



HAL
open science

Scalable Media Coding enabling Content-Aware Networking

Michael Grafl, Christian Timmerer, Hermann Hellwagner, George Gardikis,
Georgios Xilouris, Daniele Renzi, Stefano Battista, Eugen Borcoci, Daniel
Negru

► **To cite this version:**

Michael Grafl, Christian Timmerer, Hermann Hellwagner, George Gardikis, Georgios Xilouris, et al.. Scalable Media Coding enabling Content-Aware Networking. IEEE MultiMedia, 2013, 20 (2), pp.30-41. 10.1109/MMUL.2012.57 . hal-00998566

HAL Id: hal-00998566

<https://hal.science/hal-00998566>

Submitted on 2 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Scalable Media Coding enabling Content-Aware Networking

Michael Grafl, Christian Timmerer,
Hermann Hellwagner
Institute of Information Technology (ITEC)
Alpen-Adria-Universität (AAU)
Klagenfurt, Austria
e-mail: {*firstname.lastname*}@itec.aau.at

George Xilouris, Georgios Gardikis
Institute of Informatics and Telecommunications
NCSR “Demokritos”
Athens, Greece
e-mail: {xilouris, gardikis}@iit.demokritos.gr

Daniele Renzi, Stefano Battista
bSoft ltd
Macerata, Italy
e-mail: {daniele, bautz}@bsoft.net

Eugen Borcoci
Telecommunications Dept.
University POLITEHNICA of Bucharest
Bucharest, Romania
e-mail: eugen.borcoci@elcom.pub.ro

Daniel Negru
CNRS LaBRI Lab.
University of Bordeaux
Bordeaux, France
e-mail: daniel.negru@labri.fr

Abstract— Given that multimedia services are becoming increasingly popular, they are expected to play a dominant role for the Future Internet. In this context, it is essential that Content-Aware Networking (CAN) architectures, as envisaged in the frame of the Future Internet, explicitly address the efficient delivery and processing of multimedia content. This article proposes adopting a content-aware approach into the network infrastructure, thus making it capable of identifying, processing, and manipulating (i.e., adapting, caching, etc.) media streams and objects in real time towards Quality of Service/Experience (QoS/QoE) maximization. Our proposal is built upon the exploitation of scalable media coding technologies within such a content-aware networking environment and is discussed based on four representative use cases for media delivery (unicast, multicast, peer-to-peer, and adaptive HTTP streaming) and with respect to a selection of CAN challenges, specifically flow processing, caching/buffering, and QoS/QoE management.

Keywords— *scalable media coding; content-aware networking; in-network adaptation; content-aware buffering*

I. INTRODUCTION

The Future Internet (FI) [1] development has raised a rich set of research issues given the huge, global impact of this technology and new societal needs for services. A significant trend is recognized towards an information-centric orientation and, consequently, new challenging concepts are emerging. In particular, significant changes in communications and networking are proposed, including novel basic architectural principles. What are the implications of new networking principles for media streaming? How does the deployment of scalable media formats benefit from these developments? Before we answer

these questions, let us briefly revisit the approaches towards the FI and the basics of scalable media formats. The new conceptions are generally divided into revolutionary (i.e., clean-slate) and evolutionary approaches. The revolutionary approaches are often referred to as *Information-Centric Networking (ICN)*, which is used as an umbrella term for related concepts such as Content-Oriented Networking (CON) and Content-Centric Networking (CCN) [2][3]. On the other hand, evolutionary (or incremental) approaches, such as *Content-Aware Networking (CAN)*, aim at building upon existing Internet infrastructures. In this article we will explain the role of CAN for multimedia services in more detail. We will present four media streaming use cases which characterize different requirements w.r.t. content-aware processing in the network and highlight the utility of scalable media formats.

Clean-slate ICN approaches, as surveyed in [1] and [3], are very promising but raise a long list of research challenges like the degree of preservation of the classic transport (TCP/IP) layering principles, naming and addressing, content-based routing and forwarding, management and control framework, in-network caching, energy efficiency, trust, security embedded in the content objects, Quality of Service and Experience, and media flow adaptation. Additionally, new business models are needed for users, content producers, consumers, and service/network providers; deployment issues such as compatibility with existing equipment, scalability, and privacy become crucial.

