

Improving Scalable Video Delivery with Cross-layer Design

Janne Vehkaperä, Kostas Pentikousis, Mikko Majanen, Jyrki Huusko, and Johannes Peltola
VTT Technical Research Centre of Finland, Kaitoväylä 1, FI-90571 Oulu, Finland
E-mail: firstname.lastname@vtt.fi

Abstract—The layered H.264/SVC video format, which is currently under standardization, allows scalable encoding and, thus, creates new opportunities for video delivery in a unified manner. However, video streaming over different wireless access technologies (from WLAN to WiMAX to 3G WWANs) remains challenging. After briefly introducing the issues at hand, we present a cross-layer architecture developed within the EU IST PHOENIX project, which jointly optimizes source and channel rates aiming at maximizing wireless link utilization and improving the user-perceived video quality. Our proposal capitalizes on recent advances in scalable, layered video encoding and provides a platform for future development.

1. Introduction

Delivering high quality video over wireless networks is a challenging task due to the inherent variability of the access medium, which is exacerbated by user mobility. Although current wireless data networks were not designed for wide-spread video streaming usage, multimedia applications are catching on, placing a burden on current protocols. One of the main goals in the EU-funded PHOENIX research project (2004-2006) is to develop solutions that exploit the available bandwidth on varied wireless links efficiently, and allow for optimized multimedia transmission over Internet Protocol (IP) wireless networks. In order to reach this goal, the PHOENIX consortium proposed to develop a scheme which offers the means to let the application world (source coding, ciphering) “talk” to the transmission world (medium access, channel coding) over an IPv6 protocol stack, and jointly optimize the end-to-end quality of communication via one or more wireless links. The main guidelines in this effort are:

- Develop innovative schemes to enable joint optimization over end-to-end wireless links. This includes the development of flexible channel coding and modulation schemes, the adaptation of existing source coding schemes with respect to their ability for JSCC/D and the development of new ones specifically optimized for this purpose.
- Establish efficient and adaptive optimization strategies that will jointly control the coding blocks and realistically take into account the system limitations and specifications, such as the use of ciphering, the presence of one or several wireless hops, and so on.
- Build a global network architecture based on joint optimization for future wireless systems. This objective includes the development of the transparent network communication approach, which will allow applying the optimization

will allow applying the optimization strategies in any kind of IP network.

This paper is organized as follows. Section II sets the background by introducing the layered/scalable video encoding standard H.264/SVC. Section III reviews recent research results, and Section IV brings out our proposed system architecture. Finally, Section V details our current simulation methodology and reports on our preliminary results.

2. Background

After the successful collaboration in developing H.264/AVC, experts from ISO/IEC MPEG and ITU-T formed a joint team to develop a new scalable video coding (SVC) standard as an extension to AVC. The SVC extension to AVC adds signal-to-noise ratio (SNR) and spatial scalability, expanding the use of AVC to several different application settings [1].

2.1. A scalable extension of the H.264/AVC video coding standard

The main idea behind the scalable extension of H.264/AVC is to take the block-based hybrid video coding scheme one step further and achieve spatio-temporal and signal-to-noise-ratio (SNR) scalability [2]. The term *scalability* in the video coding context means that physically meaningful video information can be recovered by decoding only a portion of the compressed bit stream. For example, one should be able to recover from the compressed bit stream a video with lower resolution than the original by decoding only the lowest spatial layer and discarding other spatial layers.

In SVC, scalability is achieved by taking advantage of the layered approach. The structure of the encoding depends on which kind of scalability is needed. For example, Figure 1 depicts the block diagram of an SVC encoder with two spatial layers, which contain additional SNR enhancement layers.

In each spatial layer hierarchical motion compensation and prediction is made. The redundancy between adjacent pictures and layers is based on inter- and intra-prediction techniques. After motion-compensated prediction, transform coding is applied using the same transformation techniques as in the H.264/AVC standard. SNR and quality scalability is achieved by coding the difference between transformed and not transformed slices using progressive coding. These progressively coded slices can then be truncated at any position within each slice thus improving the user-perceived visual quality proportional to the number of bits included in the truncated slice. Meanwhile, temporal scalability is achieved using hierarchical B pictures, which provide a predictive structure already included in H.264/AVC. Motion compensated temporal filtering can also be used

but it is, for the time being, included as a non-normative option only for achieving temporal scalability. An example of hierarchical coding structure for group of pictures (GOP) which length is eight pictures is illustrated in Figure 2. All of these scalability modes can be combined to achieve three-dimensional (spatial, temporal and SNR) scalability.

