# PARTIAL TRANSCALING FOR WIRELESS PACKET VIDEO

Hayder Radha[1] and M. van der Schaar[2]
[1] Michigan State University [2]Philips Research

## Abstract

Scalable coding provides many advantages for video transmission over packet networks that lack Quality-of-Service (QoS) guarantees. Meanwhile, it is well known that the benefits of scalable coding come at the expense of degradation in video quality or reduction in coding efficiency when compared with non-scalable video compressed at a particular bitrate. Recently, the notion of *TranScaling* (TS) has been proposed [1] to improve the coding efficiency of scalable video coding. Under TS, which is a generalization of non-scalable transcoding, a scalable video stream, that covers a given bandwidth range, is mapped into one or more scalable video streams covering different bandwidth ranges. TS exploits the fact that the level of heterogeneity changes at different points of the video distribution tree over best-effort packet networks (e.g., the wireless Internet). This provides the opportunity to improve the video quality by performing the appropriate transcaling process. In this paper, we propose a new extension to the TranScaling framework by exploring *Partial TranScaling*, which provides a "Transcalar" the option of choosing a low complexity transcaling process. In particular, we propose partial-TS based approach for the MPEG-4 FGS Temporal (FGST) scalable video coding method. The proposed TS method for FGST significantly reduces the computational complexity of the original (full) TS scheme while achieving higher than 1 dB in PSNR improvements. We present some simulation results based on employing partial TS to a wide range of MPEG-4 FGST streams. In these experiments, we show 1.5 dB in PSNR improvements when compared with a non-transcaled MPEG-4 FGST video.

## 1. Introduction

New wireless LANs and mobile networks are emerging as important Internet access mechanisms. Moreover, these wireless and mobile networks are evolving to high-bitrate platforms with large 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 [2][3]. 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) [4]. 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. Meanwhile, bandwidth variation over the current Internet is a well known phenomenon that has been studied and analyzed for both the core network and over different types of wired access technologies (e.g., analog modems, cable modems, DSL, LAN, etc.) [5][6][7].

The above 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 order to address the bandwidth variation and heterogeneity aspects of the Internet and wireless networks, a variety of scalable video compression methods have been proposed and used extensively (e.g., [8]-[17], [20]). Examples of these include receiver-driven multicast multilayer coding, MPEG-4 Fine-Granular-Scalable (FGS) compression, H.263 based scalable coding, and 3-D motion-compensated wavelet based 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. In general, the wider the bandwidth range that needs to be covered by a scalable video stream, the lower the overall video quality is[1] [14].

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. Recently, we proposed a new approach for addressing the bandwidth variation issue over emerging wireless and mobile multimedia IP networks [1]. We refer to this approach as TranScaling (TS) since it represents a generalization of video transcoding. With TranScaling, one or more scalable streams covering different bandwidth ranges are derived from another scalable stream. TranScaling 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 gate-

---

[1] 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.

ways 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.

One of the major challenges of the TS framework is its computational complexity. In particular, if a gateway server needs to perform multiple TS operations, it may not be feasible to perform these operations in real-time. In this paper, we propose *Partial TranScaling*, which reduces the computational complexity of TS operations while achieving good quality improvements. Furthermore, we propose a low-complexity partial TS scheme for the MPEG-4 FGST standard. We also show several simulation results that illustrate the benefits and limitations of the proposed partial TS scheme. The remainder of the paper is organized as follows: Section 2 provides a high-level description of TS and its video multicast framework for the wireless Internet. Section 3 outlines our proposed method for partial TS and its application to MPEG-4 FGST coding. Section 4 shows some simulation results. Section 5 concludes the paper with a summary.

## 2. The TranScaling Framework and its Application to Multicasting Layered Video

The transcaling framework is very well suited for Receiver-driven Layered Multicast (RLM) applications. Therefore, we briefly outline some of the basic characteristics of the RLM framework in order to highlight how this framework can be extended to a transcaling-based solution. Then, we describe some general features of a transcaling 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) [18]. 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 as shown in Figure 1.

