ON THE DESIGN OF A SCALABLE MULTIMEDIA STREAMING SYSTEM BASED ON RECEIVER-DRIVEN FLOW AND CONGESTION AWARENESS

Iñigo Urteaga, Iraide Unanue, Javier Del Ser
TECNALIA-TELECOM, Parque Tecnologico, Ed. 202, 48170 Zamudio, Spain

Pedro Sanchez, Aitor Rodriguez
Intelligent Transport Systems and Security, IKUSI-Angel Iglesias, S.A., Paseo Miramon 170, 20009 San Sebastian, Spain

Keywords: Streaming, Scalable multimedia content, Congestion control, Flow control.

Abstract: In this position paper we present the design of an end-to-end scalable content streaming system that optimizes the quality of experience of the end-user by allowing each client to retrieve a customized multimedia stream, based on both network and client states. By taking advantage of multimedia scalability, our proposed receiver-driven architecture performs a multilayered streaming, where each client is responsible for controlling the number of multimedia layers it demands from the server. Furthermore, the streaming system proposed herein implements both congestion and flow control mechanisms, which are also delegated to the receiver. In order to properly address both network and client states and restrictions, a set of specific metrics (*Buffer State*, *Interarrival Jitter* and *Loss Event Rate*) are utilized, which have been specifically designed to match the miscellaneous characteristics of heterogeneous networks and end devices. Built upon such metrics, we present a decision algorithm that jointly performs congestion and flow control, while maximizing inter-session fairness and end-user quality of experience. The proposed architecture combines different standard protocols while guaranteeing independence between components of the streaming system.

1 INTRODUCTION

Multimedia streaming has lately gained momentum within both industry and academia in light of the forthcoming redefinition of Internet, mainly motivated by a wide variety of Internet-based applications envisioned to become vastly demanded in the following years. As to mention, in multi-point video conferencing each of a number of endpoints require personalized versions of a given content, whereas in video-on-demand the features of the multimedia content delivered to each client are established based on service quality and fees, usable bandwidth, etc.

In this context, as opposed to conventional broadcast technologies such as terrestrial or cable television, IP (*Internet Protocol*) networks are inherently heterogeneous in their underlying communication means (i.e. usually composed of combinations of wired and wireless links with distinct associated communication protocols). This heterogeneity gets even more involved if one notices that the state, traffic and characteristics of IP networks usually change dynamically in time. Besides, the rapidly growing portable device market has introduced a huge variety of streaming receivers. Based on this threefold rationale, scalable multimedia content is called to attain wide acceptance in the near future, as it provides high adaptability to all the above scenarios. As research on scalable content advances with the pioneering Scalable (H.264/SVC) and the Multiview (H.264/MVC) Video Codecs, scalable content based streaming applications will become broadly adopted.

Research on scalable multimedia streaming has so far gravitated on the use of MANE (*Media Aware Network Element*) entities which fundamentally are additional intermediate nodes used for manipulating and customizing streaming sessions, in clear contrast

This work was supported by the Spanish *Centro para el Desarrollo Tecnologico e Industrial* (CDTI) through the TELMAX project (ref. CEN20071036).

Urteaga I., Unanue I., Del Ser J., Sanchez P. and Rodriguez A. (2010).
ON THE DESIGN OF A SCALABLE MULTIMEDIA STREAMING SYSTEM BASED ON RECEIVER-DRIVEN FLOW AND CONGESTION AWARENESS.
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 39-45
DOI: 10.5220/0002987400390045

to client-server based end-to-end services. This approach has been thoroughly analyzed and studied in the literature. For instance, in (ASTRALS, 2010; Schierl et al., 2007; Renzi et al., 2008) *N* RTP (*Real Time Protocol*) sessions transmitted by the server are fused by the MANE into a single RTP flow for each client according to network conditions. Still, other contributions (Liebl et al., 2006; Tizon and Pesquet-Popescu, 2008) propose to use MANEs to perform an optimized packet scheduling and radio resource sharing over the last wireless hop of a network by mapping scalable content layer dependencies to flow priorities. Unfortunately, the aforementioned use of MANEs presents several disadvantages: 1) the insertion of an intermediate media-aware device into the streaming scenario, and 2) the need for modifying both RTP and RTCP (*Real Time Control Protocol*) packets to adapt them to the customized content. Deploying MANEs into the streaming system requires to know beforehand where the final clients are located.