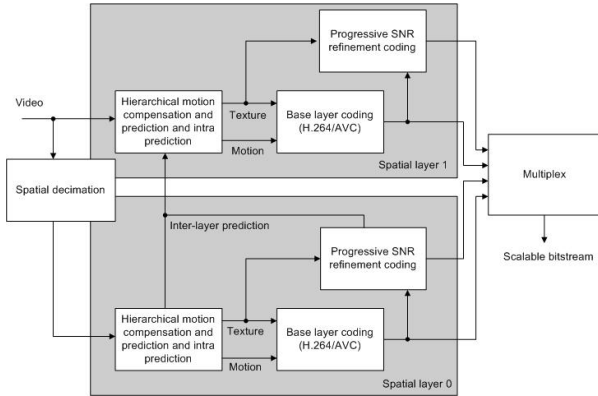


Figure 1: Block diagram for the H.264 scalable extension

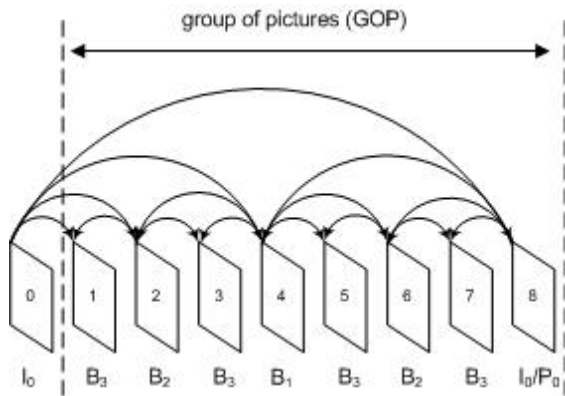


Figure 2: Hierarchical GOP structure

2.2. The dependency between layers in scalable video coding

Layers in scalable video coding are classified as a base layer and enhancement layer(s). In SVC, the base layer can be decoded using a standard H.264/AVC decoder. Information from lower layers is used to remove the redundancy between different layers. This increases coding efficiency but it also increases the importance of the lowest layers during decoding process and reduce error resiliency. If the base layer or one of the most important layers is lost, less important layers are useless because decoding them requires redundant data from the most important layers. The dependency between layers makes the prioritization of different layers during transmission suitable. The base layer also usually needs less transmission bandwidth than the enhancement layers, which is also quite important when allocating resources to different prioritization classes.

Based on the SVC layer prioritization suitability we propose a mechanism for adapting the video

transmission to rapidly changing wireless channel and network conditions. One of the main requirements for the architecture is to be general enough to work with different access networks from IEEE 802.11 (WiFi) and IEEE 802.16 (WiMAX) to 3GPP and UMTS.

3. Related work

Cross-layer optimization techniques received little attention from the research community until recently, partly due to the influence of Shannon’s joint source/channel coding theorem [3], from which one can conclude that “either reliable transmission is possible by separate source-channel coding or it is not possible at all” [4]. Of course, the success of a layered network stack, such as the TCP/IP suite, and its unanimous acceptance by academics and practitioners alike, has also effectively placed certain cross-layer designs in the same heap as “spaghetti” or monolithic design. Nevertheless, and despite rightfully cautionary notes on the topic [5], several researchers have attempted a more integrated approach in designing efficient wireless network communication systems aiming at, in particular, multimedia transmission, see [6],[7],[8] (and the references therein).

We concur with Khan *et al.* [9] that cross-layer design should be viewed as a complement, not an alternative, to layered design. Besides the efforts within the PHOENIX consortium, which is in the process of delivering a fully functional prototype that optimizes multimedia transmission in IP wireless networks, other researchers have recently published “proof of concept” implementations. For example, Haratcherev *et al.* [10] report on the effectiveness of their hybrid automatic rate control and argue for the effectiveness of a proposed cross-layer medium prediction mechanism.

Finally, Ksentini *et al.* present an architecture [11] similar to ours, but aim at IEEE 802.11e-based networks and do not take into consideration recent developments in scalable video. As mentioned earlier, we are striving for a solution that is generic enough, does not depend solely on a single data link layer, and can work with both wireless LANs and WANs.

4. Architecture

This section presents our approach for introducing cross-layer communication in the ISO-OSI stack, which allows for information exchange and joint optimization between different OSI layers. Figure 3 presents the proposed architecture, which implements a cross-layer stack between application layer source coding tools and the priority-based queuing module located at the data link layer of transmitting wireless links. The proposed design aims primarily at multimedia and video streaming optimization, and utilizes only the most important cross layer information exchange between layers in order to avoid “spaghetti”-like interlayer communication design. Our intention is to keep protocol interactions simple and implementation straightforward.

