalable, video, transcaling, streaming.

1. INTRODUCTION

The level of heterogeneity in multimedia communications has been influenced significantly by new wireless LANs and mobile networks. In addition to supporting traditional web applications, these networks are emerging as important Internet video access systems. Meanwhile, both the Internet [1, 2, 3] and wireless networks are evolving to higher bit rate platforms with even larger amount of possible variations in bandwidth and other quality of services (QoS) parameters. For example, IEEE 802.11a and HiperLAN2 wireless LANs are supporting (physical layer) bit rates from 6 Mbps to 54 Mbps (see, e.g., [4, 5]). Within each of the supported bit rates, there are further variations in bandwidth due to the shared nature of the network and the heterogeneity of the devices and the quality of their physical connections. Moreover, wireless LANs are expected to provide higher bit rates than mobile networks (including third generation) [6]. In

the meantime, it is expected that current wireless and mobile access networks (e.g., 2G and 2.5G mobile systems and sub-2 Mbps wireless LANs) will coexist with new generation systems for sometime to come. All of these developments indicate that the level of heterogeneity and the corresponding variation in available bandwidth could be increasing significantly as the Internet and wireless networks converge more and more into the future. In particular, if we consider the different wireless/mobile networks as a large multimedia heterogeneous access system for the Internet, we can appreciate the potential challenge in addressing the bandwidth variation over this system.

Many scalable video compression methods have been proposed and used extensively in addressing the bandwidth variation and heterogeneity aspects of the Internet and wireless networks (e.g., [7, 8, 9, 10, 11, 12, 13, 14, 15, 16]). Examples of these include receiver-driven multicast multilayer coding, MPEG-4 fine granularity scalability (FGS)

compression, and H.263-based scalable methods. These and other similar approaches usually generate a base layer (BL) and one or more enhancement layers (ELs) to cover the desired bandwidth range. Consequently, these approaches can be used for multimedia unicast and multicast services over wireless Internet networks.

In general, the wider the bandwidth range¹ that needs to be covered by a scalable video stream, the lower the overall video quality is² [13]. With the aforementioned increase in heterogeneity over emerging wireless multimedia Internet protocol (IP) networks, there is a need for scalable video coding and distribution solutions that maintain good video quality while addressing the high level of anticipated bandwidth variation over these networks. One trivial solution is the generation of multiple streams that cover different bandwidth ranges. For example, a content provider, which is covering a major event, can generate one stream that covers 100–500 kbps, another that covers 500–1000 kbps and yet another stream to cover 1000–2000 kbps and so on. Although this solution may be viable under certain conditions, it is desirable from a content provider perspective to generate the fewest number of streams that covers the widest possible audience. Moreover, multicasting multiple scalable streams (each of which consists of multiple multicast sessions) is inefficient in terms of bandwidth utilization over the wired segment of the wireless IP network. (In the above example, a total bit rate of 3500 kbps is needed over a link transmitting the three streams, while only 2000 kbps of bandwidth is needed by a scalable stream that covers the same bandwidth range.)

In this paper, we propose a new approach for addressing the bandwidth variation issue over emerging wireless and mobile multimedia IP networks. We refer to this approach as transcoding (TS) since it represents a generalization of video transcoding. Video transcoding implies the mapping of a non-scalable video stream into another non-scalable stream coded at a bit rate *lower* than the first stream bit rate. With TS, one or more scalable streams covering different bandwidth ranges are derived from another scalable stream. While transcoding always degrades the video quality of the already-coded (non-scalable) video, a *transcoded* video could have a significantly better quality than the (original) scalable video stream prior to the TS operation. This represents a key difference between (non-scalable) transcoding and the proposed TS framework. TS can be supported at gateways between the wired Internet and wireless/mobile access networks (e.g., at a proxy server adjunct to an access point (AP) of a wireless LAN). We believe that this approach provides an efficient method for delivering good quality video over the high-bit rate wireless LANs while maintaining efficient utilization of

the overall (wired/wireless) distribution network bandwidth. Therefore, different gateways of different wireless LANs and mobile networks can perform the desired TS operations that are suitable for their own local domains and the devices attached to them. This way, users of new higher-bandwidth LANs do not have to sacrifice in video quality due to coexisting with legacy wireless LANs or other low bit rate mobile networks. Similarly, powerful clients (e.g., laptops and PCs) can still receive high-quality video even if there are other low-bit-rate low-power devices that are being served by the same wireless/mobile network. Moreover, when combined with embedded video coding schemes and the basic tools of receiver-driven multicast, TS provides an efficient framework for video multicast over the wireless Internet.

In addition to introducing the notion of TS and describing how it can be used for unicast and multicast video services over wireless IP networks, we illustrate the level of quality improvement that TS can provide by presenting several video simulation results for a variety of TS cases. The remainder of the paper is organized as follows. Section 2 describes the wireless video TS framework with some focus on IP multicast applications. This section also highlights some of the key attributes and basic definitions of TS-based wireless systems and how they differ from traditional transcoding-based platforms. Section 3 describes hierarchical TS (HTS), which is a framework that enables transcoders to trade off video quality with complexity. HTS is described using a concrete example that is based on the MPEG-4 FGS video coding method. Then, two classes of TS are considered: full and partial. Section 4 described full TS for wireless LANs. Section 4 also shows simulation results of applying FTS on FGS streams and the level of video quality improvement one can gain by utilizing this approach. Section 5 complements Section 4 by describing partial TS and presenting results for performing PTS on the FGS temporal (FGST) coding method. Section 6 concludes the paper with a summary.

2. TRANSCALING-BASED MULTICAST (TSM) FOR VIDEO OVER THE WIRELESS INTERNET

A simple case of our proposed TS approach can be described within the context of receiver-driven layered multicast (RLM). Therefore, first, we briefly outline some of the basic characteristics of the RLM framework in order to highlight how this framework can be extended to our wireless video TS-based solution. Then, we describe some general features of a TS-based wireless Internet system.

RLM of video is based on generating a layered coded video bitstream that consists of multiple streams. The minimum quality stream is known as BL and the other streams are ELs [17]. These multiple video streams are mapped into a corresponding number of “multicast sessions.” A receiver can subscribe to one (the BL stream) or more (BL plus one or more ELs) of these multicast sessions depending on the receiver’s access bandwidth to the Internet. Receivers can subscribe to more multicast sessions or “unsubscribe” to some of the sessions in response to changes in the available

¹A more formal definition of “bandwidth range” will be introduced later in the paper.

²This is particularly true for the scalable schemes that fall under the category of signal-to-noise ratio (SNR) scalability methods. These include the MPEG-2 and MPEG-4 SNR scalability methods, and the newly developed MPEG-4 FGS method.

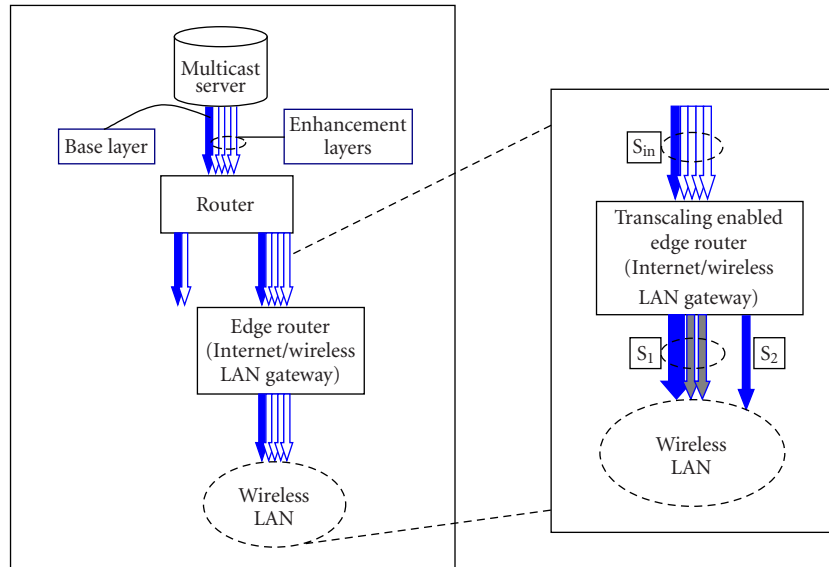


FIGURE 1: A simplified view of a wireless video TS platform within an RLM architecture.

bandwidth over time. The “subscribe” and “unsubscribe” requests generated by the receivers are forwarded upstream toward the multicast server by the different IP multicast-enabled routers between the receivers and the server. This approach results in an efficient distribution of video by utilizing minimal bandwidth resources over the multicast tree. The overall RLM framework can also be used for wireless IP devices that are capable of decoding the scalable content transmitted by an IP multicast server. The left picture of Figure 1 shows a simple example of an RLM-based system.

Similar to RLM, TS-based multicast (TSM) is driven by the receivers’ available bandwidth and their corresponding requests for viewing scalable video content. However, there is a fundamental difference between the proposed TSM framework and traditional RLM. Under TSM, an edge router³ with a TS capability (or a “transcaler”) derives new scalable streams from the original stream. A derived scalable stream could have a BL and/or EL(s) that are different from the BL and/or ELs of the original scalable stream. The objective of the TS process is to improve the overall video quality by taking advantage of reduced uncertainties in the bandwidth variation at the edge nodes of the multicast tree.

