SCALABLE VIDEO TRANSCALING FOR THE WIRELESS INTERNET[1]

Hayder Radha
Department of Electrical and Computer Engineering
Michigan State University; radha@msu.edu

Mihaela van der Schaar
Department of Electrical and Computer Engineering
University of California at Davis; mvanderschaaar@ece.ucdavis.edu

Shirish Karande
Department of Electrical and Computer Engineering
Michigan State University; karandes@msu.edu

Abstract

The rapid and unprecedented increase in the heterogeneity of multimedia networks and devices emphasizes the need for scalable and adaptive video solutions both for coding and transmission purposes. However, in general, there is an inherent tradeoff 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 transcaling, 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 transcaling process. We argue that an Internet/wireless network gateway represents a good candidate for performing transcaling. Moreover, we describe Hierarchical TranScaling (HTS), which provides a "Transcaler" the option of choosing among different levels of transcaling processes with different complexities. We illustrate the benefits of transcaling by considering the recently developed MPEG-4 Fine-Granularity-Scalability (FGS) video coding. Extensive simulation results of video transcaling over bitrate-ranges supported by emerging wireless LANs are presented.

[1] 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.

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 [14][15][16] and wireless networks are evolving to higher bitrate 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) bitrates from 6 Mbit/sec to 54 Mbit/sec (see for example [1][2]). Within each of the supported bitrates, 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 bitrates than mobile networks (including 3rd generation) [3]. In the meantime, it is expected that current wireless and mobile access networks (e.g., 2G and 2.5G mobile systems and sub-2 Mbit/sec 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., [4]-[12], [18]). 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 range[1] that needs to be covered by a scalable video stream, the lower the overall video quality is[2] [10]. With the aforementioned increase in heterogeneity over emerging wireless multimedia 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 kbit/sec, another that covers 500-1000 Kbit/sec and yet another stream to cover 1000-2000 Kbit/sec 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

[1] A more formal definition of "bandwidth range" will be introduced later in the document.
[2] This is particularly true for the scalable schemes that fall under the category of SNR (Signal-to-Noise Ratio) scalability methods. These include the MPEG-2 and MPEG-4 SNR scalability methods, and the newly developed MPEG-4 FGS method.

inefficient in terms of bandwidth utilization over the wired segment of the wireless IP network. (In the above example, a total bitrate of 3500 kbit/sec is needed over a link transmitting the three streams while only 2000 kbit/sec 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 TranScaling (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 bitrate *lower* than the first stream bitrate. With TransScaling, 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 *transcaled* 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. TransScaling 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-bitrate 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 transcaling 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-bitrate mobile networks. Similarly, powerful clients (e.g., laptops and PCs) can still receive high quality video even if there are other low-bitrate 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, transcaling provides an efficient framework for video multicast over the wireless Internet.

In addition to introducing the notion of transcaling 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 transcaling can provide by presenting several video simulation results for a variety of transcaling cases. The remainder of the paper is organized as follows: Section 2 describes the wireless-video transcaling framework with some focus on IP multicast applications. This section also highlights some of the key attributes and basic definitions of transcaling-based wireless systems and how they differ from traditional transcoding-based platforms. Section 3 describes Hierarchical TranScaling (HTS), which is a framework that enables transcalers to tradeoff video quality with complexity. HTS is described using a concrete example that is based on the MPEG-4 Fine-Granularity-Scalability (FGS) video coding method. Then, two classes of transcaling are considered: full and partial. Section 4 described full transcaling for wireless LANs. Section 4 also shows simulation results of applying full transcaling 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 transcaling and presenting results for performing partial transcaling 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 transcaling 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 transcaling based solution. Then, we describe some general features of a transcaling-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 the base-layer (BL) and the other streams are the Enhancement Layers (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 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.

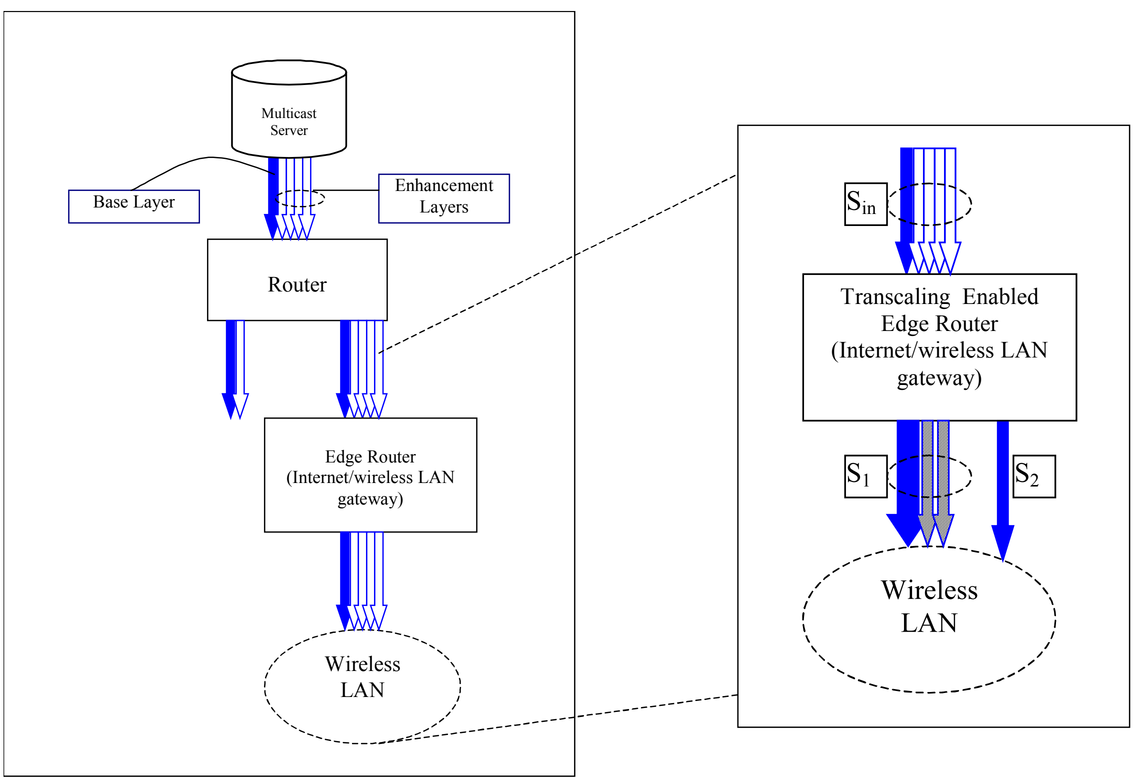


Figure 1: A simplified view of a wireless-video TranScaling platform within an RLM architecture.