Our goal is to optimize video streaming performance in a network topology that consists of wireless hops in

addition to wired links. Optimization is enabled by performing *source rate adaptation* for the video stream in case of decreased wireless channel transmission capacity. *Rate adaptation* is implemented at the data link layer allowing rapid reaction to channel state changes of the corresponding wireless channel. Adaptation is based on prioritization the transmission of different H.264 scalable extension video stream layers in the Multimedia Adaptation Layer (MAL) so that the most important base layer is transmitted first, and enhancement layers are then transmitted in decreasing order of importance.

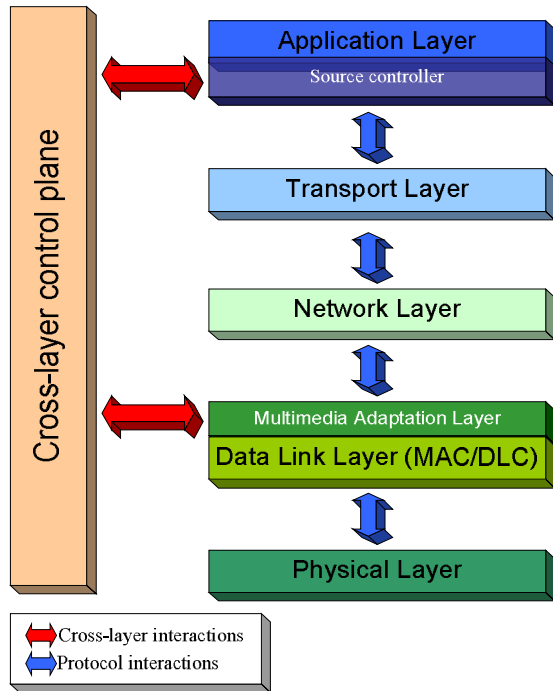


Figure 3: Protocol stack for video streaming optimization

Cross-layer communication is needed to obtain the required source sensitivity information (SSI) which is used to perform prioritization. In addition, channel state information (CSI) and network state information (NSI) are used to aid the scheduling procedure. The information transfers between the layers can be classified as either external or internal communications. The former comprises exchanging information through the network, while the latter involves the communication between different OSI layers inside the same device. The design aspects of cross layer communication include taking into account the introduced overheads and delays, as well as other functional and performance aspects, such as, synchronization issues. The video codec at the application layer provides the SSI-information by *marking* the video stream layers according to their importance for the decoded image quality. This marking information is given to the transport protocol along with the actual payload.

The packet classification based on SSI-information utilization, can be performed at the network (IP) level using Differentiated Services (DiffServ) [12],[13].

We capitalize on the availability of standards-compliant packet classification in wired and wireless networks connected with routers or layer 3 switches. In DiffServ, packets are classified and marked in order to receive a particular per-hop behavior (PHB) on nodes along their path. Different PHB are mapped using specific queue management and packet scheduling mechanisms. Sophisticated classification, marking, policing, and shaping operations need only be implemented at network boundaries including the network edges (first-hop router or source host) and administrative boundaries. Network resources are allocated to traffic streams by service provisioning policies, which govern how traffic is marked and conditioned upon entry to a DiffServ-capable network, and how that traffic is forwarded within that network. There is no need for per-flow state and signaling at every hop, which makes the DiffServ architecture scalable.

The DiffServ field, which consists of the first six bits of the IPv4 header ToS, or IPv6 header Traffic Class octet, is used for packet marking, which should map into certain PHBs. Some of the code points are assigned by IANA, others are left for experimental and local use.

For wireless networks, including also layer 2 access points and switches, which do not support layer 3 routing capabilities a lower level packet classification can be used in addition to the IP level classification.

The proposed multimedia adaptation mechanism for the scalable extension of H.264 is illustrated in Figure 4. On the transmitter side, MAL consists of a packet classifier, which classifies the incoming packets from IP-layer to queues based on frame type and priority information provided by SSI; it also includes transmit and receive buffers, and the scheduler engine.

