

## MULTIMEDIA TRANSCALING FOR THE WIRELESS INTERNET

Hayder Radha\* and J.R. Deller, Jr. <sup>▲†</sup>  
Michigan State University

### ABSTRACT

*TranScaling (TS), a generalization of non-scalable transcoding, is used to map a scalable video stream into one or more scalable streams for wireless internet transmission. TS does not result in the degradation of quality characteristic of existing techniques with similarly high levels of scalability. TS is presented and its performance illustrated by simulation studies involving the recently-developed MPEG-4 Fine-Granular-Scalable video coding scheme.*

### 1. Introduction

Scalable multimedia coding techniques are necessary to support diverse bandwidth requirements of the evolving Internet (e.g., analog modems, cable modems, DSL, LAN, etc.). 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 [6][7][8]. 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 Mbps to 54 Mbps [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 3<sup>rd</sup> 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 Mbps wireless LANs) will coexist with new generation systems for sometime to come. All of these developments indicate that the level of heterogeneity and the corresponding variation in available bandwidth could be increasing significantly as the Internet and wireless networks converge more and more into the future. In particular, if we consider the 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., [4][5][10][11]). 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. Consequently, these approaches can be used for multimedia multicast services over wireless networks.

However, a high level of scalability generally implies degradation in overall quality over the desired bandwidth. In light of the increase in heterogeneity over emerging wireless multimedia IP networks, there is a need for video coding and distribution solutions that provide high levels of scalability while maintaining quality [4]. One solution is the generation of multiple streams that cover different bandwidths. For example, a content provider can generate streams that cover 100-500, 500-1000, 1000-2000 kbps ranges, and so on. Although this solution may be viable under certain conditions, it is desirable to generate the smallest number of streams that covers the widest possible audience. Moreover, multicasting multiple scalable streams uses bandwidth inefficiently over the wired segment of the wireless IP network. (In the above example, a total bitrate of 3500 kbps is needed to transmit the three streams, while only 2000 kbps is needed by a scalable stream that covers the same bandwidth.)

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. With TS, multiple scalable streams covering different bandwidths are derived from another scalable stream. TS can be supported at gateways between the wired Internet and wireless/mobile access networks (e.g., at a proxy server adjunct to an access point of a wireless LAN). TS provides an efficient method for delivering good quality video over the wireless Internet while maintaining efficient utilization of network bandwidth. Therefore, different gateways of different wireless LANs and mobile networks can perform the desired TS operations that are suitable for their own local domains. Thus, higher-bandwidth LANs need not sacrifice video quality to coexist with legacy wireless LANs or other low-bitrate networks. Similarly, powerful clients (e.g., laptops and PCs) can receive high quality video even if there are other low-bitrate low-power devices served by the same wireless/mobile network. Moreover, when combined with embedded video-coding schemes and the basic



## 2.1 Attributes of TS-Based Systems

The TS framework has several key attributes:

1. Supporting TS at edge nodes (wireless LANs' and mobile networks' gateways) preserves the ability of the local networks to serve low-bandwidth, low-power devices (e.g., handheld devices). In the example of Figure 2, in addition to generating the scalable stream  $S_1$  (which has a higher BL than that of the input BL stream), the transcalar delivers the original BL stream to the low-bitrate devices.
2. In a TSM system, a transcalar can always revert to using the original (lower-quality) scalable video. This "fallback" feature significantly distinguishes TS from non-scalable transcoding. The "fallback" feature could be needed, for example, when a transcalar does not have enough processing power to execute the desired TS process(es). Therefore, unlike (non-scalable) transcoding-based services, TS provides a scalable framework for delivering higher-quality video.
3. Under a more general TSM framework, TS 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 can generate the desired sets of scalable streams for transmission of real-time compressed video. This general view of TSM provides a framework for distributing and scaling the TS processes throughout the multicast tree. Moreover, this general TSM framework leads to some optimization alternatives for the system. For example, depending on the bitrates determined by different edge servers (e.g., wired/wireless/mobile gateway servers), the system may have to balance computational complexity (due to the TS processes) against bandwidth efficiency (due to the possible transmission of multiple scalable streams with overlapping bitrate ranges).
4. Although TS has been described in the context of multicast services, on-demand unicast applications can also take advantage of TS. For example, a wireless or mobile gateway may perform TS on a high-demand video clip. Past experience with bandwidth variation allows the gateway server to anticipate channel requirements and generate the appropriate scalable stream through TS. This scalable stream can be stored for later viewing by the different devices served.
5. TS has its own limitations in improving video quality over an entire desired bandwidth. Nevertheless, the improvements provided by TS justify its use over a subset of the desired bandwidth range. This aspect of TS is further explained below.

