TRANSCALING: A VIDEO CODING AND MULTICASTING FRAMEWORK FOR WIRELESS IP MULTIMEDIA SERVICES

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

Abstract

The convergence of the Internet with new wireless and mobile networks is creating a whole new level of heterogeneity in multimedia communications. This increased level of heterogeneity 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 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 "Transcalar" 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-Granular-Scalability (FGS) video coding. Simulation results of video transcaling are also presented.

1. Introduction

It is well known that the current Internet exhibits a wide range of available bandwidth over both the core network and over different types of access technologies (e.g., analog modems, cable modems, DSL, LAN, etc.) [8][9][10]. Meanwhile, new wireless LANs and mobile networks are emerging as important Internet access mechanisms. Both the Internet 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 will be supporting (physical layer) bitrates from 6 Mbit/sec to 54 Mbit/sec [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 Internet and different wireless/mobile access networks as a large multimedia heterogeneous system, 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., [5][6][7][12][13]). Examples of these include receiver-driven multicast multilayer coding, MPEG-4 Fine-Granular-Scalable (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 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] [5]. With the aforementioned in-

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

crease 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, that 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 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. 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 wireless Internet while maintaining efficient utilization of the overall 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, the new higher-bandwidth LANs do not have to scarify 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 multicast video services over the wireless Internet, we illustrate the level of quality improvement that transcaling can provide by presenting several video simulation results. The remainder of the paper is organized as follows: Section 2 describes trascaling and its video multicast framework for the wireless Internet. Section 3 illustrates the usage of transcaling with the recently developed MPEG-4 Fine-Granular-Scalability (FGS) video coding method. Section 3 also includes a brief description of FGS and its tools for video streaming applications. Within the context of MPEG-4 FGS, we describe Hierarchical TranScaling (HTS) which is a framework that enables transcalaers to tradeoff video quality with complexity. Section 4 shows some simulation results of applying transcaling on FGS streams and the level of video quality improvement one can gain by utilizing this approach. Section 5 concludes the paper with a summary.

2. TranScaling based Mulicast (TSM) for Video over the Wireless Internet

A simple case of our proposed transcaling approach can be described within the context of Receiver Driven Multicast (RDM). Therefore, first, we briefly outline some of the basic characteristics of the RDM framework in order to highlight how this framework can be extended to our transcaling based solution. Then, we describe some general features of a transcaling wireless Internet system.

RDM 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) [11]. 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 RDM framework can also be

methods, and the newly developed MPEG-4 Fine-Granular-Scalability (FGS) method.

used for wireless IP devices that are capable of decoding the scalable content transmitted by an IP multicast server. Figure 1 shows a simple example of an RDM based system.

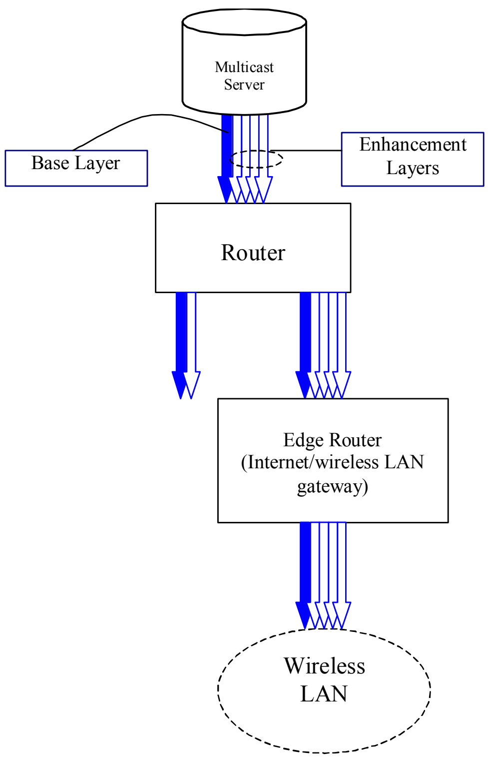

Figure 1: A simplified view of an RDM architecture.