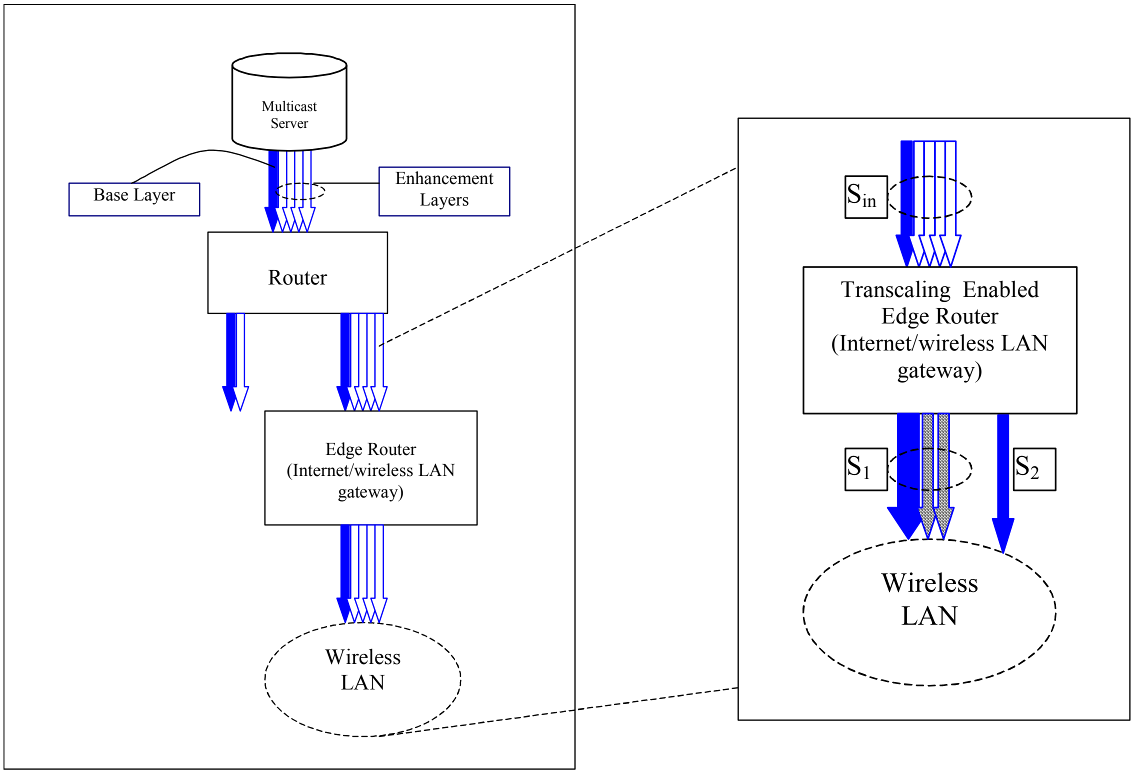

Figure 1: A simplified view of RLM architecture and how it can be extended to utilize a TranScaling framework.

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 our proposed TSM framework and traditional RLM. 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.

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 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 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.

## 2.1 TranScaling Based Systems: Features and Definitions

Before proceeding, it is important to highlight some of the basic features and definitions of the transcaling framework. Here, we outline four key attributes of the transcaling framework that are relevant to the proposed extensions of partial transcaling. (For more details regarding these and other attributes of the transcaling framework, the reader is referred to [1].)

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 transcalar delivers the original BL stream to the low-bitrate devices.
2. Under a TS system, a transcalar can always fallback to using the original (lower-quality) scalable video. This "fall-back" feature represents a key attribute of transcaling that distinguishes it from non-scalable transcoding. The "fall-back" 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.
3. In general, transcaling processes can be divided into two categories: Down TranScaling (DTS) and Up TranScaling (UTS). Let the original (input) scalable stream $S_{in}$ of a transcalar covers a bandwidth range:

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

   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. Moreover, DTS has been studied for SNR scalable video in [1]. On the other hand, up conversion has not received much attention (if any). For the remainder of this paper we will focus on up transcaling. (Unless otherwise mentioned, we will use "up transcaling" and "transcaling" interchangeably.)

---

[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.

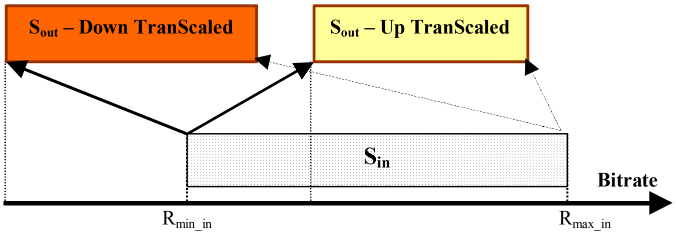

Figure 2: The distinction between DTS and UTS.