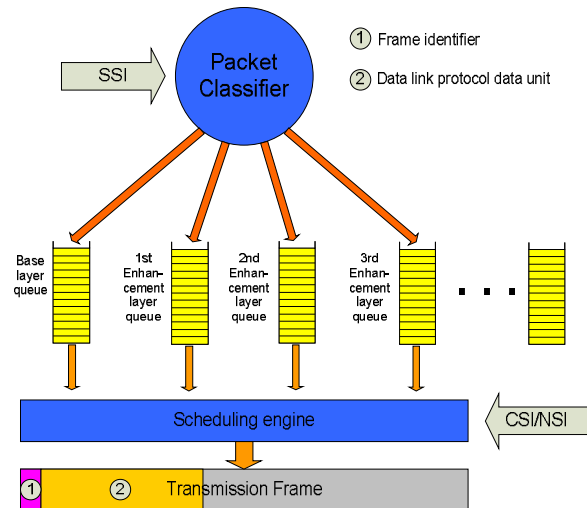


Figure 4: Transmitter side MAL architecture for scalable extension of H.264

The scheduler engine takes advantage of CSI and NSI information in transmission frame formation. Depending on the current channel and network state the scheduling engine forms the data link frame from the available base layer and enhancement layers video frames, which can be transmitted with a certain level of assurance over the

wireless network. The scheduling engine also inserts a frame identifier to the data link frame, which is further utilized at the receiver side to identify the incoming data frame format.

The proposed MAL architecture is required for fast adaptation to rapid channel and network changes. The MAL solution also provides a medium access- and physical layer-independent solution for efficient multimedia transmission. The architecture can be easily deployed e.g. with WiFi, WiMAX, and UMTS.

5. Performance Evaluation

We have studied the efficiency of the traffic prioritization scheme with MPEG-4 Fine Granular Scalability (FGS) coding algorithm in [14], where a prioritization of the enhancement layers of the scalable MPEG-4 bit stream was simulated. Different packet loss rates (PLR) for different enhancement layers were used in prioritized scheme and same PLR for every layer in non-prioritized scheme. Preliminary simulation results are illustrated in Figure 5. It can be seen from the figures that improvement on image quality can be achieved by using prioritization and the improvement is greater when PLR is larger. However, the coding efficiency of MPEG-4 FGS is quite poor which also affects the simulation results. The coding efficiency for SVC has improved and together with combined spatio-temporal and SNR scalability, it will provide better suitability to prioritized transmission scheme.

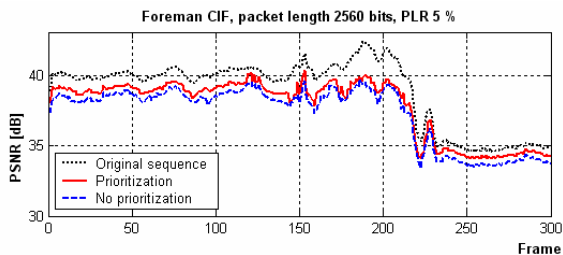


Figure 5: MPEG-4 FGS with and without prioritization when PLR=5%.

Although these results are promising, we would like to simulate a larger and, perhaps, more realistic set of network topologies while being able to compare the system performance using different video clips and the most recent video encoding software. We, thus, have developed the simulation methodology described next.

5.1. Simulation methodology

Our performance evaluation study aims at investigating the potential for improving the end-user experience using cross-layer optimizations. In particular, we want to compare the performance of our cross-layer solution with more traditional video transmission methods, and determine if it is more suitable for transmitting layered streaming video.

We are interested in the following scenario: an event is broadcasted live and encoded in real time in a scalable, layered video format, for example H.264/SVC. The server transmits the scalable video to a number of recipients, which may use devices with different display capabilities and network access. The frames generated by the encoding process are subsequently packetized forming the video stream which is transmitted over an arbitrary network topology. The network includes wireless and wired links, and may support traffic prioritization, such as Differentiated Services, at the network layer, and priority-based queuing at the data link layer. At the receiving end, the incoming stream is decoded in real time and the recipient monitors the quality of the received video feed.

As mentioned above, layered video encoding is typically very efficient in terms of the total amount of traffic injected in the network, achieving high compression rates. However, this is realized by employing computationally expensive algorithms, in particular when producing scalable video. For example, encoding a 10-second video clip in CIF resolution (352×288 pixels) with 2 spatial, 5 temporal and 3 quality layers, requires several hours on a Pentium 4-class PC using the most current reference implementation of the H.264/SVC standard. Clearly, the scenario we are interested in cannot be realized today, and as such, we resort to simulation [15] in order to qualitatively evaluate the cross-layer optimizations.

The simulation methodology we use attempts to take advantage of the state of the art components in both scalable video encoding and network simulation. As illustrated in Figure 6, we first use a real video clip and encode it using a prototype H.264/SVC encoder. Note that the SVC extension is still under standardization and all encoder implementations are still in the prototype phase of development. In addition to the real encoding, we extract the packet stream that would have been generated if the server transmitted the video as described in the scenario. We make sure that the packet stream is compatible with different network technologies, obey maximum transmission unit (MTU) limitations, calculate headers, and so on. That is, we do not simply calculate GOP sizes, but we fragment the frames generating actual packet payloads.