Similar to RDM, 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 our proposed TSM framework and traditional RDM. Under TSM, an edge router[1] with a transcaling capability (or a "transcalar") 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.

[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.

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. Figure 2 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 transcalar 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. By interacting with the access-point, the gateway server can determine the bandwidth range needed for serving its devices efficiently. As illustrated in the simulation section, this approach could improve the video quality delivered to higher-bitrate devices significantly.

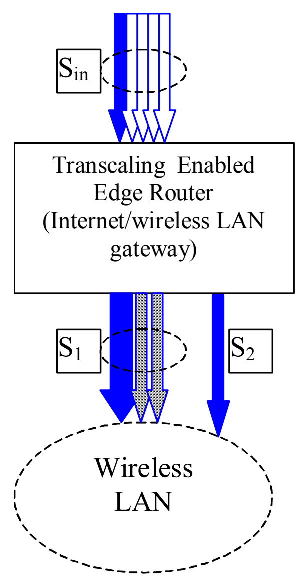

Figure 2: A simple architecture of a transcaling based node.

[2] Here, a "layered" or "scalable" stream consists of multiple sub-streams.

2.1 Attributes of Transcaing Based Systems

Here, we highlight the following attributes of the proposed 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 2. 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 transcalar delivers the original BL stream to the low-bitrate devices.
2. The proposed TSM system falls under the umbrella of active networks[1] where, in this case, the transcalar provides network-based added value services [4]. 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. Under our proposed TSM system, a transcalar 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 transcalar 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. Under a more general TSM framework, transcaling can take place at any node in the upstream path toward the multicast server. In fact, if the multicast server is covering a live event, then the scalable encoder system, which is compressing the video in realtime, can generate the desired sets of scalable streams. This general view of TSM provides a framework for distributing and scaling the desired transcaling processes throughout the multicast tree. Moreover, this general TSM framework leads to some optimization alternatives for the system. For example, depending on the bitrate ranges determined by the different edge servers (e.g., wired/wireless/mobile gateway servers), the system may have to trade off computational complexity (due to the transcaling processes) with bandwidth efficiency (due to the possible transmission of multiple scalable streams that have overlapping bitrate ranges over certain links)[2].
5. Although we have focused on describing our proposed transcaling approach in the context of multicast services, on-demand unicast applications can also take advantage of trasnscaling. 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.
6. As shown in the simulation section, 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.

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).

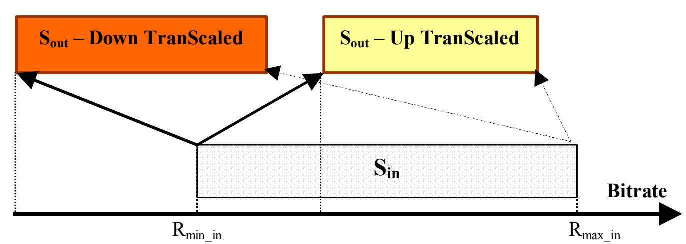

Figure 3: The distinction between DTS and UTS.

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

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

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

[2] Currently, we are investigating these optimization aspects of TSM based systems, and consequently, we will not elaborate further on these aspects in this paper.

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 3. 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. This type of down conversion has been studied by many researchers 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.)

After the above introduction to transcaling, its general features, and its potential benefits, we now give a concrete example of video transcaling within the context of the MPEG-4 FGS scalable video coding method. We start the next section with a brief introduction to MPEG-4 FGS and its coding tools developed in support of video streaming applications over the Internet and wireless networks.

3. MPEG-4 FGS Based TranScaling for the Wireless Internet

3.1 The MPEG-4 FGS Video Coding Method

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 [5]. 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.