Similar to RLM, TranScaling 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 router[1] with a transcaling capability (or a "transcaler") derives new scalable streams from the original stream. A derived scalable stream could have a base-layer and/or enhancement-layer(s) that are different from the BL and/or ELs of the original scalable stream. The objective of the transcaling 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 transcaling 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 stream[2] with a BL bitrate R_{min_in}. The bitrate range covered by this layered set of streams is $R_{range\_in}=[R_{min\_in} , R_{max\_in}]$. 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 base-layer with a bitrate R_{min_1} which is higher than the original BL bitrate: $R_{min\_1} > R_{min\_in}$. Consequently, in this example, the transcaler requires at least one additional piece of information and that is the minimum bitrate 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[3]. 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-bitrate devices significantly.

2.1 Attributes of Wireless-Video TranScaling Based Systems

Here, we highlight the following attributes of the proposed wireless-video transcaling framework:

1. Supporting transcaling 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 bitrate that is higher than the bitrate of the input BL stream), the transcaler delivers the original BL stream to the low-bitrate devices.

[1] The "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.
[2] Here, a "layered" or "scalable" stream consists of multiple sub-streams.
[3] Determining the particular bitrate 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.

2. The TSM system (described above) falls under the umbrella of active networks[1] where, in this case, the transcaler provides network-based added value services [13]. 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.

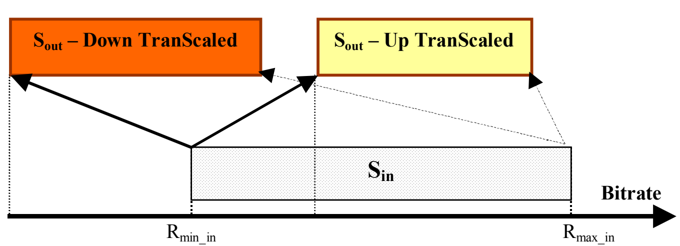
3. A wireless-video transcaler can always fallback to using the original (lower-quality) scalable video. This "fallback" feature represents a key attribute of transcaling that distinguishes it from non-scalable transcoding. The "fallback" feature could be needed, for example, when the Internet-wireless gateway (or whoever the transcaler happens to be) do not have enough processing power for performing the desired transcaling process(es). Therefore, and unlike (non-scalable) transcoding-based services, transcaling provides a scalable framework for delivering higher quality video. A more graceful transcaling 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 transcaling approach in the context of multicast services, on-demand unicast applications can also take advantage of transcaling. For example, a wireless or mobile gateway may perform transcaling 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 transcaling. 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, transcaling has its own limitations in improving the video quality over the whole desired bandwidth range. Nevertheless, the improvements that transcaling provides is significant enough to justify its merit over a subset of the desired bandwidth range. This aspect of transcaling will be explained further later in the paper.

6. Transcaling can be applied to any form of scalable streams (i.e., SNR, temporal, and/or spatial). In this paper, we will show examples of transcaling operations that are applied to SNR-scalable and hybrid SNR-temporal streams over bitrates 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 transcaling. Here, we define two types of transcaling processes: Down TranScaling (DTS) and Up TranScaling (UTS).



[1] We should emphasize here that the area of active networks covers many aspects, and "added value services" is just one of these aspects.

Figure 2: The distinction between DTS and UTS.

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

$$R_{range\_in}=[R_{min\_in} , R_{max\_in}].$$

And let a transcaled stream has a range:

$$R_{range\_out}=[R_{min\_out} , R_{max\_out}]$$

Then, down transcaling – DTS – occurs when: $R_{min\_out} < R_{min\_in}$ while up transcaling – UTS – occurs when: $R_{min\_in} < R_{min\_out} < R_{max\_in}$. The distinction between down and up transcaling is illustrated in Figure 2. Down transcaling resembles traditional non-scalable transcoding in the sense that the bitrate of the output base-layer is lower than the bitrate of the input base-layer. Many researchers have studied this type of down conversion in the past[1]. However, up conversion has not received much attention (if any). Therefore, for the remainder of this paper we will focus on up transcaling. (Unless otherwise mentioned, we will use "up transcaling" and "transcaling" interchangeably.)

Another important classification of transcaling is the distinction between *full* and *partial* transcaling (see Figure 3). Our definition of Full TranScaling (FTS) implies two things: (a) All of the input stream data (base-layer stream and enhancement layer stream) is used to perform the transcaling operation; and (b) *All* pictures of both the base and enhancement layers have been modified by transcaling. Partial TranScaling (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.

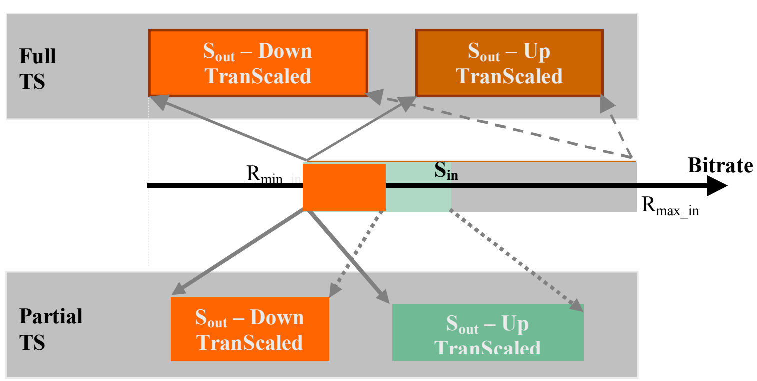


Figure 3: An example illustrating the different transcaling categories.

[1] We 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 transcaling, its general features, potential benefits, and basic definitions, we now describe Hierarchical TranScaling (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 this next section 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 Method[1]

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 [10]. As shown in Figure 4, the FGS structure consists of only two layers: a base-layer coded at a bitrate R_b and a single enhancement-layer coded using a fine-grained (or totally embedded) scheme to a maximum bitrate 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 bitrate the content will be streamed at. The streaming server on the other hand has a total flexibility in sending any desired portion of any enhancement layer frame (in parallel with the corresponding base layer 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 base-layer and the other for the enhancement layer. The base-layer is coded with the MPEG-4 motion-compensation DCT-based video encoding method (non-scalable). The enhancement-layer is coded using bitplane based embedded DCT coding.

[1] This brief sub-section is mainly provided to make the paper self-contained. Readers who are familiar with the FGS framework can skip this sub-section without impacting their understanding of the remainder of the paper.