In parallel, evolutionary approaches towards the FI like Content-Aware Networking are proposed in [4] and developed within the ALICANTE (Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments) project [5], enabling efficient *routing and*

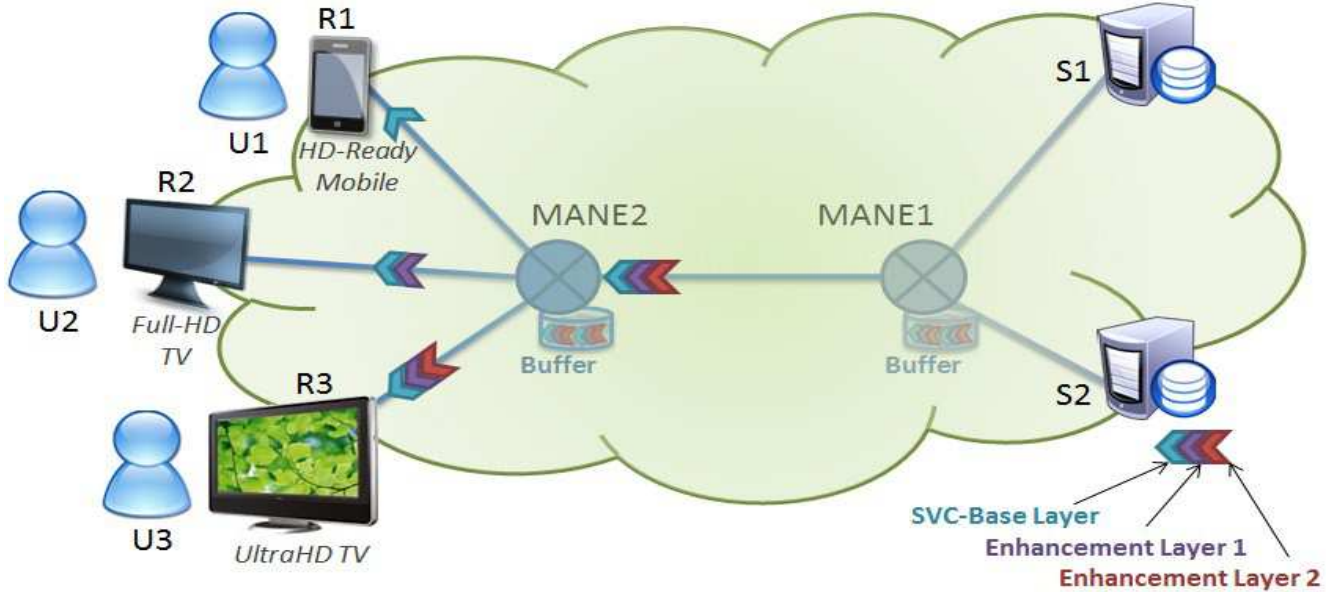


Figure 1. High-Level System Overview.

forwarding of content based on given *content and context characteristics* including the *adaptation* thereof. ALICANTE deploys content- and context-aware strategies at the network edges as discussed in [6]. A main challenge of evolutionary approaches is obviously to overcome limitations of the current Internet [1].

The ALICANTE content-aware network environment attempts at optimizing network resource utilization while maintaining the expected Quality of Service (QoS) and Quality of Experience (QoE) respectively. For this purpose:

- It establishes virtual networks on top of the physical infrastructure, which feature inherent content awareness, e.g., by dynamically providing network resources appropriate for different content types.
- It provides in-network media caching as well as real-time adaptation, exploiting scalable media coding formats, such as Scalable Video Coding (SVC), which are a vital component towards this objective thanks to their compression efficiency and flexibility [6].

Both aforementioned functions are provided by enhanced network nodes, the Media-Aware Network Elements (MANEs), which feature virtualization support, content-awareness, and media processing, as well as buffering and caching.

MANEs take advantage of SVC technology in order to achieve in-network media processing. SVC is an extension of MPEG-4 Advanced Video Coding (AVC) and requires a moderate compression overhead of around 10% over single layer coding (i.e., AVC) [7]. In SVC, the video bitstreams are encoded following a layered approach comprising an AVC-compliant base layer providing the basic quality (e.g., temporal, spatial, SNR) and one or more incrementally added enhancement layers. For example, the base layer provides the content quality needed for legacy devices or mobile devices (e.g., 720p) while, with additional

enhancement layers, high-definition (e.g., 1080p) and beyond could be reached. Currently, the next generation of SVC is being developed within MPEG based on the High Efficiency Video Coding (HEVC) technology [8].

SVC enhancement layers serve various adaptation purposes in media streaming. As a rule of thumb, spatial SVC enhancement layers support heterogeneous devices with different display resolutions, while SNR (bit-rate) and/or temporal enhancement layers rather enable dynamic adaptation towards available bandwidth.