For a wireless Internet multimedia service, an ideal location where TS can take place is at a gateway between the wired Internet and the wireless segment of the end-to-end network. The right picture of Figure 1 shows an example of a TSM system where a gateway node receives a layered video stream⁴ with a BL bit rate R_{\min_in} . The bit rate range covered by this layered set of streams is $R_{\text{range_in}} = [R_{\min_in}, R_{\max_in}]$.

³The transcaling process does not necessarily take place in the edge router itself but rather in a proxy server (or a gateway) that is adjunct to the router.

⁴Here, a layered or “scalable” stream consists of multiple substreams.

The gateway transcales the input layered stream S_{in} into another scalable stream S_1 . This new stream serves, for example, relatively high-bandwidth devices (e.g., laptops or PCs) over the wireless LAN. As shown in the figure, the new stream S_1 has a BL with a bit rate R_{\min_1} which is higher than the original BL bit rate: $R_{\min_1} > R_{\min_in}$. Consequently, in this example, the transcaler requires at least one additional piece of information, and that is the minimum bit rate R_{\min_1} needed to generate the new scalable video stream. This information can be determined, based on analyzing the wireless links of the different devices connected to the network.⁵ By interacting with the access point, the gateway server can determine the bandwidth range needed for serving its devices. As illustrated by our simulations, this approach could improve the video quality delivered to higher-bit rate devices significantly.

2.1. Attributes of wireless video-transcaling-based systems

Here, we highlight the following attributes of the proposed wireless video TS framework.

- (1) Supporting TS at edge nodes (wireless LANs’ and mobile networks’ gateways) preserves the ability of the local networks to serve low-bandwidth low-power devices (e.g., handheld devices). This is illustrated in Figure 1. In this example, in addition to generating the scalable stream S_1 (which has a BL bit rate that is

⁵Determining the particular bit rate range over an underlying (wireless or wired) network is an important aspect of any adaptive multimedia solution, including TS. However, this aspect, which could include a variety of important topics and techniques such as congestion control, bandwidth estimation, and cross-layer communication and design, is beyond the scope of this paper.

higher than the bit rate of the input BL stream), the transcaler delivers the original BL stream to the low-bit rate devices.

- (2) The TSM system (described above) falls under the umbrella of active networks⁶ where, in this case, the transcaler provides network-based added value services [18]. Therefore, TSM can be viewed as a generalization of some recent work on active based networks with (non-scalable) video transcoding capabilities of MPEG streams.
- (3) A wireless video transcaler can always fall back to using the original (lower-quality) scalable video. This “fall-back” feature represents a key attribute of TS that distinguishes it from non-scalable transcoding. The fall-back feature could be needed, for example, when the Internet wireless gateway (or whoever the transcaler happens to be) does not have enough processing power for performing the desired TS process(es). Therefore, and unlike (non-scalable) transcoding-based services, TS provides a scalable framework for delivering higher quality video. A more graceful TS framework (in terms of computational complexity) is also feasible as will be explained later in this paper.
- (4) Although we have focused on describing our proposed wireless video TS approach in the context of multicast services, on-demand unicast applications can also take advantage of TS. For example, a wireless or mobile gateway may perform TS on a popular video clip that is anticipated to be viewed by many users on-demand. In this case, the gateway server has a better idea on the bandwidth variation that it (i.e., the server) has experienced in the past, and consequently, it generates the desired scalable stream through TS. This scalable stream can be stored locally for later viewing by the different devices served by the gateway.
- (5) As illustrated by our simulation results, TS has its own limitations in improving the video quality over the whole desired bandwidth range. Nevertheless, the improvements that TS provides are significant enough to justify its merit over a subset of the desired bandwidth range. This aspect of TS will be explained further later in the paper.
- (6) TS can be applied to any form of scalable streams (i.e., SNR, temporal, and/or spatial). In this paper, we will show examples of TS operations that are applied to SNR-scalable and hybrid SNR-temporal streams over bit rates that are applicable to new wireless LANs (e.g., 802.11). The level of improvement in video quality for both cases is also presented.

Before proceeding, it is important to introduce some basic definitions of TS. Here, we define two types of TS processes: down TS (DTS) and up TS (UTS).

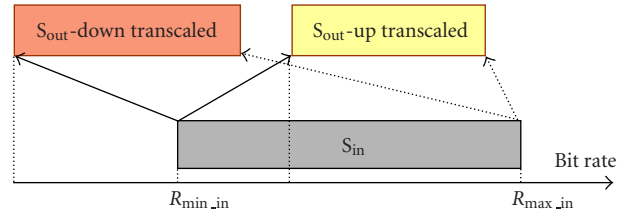


FIGURE 2: The distinction between DTS and UTS.

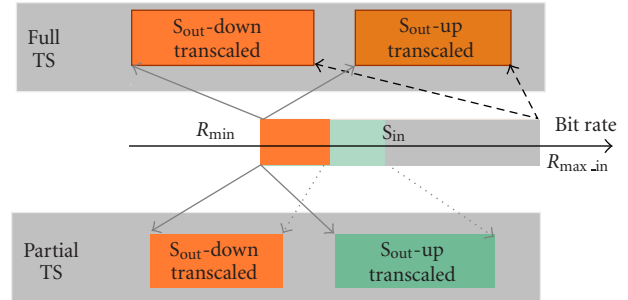


FIGURE 3: An example illustrating the different TS categories.

Let the original (input) scalable stream S_{in} of a transcaler covers a bandwidth range

$$R_{range_in} = [R_{min_in}, R_{max_in}]. \quad (1)$$

And let a transcaled stream has a range

$$R_{range_out} = [R_{min_out}, R_{max_out}]. \quad (2)$$

Then, DTS occurs when $R_{min_out} < R_{min_in}$, while UTS occurs when $R_{min_in} < R_{min_out} < R_{max_in}$. The distinction between DTS and UTS is illustrated in Figure 2. DTS resembles traditional non-scalable transcoding in the sense that the bit rate of the output BL is lower than the bit rate of the input BL. Many researchers have studied this type of down conversion in the past.⁷ However, up conversion has not received much attention (if any). Therefore, in the remainder of this paper, we will focus on UTS. (Unless otherwise mentioned, we will use UTS and TS interchangeably.)

Another important classification of TS is the distinction between FTS and PTS (see Figure 3). Our definition of FTS implies two things: (a) all of the input stream data (BL stream and EL stream) is used to perform the TS operation; and (b) *all* pictures of both BL and EL have been modified by TS. PTS is achieved if either of these two criteria is not met. Consequently, PTS provides a lower-complexity TS option that enables transcalers to trade off quality for complexity. Examples of both PTS and are covered in this paper.

⁶We should emphasize here that the area of active networks covers many aspects, and “added value services” is just one of these aspects.

⁷We should emphasize here, however, that we are not aware of any previous efforts of down converting a scalable stream into another scalable stream.

3. HIERARCHICAL TRANSCALING FOR THE WIRELESS INTERNET

After the above introduction to TS, its general features, potential benefits, and basic definitions, we now describe HTS for the wireless Internet. In order to provide a concrete example of HTS, we describe it in the context of the MPEG-4 FGS scalable video coding method. Hence, we start Section 3.1 with a very brief introduction to MPEG-4 FGS and its coding tools that have been developed in support of video streaming applications over the Internet and wireless networks.

3.1. The MPEG-4 FGS video coding method⁸

In order to meet the bandwidth variation requirements of the Internet and wireless networks, FGS encoding is designed to cover any desired bandwidth range while maintaining a very simple scalability structure [13]. As shown in Figure 4, the FGS structure consists of only two layers: a BL coded at a bit rate R_b and a single EL coded using a fine grained (or totally embedded) scheme to a maximum bit rate of R_e .

This structure provides a very efficient, yet simple, level of abstraction between the encoding and streaming processes. The encoder only needs to know the range of bandwidth $[R_{\min} = R_b, R_{\max} = R_e]$ over which it has to code the content, and it does not need to be aware of the particular bit rate the content will be streamed at. The streaming server on the other hand has a total flexibility in sending any desired portion of any EL frame (in parallel with the corresponding BL picture), without the need for performing complicated real-time rate control algorithms. This enables the server to handle a very large number of unicast streaming sessions and to adapt to their bandwidth variations in real time. On the receiver side, the FGS framework adds a small amount of complexity and memory requirements to any standard motion-compensation-based video decoder. As shown in Figure 4, the MPEG-4 FGS framework employs two encoders: one for the BL and the other for the EL. The BL is coded with the MPEG-4 motion compensation DCT-based video encoding method (non-scalable). The EL is coded using bit-plane-based embedded DCT coding.

FGS also supports temporal scalability (FGST) that allows for trade-offs between SNR and motion-smoothness at transmission time. Moreover, the FGS and FGST frames can be distributed using a single bitstream or two separate streams depending on the needs of the applications. Below, we will assume that MPEG-4 FGS/FGST video is transmitted using three separate streams: one for the BL, one for the SNR FGS frames, and the third one for the FGST frames.