The second stage involves simulations with different topologies and traffic scenarios in ns-2. The H.264/SVC encoder (D1) is aware of video semantics and can mark each packet based on, for example, whether it belongs to a basic or an enhancement layer. Clearly, not all video packets are created equal: packets from a basic layer are more important than packets from enhancement layer(s) and should, thus, receive preferential treatment, such as prioritized forwarding or reliable delivery, while in transit. In our studies, this is achieved by employing either a DiffServ network or a data link layer traffic differentiation scheme. The final stage of the methodology involves the analysis of the simulation trace, and the calculation of performance metrics

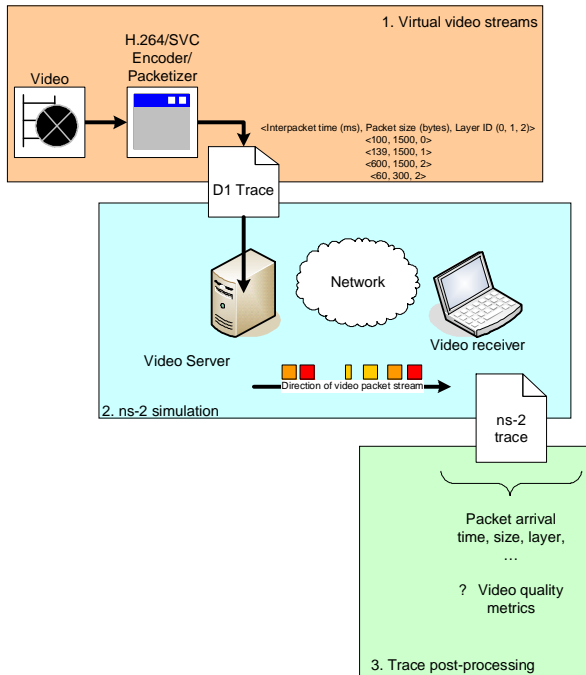


Figure 6: Simulation methodology

5.2. Results

We use *ns-2* to explore the simple scenario involving two mobile nodes connected via IEEE 802.11b. The nodes are located 120 m apart on a planar field and use Dynamic Source Routing (DSR) [16] to discover each other and maintain connectivity. The first node transmits a 10-second video packet stream corresponding to the H.264/SVC encoding of the well-known Foreman clip (generated as explained above). The clip has a resolution of 352 by 288 pixels and is encoded at 125 kB/s with the basic and five enhancement layers. The sending node has a large interface buffer, $L=250$ packets, which is well above the bandwidth-delay product in order to avoid congestion losses. We configure the buffer either as a flat drop-tail queue without packet differentiation, or as a priority queue (PriQ) that classifies packets into one of five buffers ($l_i=50$, $i=1\dots 5$, as depicted in Figure 4)

At $t=10s$ the sending node starts the transmission of the video file. At the same time, the nodes start to move away from each other in opposite directions. Connectivity is lost at about $t=11.5s$. The nodes reverse course at $t=14s$ and are again within communication range at $t=16.5s$.

We explore four different transport methods. First, we use a straightforward FTP transfer of the entire encoded video in H.264/SVC. Second, the same H.264/SVC packet stream is transmitted over UDP as generated by the encoder; packets are not differentiated based on source coding (layer) information. Third, we use the same stream but mark each packet at the source using five different DiffServ codepoints (DSCP). The basic layer and the next three enhancement layers are associated with DSCP “0” to “3”, respectively; the last two enhancements layers are both mapped to DSCP “4”. In this case

(“SVC/PriQ”), the network does support traffic prioritization (PriQ), but does not provide any notifications when connectivity is lost or packets are being dropped. Finally, we add MAL to the previous scenario (“SVC/MAL”) and repeat the simulation. In this last case, the sender is programmed to halve its transmission rate when it receives such notification, which only traverse the sender’s network stack (*internal* communication, see Section 4). Note that we do not yet have a generalized mechanism to control the video bitstream adaptation.