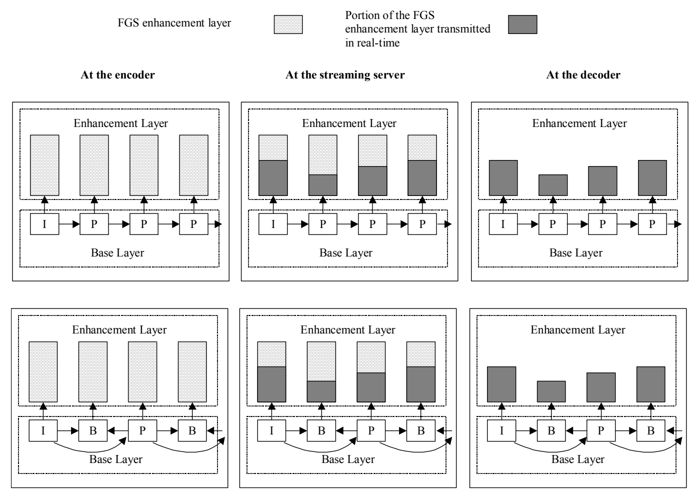

Figure 4: Examples of the MPEG-4 FGS Scalability Structure.

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.

[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.

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.

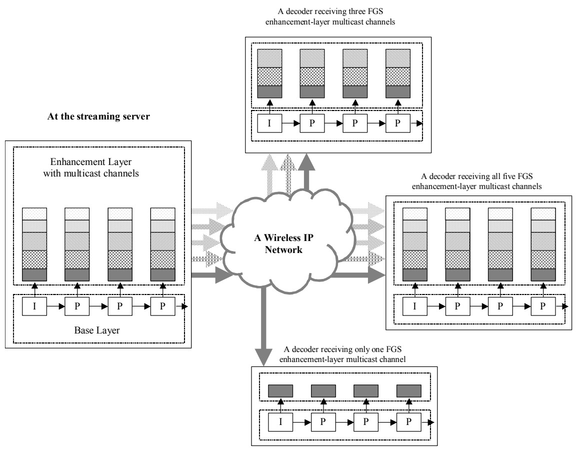

Figure 5: An example of video multicast using MPEG-4 FGS over a wireless IP network.

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

Examples of transcaling an MPEG-4 FGS stream is 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 portion of 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].

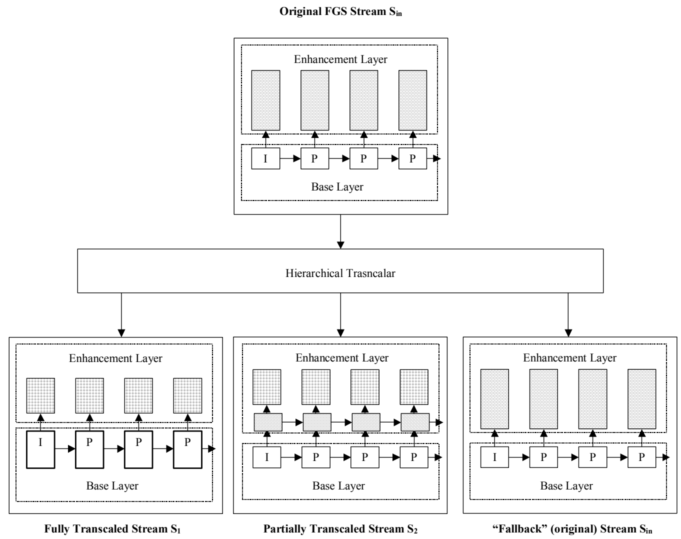

Figure 6: Examples of hierarchical transcaling of the MPEG-4 FGS Scalability Structure

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[2] 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.)

[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.

[2] 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 $R_{max\_1} < R_{max\_in}$.

As mentioned above, under "full" transcaling, 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 transcaling 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 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.

4. Simulation Results

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 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 transcalar architecture shown in Figure 7. The main objective for adopting the transcalar 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 depend 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. 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.