4. Transcaling can result in the generation of a new base-layer, which is different from the original base-layer, *and* one or more new enhancement layers. This type of Transcaling is referred to as *full* TS [1]. Otherwise, the TS operation is known as *partial* TS. As shown in [1], full TS can provide significant improvements in coding efficiency or video quality (e.g., more than 4 dB improvements in PSNR). However, full TS could be very computationally complex and may not be feasible to support in real-time under certain conditions. Consequently, partial TS can provide a viable option to improve the quality of scalable video while being relatively simple to realize in real-time. Below, we propose a partial TS approach that is suitable for the MPEG-4 FGST coding method.

## 3. Partial TranScaling for MPEG-4 FGS Temporal (FGST) Scalability

First, we provide a brief description of the MPEG-4 FGS coding framework. Second, we describe our proposed partial transcaling for the MPEG-4 FGS Temporal (FGST) scalability method.

### 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 [14]. As shown in Figure 3, 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 3, 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. 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 more details regarding the MPEG-4 FGS coding method, the reader is referred to [14][15][16].

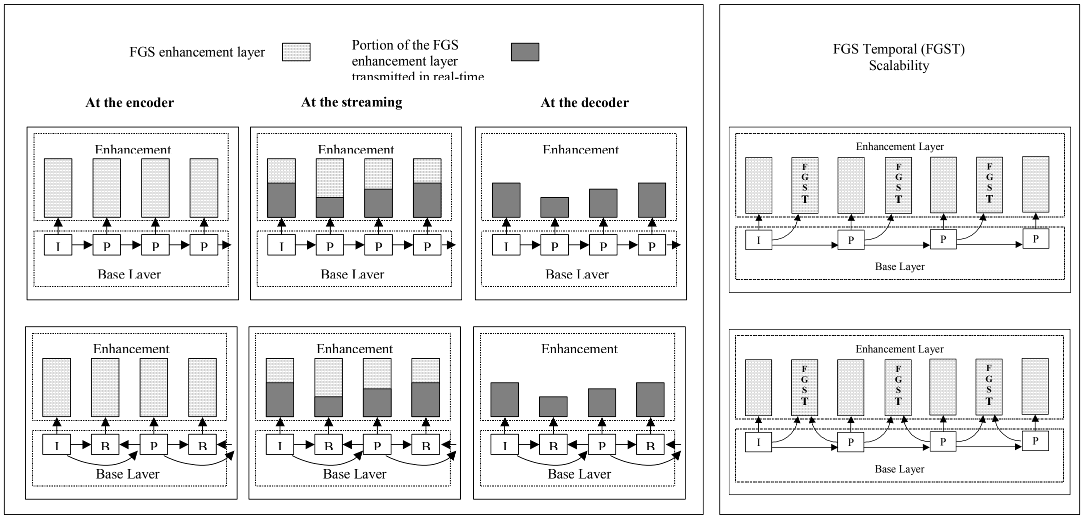

Figure 3: Examples of the MPEG-4 FGS and FGST scalability structures. Examples of the hybrid temporal-SNR scalability structures are shown on the right side of the figure. Both bi-directional (lower right structure) and forward-prediction (top right figure) FGST picture types are supported by the MPEG-4 FGS/FGST standard.

### 3.2 Partial TranScaling of FGST Video Streams

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 transcalar [1]. 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 4.

As shown in Figure 4, 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.