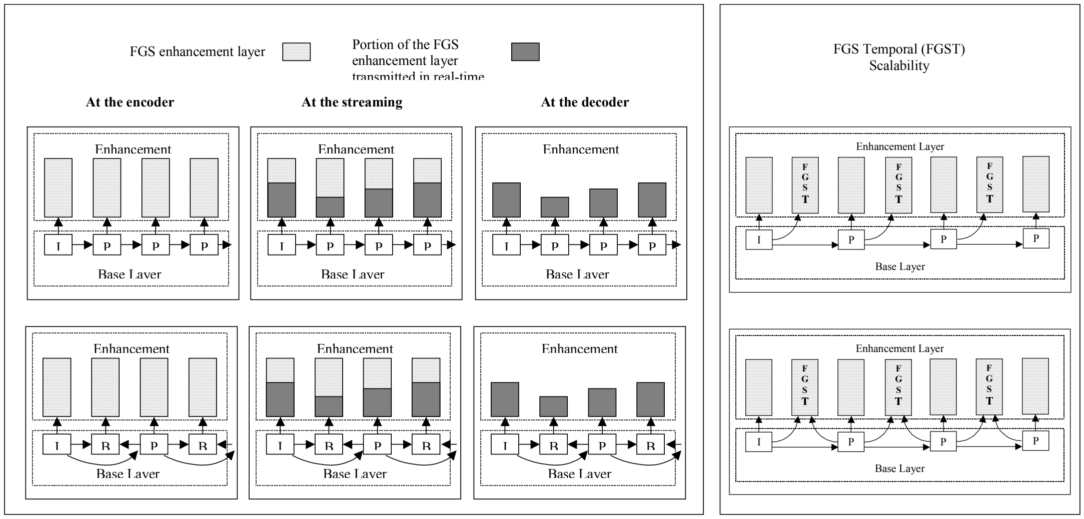


Figure 4: Examples of the MPEG-4 FGS and FGST scalability structures. Examples of the hybrid temporal-SNR scalability structures are shown on the right side of the figure. Both bi-directional (lower right structure) and forward-prediction (top right figure) FGST picture types are supported by the MPEG-4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

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 base-layer, one for the SNR FGS frames, and the third one for the FGST frames.

For receiver-driven multicast applications, 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 enhancement layer 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 "base-layer channel" and to any number of FGS enhancement-layer channels that the receiver is capable of accessing (depending for example on the receiver access bandwidth). It is important to note that regardless of the number of FGS enhancement-layer channels that the receiver subscribes to, the decoder has to decode only a single enhancement-layer.

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

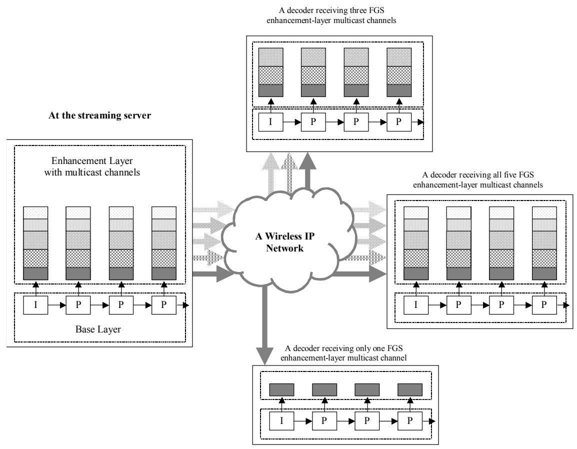


Figure 5: An example of video multicast using MPEG-4 FGS over a wireless IP network.

3.2 Hierarchical TranScaling (HTS) of MPEG-4 FGS Video Streams

Examples of transcaling 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 base layer BL_{in} of S_{in} (with bitrate 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 bitrate of the the EL_{in} used to generate the new base layer BL_1, then this new BL's bitrate R_{min_1} satisfies the following:

$$R_{min\_in} < R_{min\_1} < R_{min\_in} + R_{e1}$$

Consequently, and based on the definition we adopted earlier for "up transcaling" and "down transcaling", this example represents an "up transcaling" scenario. Furthermore, in this case, both the base and enhancement layers of the input stream S_{in} has been modified. Consequently, this represents a "full" transcaling scenario. Full transcaling 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[1].

[1] To reduce the complexity of full transcaling 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 transcoding.

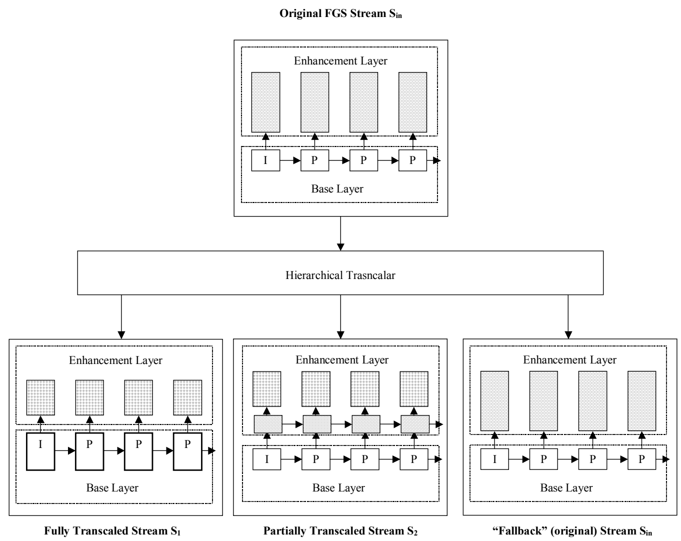


Figure 6: Examples of Hierarchical TranScaling (HTS) of the MPEG-4 FGS scalability structure with a Full TranScaling (FTS) option.

The residual signal between the original stream S_{in} and the new BL_1 stream is coded using FGS enhancement-layer compression. Therefore, this is an example of transcaling an FGS stream with a bitrate range $R_{range\_in}=[R_{min\_in} , R_{max\_in}]$ to another FGS stream with a bitrate range $R_{range\_1}=[R_{min\_1} , R_{max\_1}]$. It is important to note that the maximum bitrate R_{max_1} can be (and should be) selected to be smaller than the original maximum bitrate[1] R_{max_in}:

$$R_{max\_1} < R_{max\_in}$$

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 bitrate $R >> R_{max\_1}$. Consequently, transcaling could enable a device which has a bandwidth $R >> R_{max\_1}$ to receive a better (or at least similar) quality video while saving some bandwidth. (This access bandwidth can be used, for example, for other auxiliary or non-realtime applications.)