The FTP sender uses TCP NewReno and the receiver employs the delayed acknowledgements algorithm [17]. TCP is unaware of layer information and cannot prioritize frame transmissions based on application-level criteria; it is included in our presentation for completeness. Figure 7 presents the throughput calculated at 100ms intervals at the receiver side, while Figure 8 presents the timeline of packet arrivals and drops. Note that this particular scenario is quite favorable for TCP: the sender is not application-limited (the entire video clip is available for download at $t=0s$), and no random or congestion-related losses occur. TCP takes advantage of all available network capacity and, due to its greedy nature, over-shoots well above the average encoding data rate. During the period of lost connectivity, the TCP sender times out five times (between $t=11.7s$ and $t=17.7s$; see lower part of Figure 8). TCP resumes successful transmission a bit after the last timeout, but a valuable period of more than 1.2s has been lost. Nevertheless, TCP manages to transfer a significant part of the video before connectivity is lost, and (by transmitting at full throttle) finishes the transfer before $t=20s$.

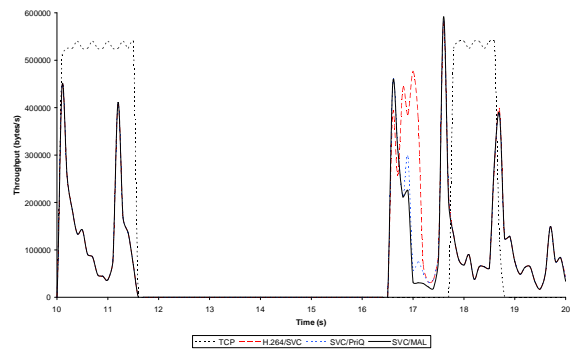


Figure 7: Video transfer throughput

Prior to the lost connectivity event ($t=10\dots 11.5s$), the three SVC “variants” proceed streaming in lock step (Figures 7 & 8). Once connectivity is lost, the buffer at the sender’s side starts to fill up and then starts to overflow. In the case of “H.264/SVC” over drop-tail this occurs at $t=13.28s$. In the case of SVC/PriQ and SVC/MAL packets start to being dropped more than a second earlier. However, as highlighted in Figure 8, in the latter case the dropped packets belong to lower priority (enhancement) layers. In particular, the majority of dropped packets in the case of SVC/PriQ and SVC/MAL belong to layers 3, 4 and 5; no packets from the basic layer or the first enhancement layer are dropped.

In contrast, the timeline indicates that SVC over-non-priority queues may be of little value as it is bound to lose packets for the basic layer in cases of lost connectivity. Figure 9 zooms in the timeline of packet arrivals and packet drops for $t=16\dots17s$. The lower part reinforces the point made about layer prioritization: without it, packets from the basic layer are dropped, while others from the fourth and fifth enhancement layers are being delivered over the air interface. This is very critical because it is challenging the SVC decoder to recover from packet losses that belong to the basic layer, and may be forced to discard any following packets from the same frame. This simple scenario indicates that in order for SVC to deliver an improved user experience the network must employ some kind of traffic prioritization.

However, simple traffic prioritization is not sufficient. For example in the case of SVC/PriQ, once connectivity is restored, although packets from the basic layer are being received, packets from enhancement layers continue to be transmitted by the sender thereby overflowing the interface buffer. Source rate adaptation is clearly called upon. The upper part of Figure 8 & 9 show that by halving the source rate less stress is placed on the queues for the lowest priority packets. These savings translate into a smoother transition into full throttle once connectivity is restored (Figure 7, $t>16s$).

6. Summary and Future Work

We introduced an architecture, which centers on our Multimedia Adaptation Layer (MAL) and aims at delivering improved streaming video performance over wireless networks. MAL builds upon recent advances in scalable, layered video standards and employs standards-based network and MAC-layer traffic prioritization mechanisms. We presented our methodology for evaluating MAL (and other similar approaches) based on cutting-edge, prototypical H.264/SVC video encoding software, and used the most-widely used network simulator to evaluate SVC/MAL in a simple scenario, showcasing the advantages of using real traces in evaluating video streaming adaptation mechanisms.

We are actively working on developing source rate adaptation algorithms based on peer-, network-, and media access notifications. Of main interest is the development of adaptation mechanisms that improve image quality. We are looking into more complex simulation scenarios, which involve more sources, transmission-, and congestion-related losses, and developing metrics that capture the user-perceived quality in a more formalized way.

ACKNOWLEDGEMENT

This work was carried out in the PHOENIX project, which was partially funded by the European Commission within the European Union Sixth Framework Programme and Information Society

Technologies. The authors would like to thank their colleagues, who have participated in this project. They provided valuable work contributions for the development of the PHOENIX system.