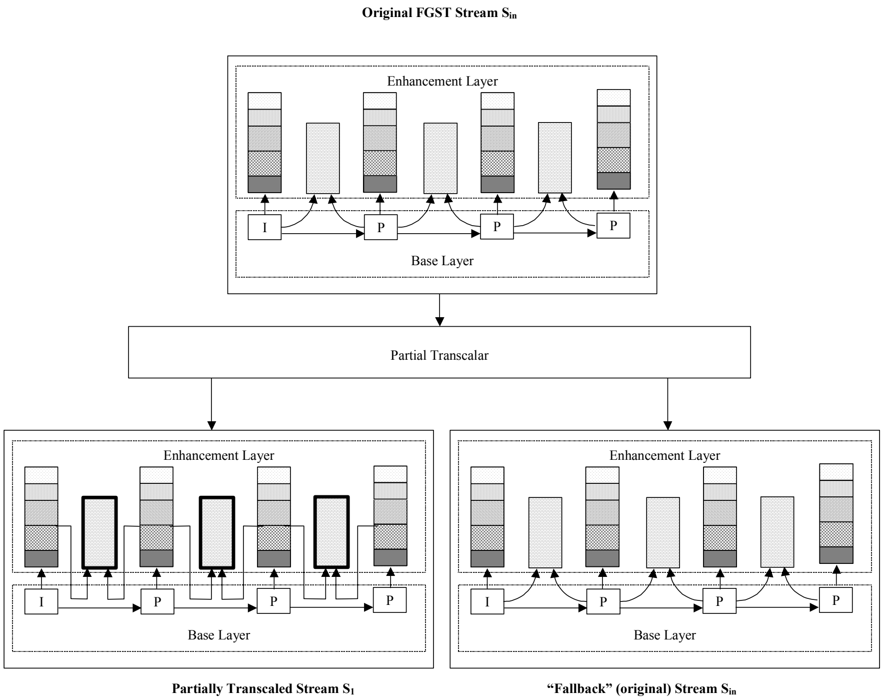

Figure 4: The proposed partial transcaling of the MPEG-4 FGST scalability structure. The FGST frames are the only part of the original scalable stream that is fully re-encoded under the proposed partial transcaling scheme.

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 $BL_{in}$ 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 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.

It should be also noted that the maximum bitrate $R_{max\_1}$ can be (and should be) selected to be smaller than the original maximum bitrate $R_{max\_in}$: $R_{max\_1} < R_{max\_in}$. 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, in general, and partial transcaling in particular 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 the proposed transcaling, the FGST enhancement layers of the original FGS stream $S_1$ have been modified. Although the original motion vectors can be reused here, this process may still be computationally complex for some gateway servers. In this case, the gateway could always fallback to the original FGS stream, and consequently, this provides some level of computational scalability. 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 the partial transcaling scenario, one may perform the desired transcaling on a fewer number of frames. This represents some form of temporal scalability in computational-complexity.

## 4. Simulation Results

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. We coded several video sequences using the MPEG-4 FGST scheme. These sequences were then modified using the partial transcalar scalability structure that employs a portion of the enhancement-layer for FGST prediction as shown in Figure 4. The main objective of our experiments is to illustrate the potential of partial transcaling and highlight some of its key advantages and limitations. We should emphasize here that all the results shown 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.

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 the current FGS/FGST implementation. 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. 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, several wireless devices will share the available bandwidth. Consequently, 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 and 10 frames/sec. The GOP size is 2sec 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 5 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 5 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).

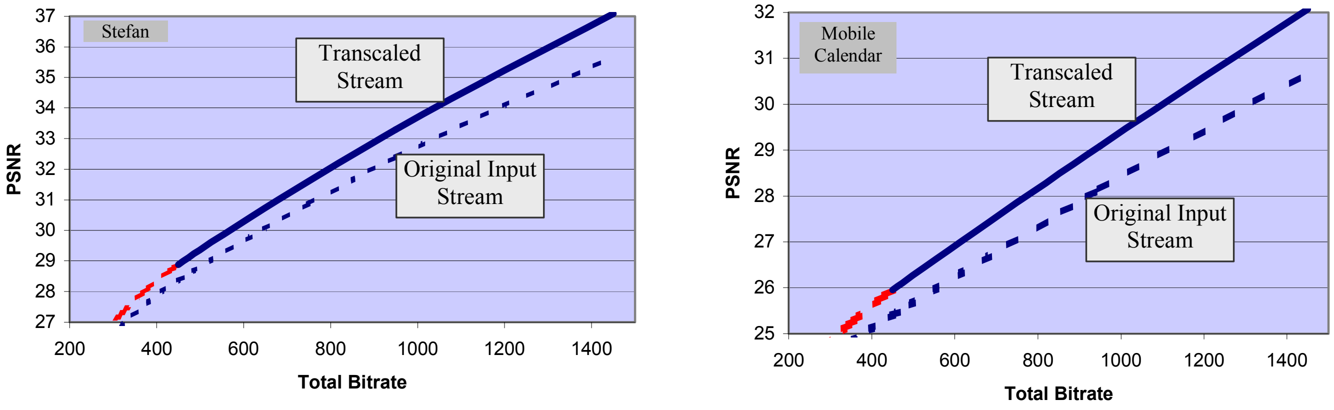

Figure 5: Performance of partial transcaling of the two sequences: *Stefan* and *Mobile*.