The aim of this article is to describe the role of scalable media coding formats – such as SVC – in Content-Aware Networks and to propose new solutions for some use cases. Therefore, we will describe a set of use cases (Section II) and provide an analysis thereof regarding a selection of CAN challenges (Section III), specifically flow processing, caching/buffering, and QoS/QoE management. Finally, we provide conclusions in Section IV.

II. USE CASES

In this section we will illustrate use cases highlighting the benefits of using SVC in CAN ranging from *unicast* and *multicast* to *P2P* and adaptive *HTTP streaming*.

A simplified and generic high-level system overview for the use cases in question is depicted in Figure 1 comprising the following entities: two senders (S1, S2), two MANEs (MANE1, MANE2), and three receivers (R1, R2, R3) with different terminal and (potentially) network capabilities, to which three end-users (U1, U2, U3) are connected. Our discussion of the use cases addresses streaming of non-live content (e.g., Video on Demand), unless noted otherwise. Please note that in more complex scenarios, more senders, even more receivers, and additional MANEs distributed over multiple autonomous network domains may be deployed. These use cases are subsequently analyzed in Section III with respect to content-aware networking aspects.

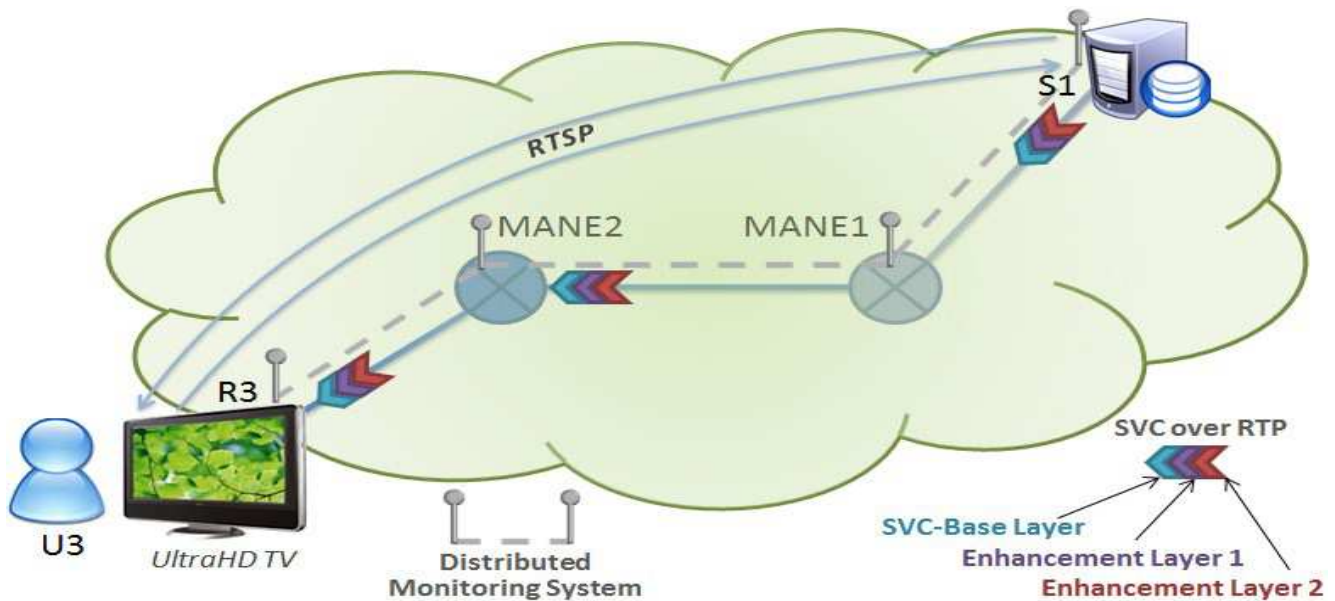


Figure 2. Unicast Streaming in Content-Aware Networks.

A. Unicast Streaming

For the unicast use case we have only one sender (e.g., S1), which streams the scalable video content to a single receiver (e.g., R3), like in a traditional Video on Demand application (see Figure 2). This layered media coding approach enables MANEs along the path to perform content-aware operations such as in-network content adaptation. For example, a MANE can react to changing network conditions (based on information provided by a network monitoring system) by dropping enhancement layers of the SVC stream. In current deployments, RTP is typically used as the transport protocol and RTSP is used for session control. Note that in the unicast use case the SVC stream is typically sent via single-session transmission mode over RTP, i.e., all SVC layers are packed into one RTP session.