Before proceeding, it is important to introduce some basic TS definitions. Two types of TS processes are defined: *down TS* (DTS) and *up TS* (UTS).

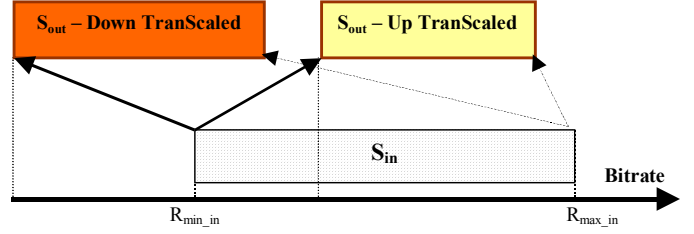


Figure 3: The distinction between DTS and UTS.

Suppose that the input scalable stream  $S_{in}$  to a transcalar covers a bandwidth range

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

and that the output transcaled stream has a range

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

Then, DTS occurs when  $R_{min\_out} < R_{min\_in}$  while UTS occurs when  $R_{min\_in} < R_{min\_out} < R_{max\_in}$ . The distinction between DTS and UTS is illustrated in Figure 3. DTS resembles traditional non-scalable transcoding in the sense that the bitrate of the output BL is lower than the bitrate of the input BL. This type of down conversion has been studied by many researchers<sup>1</sup>. However, up conversion has received little or no attention. Therefore, for the remainder of this paper we will focus on UTS. (Unless otherwise mentioned, we will use "UTS" and "TS" interchangeably.)

## 3. Simulation Results

To illustrate the level of quality improvements that TS can provide for wireless Internet multimedia applications, we present some simulation results of MPEG-4 FGS video TS.

Several video sequences were coded using the draft standard MPEG-4 FGS scheme. These sequences were then modified using the full TS architecture shown in Figure 4. Although a more elaborate TS-based algorithm can be used (e.g., refinement of motion vectors instead of a full re-computation of them), the main objective for adopting the transcalar shown in the figure is to illustrate the potential of video TS and to highlight some of its key advantages and limitations.

The improvements achieved by TS depend on several factors including the type of video sequence that is being transcaled. For example, 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 TS. On the other end, sequences that contain detailed textures and exhibit a high degree of correlation among successive frames could benefit from TS significantly. Overall, most sequences gained visible quality improvements from TS.

<sup>1</sup> The authors are unaware of any previous efforts to down convert a scalable stream into another scalable stream.

Another key factor is the ranges of bitrates used for the input and output streams. New wireless LANs (e.g., 802.11a or HiperLAN2) could have bitrates on the order of tens of Mbps. Although it is feasible that such high bitrates may be available to some devices at certain points in time, it is unreasonable to assume that a video sequence could be coded for sustained periods at such high bitrates. Moreover, most video sequences can be coded efficiently at bitrates below 10 Mbps. Consequently, the FGS sequences used in the simulations below were compressed at maximum bitrates (i.e.,  $R_{\max\_in}$ ) lower than 10 Mbps. For the BL bitrate  $R_{\min\_in}$ , we used different values in the range of few hundreds of kbps (e.g., between 200 and 500 kbps).

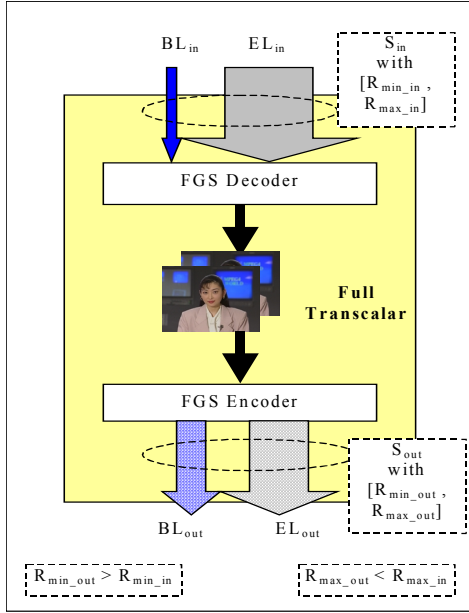


Figure 4: The full transcalar architecture used for generating the simulation results shown here.

First, we present the results of TS an MPEG 4 FGS stream (“Mobile”) that has been coded originally with  $R_{\min\_in}=250$  kbps and  $R_{\max\_in}=8$  Mbps. The transcalar used a new BL bitrate  $R_{\min\_out}=1$  Mbps. The peak signal-to-noise ratio (PSNR) performance of the two streams as functions of the bitrate is shown in Figure 5.