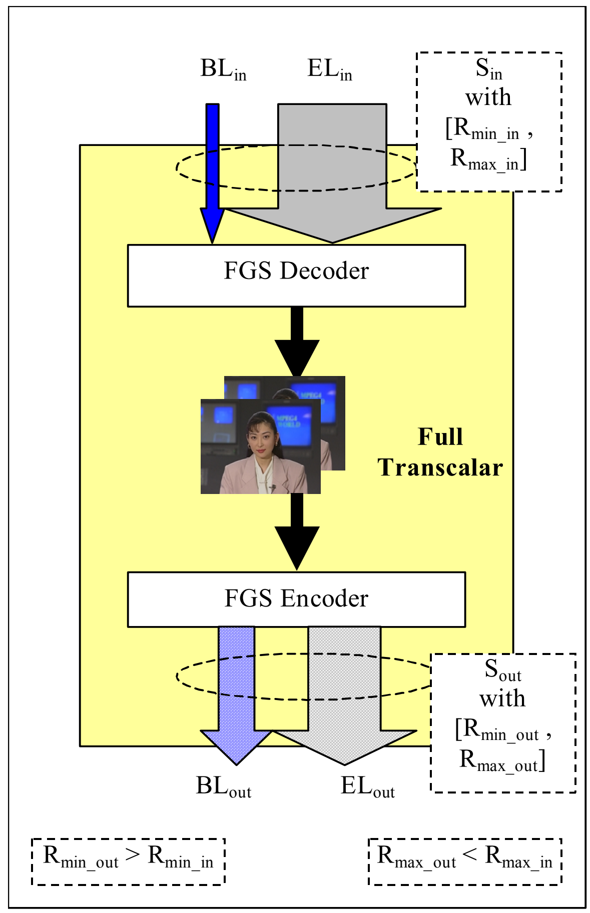

Figure 7: The full transcalar architecture used for generating the simulation results shown here.

[1] It is clear that other elaborate algorithms can be used for performing transcaling. However, these elaborate algorithms could bias some of our important findings regarding the performance of transcaling and related conclusions. 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 many other examples.

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}) lower than 10 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 200 and 500 kbit/sec).

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 transcalar used a new base-layer bitrate R_{min_out}=1 Mbit/sec. 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 [5][6].)

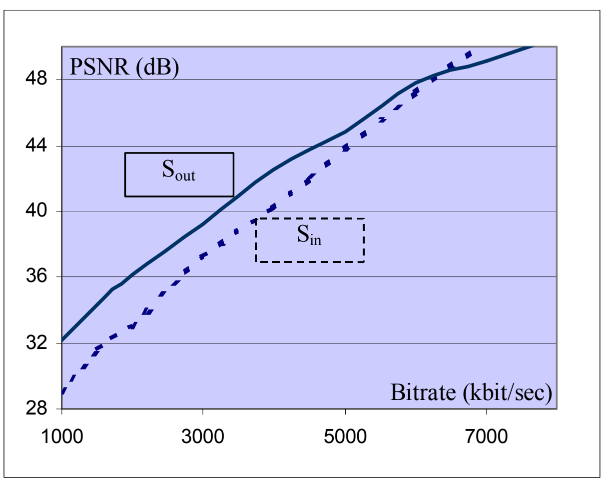

Figure 8: Performance of transcaling the "Mobile" sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=250 kbit/sec into a stream with a base-layer R_{min_out}= 1 Mbit/sec.

[1] The exceptions to this statement are high-definition video sequences which could benefit from bitrates around 20 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.