As mentioned above, under "full" transcaling, all pictures of both the base and enhancement layers 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 fallback to the original FGS stream, and consequently, this provides some level of computational scalability.

Furthermore, FGS provides another option for transcaling. Here, the gateway server can transcales the enhancement layer only. This is achieved by (a) decoding a portion of the enhancement layer of one picture, and (b) using that decoded portion to predict the next picture of the enhancement layer, and so on. Therefore, in this case, the base layer of the original FGS stream S_{in} is not modified and the computational complexity is reduced compared to full transcal-

[1] It is feasible that the *actual* maximum bitrate of the transcaled stream S_1 is higher than the maximum bitrate of the original input stream S_{in}. However, and as expected, this increase in bitrate 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 bitrate $R_{max\_1} < R_{max\_in}$.

ing of the whole FGS stream (i.e., both base and enhancement layers). Similar to the previous case, the motion vectors from the base layer can be reused here for prediction within the enhancement layer to reduce the computational complexity significantly.

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

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

4. Full TranScaling for High-bitrate Wireless LANs

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

We coded several video sequences using the draft standard of the MPEG-4 FGS encoding scheme. These sequences were then modified using the full transcaler architecture shown in Figure 7. The main objective for adopting the transcaler shown in the figure is to illustrate the potential of video transcaling and highlight some of its key advantages and limitations[1].

The level of improvements achieved by transcaling depends on several factors. These factors include the type of video sequence that is being transcaled. For example, certain video sequences with a high degree of motion and scene changes are coded very efficiently with FGS [10]. Consequently, these sequences may not benefit significantly from transcaling. On the other end, sequences that contain detailed textures and exhibit a high degree of correlation among successive frames could benefit from transcaling significantly. Overall, most sequences gained visible quality improvements from transcaling.

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

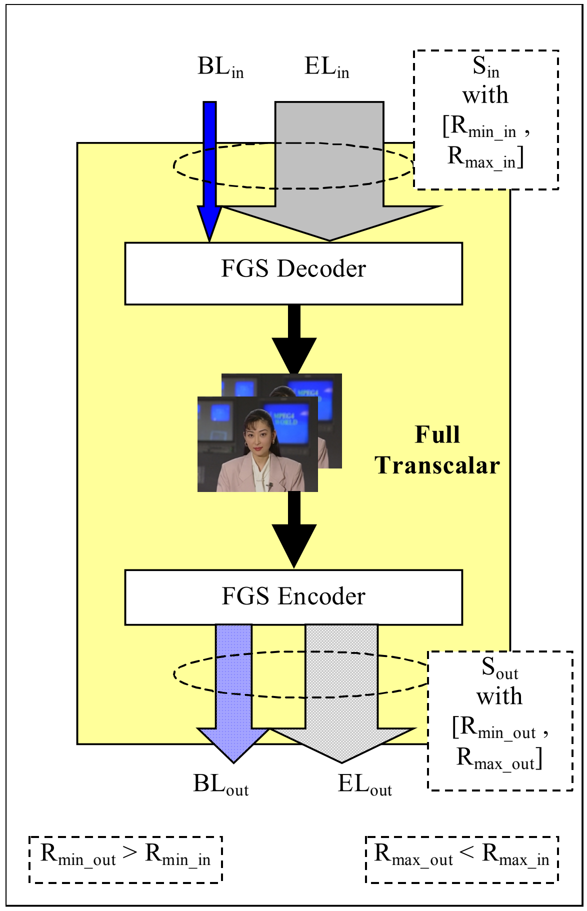


Figure 7: The full transcaler architecture used for generating the simulation results shown here.

Another key factor is the range of bitrates used for both the input and output streams. Therefore, we first need to decide on a reasonable set of bitrates that should be used in our simulations. As mentioned in the introduction, new wireless LANs (e.g., 802.11a or HiperLAN2) could have bitrates on the order of tens of Mbits/second (e.g, more than 50 Mbit/sec). Although it is feasible that such high bitrates 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 bitrates. Moreover, in practice, most video sequences[1] can be coded very efficiently at bitrates below 10 Mbits/sec. Consequently, the FGS sequences we coded were compressed at maximum bitrates (i.e., R_{max_in}) at around 6-8 Mbits/sec. For the base-layer bitrate R_{min_in} , we used different values in the range of few hundreds kbit/sec (e.g., between 100 and 500 kbit/sec). Video parameters, which are suitable for the base-layer bitrates, were selected. All sequences were coded using CIF resolution and 10-15 frames/sec[2].

First, we present the results of transcaling an FGS stream ("Mobile") that has been coded originally with R_{min_in}=250 kbit/sec and R_{max_in}=8 Mbit/sec. The transcaler used a new base-layer bitrate R_{min_out}=1 Mbit/sec. This

[1] The exceptions to this statement are high-definition video sequences, which could benefit from bitrates around 20 Mbit/sec.
[2] Our full transcaler used the exact same video parameters of the original video sequence (except bitrates) in order to avoid biasing the results.

example could represent a stream that was coded originally for transmission over lower bitrate systems (e.g., cable modem or legacy wireless LANs) and is being transcaled for transmission over new higher-bitrate LANs. The Peak SNR (PSNR) performance of the two streams as functions of the bitrate is shown in Figure 8. (For more information about the MPEG-4 FGS encoding and decoding methods, the reader is referred to [10][11].)

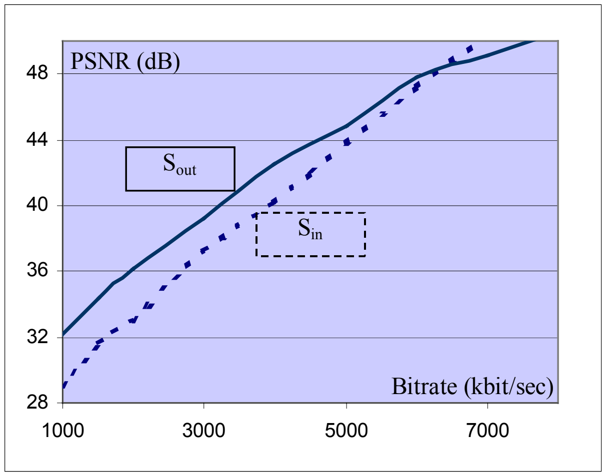


Figure 8: Performance of transcaling the "Mobile" sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=250 kbit/sec into a stream with a base-layer R_{min_out}= 1 Mbit/sec.