For receiver-driven multicast applications (Figure 5), FGS provides a flexible framework for the encoding, streaming, and decoding processes. Identical to the unicast case, the encoder compresses the content using any desired range of

bandwidth $[R_{\min} = R_b, R_{\max} = R_e]$. Therefore, the same compressed streams can be used for both unicast and multicast applications. At time of transmission, the multicast server partitions the FGS EL into any preferred number of “multicast channels” each of which can occupy any desired portion of the total bandwidth. At the decoder side, the receiver can “subscribe” to the “BL channel” and to any number of FGS EL channels that the receiver is capable of accessing (depending, e.g., on the receiver access bandwidth). It is important to note that regardless of the number of FGS EL channels that the receiver subscribes to, the decoder has to decode only a single EL.

The above advantages of the FGS framework are achieved while maintaining good coding efficiency results. However, similar to other scalable coding schemes, FGS’s overall performance can degrade as the bandwidth range that an FGS stream covers increases.

3.2. Hierarchical transcoding of MPEG-4 FGS video streams

Examples of TS an MPEG-4 FGS stream are illustrated in Figure 6. Under the first example, the input FGS stream S_{in} is transcaled into another scalable stream S_1 . In this case, the BL_{in} of S_{in} (with bit rate $R_{\min, in}$) and a certain portion of the EL_{in} are used to generate a new base layer, BL_1 . If R_{e1} represents the bit rate of the EL_{in} used to generate the new BL_1 , then this new BL’s bit rate $R_{\min, 1}$ satisfies the following:

$$R_{\min, in} < R_{\min, 1} < R_{\min, in} + R_{e1}. \quad (3)$$

Consequently, and based on the definition we adopted earlier for UTS and DTS, this example represents a UTS scenario. Furthermore, in this case, both the BL and EL of the input stream S_{in} has been modified. Consequently, this represents a FTS scenario. FTS can be implemented using cascaded decoder-encoder systems (as we will show in the simulation results section). This, in general could provide high-quality improvements at the expense of computational complexity at the gateway server.⁹

The residual signal between the original stream S_{in} and the new BL_1 stream is coded using FGS EL compression. Therefore, this is an example of TS an FGS stream with a bit rate range $R_{\text{range}, in} = [R_{\min, in}, R_{\max, in}]$ to another FGS stream with a bit rate range $R_{\text{range}, 1} = [R_{\min, 1}, R_{\max, 1}]$. It is important to note that the maximum bit rate $R_{\max, 1}$ can be (and should be) selected to be smaller than the original maximum bit rate¹⁰ $R_{\max, in}$:

$$R_{\max, 1} < R_{\max, in}. \quad (4)$$

⁹To reduce the complexity of FTS, one can reuse the motion vectors of the original FGS stream S_{in} . Reusing the same motion vectors, however, may not provide the best quality as has been shown in previous results for non-scalable TS.

¹⁰It is feasible that the *actual* maximum bit rate of the transcaled stream S_1 is higher than the maximum bit rate of the original input stream S_{in} . However, and as expected, this increase in bit rate does not provide any quality improvements as we will see in the simulation results. Consequently, it is important to truncate a transcaled stream at a bit rate $R_{\max, 1} < R_{\max, in}$.

⁸This brief subsection is mainly provided to make the paper self-contained. Readers who are familiar with the FGS framework can skip this subsection without affecting their understanding of the remainder of the paper.

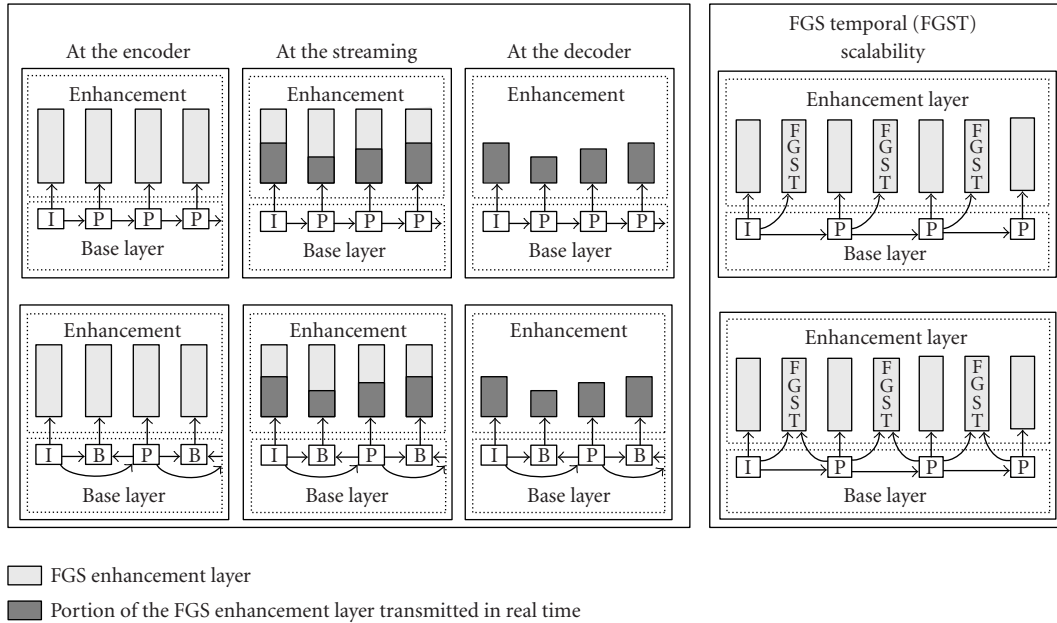


FIGURE 4: Examples of the MPEG-4 FGS and FGST scalability structures. Examples of the hybrid temporal SNR scalability structures are shown on the right-hand side of the figure. Both of bidirectional (lower right structure) and forward-prediction (top right figure) FGST picture types are supported by the MPEG-4 FGS/FGST standard.

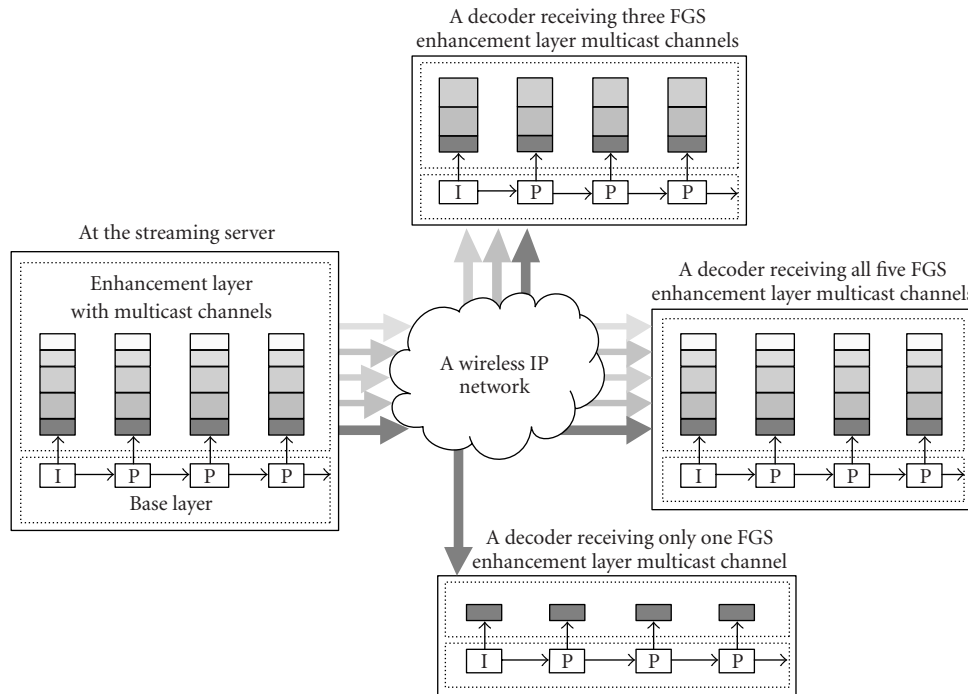


FIGURE 5: An example of video multicast using MPEG-4 FGS over a wireless IP network.

As we will see in the simulation section, the quality of the new stream S_1 at $R_{\max-1}$ could still be higher than the quality of the original stream S_{in} at a higher bit rate $R \gg R_{\max-1}$. Consequently, TS could enable a device which has a band-

width $R \gg R_{\max-1}$ to receive a better (or at least similar) quality video while saving some bandwidth. (This access bandwidth can be used, e.g., for other auxiliary or non-real-time applications.)