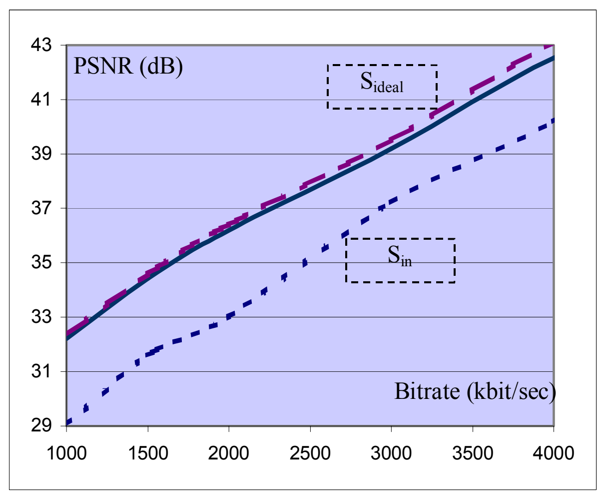

Figure 9: Comparing the performance of the "Mobile" transcaled stream (shown in Figure 8) with an "ideal" FGS stream. The performance of the transcaled stream is represented by the solid line.

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.

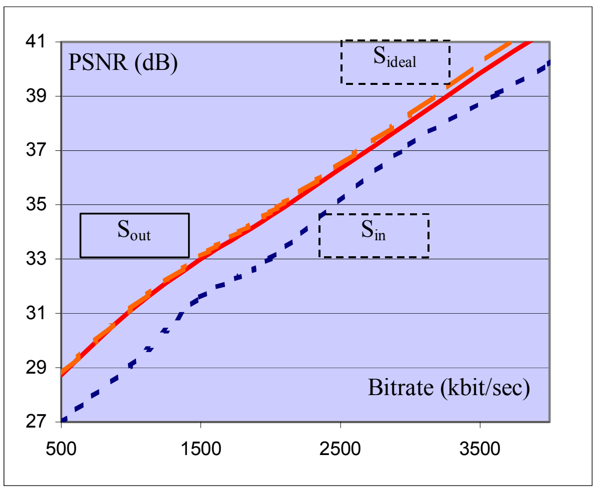

Figure 10: Performance of transcaling the "Mobile" sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=250 kbit/sec into a stream with a base-layer R_{min_out}= 500 kbit/sec.

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 transcalar. For example, and as mentioned above, FGS provides different levels of performance depending on the type of video sequence [5]. 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}.

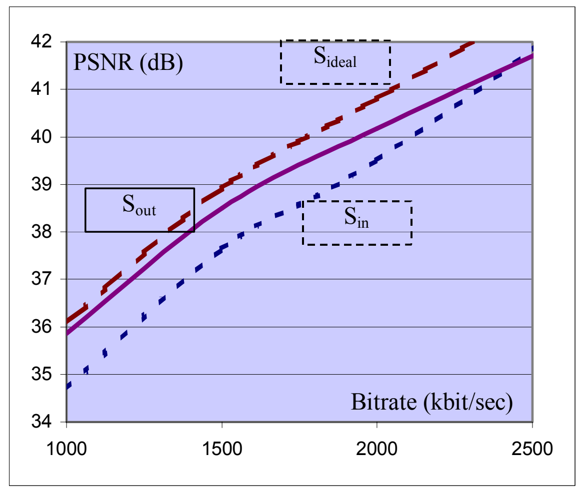

Figure 11: Performance of transcaling the "Coastguard" sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=250 kbit/sec into a stream with a base-layer R_{min_out}= 1000 kbit/sec.

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 transcalar 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 transcalar; and/or (c) sudden local congestion over a wireless LAN is observed, and consequently reducing the minimum bitrate needed. In this case, the transcalar 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 transcalar 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.