It is clear from the figure that there is a significant improvement in quality (close to 4 dB) in particular at bitrates close to the new base-layer rate of 1 Mbit/sec. The figure also highlights that the improvements gained through transcaling are limited by the maximum performance of the input stream S_{in}. As the bitrate gets closer to the maximum input bitrate (8 Mbit/sec), 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 bitrate range (i.e., above 1 Mbit/sec), the performance of the transcaled stream is significantly higher. In order to appreciate the improvements gained through transcaling, 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 pre-compressed stream such as S_{in}). In this example, an ideal FGS stream is generated from the original sequence with a base-layer of 1 Mbit/sec. Figure 9 shows the comparison between the transcaled stream and an "ideal" FGS stream over the range 1 to 4 Mbit/sec. As shown in the figure, the performances of the transcaled and ideal streams are virtually identical over this range.

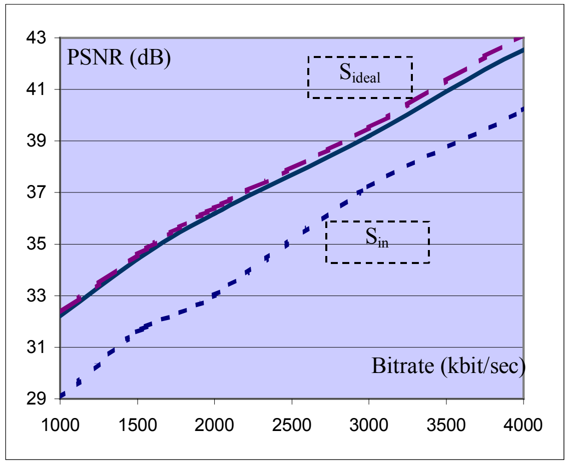


Figure 9: Comparing the performance of the "Mobile" transcaled stream (shown in Figure 8) with an "ideal" FGS stream. The performance of the transcaled stream is represented by the solid line.

By increasing the range of bitrates 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 base-layer bitrate of R_{min_in}=250 kbit/sec, we transcaled this stream with a new base-layer bitrate R_{min_out}=500 kbit/sec (i.e., lower than the 1 Mbit/sec base-layer bitrate of the transcaling example described above). Figure 10 shows the PSNR performance of the input, transcaled, and "ideal" streams. Here, the PSNR improvement is as high as 2 dB around the new base-layer bitrate 500 kbit/sec. 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 transcaled stream does saturates toward the performance of the input stream S_{in} at higher bitrates, and, overall, the performance of the transcaled stream is very close to the performance of the "ideal" FGS stream.

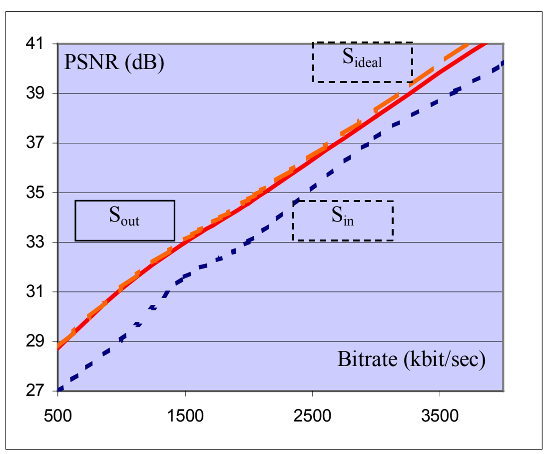


Figure 10: Performance of transcaling the "Mobile" sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=250 kbit/sec into a stream with a base-layer R_{min_out}= 500 kbit/sec.

Therefore, transcaling 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 bitrate 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 [10]. Figure 11 illustrates the performance of transcaling the “Coastguard” MPEG-4 test sequence. The original MPEG-4 stream S_{in} has a base-layer bitrate R_{min}=250 kbit/sec and a maximum bitrate of 4 Mbit/sec. Overall, FGS (without transcaling) provides a better quality scalable video for this sequence when compared with the performance of the previous sequence (Mobile). Moreover, the maximum bitrate used here for the original FGS stream (R_{max_in} = 4 Mbit/sec) is lower than the maximum bitrate 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 bitrate for the original FGS stream S_{in}) led to the following: the level of improvements achieved in this case through transcaling is lower than the improvements we observed for the Mobile sequence. Nevertheless, significant gain in quality (more than 1 dB at 1 Mbit/sec) can be noticed over a wide range over the transcaled bitstream. Moreover, we observe here the same “saturation-in-quality” behavior that characterized the previous Mobile sequence experiments. As the bitrate gets closer to the maximum rate R_{max_in}, the performance of the transcaled video approaches the performance of the original stream S_{in}.

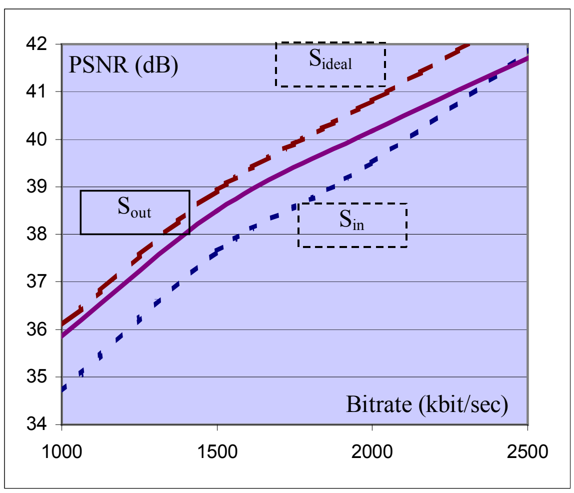


Figure 11: Performance of transcaling the “Coastguard” sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=250 kbit/sec into a stream with a base-layer R_{min_out}= 1000 kbit/sec.

The above results for transcaling were observed for a wide range of sequences and bitrates. So far, we have focused our attention on the performance of “up transcaling” (UTS), which we have referred to throughout this section simply by using the word “transcaling”. Now, we shift our focus to some simulation results for “down-transcaling”.

As explained above, Down TranScaling (DTS) can be used to convert a scalable stream with a base-layer bitrate R_{min_in} into another stream with a smaller base layer bitrate $R_{min\_out} < R_{min\_in}$. 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 bitrate (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 bitrate needed. In this case, the transcaler has to generate a new scalable bitstream with a lower base layer $R_{min\_out} < R_{min\_in}$. Below, we show some simulation results for down transcaling.