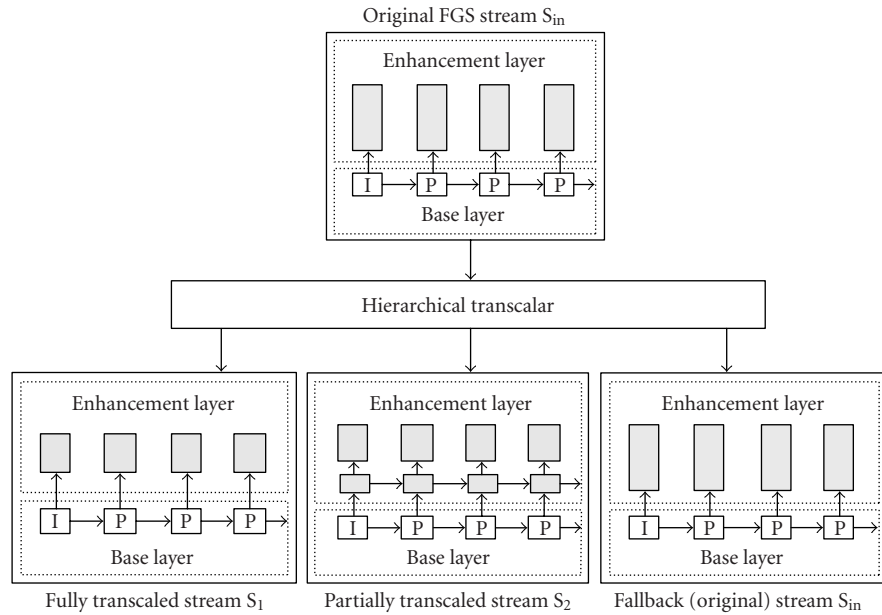


FIGURE 6: Examples of HTS of the MPEG-4 FGS scalability structure with an FTS option.

As mentioned above, under FTS, all pictures of both the BL and EL of the original FGS stream S_1 have been modified. Although the original motion vectors can be reused here, this process may be computationally complex for some gateway servers. In this case, the gateway can always fall back to the original FGS stream, and consequently, this provides some level of computational scalability.

Furthermore, FGS provides another option for TS. Here, the gateway server can transcode the EL only. This is achieved by (a) decoding a portion of the EL of one picture, and (b) using that decoded portion to predict the next picture of the EL, and so on. Therefore, in this case, the BL of the original FGS stream S_{in} is not modified and the computational complexity is reduced compared to FTS of the whole FGS stream (i.e., both BL and EL). Similar to the previous case, the motion vectors from the BL can be reused here for prediction within the EL to reduce the computational complexity significantly.

Figure 6 shows the three options described above for supporting HTS of FGS (SNR only) streams: FTS, PTS, and the fallback (no TS) option. Depending on the processing power available to the gateway, the system can select one of these options. The TS process with the higher complexity provides bigger improvements in video quality.

It is important to note that within each of the above TS options, one can identify further alternatives to achieve more graceful TS in terms computational complexity. For example, under each option, one may perform the desired TS on a fewer number of frames. This represents some form of temporal TS. Examples of this type of temporal TS and corresponding simulation results for wireless LANs bit rates are described in Section 5. Before proceeding, we show simulation results for FTS in the following section.

4. FULL TRANSCALING FOR HIGH-BIT-RATE WIRELESS LANs

In order to illustrate the level of video quality improvements that TS can provide for wireless Internet multimedia applications, in this section, we present some simulation results of FGS-based FTS.

We coded several video sequences using the draft standard of the MPEG-4 FGS encoding scheme. These sequences were then modified using the full transcoder architecture shown in Figure 7. The main objective for adopting the transcoder shown in the figure is to illustrate the potential of video TS and highlight some of its key advantages and limitations.¹¹

The level of improvements achieved by TS depends on several factors. These factors include the type of video sequence that is being transcoded. For example, certain video sequences with a high degree of motion and scene changes are coded very efficiently with FGS [13]. Consequently, these sequences may not benefit significantly from TS. On the other end, sequences that contain detailed textures and exhibit a high degree of correlation among successive frames could benefit from TS significantly. Overall, most sequences gained visible quality improvements from TS.

Another key factor is the range of bit rates used for both the input and output streams. Therefore, we first need to

¹¹Other elaborate architectures or algorithms can be used for performing FTS. However, these elaborate algorithms will bias some of our findings regarding the full potential of TS and its performance. Examples of these algorithms include refinement of motion vectors instead of a full re-computation of them; TS in the compressed DCT domain; and similar techniques.

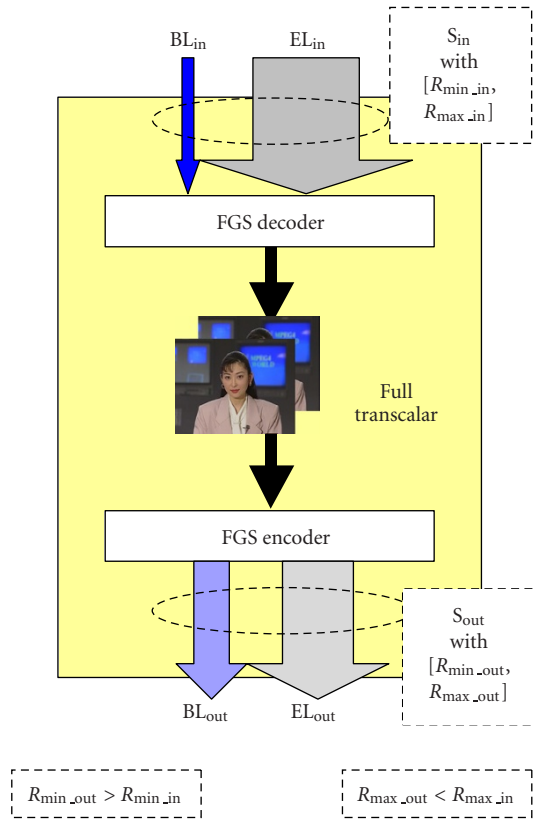


FIGURE 7: The full transcaler architecture used for generating the simulation results shown here.

decide on a reasonable set of bit rates that should be used in our simulations. As mentioned in the introduction, new wireless LANs (e.g., 802.11a or HiperLAN2) could have bit rates on the order of tens of Mbps (e.g., more than 50 Mbps). Although it is feasible that such high bit rates may be available to one or few devices at certain points in time, it is unreasonable to assume that a video sequence should be coded at such high bit rates. Moreover, in practice, most video sequences¹² can be coded very efficiently at bit rates below 10 Mbps. Consequently, the FGS sequences we coded were compressed at maximum bit rates (i.e., R_{\max_in}) at around 6–8 Mbps. For the BL bit rate R_{\min_in} , we used different values in the range of few hundreds kbps (e.g., between 100 and 500 kbps). Video parameters, which are suitable for the BL bit rates, were selected. All sequences were coded using CIF resolution and 10–15 frames/s.¹³

First, we present the results of TS an FGS stream (“Mobile”) that has been coded originally with $R_{\min_in} = 250$ kbps and $R_{\max_in} = 8$ Mbps. The transcaler used a new BL bit rate $R_{\min_out} = 1$ Mbps. This example could represent a stream that was coded originally for transmission over lower bit

¹²The exceptions to this statement are high-definition video sequences, which could benefit from bit rates around 20 Mbps.

¹³Our full transcaler used the exact same video parameters of the original video sequence (except bit rates) in order to avoid biasing the results.

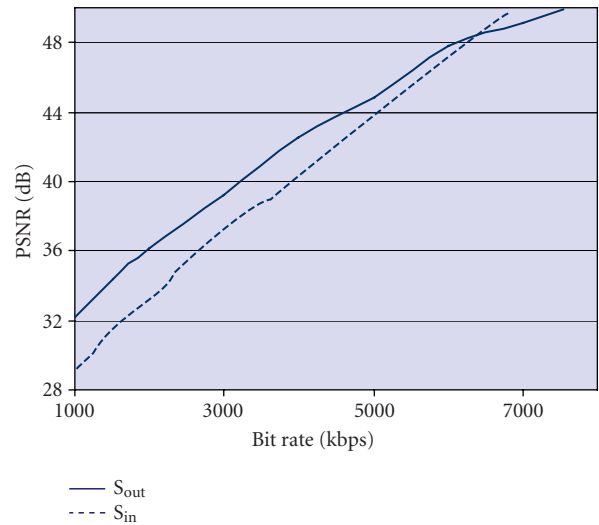


FIGURE 8: Performance of transcaling the Mobile sequence using an input stream S_{in} with a BL bit rate $R_{\min_in} = 250$ kbps into a stream with a BL $R_{\min_out} = 1$ Mbps.

rate systems (e.g., cable modem or legacy wireless LANs) and is being transcaled for transmission over new higher bit rate LANs. The peak SNR (PSNR) performance of the two streams as the functions of the bit rate is shown in Figure 8. (For more information about the MPEG-4 FGS encoding and decoding methods, the reader is referred to [13, 14].)

It is clear from the figure that there is a significant improvement in quality (close to 4 dB) in particular at bit rates close to the new BL rate of 1 Mbps. The figure also highlights that the improvements gained through TS are limited by the maximum performance of the input stream S_{in} . As the bit rate gets closer to the maximum input bit rate (8 Mbps), the performance of the transcaled stream saturates and gets closer (and eventually degrades below) the performance of the original FGS stream S_{in} . Nevertheless, for the majority of the desired bit rate range (i.e., above 1 Mbps), the performance of the transcaled stream is significantly higher. In order to appreciate the improvements gained through TS, we can compare the performance of the transcaled stream with that of an “ideal FGS” stream. Here, an ideal FGS stream is the one that has been generated from the original uncompressed sequence (i.e., not from a precompressed stream such as S_{in}). In this example, an ideal FGS stream is generated from the original sequence with a BL of 1 Mbps. Figure 9 shows the comparison between the transcaled stream and an ideal FGS stream over the range 1 to 4 Mbps. As shown in the figure, the performances of the transcaled and ideal streams are virtually identical over this range.