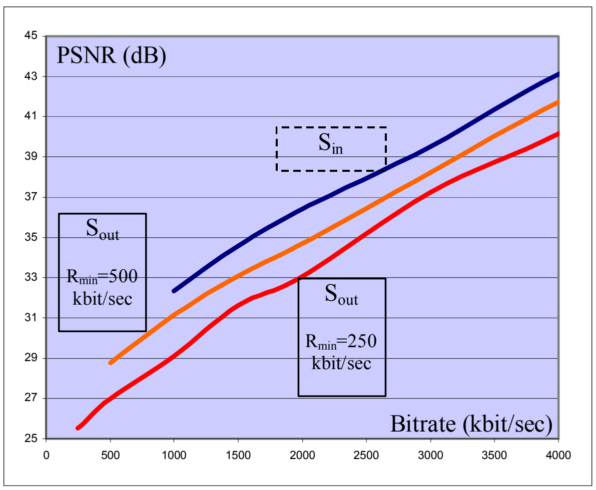

Figure 12: Performance of down-transcaling (DTS) the "Mobile" sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=1 Mbit/sec into two streams with base-layers R_{min_out}= 500 and 250 kbit/sec.

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 transcalar 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 transcalar 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 transcalar needs to employ a single scalable stream to transmit its content to its clients (e.g., multicast with a limited total bandwidth constraint), a transcalar 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 transcalar 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}.

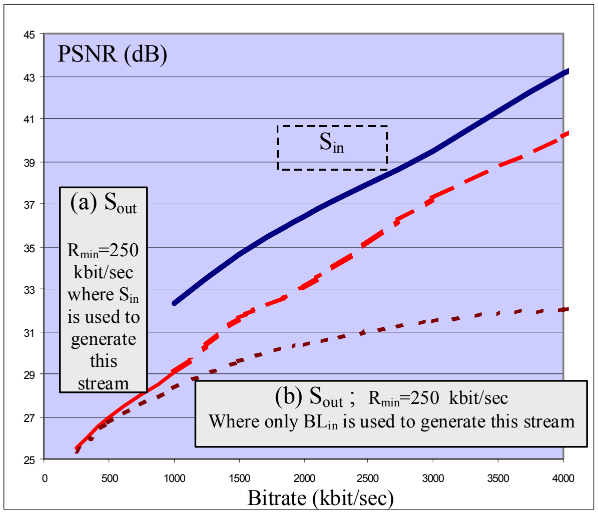

Figure 13: Performance of down-transcaling (DTS) the "Mobile" sequence using an input stream S_{in} with a base-layer bitrate R_{min_in}=1 Mbit/sec. Here, two DTS operations are compared: (a) the whole input stream S_{in} (base+enhancement) is used ; (b) only the base-layer BL_{in} of S_{in} is used 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. 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 "transcalar" the option of choosing among different levels of transcaling processes with different complexities. This enables transcalars to tradeoff video quality with computational complexity. We illustrated the benefits of transcaling by considering the recently developed MPEG-4 Fine-Granular-Scalability (FGS) video coding.

We examined two forms of transcaling: 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.

Currently, we are studying and analyzing several aspects of transcaling. This includes: (a) some optimization work on trading-off complexity versus quality throughout the distribution tree of the desired video content; (b) transcaling in the context of other scalable video schemes (e.g., wavelet based); (c) the development of a theoretical framework for understanding and predicting the behavior of different aspects of transcaling; and (d) other issues related to efficient algorithms for performing the desired transcaling processes.

Acknowledgments

The author would like to thank Mr. Shirish Karande from Michigan State University for his help and support in generating some of the simulation results presented in this paper.

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

[5] H. Radha, et al, "The MPEG-4 FGS Video Coding Method for Multmedia Streaming over IP," IEEE Transactions on Multimedia, March 2001.

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

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

[8] 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.

[9] V. Paxson, "End-to-End Internet Packet Dynamics*," Proc. ACM SIG-COM*, vol. 27, no. 4, p. 13-52, Oct. '97.

[10] 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.

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

[12] 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.

[13] B. Girod, K. W. Stuhlmüller, M. Link and U. Horn, "Packet Loss Resilient Internet Video Streaming", *VCIP'99, Proc. SPIE*, vol. 3653, p. 833-844, January 1999.