Since the success of streaming services is only achievable if respecting the self-regulatory nature of transmissions within the Internet, it is mandatory to avoid either overloading or under-utilizing network resources. This justifies the need for providing congestion control techniques. Several congestion control mechanisms have been presented for streaming applications in the literature, e.g. see (Feamster et al., 2001; Ma and Ooi, 2007; Mujica-V. et al., 2004; Papadimitriou and Tsaoussidis, 2007) and references therein. However, most of them are source-based (i.e. the transmitting node is in charge for implementing congestion-aware techniques), which requires active probing, information piggybacking or acknowledgement mechanisms. Several drawbacks can be inferred from the application of source-based congestion protocols to heterogeneous IP networks. On one hand, demanding feedback from the client implies an overhead in terms of both processing complexity at the client and bandwidth over-utilization. On the other hand, in networks where both wired and wireless technologies coexist links with highly asymmetric characteristics are likely to appear. Therefore, taking the two-way path into account is not desirable in heterogeneous networks, since asymmetric characteristics of paths cannot be reliably estimated at the server side. Thus, worst conditions prevail in two-way path congestion control, as it is not possible to distinguish whether the problem arises in the uplink or the downlink. The heterogeneous nature of future networks implies the need for new receiver-driven congestion control mechanisms. We here propose to discard considering two-way paths and, in-contrast, to only account for the down-link state in our receiver-driven network congestion.

This position paper outlines the key design principles of a receiver-driven streaming system based on scalable multimedia content. Both the management of the multimedia content and the congestion and flow control logic are placed on the client, hence minimizing the computational complexity of the server. In our approach, not only streaming standards are kept unmodified, but we also profit from the information already embedded by such protocols. Furthermore, each constituent component of our proposed architecture is independent from each other. Another novel contribution of our work hinges on the metrics utilized for the congestion and flow control mechanisms, for which we introduce a novel LER (*Loss Event Rate*) metric which is proven to offer enhanced stability to bursty losses with respect to conventional packet loss rate metrics.

The remainder of this manuscript is organized as follows: first Section 2 introduces the reader to the fundamentals of scalable multimedia streaming[1], whereas Section 3 presents our novel receiver-driven end-to-end streaming system proposal for distributing scalable media content. Finally, concluding remarks and future research lines are drawn in Section 4.

2 SCALABLE STREAMING

Due to the heterogeneity of the actual networks and the proliferation of a wide range of final devices, it is essential to adapt the streaming content for each specific context. Early approaches have been based on storing a number of replicas of the same original content or, alternately, on transcoding the original content in a case-by-case basis. Recently, research efforts have been conducted towards the generation of inherently scalable multimedia content as a means to provide different versions of the same multimedia content, without resorting to multiple successive transcoding tasks. Consequently, processing redundancy and storage occupancy of the encoded multimedia content are minimized.

This growing interest in scalable codification has led to several research lines: the SVC (H.264/SVC, 2009) and MVC (H.264/MVC, 2009) extensions of the so-called *Advanced Video Codec* (H.264/AVC). The H.264/SVC standard attains high compression rates while simultaneously combining three scalability levels into a single encoded bitstream, namely spatial (resolution), temporal (frame rate) and *signal-to-noise ratio* (SNR) scalability (fidelity). H.264/SVC

[1]The authors recommend the reader to skip Section 2 if familiar with the concepts tackled therein.

encodes a given video content to a layered structure consisting of a base layer (comprising the lower levels of each of the mentioned scalabilities) and a number of enhancement layers. The goal of the enhancement layers is to progressively refine (in terms of each aforementioned scalabilities) the base layer so as to obtain better end-user Quality of Experience (QoE) degrees. The H.264/MVC video codec (currently under development) is also based on a layered bitstream structure. However, it additionally provides multiple views of the same scene which allows rendering 3D perspectives. Thanks to the scalability of this codec, it is possible to choose a specific viewpoint of a scene, while keeping high encoding efficiency through inter-view predictions.