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 bitrate range R_{min_in}=1 Mbit/sec to R_{max_in}=8 Mbit/sec. Figure 12 illustrates the performance of the down-transcaling operation for two bitstreams: One stream was generated by down-transcaling the original FGS stream (with a base-layer of 1 Mbit/sec) into a new scalable stream coded with a base-layer of R_{min_out}= 500 kbit/sec. The second stream was generated using a new base layer R_{min_out}= 250 kbit/sec. As expected, the down-transcaling operation degrades the overall performance of the scalable stream.

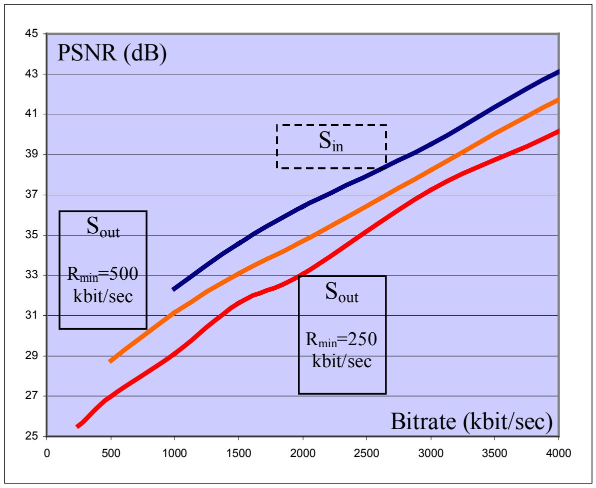


Figure 12: Performance of down-transcaling (DTS) the "Mobile" sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=1 Mbit/sec into two streams with base-layers R_{min_out}= 500 and 250 kbit/sec.

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 the other hand, clients with access bandwidth less than the original base-layer bitrate R_{min_in}, can only use the down-transcaled bitstream.

As mentioned in a previous section, down-transcaling (DTS) is similar to traditional transcoding, which converts a non-scalable bitstream into another non-scalable stream with a lower bitrate. 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 base-and-enhancement layers or (b) the base-layer only to perform the desired down-conversion. This, for example, may be used to reduce the amount of processing power needed for the down-transcaling operation. In this case, the transcaler has the option of performing only one decoding process (on the base-layer only versus decoding both the base and enhancement layers). However, using the base-layer 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 transcaling using (a) the entire input stream S_{in} (i.e., base plus enhancement) and (b) the base-layer 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 kbit/sec, R_{min_in}=500 kbit/sec]. 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 the base-layer BL_{in} (only) to generate the new down-transcaled stream is a viable option.

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 base-layer and any portion of the enhancement layer to generate the new down-transcaled scalable bitstream. The larger the portion of the enhancement layer used for down-transcaling, the higher the quality of the resulting scalable video. Therefore, and since partial decoding of the enhancement-layer 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 up- and down-transcaling.

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 the base layer 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}.

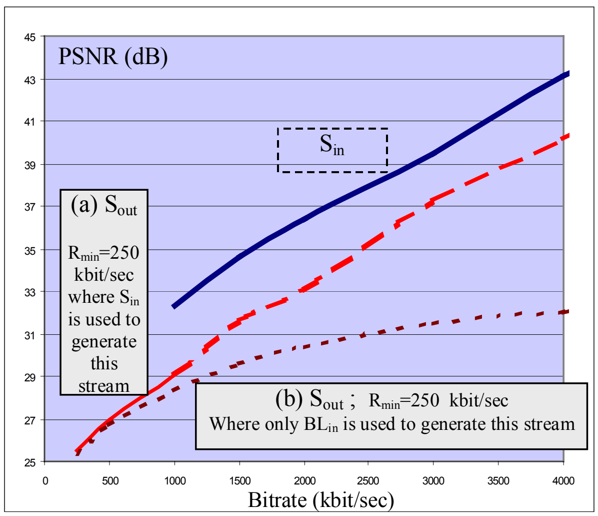


Figure 13: Performance of down-transcaling (DTS) the "Mobile" sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=1 Mbit/sec. Here, two DTS operations are compared: (a) the whole input stream S_{in} (base+enhancement) is used ; (b) only the base-layer BL_{in} of S_{in} is used to generate the down-transcaled stream. In both cases, the new DTS stream has a base-layer bitrate R_{min_out}= 250 kbit/sec.

5. Partial TranScaling for High-bitrate Wireless LANs

As described above, the MPEG-4 FGST framework supports SNR (regular FGS), temporal (FGST frames), and hybrid SNR-temporal scalabilities. At low bitrates (i.e., bitrates close to the base-layer bitrate), receivers can benefit from the standard SNR FGS scalability by streaming the base-layer 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 non-transcaled FGST streams. One of the reasons for the relatively high penalty in quality associated with the traditional FGST-based coding is that, at high bitrates, the FGST frames are predicted from low-quality (low bitrate) base-layer 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 transcaling operation that provides the desirable improvements in quality. One approach for maintaining low complexity transcaling is to eliminate the need for re-encoding the base-layer. Consequently, this eliminates the need for re-computing new motion vectors, which is the most costly part of a full transcaler that elects to perform this re-computation. Meanwhile, improvements can be achieved by using higher-quality (higher bitrate) SNR FGS pictures to predict the FGST frames. This reduces the entropy of the bi-directionally 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 partial transcaling scheme for FGST are depicted in Figure 14.

As shown in Figure 14, and similar to the full transcaling case, there are two options for supporting transcaling of FGST streams: the partial transcaling option and the fallback (no transcaling) 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 bitplanes of that frame. It is important to note that at higher bitrates, larger number of FGS SNR bitplanes will be streamed, and consequently these bitplanes can be used to predict the FGST frames. Therefore, under a Receiver-driven Layered Multicast (RLM) framework, receivers that "subscribe" to the transcaled FGST stream should also "subscribe" to the appropriate number of FGS SNR bitplanes.