By increasing the range of bit rates that need to be covered by the transcaled stream, one would expect that its improvement in quality over the original FGS stream should get lower. Using the same original FGS (Mobile) stream coded with a BL bit rate of $R_{\min_in} = 250$ kbps, we transcaled this stream with a new BL bit rate $R_{\min_out} = 500$ kbps (i.e., lower than the 1 Mbps BL bit rate of the TS example described

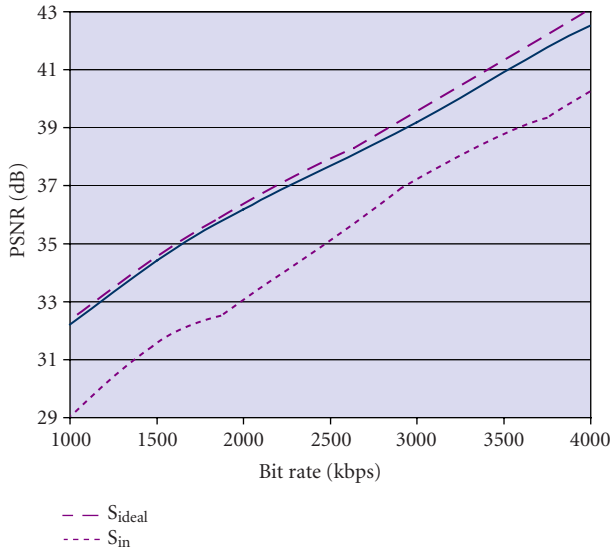


FIGURE 9: Comparing the performance of the Mobile transcoded stream (shown in Figure 8) with an ideal FGS stream. The performance of the transcoded stream is represented by the solid line.

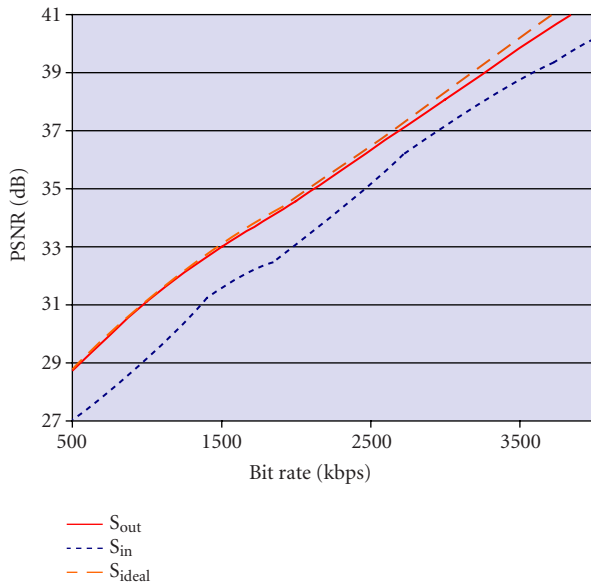


FIGURE 10: Performance of transcaling the Mobile sequence using an input stream S_{in} with a BL bit rate $R_{min,in} = 250$ kbps into a stream with a BL $R_{min,out} = 500$ kbps.

above). Figure 10 shows the PSNR performance of the input, transcoded, and ideal streams. Here, the PSNR improvement is as high as 2 dB around the new BL bit rate 500 kbps. These improvements are still significant (higher than 1 dB) for the majority of the bandwidth range. Similar to the previous example, we can see that the transcoded stream does saturates toward the performance of the input stream S_{in} at higher bit rates, and, overall, the performance of the transcoded stream is very close to the performance of the ideal FGS stream.

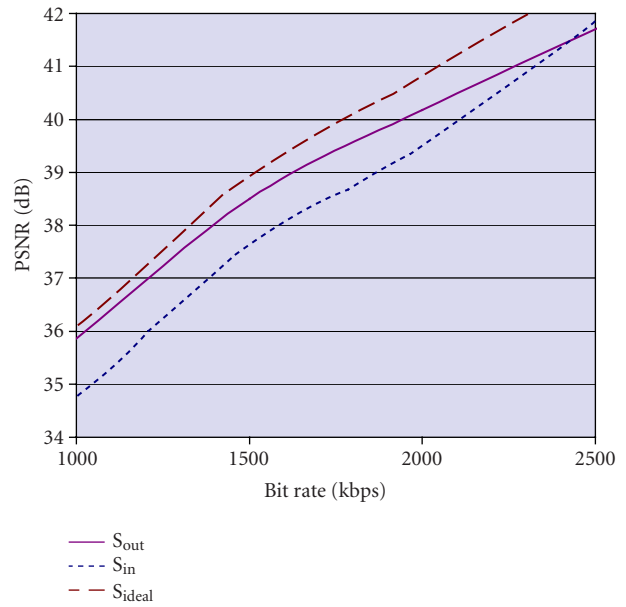


FIGURE 11: Performance of transcaling the Coastguard sequence using an input stream S_{in} with a BL bit rate $R_{min,in} = 250$ kbps into a stream with a BL $R_{min,out} = 1000$ kbps.

Therefore, TS provides rather significant improvements in video quality (around 1 dB and higher). The level of improvement is a function of the particular video sequences and the bit rate ranges of the input and output streams of the transcaler. For example, and as mentioned above, FGS provides different levels of performance depending on the type of video sequence [13]. Figure 11 illustrates the performance of TS the “Coastguard” MPEG-4 test sequence. The original MPEG-4 stream S_{in} has a BL bit rate $R_{min} = 250$ kbps and a maximum bit rate of 4 Mbps. Overall, FGS (without TS) provides a better quality scalable video for this sequence when compared with the performance of the previous sequence (Mobile). Moreover, the maximum bit rate used here for the original FGS stream ($R_{max,in} = 4$ Mbps) is lower than the maximum bit rate used for the above Mobile sequence experiments. Both of these factors (i.e., a different sequence with a better FGS performance and a lower maximum bit rate for the original FGS stream S_{in}) led to the following: the level of improvements achieved in this case through TS is lower than the improvements we observed for the Mobile sequence. Nevertheless, significant gain in quality (more than 1 dB at 1 Mbps) can be noticed over a wide range over the transcoded bitstream. Moreover, we observe here the same “saturation-in-quality” behavior that characterized the previous Mobile sequence experiments. As the bit rate gets closer to the maximum rate $R_{max,in}$, the performance of the transcoded video approaches the performance of the original stream S_{in} .

The above results for TS were observed for a wide range of sequences and bit rates. So far, we have focused our attention on the performance of UTS, which we have referred to throughout this section simply by using the word TS. Now, we shift our focus to some simulation results for DTS.

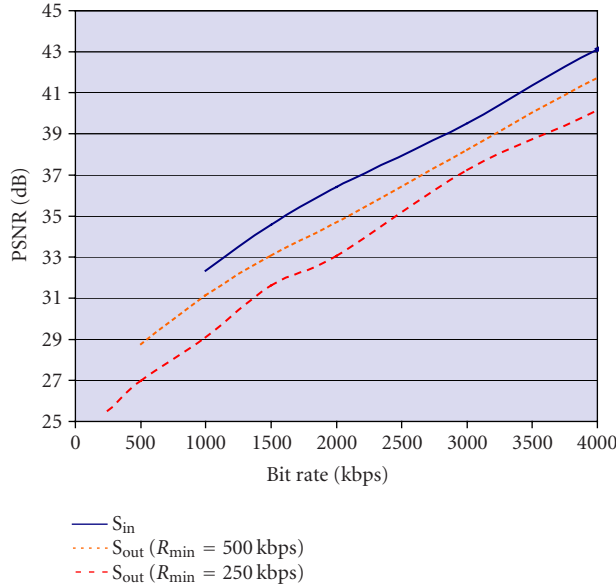


FIGURE 12: Performance of down transcaling the “Mobile” sequence using an input stream S_{in} with a BL bit rate $R_{min_in} = 1$ Mbps into two streams with BL $R_{min_out} = 500$ and 250 kbps.

As explained above, DTS can be used to convert a scalable stream with a BL bit rate R_{min_in} into another stream with a smaller BL bit rate $R_{min_out} < R_{min_in}$. This scenario could be needed, for example, if (a) the transcaler gateway misestimates the range of bandwidth that it requires for its clients, (b) a new client appears over the wireless LAN, where this client has access bandwidth lower than the minimum bit rate (R_{min_in}) of the bitstream available to the transcaler; and/or (c) sudden local congestion over a wireless LAN is observed, and consequently, reducing the minimum bit rate needed. In this case, the transcaler has to generate a new scalable bitstream with a lower BL $R_{min_out} < R_{min_in}$. Below, we show some simulation results for DTS.