Additionally, the Internet Streaming Media Alliance (ISMA, 2010) promotes the use of certain standard protocols for streaming applications: *Real Time Streaming Protocol* (RTSP), *Session Description Protocol* (SDP), *Real Time Protocol* (RTP) and *Real Time Control Protocol* (RTCP). RTSP is responsible for establishing and controlling the streaming sessions in real time. SDP describes the whole streaming session, as it characterizes the content and the streaming session itself. RTP delivers the multimedia content to destination in combination with RTCP, which communicates statistics and control information for each RTP session. Finally, it is important to recall that standardization organisms periodically evaluate and update the set of recommended standards to meet the requirements and constraints of newly designed multimedia codecs.

3 SYSTEM DESIGN

A block diagram of the proposed end-to-end streaming system design is depicted in Figure 1. As shown, the first processing steps of the streaming system consist basically of multimedia content encoding and encapsulation. Without loss of generality we have hereby adopted the H.264/SVC codec for ensuring scalability at the encoding process. The only requirement imposed by our design is that real-time layer switching must be supported during the decoding process. As for the encapsulation, the MP4 file format has been selected due to 1) the H.264/SVC specific extension (AVC, 2008) included in such standard, and 2) the so-called *hint tracks*. *Hint tracks* enable a media-unaware streaming server by indicating how to perform the streaming disregarding the content itself, thus alleviating the server from the computational burden derived from analyzing the streaming peculiarities of each specific content.

Our system performs a multilayered streaming where the M layers conforming the scalable content are mapped to N RTP sessions $\{RTP_s^i\}_{i=0}^{N-1}$. Observe that even if the end-user appreciates a single multimedia stream at reproduction, the content is received in $n \leq N$ parallel RTP sessions, where n denotes the actual number of demanded RTP_s^i by a given client. As defined in (RTP, 2010), the mapping between SVC layers and RTP_s^i can be done by following distinct criteria. In our system the mapping rule is provided to the server through *hint tracks* in the encapsulation.

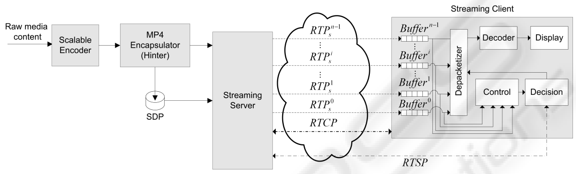

Figure 1: Block diagram of the proposed end-to-end scalable content streaming system.

Thanks to multilayered streaming, the characteristics of received multimedia content are dependent on the actual number of transmitted RTP_s^i and therefore, diverse end-user requirements can be easily met. Our system design allows each client to select the subset of RTP_s^i that better fulfills its needs, as described in Subsection 3.1. Furthermore, the client performs a receiver-driven congestion and flow control by adapting n, based on both network and client conditions (Subsection 3.2). The justification for this receiver-driven approach is to avoid complex and highly-loaded servers by balancing the computational load between clients. Besides, piggybacking otherwise necessary client information and network state parameters to the server is circumvented. Finally, it should be clear that sharp and frequent transitions among video layers are extremely displeasing for the QoE. In such situation a smoother video of reduced bit rate is then preferred rather than an inconsistent and jerky high quality video. In Subsection 3.2.2 we outline several criteria to achieve smooth multimedia reproduction aimed at maintaining a satisfactory QoE.