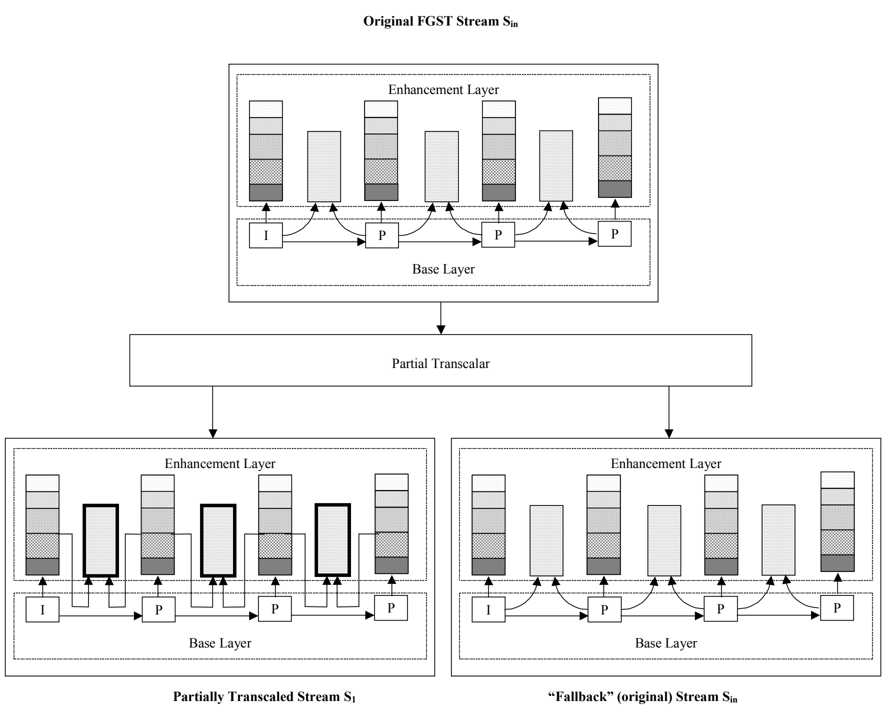


Figure 14: The proposed partial transcaling of the MPEG-4 FGST scalability structure. The FGST frames are the only part of the original scalable stream that is fully re-encoded under the proposed partial transcaling scheme.

Under the above-proposed partial transcaling, the input FGST stream S_{in} is transcaled into another scalable stream S_1. In this case, the base layer BLin of S_{in} (with bitrate 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 transcaling an FGST stream with a bitrate range R_{range_in}=[R_{min_in} , R_{max_in}] to another FGST stream with a bitrate 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 “up transcaling” and “down transcaling”, this is example represents an “up transcaling” scenario. Furthermore, in this case, only the FGST enhancement layers of the input stream S_{in} has been modified. Consequently, this represents a “partial” transcaling scenario. Partial transcaling 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 an “up transcaling” scenario here, low-bandwidth receivers can still use the base-layer of the new transcaled stream, which is identical to the original base-layer. These receivers 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 bitrate 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 partial transcaling can provide for wireless Internet applications, in this section, we present some simulation results of the FGST based partial transcaling method described above. As in the full TS 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 enhancement-layer for FGST prediction as shown in Figure 14. We should emphasize here the following: (a) Unlike the full TS results shown above, all the results presented in this section are based on re-using the same motion vectors that were originally computed by the base-layer 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 bitrates (i.e., R_{max_in}) lower than 2 Mbits/sec. For the base-layer bitrate R_{min_in} , we used 50-100 kbit/sec. Other video parameters, which are suitable for the base-layer bitrates, were selected. All sequences were coded using CIF resolution; however, and since the bitrate ranges are smaller than the full TS experiments, 10 frames/sec were used in this case. The GOP size is 2-second long and M=2 (i.e. one FGST bi-directionally predicted frame can be inserted between two I and P reference frames).

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

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

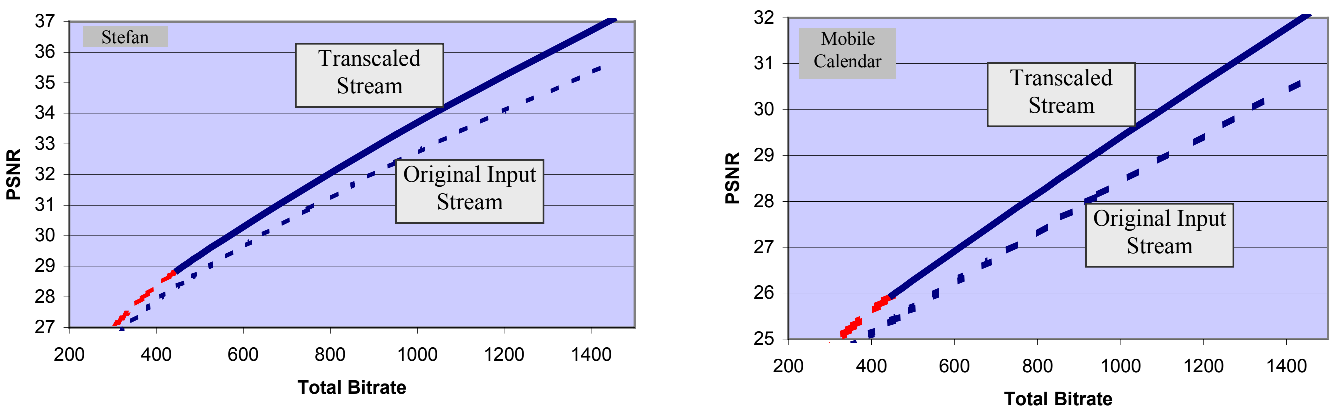


Figure 15: Performance of partial transcaling of the two sequences: *Stefan* and *Mobile*.

As mentioned above, the level of gain obtained from the proposed partial transcaling operation depends on the type of video sequence. Moreover, the number of FGS bitplanes 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 partial transcaling 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 bitrates. Figure 16 also shows the impact of using different number of FGS bitplanes from predicting the FGST frames. It is clear from both figures that, in general, larger number of bitplanes provides higher gain in performance. However, it is important to note that this increase in PSNR gain (as the number of FGS bitplanes 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 partial transcaling, in general partial transcaling in particular without the bias of different video parameters and related algorithms.