B. Multicast Streaming

The second use case is multicast streaming, which is characterized by a single sender providing the same content to multiple receivers. This use case is obtained if one sender (e.g., S2 in Figure 1) is streaming the content to heterogeneous trees of MANEs and subsequently to multiple receivers (e.g., R1, R2, R3). The term *heterogeneous trees* denotes a set of trees, allocated for different SVC layers. All trees have the same root (e.g., S2) but different leaves, depending on the transported SVC layer (e.g., the SVC base layer is delivered to all receivers, while the highest SVC layer is only received by R3), as shown in Figure 3.

Scalable media formats enable the realization of this use case via receiver-driven layered multicast (RDLM) [9] and with SVC this approach is becoming efficient enough to surpass simulcast [6]. In RDLM, different layers are transmitted over separate multicast groups. RTP realizes this via the multi-session transmission mode, where SVC layers are separated into multiple RTP sessions at the sender side,

and re-arranged to the proper SVC bitstream at the receiver side. Each receiver only subscribes to those layers that it supports and that its network link can handle.

Again, a MANE can react to changing network conditions by adjusting the number of layers to which it is subscribed. Such an approach simplifies the adaptation operations. MANEs can transparently neglect the video header information, since the mapping of SVC layers to multicast groups is realized at a lower level, simplifying the process of content adaptation. In other words, a MANE simply adjusts the number of subscribed RTP sessions without having to inspect each and every RTP packet header.

C. Peer-to-Peer Streaming

In a P2P streaming use case, multiple senders exist and every sender provides some parts of the content called chunks or pieces, while one or possibly more receivers consume the content. A scalable media format enables each receiver to request only those layers which are supported by its media player.

In contrast to conventional P2P content distribution, P2P streaming has the timing constraint that every piece must arrive before its playout deadline expires. P2P streaming systems typically use a sliding window of pieces which are currently relevant for the receivers. Within this sliding window, a piece-picking algorithm at the receiver side takes care of downloading those pieces that provide the highest quality to the end-user. The piece-picking algorithm ensures that the base layer is always received before the deadline, determines enhancement layers that can be downloaded under the current network conditions, and takes care of the peer selection for each piece [10].

While a P2P system is traditionally organized as an overlay network that is transparent to the core network, a content-aware network will allow MANEs to participate in the streaming process in several ways. Figure 4 shows an

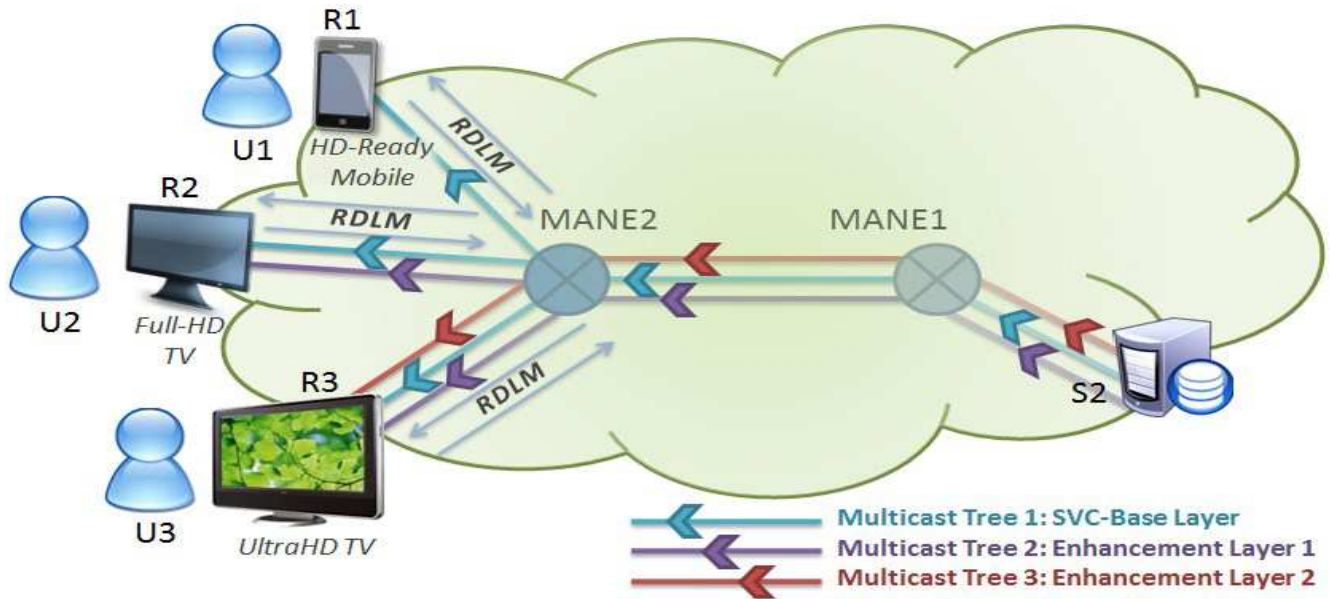


Figure 3. Multicast Streaming in Content-Aware Networks.

outline of this use case, showing senders, receivers, and the supporting MANEs.