We employed the same full transcaler architecture shown in Figure 7. We also used the same Mobile sequence coded with MPEG-4 FGS and with a bit rate range $R_{min_in} = 1$ Mbps to $R_{max_in} = 8$ Mbps. Figure 12 illustrates the performance of the DTS operation for two bitstreams: one stream was generated by DTS the original FGS stream (with a BL of 1 Mbps) into a new scalable stream coded with a BL of $R_{min_out} = 500$ kbps. The second stream was generated using a new base layer $R_{min_out} = 250$ kbps. As expected, the DTS operation degrades the overall performance of the scalable stream.

It is important to note that, depending on the application (e.g., unicast versus multicast), the gateway server may utilize both the new generated (down-transcaled) stream and the original scalable stream for its different clients. In particular, since the quality of the original scalable stream S_{in} is higher than the quality of the down-transcaled stream S_{out} over the range $[R_{min_in}, R_{max_in}]$, then it should be clear that clients with access bandwidth that falls within this range can benefit from the higher quality (original) scalable stream S_{in} . On

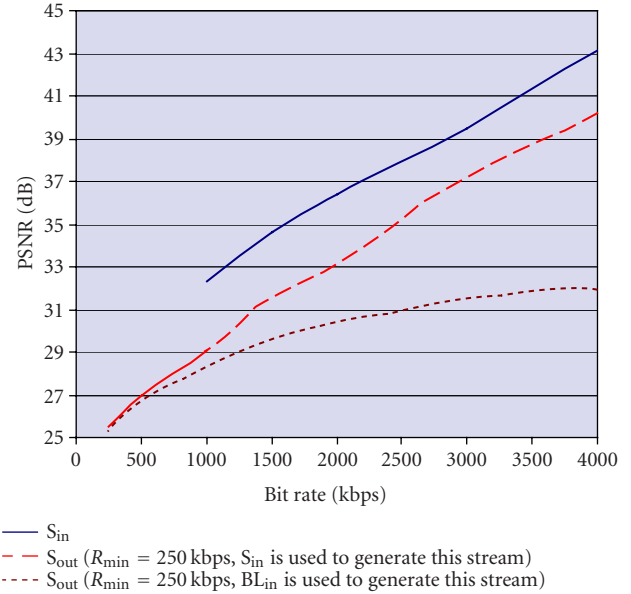


FIGURE 13: Performance of down transcaling the Mobile sequence using an input stream S_{in} with a BL bit rate $R_{min_in} = 1$ Mbps. Here, two DTS operations are compared, respectively, the whole input stream S_{in} (base + enhancement) is used, and only the BL_{in} of S_{in} is used to generate the down-transcaled stream. In both cases, the new DTS stream has a BL bit rate $R_{min_out} = 250$ kbps.

the other hand, clients with access bandwidth less than the original BL bit rate R_{min_in} , can only use the down-transcaled bitstream.

As mentioned in Section 2, DTS is similar to traditional transcoding, which converts a non-scalable bitstream into another non-scalable stream with a lower bit rate. However, DTS provides new options for performing the desired conversion that are not available with non-scalable transcoding. For example, under DTS, one may elect to use (a) both the BL and EL or (b) the BL only to perform the desired down conversion. This, for example, may be used to reduce the amount of processing power needed for the DTS operation. In this case, the transcaler has the option of performing only one decoding process (on the BL only versus decoding both the BL and EL). However, using the BL only to generate a new scalable stream limits the range of bandwidth that can be covered by the new scalable stream with an acceptable quality. To clarify this point, Figure 13 shows the performance of TS using (a) the entire input stream S_{in} (i.e., base plus enhancement) and (b) BL_{in} (only) of the input stream S_{in} . It is clear from the figure that the performance of the transcaled stream generated from BL_{in} saturates rather quickly and does not keep up with the performance of the other two streams. However, the performance of the second stream (b) is virtually identical over most of the range $[R_{min_out} = 250 \text{ kbps}, R_{min_in} = 500 \text{ kbps}]$. Consequently, if the transcaler is capable of using both the original stream S_{in} and the new transcaled stream S_{out} for transmission to its clients, then employing BL_{in} (only) to generate the new down-transcaled stream is a viable option.

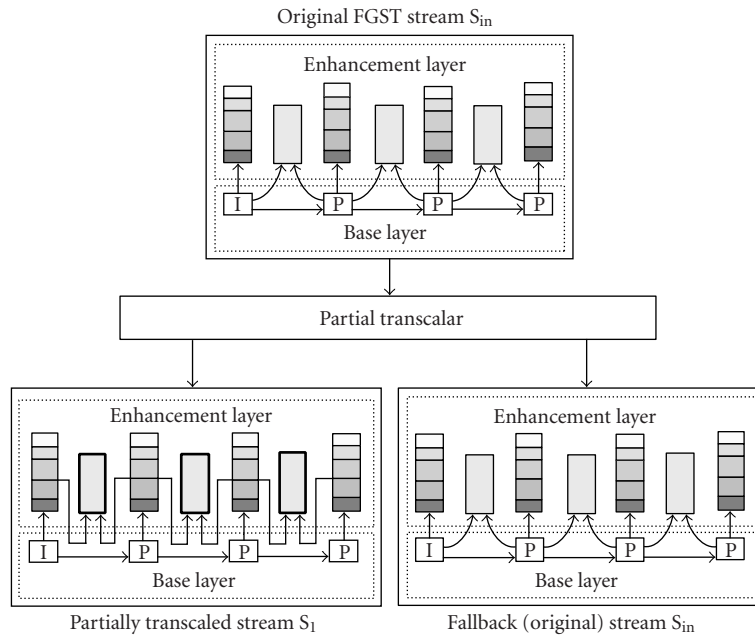


FIGURE 14: The proposed partial TS of the MPEG-4 FGST scalability structure. The FGST frames are the only part of the original scalable stream that is fully reencoded under the proposed partial TS scheme.

It is important to note that, in cases when the transcaler needs to employ a single scalable stream to transmit its content to its clients (e.g., multicast with a limited total bandwidth constraint), a transcaler can use the BL and any portion of the EL to generate the new down-transcaled scalable bitstream. The larger the portion of the EL used for DTS, the higher the quality of the resulting scalable video. Therefore, and since partial decoding of the EL represents some form of computational scalability, an FGS transcaler has the option of trading-off quality versus computational complexity when needed. It is important to note that this observation is applicable to both UTS and DTS.

Finally, by examining Figure 13, one can infer the performance of a wide range of down-transcaled scalable streams. The lower-bound quality of these downscaled streams is represented by the quality of the bitstream generated from BL_{in} only (i.e., case (b) of S_{out}). Meanwhile, the upper-bound of the quality is represented by the downscaled stream (case (a) of S_{out}) generated by the full input stream S_{in} .

5. PARTIAL TRANSCALING FOR HIGH-BIT-RATE WIRELESS LANS

As described above, the MPEG-4 FGST framework supports SNR (regular FGS), temporal (FGST frames), and hybrid SNR-temporal scalabilities. At low bit rates (i.e., bit rates close to the BL bit rate), receivers can benefit from the standard SNR FGS scalability by streaming the BL and any desired portion of the SNR FGS enhancement-layer frames. As the available bandwidth increases, high-end receivers can benefit from both FGS and FGST pictures. It is important for these high-end receivers to experience higher-quality video

when compared to the video quality of nontranscaled FGST streams. One of the reasons for the relatively high penalty in quality associated with the traditional FGST-based coding is that, at high bit rates, the FGST frames are predicted from low-quality (low bit rate) BL frames. Consequently, the resulting motion-compensated residual error is high, and thus a large number of bits are necessary for its compression.

In addition to improving the coding efficiency, it is crucial to develop a low complexity TS operation that provides the desirable improvements in quality. One approach for maintaining low complexity TS is to eliminate the need for reencoding the BL. Consequently, this eliminates the need for recomputing new motion vectors, which is the most costly part of a full transcaler that elects to perform this recomputation. Meanwhile, improvements can be achieved by using higher-quality (higher bit rate) SNR FGS pictures to predict the FGST frames. This reduces the entropy of the bidirectionally predicted FGST frames and, consequently leads to more coding efficiency or higher PSNR values. Examples of the input and output scalability structures of the proposed PTS scheme for FGST are depicted in Figure 14.