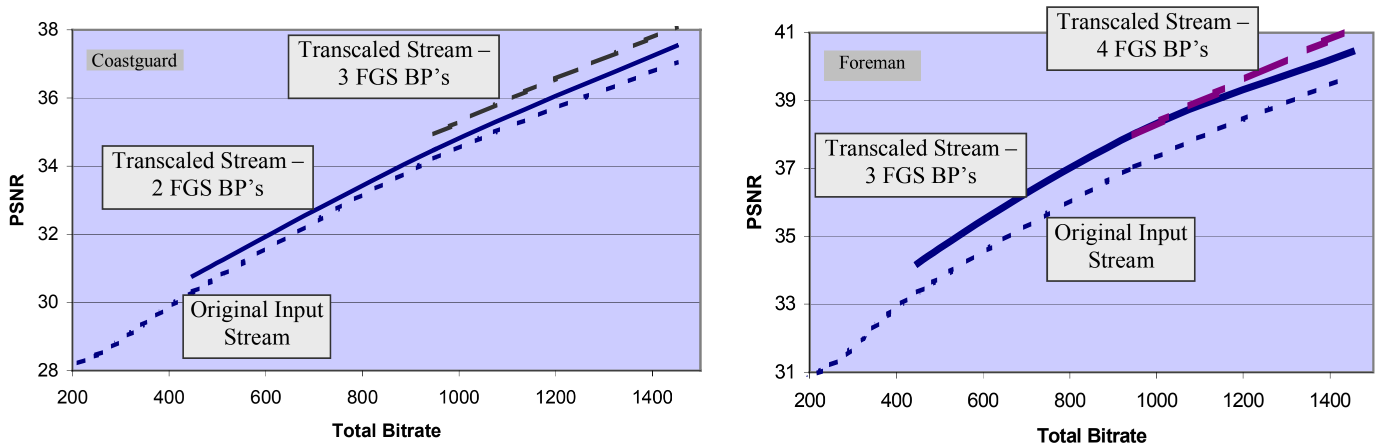


Figure 16: Performance of partial transcaling of the two sequences: *Coastguard* and *Foreman*.

6. Summary and Future Work

In this paper, we introduced the notion of transcaling, which is a generalization of (non-scalable) transcoding. With TranScaling (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 transcaling process.

We argued that an Internet/wireless network gateway represents a good candidate for performing transcaling. Moreover, we described Hierarchical TranScaling (HTS), which provides a "transcaler" the option of choosing among different levels of transcaling processes with different complexities. This enables transcalers to tradeoff video quality with computational complexity. We illustrated the benefits of *full* and *partial* transcaling by considering the recently developed MPEG-4 Fine-Granularity-Scalability (FGS) video coding.

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

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

- A thorough analysis for an optimum rate-distortion (RD) approach for the transcaling 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 [21], 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?"
- 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 [22][23].
- Optimum networked TS that tradeoffs 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 for example [24][25][26][27][28]).

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.

References

[1] B. H. Walke, et al, "IP over Wireless Mobile ATM – Guaranteed Wireless QoS by HiperLAN/2," Proceedings of the IEEE, January 2001.

[2] "High Speed Physical Layer in the 5 GHz Band," Draft Supplement to IEEE 802.11, 1999.

[3] R. Prasad, et al, " Third Generation Mobile Communication Systems, " Artech House, March 2000.

[4] M.-T. Sun and A. Reibmen, Editors, Compressed Video over Networks, Marcel Dekker, Inc., 2000.

[5] B. Girod and N. Farber, "Wireless Video," Chapter in Compressed Video over Networks, Marcel Dekker, Inc., 2000.

[6] H. Radha, C. Ngu, T. Sato, and M. Balakrishnan, "Multimedia over Wireless," Chapter in Advances in Multimedia: Systems, Standards, and Networks, Marcel Dekker, Inc., 2000.

[7] M.R. Civanlar, "Internet Video," Chapter in Advances in Multimedia: Systems, Standards, and Networks, Marcel Dekker, Inc., 2000.

[8] W. Tan and A. Zakhor, "Real-Time Internet Video Using Error Resilient Scalable Compression and TCP-Friendly Transport Protocol", IEEE Trans. on Multimedia, vol. 1, no. 2, June 1999.

[9] H. Radha Y. Chen, K. Parthasarathy, and R. Cohen, "Scalable Internet Video Using MPEG-4," Signal Processing: Image Communication, Sept. 1999.

[10] H. Radha and M. van der Schaar, "The MPEG-4 FGS Video Coding Method for Multimedia Streaming over IP," IEEE Transactions on Multimedia, March 2001.

[11] ISO/IEC 14496-2, "Information Technology – Coding of Audio-Visual Objects: Visual", International Standard, ISO/IEC JTC1/SC29/WG11, March 2000.

[12] D. Wu, et al. "Scalable Video Coding and Transport over Broadband Wireless Networks," Proceedings of the IEEE, January 2001.

[13] IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Active and Programmable Networks, March 2001.

[14] M. Allman and V. Paxson, "On estimating end-to-end network path properties," Proc. ACM SIGCOMM '99 Conf., Cambridge, Mass., Sept 1999, vol. 29, no. 4, pp 263-274, October 1999.

[15] V. Paxson, "End-to-end Internet packet dynamics," Proc. ACM SIGCOMM '97 Conf., Cannes, France, Sept 1997, vol. 27, no. 4, pp 139-52, October 1997.

[16] D. Loguinov and H. Radha, "Measurement Study of Low-bitrate Internet Video Streaming," ACM SIGCOMM – Internet Measurement Workshop, November 2001.

[17] S. McCanne, V. Jackobson, and M. Vetterli, "Receiver-driven Layered Multicast," Proc. SIGCOMM'96, Standford, CA, Aug. 1996, pp. 117-30.

[18] S. McCanne, M. Vetterli, and V. Jacobson, "Low-Complexity Video Coding for Receiver-Driven Layered Multicast," IEEE JSAC, vol. 16, no. 6, Aug. 1997, pp. 983-1001.

[19] H. Radha, "TranScaling: A Video Coding and Multicasting Framework for Wireless IP Multimedia Services," Proceedings of the ACM SIGMOBILE Workshop on Wireless Mobile Multimedia, pp. 13-23, July 2001.

[20] H. Radha and M. van der Schaar, “Up-TranScaling for Improved Fine-Granular-Temporal-scalability”, Proc. Of Packet Video, April 2002.

[21] M. Dai, D. Loguinov, and H. Radha, "Statistical Analysis and Distortion Modeling of MPEG-4 FGS," IEEE International Conference on Image Processing (ICIP), September 2003.

[22] S. A. Khayam, S. S. Karande, M. Krappel, and H. Radha, "Cross-Layer Protocol Design for Real-Time Multimedia Applications over 802.11b Networks," IEEE International Conference on Multimedia and Expo (ICME), July 2003.

[23] 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 - Special Issue on Recent Advances in Wireless Video, vol.18, no.7, pp. 575-595, August 2003.

[24] Session on “Overlay Networks,” SIGCOMM 2002, August 2002.

[25] Special Issue on “Network Support for Multicast Communications,” IEEE Journal on Selected Area in Communications (JSAC), October 2002.

[26] Session on “Overlay Routing and Multicast,” INFOCOM 2003,” April 2003.

[27] Special Session on “Networked Video,” (SS-LI), IEEE International Conference on Multimedia and Expo (ICME), July 2003.

[28] Lecture Session on “Multimedia Streaming,” (MCN-L5), IEEE International Conference on Multimedia and Expo (ICME), July 2003.