A MANE can participate in P2P streaming by caching pieces in a content-aware manner or by acting as a peer itself as discussed later in this article (in Section III.A).

D. Adaptive HTTP Streaming

The previous use cases have shown streaming scenarios with various numbers of senders and receivers. In order to overcome common shortcomings of RTP-based streaming such as network address translation (NAT) and firewall issues, this use case introduces adaptive HTTP streaming (e.g., MPEG-DASH) in the context of CAN. In HTTP streaming, the content is typically fragmented into segments which are downloaded by the receiver via individual HTTP (partial) GET requests. This approach allows for a stateless sender and enables at the same time caching at the MANEs and dynamic content adaptation at the client. Based on several industry solutions, MPEG has recently standardized Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [11].

HTTP streaming is typically used in unicast mode, but multicast or even P2P streaming modes are also possible.

In unicast mode, the sender provides a so-called manifest file of the content, which describes the structure of the media segments and the available media representations. A media representation denotes a particular encoding configuration of the content, e.g., bit-rate or resolution. For layered coding formats such as SVC, those representations can define either the individual layers or even sub-sets of layers of the bitstream. The receiver selects the appropriate representation based on its (rendering) capabilities and starts requesting continuous segments of the content from the sender. MANEs along the network path can act as caches or as content delivery network (CDN) nodes, as shown in Figure 5.

Although HTTP is a unicast protocol, the concept of HTTP streaming can also be applied to multicast streaming. If MANEs along the network path between sender and receivers cache the content segments for subsequent requests by other receivers, the result will be a multicast-like tree. The technical considerations of this approach are discussed later in Section III.B.

The concept of HTTP streaming can even be applied to multisource streaming scenarios similar to P2P streaming. The manifest file can contain multiple sources for each segment including dynamic updates thereof. The receiver may select any of them to download the segments, thus, balancing the load among the senders.

III. ANALYSIS OF USE CASES

We have described different use cases for multimedia streaming and how they can be applied in content-aware networks. In this section we will provide an analysis concerning content-aware network operations, such as *flow processing, caching and buffering*, and *QoS/QoE management* for the use cases in question and present some recent scientific advances.

A. Flow Processing

In the **unicast use case**, the usage of scalable media formats like SVC in a content-aware network brings three main advantages.

First, the sender can easily adapt the content to the receiver's capabilities by only sending those layers that are actually supported by the receiver (e.g., in terms of spatial resolution).

Second, a MANE can perform efficient in-network adaptation of the content in reaction to network fluctuations. That is, when a MANE detects a decrease in available downstream bandwidth that prevents the entire content from being transmitted, it can drop some higher layers of the