As shown in Figure 14, and similar to the full TS case, there are two options for supporting TS of FGST streams: the PTS option and the fallback (no TS) option. Depending on the processing power available to the gateway, the system can select one of these options. Every FGS SNR frame is shown with multiple layers, each of which can represent one of the bit planes of that frame. It is important to note that at higher bit rates, larger number of FGS SNR bit planes will be streamed, and consequently, these bit planes can be used to predict the FGST frames. Therefore, under an RLM framework, receivers that subscribe to the transcaled FGST stream

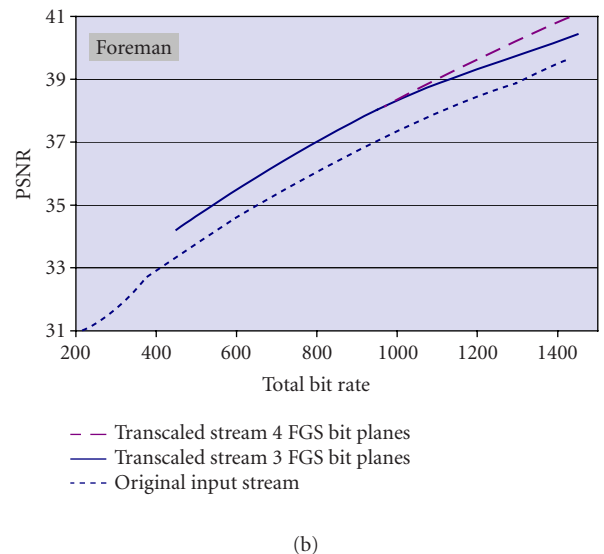
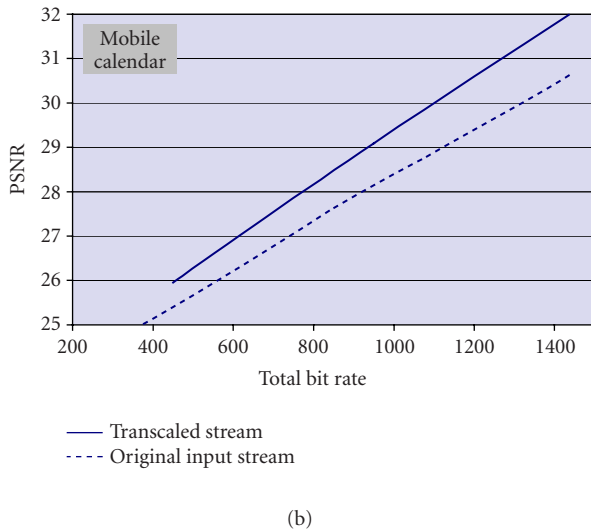
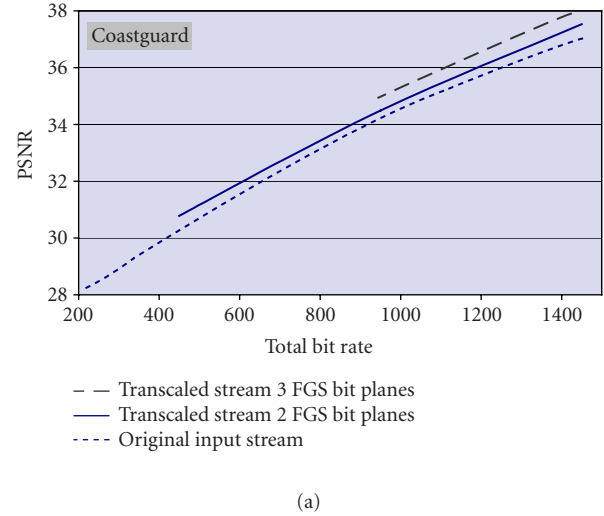
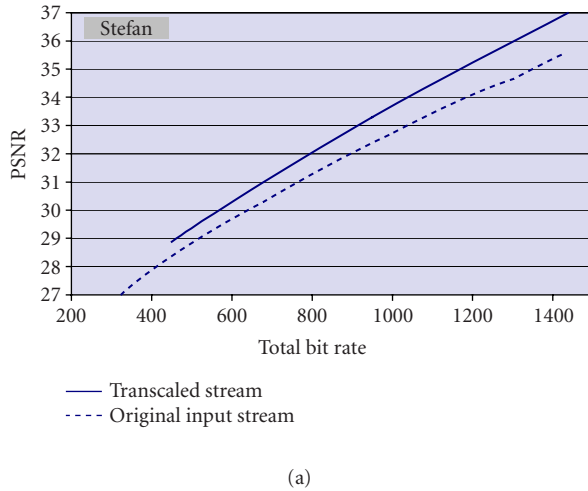


FIGURE 15: Performance of PTS of two sequences: Stefan and Mobile.

FIGURE 16: Performance of PTS of two sequences: Coastguard and Foreman.

should also subscribe to the appropriate number of FGS SNR bit planes.

Under the above-proposed PTS, the input FGST stream S_{in} is transcaled into another scalable stream S_1 . In this case, BL_{in} of S_{in} (with bit rate R_{min_in}) and a certain portion of the EL_{in} are used as reference frames for an improved FGST performance. Therefore, this is an example of TS an FGST stream with a bit rate range $R_{range_in} = [R_{min_in}, R_{max_in}]$ to another FGST stream with a bit rate range $R_{range_1} = [R_{min_1}, R_{max_1}]$, where $R_{min_in} < R_{min_1}$. Consequently, and based on the definition we adopted earlier for UTS and DTS, this example represents a UTS scenario. Furthermore, in this case, only the FGST ELs of the input stream S_{in} has been modified. Consequently, this represents a PTS scenario. PTS can be implemented by using cascaded decoder-encoder systems for only part of the original scalable stream. It is important to note that, although we have a UTS scenario here, low-bandwidth receivers can still use the BL of the new transcaled stream, which is identical to the original BL. These re-

ceivers can also stream any desired portions of the FGS SNR frames. However, and as mentioned above, receivers that take advantage of the improved FGST frames have a new (higher) minimum bit rate stream ($R_{min_1} > R_{min_in}$) that is needed to decode the new FGST frames.

5.1. Simulation results for partial transcaling of FGST streams

In order to illustrate the level of video quality improvements that PTS can provide for wireless Internet applications, in this section, we present some simulation results of the FGST based PTS method described above. As in FTS experiments, we coded several video sequences using the MPEG-4 FGST scheme. These sequences were then modified using the partial transcaler scalability structure that employs a portion of the EL for FGST prediction as shown in Figure 14. We should emphasize here the following. (a) Unlike the FTS

results shown above, all the results presented in this section are based on reusing the same motion vectors that were originally computed by the BL encoder at the source. This is important for maintaining a low-complexity operation that can be realized in real time. (b) The FGS/FGST sequences we coded were compressed at maximum bit rates (i.e., R_{\max_in}) lower than 2 Mbps. For the BL bit rate R_{\min_in} , we used 50–100 kbps. Other video parameters, which are suitable for the BL bit rates, were selected. All sequences were coded using CIF resolution; however, and since the bit rate ranges are smaller than the FTS experiments, 10 frames/s were used in this case. The GOP size is 2-second long and $M = 2$ (i.e., one FGST bidirectionally predicted frame can be inserted between two I and P reference frames).

The PSNR performance of four well-known MPEG-4 streams: Foreman, Coastguard, Mobile, and Stefan have been simulated and measured for both original FGST (nontranscaled) and partially transcaled bitstreams over a wide range of bit rates.

Figure 15 shows the performance of the Stefan and Mobile (calendar) and compares the PSNR of the input nontranscaled stream with the partially transcaled streams' PSNR results. Both of these video sequences benefited from the PTS operation described above and gained as much as 1.5 dB in PSNR, in particular, at high bit rates. Three FGS bit planes were used (in addition to the BL) for predicting the FGST frames. Consequently, taking advantage of PTS requires that the receiver have enough bandwidth to receive the BL plus a minimum of three FGS bit planes. This explains why the gain in performance shown in Figure 15 begins at higher rates than the rate of the original BL bit rates (which are in the 50–100 kbps range as mentioned above).

As mentioned above, the level of gain obtained from the proposed PTS operation depends on the type of video sequence. Moreover, the number of FGS bit planes used for predicting the FGST frames influence the level of improvements in PSNR. Figure 16 shows the performance of the Coastguard and Foreman sequences. These sequences are usually coded more efficiently with FGS than the other two sequences shown above (Stefan and Mobile). Consequently, the improvements obtained by employing PTS on the Coastguard and Foreman sequences are less than the improvements observed in the above plots. Nevertheless, we are still able to gain about 1 dB in PSNR values at higher bit rates. Figure 16 also shows the impact of using different number of FGS bit planes from predicting the FGST frames. It is clear from both figures that, in general, larger number of bit planes provides higher gain in performance. However, it is important to note that this increase in PSNR gain (as the number of FGS bit planes used for prediction increases) could saturate as shown in the Foreman performance plots.

Furthermore, we should emphasize here that many of the video parameters used at the partial transcaler do not represent the best choice in a rate-distortion sense. For example, all of the results shown in this section are based on allocating the same number of bits to both the FGS and transcaled FGST frames. It is clear that a better rate allocation mechanism can be used. However, and as mentioned above, the

main objective of this study is to illustrate the benefits and limitations of TS, in general, PTS, in particular, without the bias of different video parameters and related algorithms.

6. SUMMARY AND FUTURE WORK

In this paper, we introduced the notion of TS, which is a generalization of (non-scalable) transcoding. With TS, a scalable video stream, that covers a given bandwidth range, is mapped into one or more scalable video streams covering different bandwidth ranges. Our proposed TS framework exploits the fact that the level of heterogeneity changes at different points of the video distribution tree over wireless and mobile Internet networks. This provides the opportunity to improve the video quality by performing the appropriate TS process.

We argued that an Internet/wireless network gateway represents a good candidate for performing TS. Moreover, we described HTS, which provides a transcaler with the option of choosing among different levels of TS processes with different complexities. This enables transcalers to trade off video quality with computational complexity. We illustrated the benefits of FTS and PTS by considering the recently developed MPEG-4 FGS video coding.