RERERENCES

- [1] Joint Scalable Video Model (JSVM) 4.0 Reference Encoding Algorithm Description, ISO/IEC JTC 1/SC 29/WG 11, N7556, October 2005, Nice, France.
- [2] H. Schwarz, D. Marpe, and T. Wiegand: "Basic Concepts for Supporting Spatial and SNR Scalability in the Scalable H.264/MPEG4-AVC Extension", *Proceedings of IWSSIP 2005*, Chalkida, Greece, September 22-24, 2005.
- [3] C. E. Shannon, "A mathematical theory of communication", *Bell System Technical Journal*, Vol. 27, pp. 379-423 and 623-656, July and October 1948.
- [4] S. Vembu; S. Verdu, Y. Steinberg, "The source-channel separation theorem revisited", *IEEE Transactions on Information Theory*, Vol. 41, No. 1, pp.44-54, January 1995.
- [5] V. Kawadia and P.R. Kumar, "A cautionary perspective on cross-layer design", *IEEE Wireless Communications*, Vol. 12, No. 1, pp. 3-11, February 2005.
- [6] W. Stark, Hua Wang, A. Worthen, S. Lafortune, D. Teneketzis, "Low-energy wireless communication network design", *IEEE Wireless Communications*, Vol. 9, No. 4, pp. 60-72, August 2002.
- [7] S. Shakkottai, T.S. Rappaport, P.C. Karlsson, "Cross-layer design for wireless networks", *Communications Magazine, IEEE*, Vol. 41, No. 10, pp. 74-80, October 2003.
- [8] M. van Der Schaar and N. Sai Shankar, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms", *IEEE Wireless Communications*, Vol. 12, No. 4, pp. 50-58, August 2005
- [9] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, W. Kellerer, "Application-driven cross-layer optimization for video streaming over wireless networks", *IEEE Communications Magazine*, Vol. 44, No. 1, pp. 122-130, January 2006.
- [10] I. Haratcherev, J. Taal, K. Langendoen, R. Lagendijk, H. Sips, "Optimized video streaming over 802.11 by cross-layer signaling", *IEEE Communications Magazine*, Vol. 44, No. 1, pp. 115-121, January 2006.
- [11] A. Ksentini, M. Naimi, A. Gueroui, "Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture", *IEEE Communications Magazine*, Vol. 44, No. 1, pp. 107-114, January 2006.
- [12] K. Nichols, S. Blake, F. Baker, and D. Black: *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, RFC 2474, December 1998.
- [13] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Services*, RFC 2475, December 1998.
- [14] J. Vehkaperä and J. Peltola, "Optimized Decoding Scheme for Erroneous MPEG-4 FGS Bitstream", *Proceedings of IEEE ISCAS 2005*, Kobe, Japan, May 23-26, 2005.
- [15] P. Seeling, M. Reisslein, and B. Kulapala, "Network Performance Evaluation Using Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A Tutorial", *IEEE Communications Surveys*

& Tutorials, Vol. 6, No. 2, pp. 58-78, Third Quarter 2004.

[16] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", in *Ad Hoc*

Networking, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.

[17] S. Floyd, T. Henderson, and T. Gurtov, *The NewReno Modification to TCP's Fast Recovery Algorithm*, RFC 3782, April 2004.