3.1 Content Streaming Procedure

In our proposal the client is the unique responsible for (throughout the whole streaming session) dynamically controlling n, i.e. the number of scalable multimedia layers to be received. We remark that our system follows IETF's specifications concerning scalable content over streaming protocols (RTP, 2010). The presented streaming process begins with the client demanding information to the server about some specific multimedia content by sending a RTSP DESCRIBE request. The server responds to the client with the SDP description of the required content over RTSP. This SDP description contains all the information regarding that particular multimedia streaming session: number and characteristics of each RTP_s^i that conform the streaming session, dependencies between different RTP_s^i, and so on. At this point the client is capable of selecting a subset of RTP_s^i, depending on its processing and memory capabilities. Once this is set, the client triggers the streaming process by sending the RTSP SETUP and RTSP PLAY commands to the server. Throughout the whole streaming session, the client is able to cancel or demand (using RTSP commands) each RTP_s^i described in the SDP, as long as dependencies among scalable content layers are met. Once the content is received in the client, the RTP packets corresponding to different RTP_s^i are merged and ordered in a single bitstream, which is next depacketized and decoded. Finally, the multimedia content is displayed.

The determination of the optimal number of scalable layers and their mapping to RTP_s^i sessions is both application and content dependent, however our system is generically designed (independent from specific mappings). Hence, we assume that RTP_s^i sessions (and, consequently, scalable layers) are correctly ordered beforehand in the encapsulation process, so the user simply needs to comply with the information provided by the SDP. In order to maximize the QoE, we propose a soft and stable layer switching mechanism further detailed in Subsection 3.2.2.

3.2 Flow and Congestion Control

In IP networks, several traffic types and flows compete for the available scarce resources, which requires avoiding either traffic overload or the underutilization of the network resources. Congestion can be induced by both attempting to oversubscribe the processing capabilities of intermediate nodes or by over-demanding network link capabilities. This rationale, along with the diverse memory and processing characteristics presented by end clients, motivates the need for appropriate congestion and flow control mechanisms in streaming systems. However, multilayered streaming imposes several considerations to be taken into account. First, both congestion and flow control must be based not only on a single flow, but on several parallel RTP_s^i. Second, each RTP_s^i has a fixed transmission bit rate enforced by the scalable content requirements. Thus, RTP_s^i by themselves cannot expand nor reduce their bandwidth usage.

Consequently, congestion and flow control for multilayered streaming can only be accomplished based on discrete bitrate intervals. This certainly poses several design challenges gravitating on the tradeoff between reactivity to network and client dynamic characteristics (which justifies relatively short control periods) and the QoE degradation due to layer switching. We intend to balance this tradeoff by benefiting from the specific features of scalable content streaming, which gives rise to a novel receiver-driven congestion and flow control mechanism.

3.2.1 Metrics

The proposed metrics are restricted to the information available at the client side. Therefore, procedures such as message piggybacking or probing (e.g. for bandwidth estimation) are discarded. Network state is inferred by extracting information from the received RTP_s^i packets, while reception buffers are monitored for estimating the client state. The following metrics will be sampled and computed for each received RTP_s^i ($i \in \{0,\ldots,n-1\}$) every T_s seconds which, at the early stage of this research, is believed to be a multiple of the GOP (*Group of Pictures*) size:

A) *Buffer state*, B_t^i: it quantifies the load at the receiver for each received RTP_s^i. Let $b_t^i \in [0,1]$ denote the buffer state of session RTP_s^i at time t. At this early stage of our research we define $B_t^i \doteq \Gamma(b_t^i, \Delta b_t^i) \in [-1,1]$, where $\Delta b_t^i \doteq b_t^i - b_{t-T_s}^i$. $\Gamma(\cdot)$ is a monotonically increasing function with its two parameters. Notice that this generic definition of B_t^i not only accounts for the current state of the buffer, but also accommodates sharp changes on it.

B) *Interarrival jitter*, J_t^i : the interarrival jitter is defined as the mean deviation of the difference (D) in packet spacing at the receiver compared to the sender for a pair of packets. In our case, computation is done based solely on the timestamp values of received RTP packets. Let S_p and R_p denote the timestamps for the p-th RTP packet at transmission and reception, respectively. The packet spacing difference at session RTP_s^i for packets p and q will be given by

$$D_{(p,q)}^i \doteq (R_q^i\text{-}R_p^i)\text{-}(S_q^i\text{-}S_p^i) = (R_q^i\text{-}S_q^i)\text{-}(R_p^i\text{-}S_p^i). \quad (1)$$