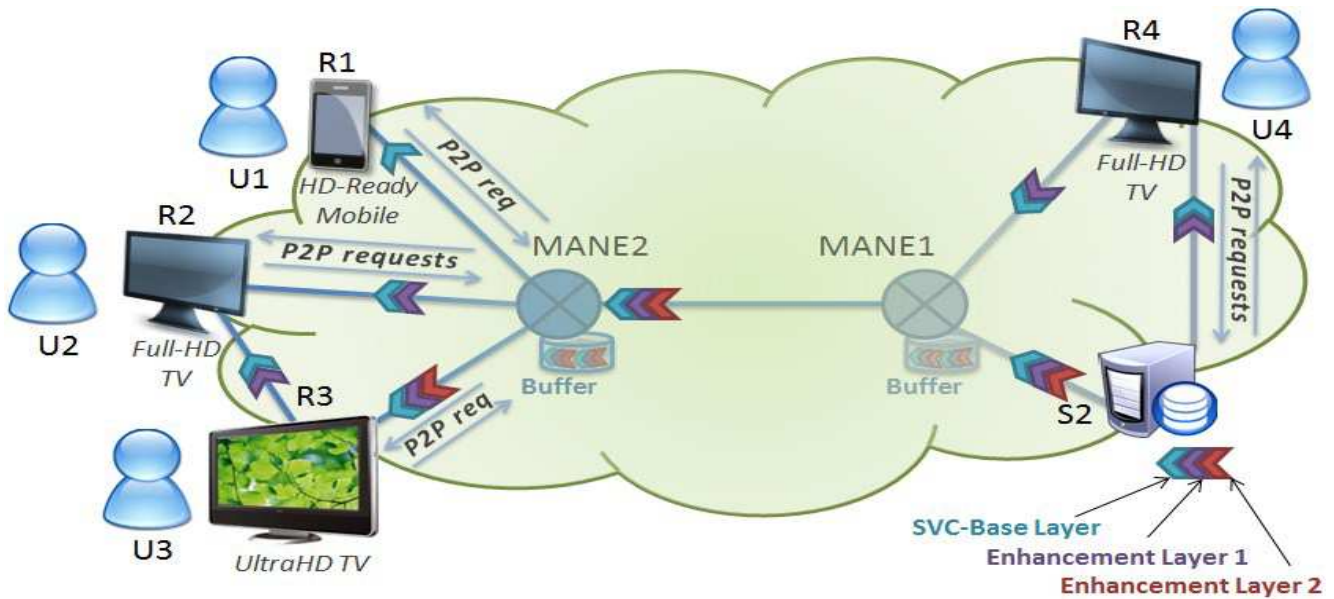


Figure 4. P2P Streaming in Content-Aware Networks.

media stream, assuring continuous playout of at least the base quality at the receiver. Although the end-user receives the content at a lower bit-rate, the actual QoE may increase compared to the alternative which would cause the playout either to stall or to show too many visual artifacts due to high packet loss rate. As soon as the network conditions return to normal, the MANE can re-increase the number of forwarded layers. Each decision about dropping or forwarding SVC layers is triggered by a distributed network monitoring system, which detects network fluctuations and raises appropriate alarms.

The choice which SVC layers to drop or to forward is solved by an adaptation decision-taking engine (ADTE). The ADTE is actually not specific to SVC adaptation but is used for steering any adaptation of content – be it at the MANE or outside the network at the sender or receiver. Based on context parameters and the description of possible adaptation options, the ADTE runs an optimization algorithm that finds the best-suited choice for the current situation. In the case of in-network SVC adaptation, the set of context parameters is reduced to the network parameters and possible adaptations are limited to SVC layers, making this task rather simple and fast to compute.

Third, a MANE can signal its monitoring information about the network condition upstream to the sender, allowing for sender-side adaptation. While in-network adaptation is a good solution for mitigating short-term network fluctuations, it wastes bandwidth between the sender and the MANE in case of longer periods of decreased available bandwidth. In other words, if a higher layer packet is to be discarded at a MANE anyway, it is useless to transmit it to that MANE in the first place. Note, however, that network-aware adaptation at the sender needs at least one round-trip time (from MANE to sender) to take effect.

In the **multicast use case**, MANEs can adapt to changing network conditions by subscribing to or unsubscribing from

multicast groups containing SVC enhancement layers. Conventional layered multicast is receiver-driven [9], i.e., the receivers control the subscriptions to multicast groups. Hence, in-network adaptation is achieved implicitly as the receiver controls it through subscription to appropriate SVC layers. MANEs aggregate and combine subscriptions from downstream entities – both receivers and MANEs – using them for subscribing to appropriate SVC layers upstream. ALICANTE adopts and extends the RDLM approach for the distribution of video content in multicast-based scenarios.

There are two possibilities for MANEs to assist the network-aware adaptation of multicast streaming. Either, downstream forwarding of one or more SVC layers can be temporarily truncated in case of congestion at an outgoing link as discussed in [12], or a MANE can control multicast group subscriptions by sending prune or graft messages to upstream neighbors as defined in RFC 3973 [13].

MANEs can also improve multicast functionalities of existing network infrastructures. If native multicast is not supported, MANEs may perform overlay multicast with adjacent MANEs, so that they become bridges between native and overlay multicast, as it is done in ALICANTE [5]. Furthermore, ALICANTE supports traffic engineering as well as content and service classification and differentiation mechanisms (i.e., DiffServ and MPLS) that enable selective treatment of SVC layers, e.g., increasing priority and robustness of the base layer.

For the **P2P streaming use case**, a MANE may act as a peer, autonomously requesting pieces which it deems relevant for any connected receivers. Running a P2P engine on a MANE increases the processing requirements for this entity but it also offers a flexible and powerful way to participate in P2P streaming. The MANEs thus form a P2P overlay network (at the CAN layer) that may closely cooperate with the overlay network at the application layer.