It is clear from the figure that there is a significant improvement in quality ( $\approx 4$  dB) in particular at bitrates close to the new BL rate of 1 Mbps. The figure also highlights that the improvements gained through TS are limited by the maximum performance of the input stream  $S_{in}$ . As the bitrate approaches the maximum input bitrate (8 Mbps), the performance of the transcaled stream saturates and gets closer to (and eventually degrades below) the performance of the original FGS stream  $S_{in}$ . Nevertheless, for the majority of the desired bitrate range (above 1 Mbps), the performance of the transcaled stream is significantly higher.

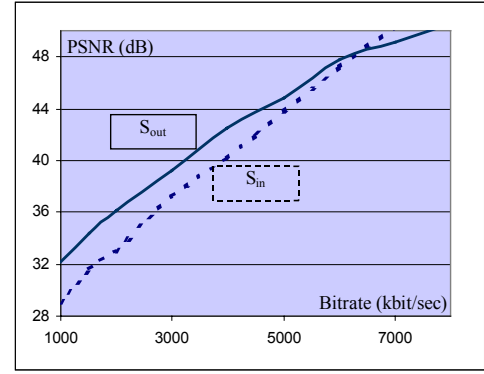


Figure 5: Performance of transcaling the “Mobile” sequence using an input stream  $S_{in}$  with a BL bitrate  $R_{\min\_in}=250$  kbps into a stream with a BL  $R_{\min\_out}=1$  Mbps.

In order to appreciate the improvements gained through TS, we can compare the performance of the transcaled stream with that of an “ideal FGS” stream. An “ideal FGS” stream is 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 BL of 1 Mbps. Figure 6 shows the comparison between the transcaled stream and an ideal FGS stream over the range 1 to 4 Mbps. The performances of the transcaled and ideal streams are virtually identical over this range.

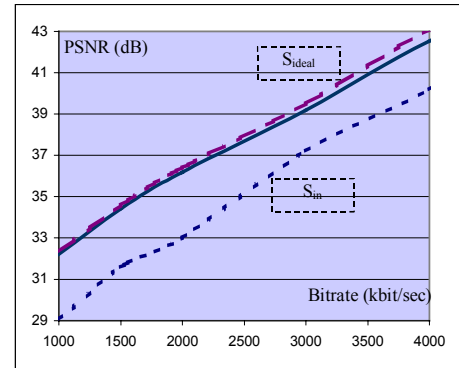


Figure 6: Comparing the performance of the “Mobile” transcaled stream (shown in Figure 5) with an “ideal FGS” stream. The performance of the transcaled stream is represented by the solid line.

As the number of bitrate ranges to be covered by the transcaled stream increases, one would expect that quality improvement with respect to the original FGS stream would diminish. We transcaled same original FGS (“Mobile”) stream with a new BL bitrate  $R_{\min\_out}=500$  kbps (i.e., lower than the 1 Mbps BL bitrate of the TS example described above). Figure 7 shows the PSNR performance of the input, transcaled, and “ideal” streams. Here, the PSNR improvement is as high as 2 dB around the new BL bitrate 500 kbps. These improvements are significant ( $>1$  dB) for the majority of the bandwidth range. As in the previous example, the transcaled stream 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. Therefore, TS provides rather significant improvements in video quality ( $> 1$  dB). 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. It is important to note that, depending on the application (e.g., unicast versus multicast), the gateway server may use *both* the newly generated (DTS) 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 clients with access bandwidths in 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 BL bitrate  $R_{min\_in}$ , can only use the down-transcaled bitstream.

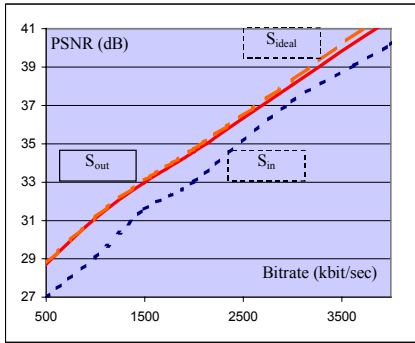


Figure 7: Performance of transcaling the “Mobile” sequence using an input stream  $S_{in}$  with a BL bitrate  $R_{min\_in}=250$  kbps into a stream with a BL  $R_{min\_out}=500$  kbps.

Similar results to those noted above for TS were observed for a wide range of sequences and bitrates.

The experiments above have focused on the performance of UTS which we have referred to throughout this section by simply “TS.” We now focus on some simulation results for DTS.

We employed the same full transcalar architecture shown in Figure 4. We also used the same Mobile sequence coded with MPEG-4 FGS and with a bitrate range  $R_{min\_in}=1$  Mbps to  $R_{max\_in}=8$  Mbps. Figure 8 illustrates the performance of the DTS operation for two bitstreams: One stream was generated by DTS the original FGS stream (with a BL of 1 Mbps) into a new scalable stream coded with a BL of  $R_{min\_out}=500$  kbps. The second stream was generated using a new BL  $R_{min\_out}=250$  kbps. As expected, the DTS operation degrades the overall performance of the scalable stream.