The interarrival jitter for the received packet p within session RTP_s^i, denoted as j_p^i, is given by

$$j_p^i = j_{p-1}^i + (|D_{p-1,p}^i| - j_{p-1}^i)/16, \quad (2)$$

and the continuous interarrival jitter at time t for RTP_s^i, denoted as j_t^i, will be set equal to the interarrival jitter j_p^i of the last received packet for each RTP_s^i. It should be noted that j_t^i is continuously updated upon reception of each RTP packet. Then, every T_s seconds the overall *Interarrival jitter* J_t^i is computed as $J_t^i \doteq \Psi(j_t^i, \Delta j_t^i) \in [-1,1]$, where $\Delta j_t^i \doteq j_t^i - j_{t-T_s}^i$

and $j_t^i \in [0, j_{MAX}]$, with j_{MAX} denoting the maximum permissible delay for packet decoding. Similar to $\Gamma(\cdot)$, $\Psi(\cdot)$ is a monotonically increasing function with its two parameters.

C) *Loss Event Rate*, LER_t^i : it defines the rate at which packet loss events occur. Although the fraction between sent and received packets is typically used as a congestion indicator, it does present several drawbacks. When bursty losses occur, the value of such fraction metric decreases sharply from which serious network congestion is deduced. However, successive losses do not necessarily involve severe congestion, specially in wireless communications subject to interference and collisions. To overcome this issue, the novel packet *Loss Event Rate* LER_t^i metric proposed here comprises both isolated and bursty losses within a predetermined evaluation interval T_{le}^{eval}. In other words, LER_t^i is the frequency of packet loss events (either single or multiple) during a T_{le}^{eval} period for each RTP_s^i measured at time $t = kT_{le}^{eval}$.

To compute this metric, every T_{le}^{eval} seconds the client detects any packet loss based exclusively on checking the sequence number information provided by incoming RTP packets. Two variables are progressively updated after the loss detection process: I_{le}^{last} and I_{le}^{new}. The first refers to the index of the last evaluation period with either bursty or isolated packet losses, whereas the second is updated to the current evaluating interval index if any packet loss is detected. Based on these two variables, the instantaneous loss event rate $ILER_k^i$ for the k-th evaluation interval is computed as

$$ILER_k^i \doteq \begin{cases} 0 & \text{if no packet loss detected,} \\ \frac{1}{I_{le}^{new} - I_{le}^{last}} & \text{otherwise,} \end{cases} \tag{3}$$

from which a global weighted *Loss Event Rate* LER_t^i is recursively computed every T_{le}^{eval} seconds as

$$LER_t^i \doteq \delta \cdot ILER^i_{\lfloor t/T_{le}^{eval} \rfloor} + (1-\delta) \cdot LER^i_{t-T_{le}^{eval}}. \tag{4}$$

In the above definition, $\delta \in (0,1]$ is an arbitrary parameter that trades exhaustive traceability of the packet losses ($\delta = 1$) for the smooth estimation of the packet loss trend ($\delta \to 0$). It is also assumed that $LER_t^i \in [0,1]$: if no losses occur, $LER_t^i = 0$ and, otherwise, if every T_{le}^{eval} any packet is lost, $LER_t^i = 1$.

3.2.2 Decision Criteria

The congestion and flow control mechanism builds upon the above defined B_t^i (flow), J_t^i and LER_t^i (congestion) metrics. In fact, J_t^i is a significant indicator for initial network congestion. When the network is unable to correctly process traffic data, the packet delay increases even in absence of packet losses. When network congestion increases further, packet losses occur as the LER_t^i metric would reflect.