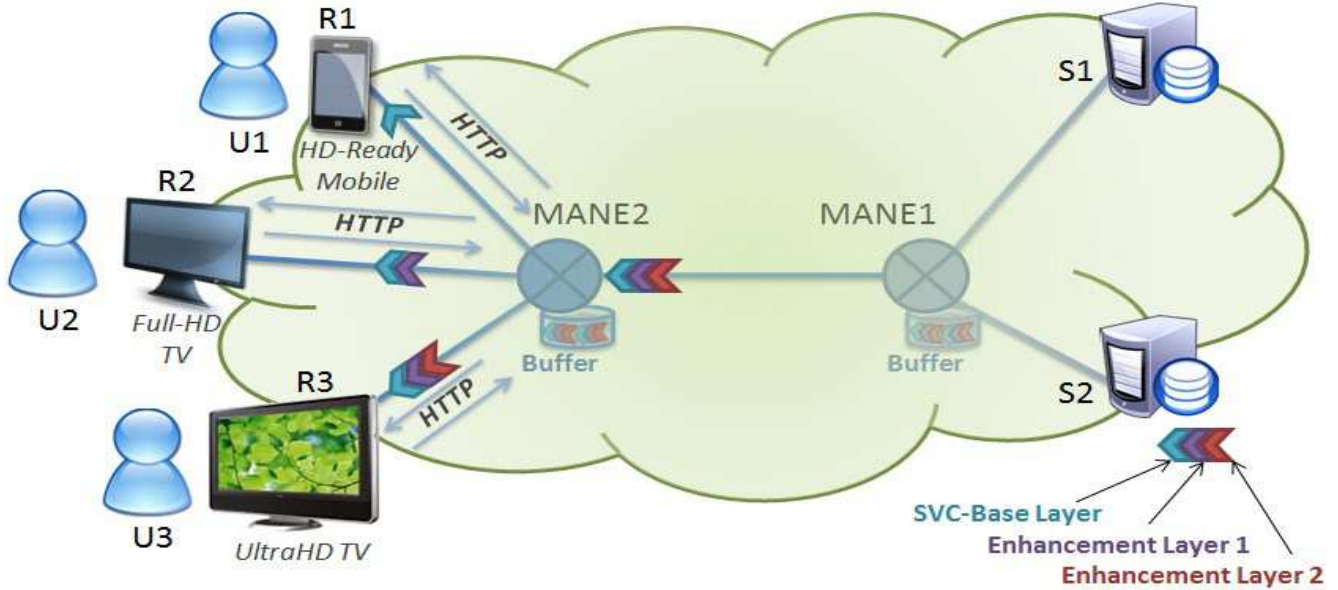


Figure 5. Adaptive HTTP Streaming in Content-Aware Networks.

The aforementioned flow processing policies are also applicable to **adaptive HTTP streaming** with some noticeable differences. TCP uses reliable transmission which is not suitable for in-network adaptation achieved through enhancement layer dropping. If a MANE simply drops TCP packets of an enhancement layer to avoid network congestion, it would trigger the sender to retransmit the packets after TCP timeout. For the streaming session, the retransmission of the packet wastes bandwidth and even if the packet reached the receiver eventually, it would probably arrive after the playout deadline. Thus, for HTTP streaming a MANE shall act as a (transparent) proxy cache in combination with CDN functionality as will be described in the following section. As the adaptation logic is entirely located at the receiver side, in-network adaptation is achieved implicitly – similar to the multicast use case – by means of HTTP requests for layers that are supported by the receiver. Requests for individual SVC layers can be answered by different network nodes (or by the sender), depending on where these layers are buffered. Hence, adaptation occurs within the network, but without active participation by the MANEs.

The aforementioned in-network adaptation mechanisms – implicit or explicit – provide a powerful tool for mitigating the effects of network fluctuations. Furthermore, the adaptation decision-taking (i.e., the selection of which SVC layers to forward) has to be performed in a distributed manner. That is, each MANE computes its local adaptation decision and coordinates it with the other nodes in the network. Efficient, scalable signaling and coordination of adaptation decisions is still an open research challenge [6].

B. Caching and Buffering

MANEs can buffer previously requested content and may even act as CDN caches, i.e., proactively moving the content closer to the receivers. Note that the storage requirements for

CDN-enabled MANEs are considerably higher than for mere buffering support.

In the **unicast use case**, a CDN-enabled MANE can proactively perform caching of popular content. In particular, prefix caching decreases start-up delay while also reducing network traffic. When a receiver requests the content, the MANE starts streaming from its cache while requesting the suffix of the content from the sender [14].

The usage of SVC offers a tradeoff between quality and availability to the MANE. The prefix cache may contain only the base layer for less popular content. Thus, the end-user starts receiving only the base layer, but with a low start-up delay, and later the enhancement layers from the sender are added.

Proactive caching can also be used in the **multicast use case** to reduce mainly start-up delay but also network traffic. Note that proactive caching is not applicable to live streaming sessions. Moreover, all receivers are served simultaneously via multicast RTP streams, abolishing the need for buffering at MANEs.