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 6 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 6 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. This leads to an interesting optimization problem, which is the selection of the optimum number of bitplanes that are needed for achieving the highest level of PSNR gain under certain computational complexity constraints. This optimization problem is currently under investigation.

Furthermore, we should emphasize here that many of the video parameters used at the partial transcalar do not represent the optimum 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, and of FGST-based partial transcaling in particular without the bias of different video parameters.

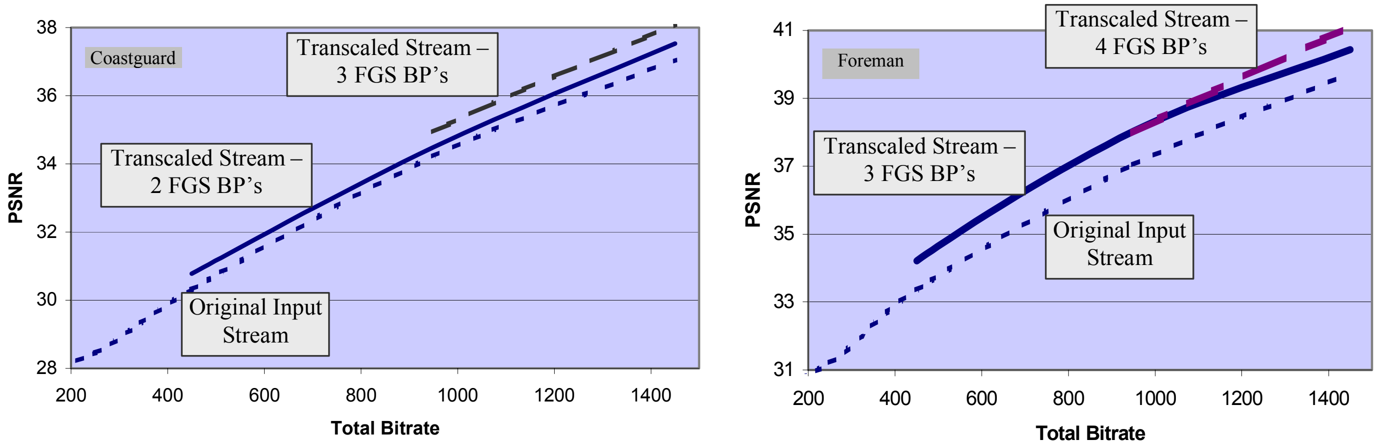

Figure 6 Performance of partial transcaling of the two sequences: *Coastguard* and *Foreman*.

## 5. Summary

In this paper, we proposed the notion of partial transcaling, which is an extension of our earlier framework of TranScaling. 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 (full or partial) 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.

We proposed partial TS in the context of the MPEG-4 FGST video-coding standard. In particular, we outlined a partial TS mechanism that maintains the original base-layer of the input FGST stream. The proposed scheme also maintains the SNR FGS pictures of the input scalable stream. Meanwhile, PSNR improvements are gained by using additional bitplanes from the SNR FGS pictures to predict the new FGST frames. Therefore, the FGST frames are the only frames that need to be fully re-encoded. Our simulation results showed PSNR improvements higher than 1.5 dB at high bitrates (higher than 1 Mbit/sec). These results were obtained based on re-using the same motion vectors coded within the original base-layer stream compressed at the source.

## References

[1] 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.

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

[3] “High Speed Physical Layer in the 5 GHz Band,” Draft Supplement to IEEE 802.11, 1999.

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

[5] 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.

[6] 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.

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

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

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

[10] H. Radha, et al. “Multimedia over Wireless,” Chapter in Advances in Multimedia: Systems, Standards, and Networks, Marcel Dekker, Inc., 2000.

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

[12] W. Tan, 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.

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

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

[15] M. van der Schaar and H. Radha, “A hybrid temporal-SNR Fine-Granular Scalability for Internet video,” IEEE Transactions on Circuits and Systems for Video Technology, March 2001.

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

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

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

[19] S. McCanne, V. Jackobson, and M. Vetterli, “Receiver-driven Layered Multi-cast,” Proc. SIGCOMM’96, Standford, CA, Aug. 1996, pp. 117-30.

[20] 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.