As multilayered streaming is considered in our system design, the whole set of $\{RTP_s^i\}_{i=0}^{n-1}$ must be considered at the receiver. However, note that each session does not have the same relevance due to the dependencies between scalable content layers (e.g. as RTP_s^0 contains the base layer, such session must be given full processing priority). Thereby, every $t = kT_s$ seconds a set of accumulated metrics $(\overline{B}_t^n, \overline{J}_t^n, \overline{LER}_t^n)$ for the n RTP_s^i sessions is obtained by applying different weights α^i, namely

$$\overline{B}_t^n = \sum_{i=0}^{n-1} \alpha^i \cdot B_t^i \quad \text{(Accumulated Buffer State)}, \tag{5}$$

$$\overline{J}_t^n = \sum_{i=0}^{n-1} \alpha^i \cdot J_t^i \quad \text{(Accumulated Jitter)}, \tag{6}$$

$$\overline{LER}_t^n = \sum_{i=0}^{n-1} \alpha^i \cdot LER_t^i \quad \text{(Accumulated } LER\text{)}. \tag{7}$$

It should be clear that since session RTP_s^0 contains the base layer, $\max\{\alpha^i\}_{i=0}^{n-1} = \alpha^0$. Also observe that the values of the weights for the three metrics within a given session index are set equal. Nevertheless, balancing the importance between $\overline{B}_t^n$, $\overline{J}_t^n$ and $\overline{LER}_t^n$ is accomplished by utilizing different coefficients in the metric fusion stage, which merges the above accumulated metrics into an overall flow-congestion indicator ζ_t as

$$\zeta_t = \Omega(\overline{B}_t^n, \overline{J}_t^n, \overline{LER}_t^n) = \gamma_B \cdot \overline{B}_t^n + \gamma_J \cdot \overline{J}_t^n + \gamma_L \cdot \overline{LER}_t^n, \tag{8}$$

where it should be remarked that $\Omega(\cdot)$ can be set to any other (not necessarily linear) combination of the accumulated metrics. Finally, a decision rule is taken every T_s based on ζ_t. The decision logic determines whether a new RTP_s^i can be demanded from the server (i.e. n is increased to $n+1$) without degrading the performance of both client and network, or if it is instead mandatory to reduce the number of sessions received ($n = n-1$), i.e.

$$n = \begin{cases} n+1 & \text{if } \zeta_t < \zeta_a(n), \\ n-1 & \text{if } \zeta_t > \zeta_r(n). \end{cases} \tag{9}$$

Note that decision limits $\zeta_a(n)$ (*add*) and $\zeta_r(n)$ (*remove*) are not static values but depend on the number of RTP_s^i sessions received by the client(n). By following this approach inter-session fairness is guaranteed, since we facilitate the demand of new RTP_s^i for low-quality streams, while restraining high quality streams from demanding more RTP_s^i sessions. Therefore, $\zeta_a(n)$ is a monotonically increasing function with n, bounded in the range $[0+\varepsilon_1, 1-\varepsilon_2]$, while $\zeta_r(n)$ is a monotonically decreasing function with n with support $[0+\varepsilon_3, 1-\varepsilon_4]$, where all ε's are design parameters. Unfortunately, our decision logic still poses the hazard of entering an unstable state when iterating between adjacent RTP_s^i. Since frequent layer switching degrades the QoE at content reproduction, we propose a safeguard mechanism: only if the stability of the system (based on proposed both network and client metrics) is guaranteed during a predetermined interval $\mathcal{T}_{st}$, the scalable layers contained in the newly received RTP_s^i are served to the decoder and finally, delivered to the end-user.

The definition of the $\Omega(\cdot)$, $\Psi(\cdot)$ and $\Gamma(\cdot)$ functions, as well as the obtention of optimum values for the decision limits $\zeta_a(n), \zeta_r(n)$ and the intervals T_s and $\mathcal{T}_{st}$ are not straightforward. In order to perform a satisfactory receiver-driven flow and congestion control, the following guidelines should be met:

- The receiver-driven control procedure should be responsive to sudden changes on any of the above metrics, and allow RTP_s^i session dropping as the value of any of such metrics becomes critical, i.e. $B_t \to 1$, $J_t \to 1$ or $LER_t \to 1$.
- The decision rule must be specially sensitive to the buffer state, as it dominates client's performance even in absence of network congestion.
- Iterating between adjacent RTP_s^i should be circumvented to avoid continuous layer switching which in turn degrades QoE.
- Inter-session fairness should be achieved. It is preferable to have equal-quality multimedia flows than streaming sessions with strongly asymmetric quality levels.