Under FTS, we examined two forms: UTS (which we simply refer to as TS) and DTS. With UTS, significant improvements in video quality can be achieved as we illustrated in the simulation results section. Moreover, several scenarios for performing DTS were evaluated. Under PTS, we illustrated that a transcaler can still provide improved video quality (around 1 dB in improvements) while significantly reducing the high complexity associated with FTS. Consequently, we believe that the overall TS framework provides a viable option for the delivery of high-quality video over new and emerging high bit rate wireless LANs such 802.11a and 802.11b.

This paper has focused on the applied, practical, and proof-of-concept aspects of TS. Meanwhile, the proposed TS framework opens the door for many interesting research problems, some of which we are currently investigating. These problems include the following.

- (1) A thorough analysis for an optimum rate-distortion (RD) approach for the TS of a wide range of video sequences is under way. This RD-based analysis, which is based on recent RD models for compressed scalable video [19], will provide robust estimation for the level of quality improvements that TS can provide for a given video sequence. Consequently, an RD-based analysis will provide an in-depth (or at least an educated) answer for: "when TS should be performed and on what type of sequences?"
- (2) We are exploring new approaches for combining TS with other scalable video coding schemes such as 3D motion-compensated wavelets. Furthermore, TS in the context of cross-layer design of wireless networks is being evaluated [20, 21].
- (3) Optimum networked TS that trades off complexity and quality in a distributed manner over a network of proxy video servers. Some aspects of this analysis

include distortion-complexity models for the different (full and partial) TS operations introduced in this paper. Moreover, other aspects of a networked TS framework will be investigated in the context of new and emerging paradigms such as overlay networks and video communications using path diversity (see, e.g., [22, 23, 24, 25, 26]).

ACKNOWLEDGMENTS

The authors would like to thank three anonymous reviewers who provided very constructive and valuable feedback on an earlier version of this paper. Many thanks to Professor Zixiang Xiong for his help and guidance throughout the review process. Parts of this work were presented at the ACM SIGMOBILE 2001 Workshop on Wireless Mobile Multimedia, Rome, Italy (in conjunction with MOBICOM 2001), the IEEE CAS 2001 Workshop on Wireless Communications and Networking, University of Notre Dame, and the Packet Video Workshop 2002.

REFERENCES

- [1] M. Allman and V. Paxson, "On estimating end-to-end network path properties," in *ACM SIGCOMM '99*, pp. 263–274, Cambridge, Mass, USA, September 1999.
- [2] V. Paxson, "End-to-end Internet packet dynamics," in *ACM SIGCOMM '97*, pp. 139–152, Cannes, France, September 1997.
- [3] D. Loguinov and H. Radha, "Measurement study of low-bitrate Internet video streaming," in *ACM SIGCOMM Internet Measurement Workshop*, pp. 281–293, San Francisco, Calif, USA, November 2001.
- [4] B. Walke, N. Esseling, J. Habetha, et al., "IP over wireless mobile ATM – guaranteed wireless QoS by HiperLAN/2," *Proceedings of the IEEE*, vol. 89, no. 1, pp. 21–40, 2001.
- [5] IEEE 802.11, "High Speed Physical Layer in the 5 GHz Band," 1999.
- [6] R. Prasad, W. Mohr, and W. Konhäuser, Eds., *Third Generation Mobile Communication Systems*, Artech House, Boston, Mass, USA, 2000.
- [7] M.-T. Sun and A. Reibmen, Eds., *Compressed Video over Networks*, Marcel Dekker, NY, USA, 2000.
- [8] B. Girod and N. Farber, "Wireless video," in *Compressed Video over Networks*, Marcel Dekker, NY, USA, 2000.
- [9] H. Radha, C. Ngu, T. Sato, and M. Balakrishnan, "Multimedia over wireless," in *Advances in Multimedia: Systems, Standards, and Networks*, Marcel Dekker, NY, USA, March 2000.
- [10] M. R. Civanlar, "Internet video," in *Advances in Multimedia: Systems, Standards, and Networks*, Marcel Dekker, NY, USA, March 2000.
- [11] W. Tan and A. Zakhori, "Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Trans. Multimedia*, vol. 1, no. 2, pp. 172–186, 1999.
- [12] H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen, "Scalable Internet video using MPEG-4," *Signal Processing: Image Communication*, vol. 15, no. 1–2, pp. 95–126, 1999.
- [13] H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 FGS video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 53–68, 2001.
- [14] ISO/IEC 14496-2, "Information Technology – Coding of Audio-Visual Objects – part 2: Visual," International Standard, ISO/IEC JTC 1/SC 29/WG 11, March 2000.
- [15] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Scalable video coding and transport over broadband wireless networks," *Proceedings of the IEEE*, vol. 89, no. 1, pp. 6–20, 2001.
- [16] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 6, pp. 983–1001, 1997.
- [17] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. Special Interest Group on Data Communications*, pp. 117–130, Stanford, Calif, USA, August 1996.
- [18] K. L. Calvert, A. T. Campbell, A. A. Lazar, D. Wetherall, and R. Yavatkar eds., "Special issue on active and programmable networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 3, 2001.
- [19] M. Dai, D. Loguinov, and H. Radha, "Statistical analysis and distortion modeling of MPEG-4 FGS," in *IEEE International Conference on Image Processing*, Barcelona, Spain, September 2003.
- [20] S. A. Khayam, S. S. Karande, M. Krappel, and H. Radha, "Cross-layer protocol design for real-time multimedia applications over 802.11b networks," in *IEEE International Conference on Multimedia and Expo*, Baltimore, Md, USA, July 2003.
- [21] S. A. Khayam, S. S. Karande, H. Radha, and D. Loguinov, "Performance analysis and modeling of errors and losses over 802.11b LANs for high-bitrate real-time multimedia," *Signal Processing: Image Communication*, vol. 18, no. 7, pp. 575–595, 2003.
- [22] "Session on overlay networks," in *Proc. Special Interest Group on Data Communications*, Pittsburgh, Pa, USA, August 2002.
- [23] D. Towsley, C. Diot, B. N. Levine, and L. Rizzo eds., "Special issue on network support for multicast communications," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, 2002.
- [24] "Session on overlay routing and multicast," in *INFOCOM 2003*, San Francisco, Calif, USA, April 2003.
- [25] J. Taal, I. Haratcherev, K. Langendoen, and R. Legendijk, "Special session on networked video (SS-LI)," in *IEEE International Conference on Multimedia and Expo*, Baltimore, Md, USA, July 2003.
- [26] "Lecture session on multimedia streaming: MCN-L5," in *IEEE International Conference on Multimedia and Expo*, Baltimore, Md, USA, July 2003.

Hayder Radha received his B.S. degree (with honors) from Michigan State University (MSU) in 1984, his M.S. degree from Purdue University in 1986, and his Ph.M. and Ph.D. degrees from Columbia University in 1991 and 1993, all in electrical engineering. He joined MSU in 2000 as an Associate Professor in the Department of Electrical and Computer Engineering. Between 1996 and 2000, Dr. Radha was with Philips Research, USA, first as a Principal Member of the research staff and then as a Consulting Scientist. In 1997, Dr. Radha initiated the Internet video project and led a team of researchers working on scalable video coding, networking, and streaming algorithms. Prior to working at Philips Research, Dr. Radha was a member of technical staff at Bell Labs, where he worked between 1986 and 1996 in the areas of digital communications, signal/image processing, and broadband multimedia. Dr. Radha is the recipient of the Bell Labs Distinguished Member of Technical Staff Award and the Withrow Junior Distinguished Scholar Award. He was appointed as a Philips



Research Fellow in 2000. His research interests include image and video coding, wireless communications, and multimedia networking. He has 25 patents (granted or pending) in these areas.

Mihaela van der Schaar is currently an Assistant Professor in the Electrical and Computer Engineering Department at the University of California, Davis. She received her Ph.D. degree in electrical engineering from Eindhoven University of Technology, the Netherlands. Between 1996 and June 2003, she was a Senior Member Research Staff at Philips Research in the Netherlands and USA. In 1998, she worked in the Wireless Communications and Networking Department. From January to September 2003, she was also an Adjunct Assistant Professor at Columbia University. In 1999, she became an active participant to the MPEG-4 standard, contributing to the scalable video coding activities. She is currently chairing the MPEG Ad-hoc group on Scalable Video Coding, and is also cochairing the Ad-hoc group on Multimedia Test Bed. Her research interests include multimedia coding, processing, networking, and architectures. She has authored more than 70 book chapters, and conference and journal papers and holds 9 patents and several more pending. She was also elected as a member of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society and is an Associate Editor of IEEE Transactions on Multimedia and an Associate Editor of Optical Engineering.



Shirish Karande is currently pursuing his Ph.D. at Michigan State University (MSU). He received his B.E. degree in electronics and telecommunications from University of Pune in 2000, and his M.S. degree in electrical engineering from MSU in 2003. He was the recipient of the Government of India National Talent Search (NTS) Merit Scholarship from 1994 to 2000. His research interests include scalable source coding, channel coding, and wireless networking.