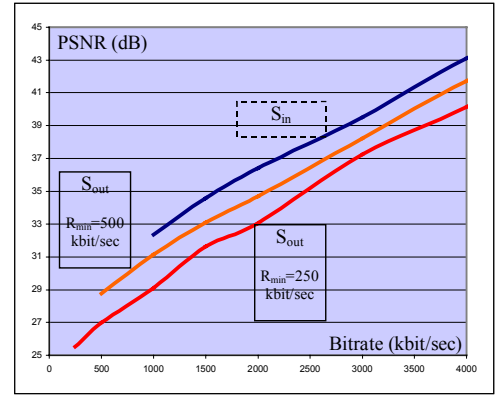


Figure 8: Performance of down-transcaling the “Mobile” sequence using an input stream  $S_{in}$  with a BL bitrate  $R_{min\_in}=1$  Mbps into two streams with BLs  $R_{min\_out}=500$  and  $250$  kbps.

As noted above, 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 conversion that are not available with non-scalable transcoding. For example, under DTS, one may elect to use both the BL and ELs, or the BL only, to perform the desired down-conversion. This strategy may be used, for example, to reduce the amount of processing power needed for the DTS operation. In this case, the transcalar has the option of performing only one decoding process (on the BL only versus decoding both the BL and ELs). However, using the BL only to generate a new scalable stream limits the range of bandwidth that can be covered by the new scalable stream with an acceptable quality. To clarify this point, Figure 9 shows the performance of TS 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  $[R_{min\_out}=250$  kbps,  $R_{min\_in}=500$  kbps]. Consequently, if the transcalar is capable of using both the original stream  $S_{in}$  and the 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.

In cases in which the transcalar must employ a single scalable stream to transmit content (e.g., multicast with a limited total bandwidth constraint), a transcalar can use the BL and any portion of the EL to generate the new down-transcaled scalable bitstream. The larger the portion of the EL used for DTS, the higher the quality of the resulting scalable video. Therefore, and since partial decoding of the EL represents a form of computational scalability, an FGS transcalar has the option of trading-off quality versus computational complexity when necessary. This observation applies to both UTS and DTS.

Finally, by examining Figure 9, 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  $S_{in}$ .

#### 4. Summary

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

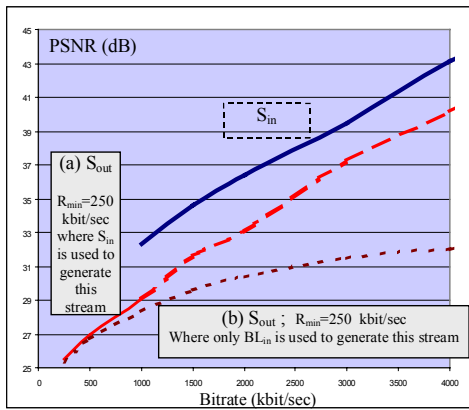


Figure 9: Performance of down-transcaling the “Mobile” sequence using an input stream  $S_{in}$  with a BL bitrate  $R_{min,in}=1$  Mbps. Here, two DTS operations are compared: (a) the whole input stream  $S_{in}$  (base+enhancement) is used ; (b) only the base-layer  $BL_{in}$  of  $S_{in}$  is used to generate the down-transcaled stream. In both cases, the new DTS stream has a BL bitrate  $R_{min,out}=250$  kbps.

#### 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] H. Radha, *et al.*, “The MPEG-4 FGS Video Coding Method for Multimedia Streaming over IP,” *IEEE Transactions on Multimedia*, March 2001.
- [5] D. Wu, *et al.*, “Scalable Video Coding and Transport over Broadband Wireless Networks,” *Proceedings of the IEEE*, January 2001.
- [6] 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.
- [7] V. Paxson, “End-to-End Internet Packet Dynamics,” *Proc. ACM SIG-COM*, vol. 27, no. 4, p. 13-52, Oct. '97.
- [8] 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.
- [9] S. McCanne, V. Jacobson, and M. Vetterli, “Receiver-driven Layered Multi-cast,” *Proc. SIGCOMM'96*, Stanford, CA, Aug. 1996, pp. 117-30.
- [10] 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.
- [11] 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.

\* Michigan State University; WAVES Laboratory; Department of Electrical & Computer Engineering / 2120 EB; East Lansing, MI 48824-1226. radha@msu.edu

▲ Speech Processing Laboratory at address above. deller@msu.edu

† J.D. was supported in part by the National Science Foundation under Cooperative Agreement No. IBIS-9817485. Opinions, findings, or recommendations expressed are those of the authors and do not necessarily reflect the views of the NSF.