4 FUTURE RESEARCH

In this paper we have presented a novel end-to-end scalable content based streaming system aimed at maximizing the end-user's QoE. Our system profits from the virtues of the scalable content to perform a multilayered streaming, where each client is able to retrieve a personalized content. Being scalable encoding the only limitation imposed to our system, we determine to keep intact the involved streaming standards and maximize system component independence. Furthermore, due to our receiver-driven congestion and flow control algorithm, the streaming session is adapted to both dynamic changes in network's state and to client's limitations. Our presented control metrics (*Buffer State*, *Interarrival Jitter* and *Loss Event Rate*) are restricted to information already available in streaming clients. Besides, recall that the *Loss Event Rate* has been specifically designed to improve congestion control performance over heterogeneous networks.

Further investigation will be conducted towards the definition of the weights α^i, the $\Omega(\cdot)$, $\Psi(\cdot)$ and $\Gamma(\cdot)$ functions, and the decision limits $\zeta_a(n)$ and $\zeta_r(n)$. To this end, a threefold criteria will be adopted: 1) to be responsive to both sudden changes and critical values of network's and client's state metrics; 2) to emphasize on client's buffer state during the control procedure; and 3) to ensure inter-session fairness among the streaming clients.

REFERENCES

ASTRALS (2010). ASTRALS Project (FP6-IST-028097): *http://www.ist-astrals.org/*.

AVC (2008). File format support for Scalable Video Coding, "Information technology - Coding of audio-visual objetcs - Part 15: Advanced Video Coding (AVC) file format AMENDMENT 2", ISO/IEC 14496-15:2004/AMD2:2008(E).

Feamster, N., Bansal, D., and Balakrishnan, H. (2001). "On the Interactions Between Layered Quality Adaptation and Congestion Control for Streaming Video". In *11th International Packet Video Workshop*.

H.264/MVC (2009). H.264/MVC, "Ammendment H of Information technology - Coding of audio-visual objects - Part 10: Advanced video coding", ISO/IEC 14496-10:2009(E).

H.264/SVC (2009). H.264/SVC, "Ammendment G of Information technology - Coding of audio-visual objects - Part 10: Advanced video coding", ISO/IEC 14496-10:2009(E).

ISMA (2010). Internet Streaming Media Alliance: *http://www.isma.tv/*.

Liebl, G., Schierl, T., Wiegand, T., and Stockhammer, T. (2006). "Advanced Wireless Multiuser Video Streaming using the Scalable Video Coding Extensions of H.264/MPEG4-AVC". In *ICME*, pages 625–628. IEEE.

Ma, L. and Ooi, W. T. (2007). "Congestion Control in Distributed Media Streaming". In *26th Annual IEEE Conference on Computer Communications (INFOCOM 2007)*.

Mujica-V., V. E., Sisalem, D., Popescu-Zeletin, R., and Wolisz, A. (2004). "TCP-Friendly Congestion Control over Wireless Networks". In *European Wireless 2004*.

Papadimitriou, P. and Tsaoussidis, V. (2007). "SSVP: A Congestion Control Scheme for Real-time Video Streaming". *Computer Networks*, 51(15):4377–4395.

Renzi, D., Amon, P., and Battista, S. (2008). "Video Content Adaptation Based on SVC and Associated RTP

Packet Loss Detection and Signaling". In *Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, pages 97–100.

RTP (2010). RTP Payload Format for SVC Video ("draft-ietf-avt-rtp-svc-20.txt") *http://tools.ietf.org/html/draft-ietf-avt-rtp-svc-20*.

Schierl, T., Hellge, C., Mirta, S., Grüneberg, K., and Wiegand, T. (2007). "Using H.264/AVC-based Scalable Video Coding (SVC) for Real Time Streaming in Wireless IP Networks". In *IEEE International Symposium on Circuits and Systems (ISCAS)*.

Tizon, N. and Pesquet-Popescu, B. (2008). "Scalable and Media Aware Adaptive Video Streaming over Wireless Networks". *EURASIP Journal on Advances in Signal Processing*, 2008(168).