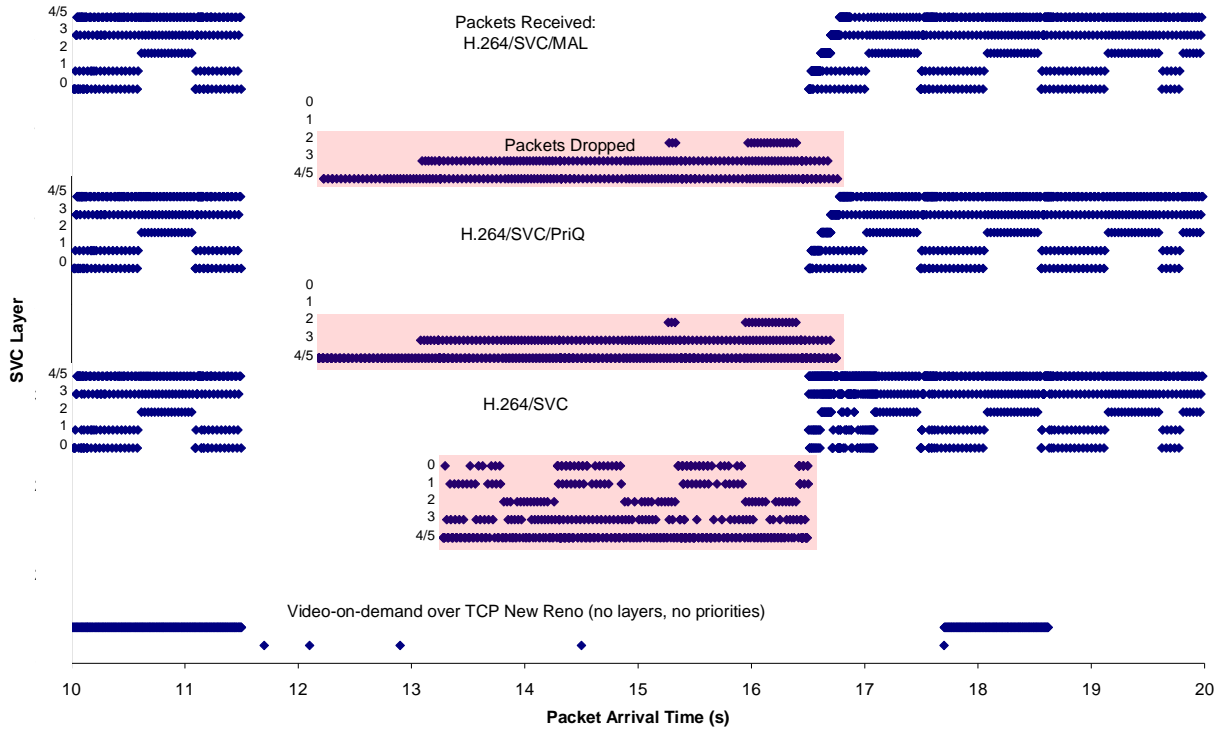


Figure 8: Timeline of packet arrivals

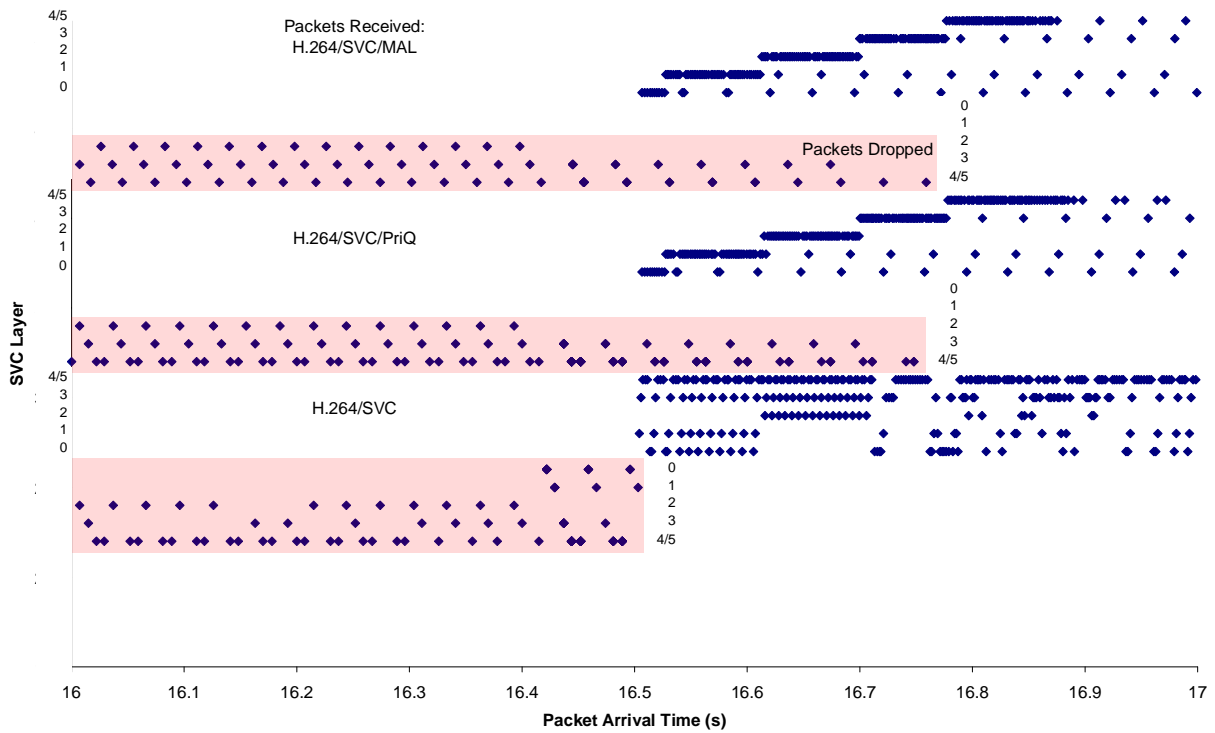


Figure 9: Timeline of packet arrivals (zoom in)