In the **P2P streaming use case**, a MANE can aggregate requests for a piece and buffer downloaded pieces for subsequent requests. Especially in live scenarios, almost all receivers share the same time window for the content; thus, each piece will be highly popular for a short time span. By buffering a piece during this time frame, the MANE will be able to reduce network utilization and latency even with a limited buffer size. In most cases, such behavior is transparent to the peers within the traditional, application layer P2P overlay network.

Additionally, the MANE may also aggregate requests for the same piece to different senders and only forward one request which we call *content-aware buffering*. Unlike conventional, the MANE may intercept requests and transmit a buffered piece instead of forwarding them. This approach would constitute an evolutionary implementation of the CCN

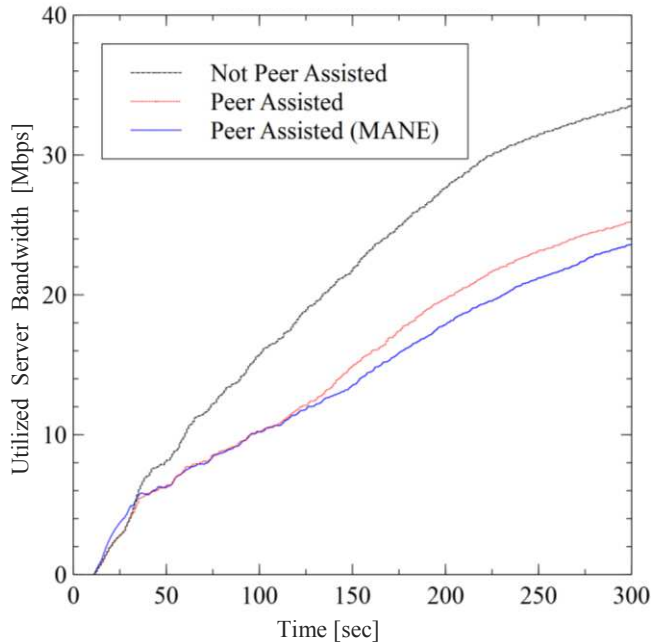


Figure 6. Simulation of peer-assisted HTTP streaming with MANEs as peers.

functionality [2]. However, a small drawback of this approach is that the peer selection of the first receiver might not always be the optimal selection. But once the MANE has downloaded and buffered the entire piece, the issue is alleviated.

A MANE might also act as a peer, proactively requesting pieces that may be needed in the near future by any receivers connected to it. Thus, the MANE increases the replication of the content and moves it closer to the receivers. However, this puts some additional performance and storage requirements on the MANE.

Caching and buffering are also integral parts of the **adaptive HTTP streaming use case**. In unicast mode, a MANE can provide CDN functionalities similar to the unicast use case discussed above. In contrast to RTP-based streaming, HTTP streaming immediately benefits from existing HTTP caching infrastructures [15] that may be deployed on top of content-aware networks. The multicast mode relies on buffering and request aggregation at the MANE for bandwidth-efficient streaming. As mentioned before, intelligent buffering at MANEs along the network path between sender and receivers constructs a bandwidth-efficient multicast tree. In order for the buffer size at the MANE to remain inside a reasonable limit, two requirements must be met. On the one hand, all receivers must share the same time window so that the popularity of a segment is temporarily limited. This time window can be signaled in the manifest file, as it is typically the case for live streaming services [11]. On the other hand, the MANE has to be aware of the streaming session in order to buffer the segments accordingly. The straightforward solution is for the MANE to parse the manifest file and to retrieve such information from there. An alternative solution would be that the MANE

learns about the best buffering policy from a statistical analysis of the stream.

In the multisource mode of HTTP streaming, buffering at MANEs has similar effects as in P2P streaming. That is, MANEs aggregate requests (even to different senders) and perform content-aware buffering of downloaded segments for the duration of the sliding window of the streaming session. An open research challenge is the impact of the discussed request aggregation on the load balancing strategies between the senders.

In a recent study, Lederer et al. have proposed a peer-assisted HTTP streaming architecture compliant with MPEG-DASH [16]. For each segment, the server lists a selection of possible peers in the manifest file. Those peers have already downloaded the segment and provide it through local HTTP servers. Other clients download segments from those peers if their buffer fill level guarantees smooth playback. Even under the consideration that clients have asymmetric Internet connections with significantly lower uplink bandwidth than downlink bandwidth, the solution reduces server bandwidth by up to 25%. While that work [16] focuses on conventional client peers, MANEs can act as peers just as well. Since MANEs are usually not limited by asymmetric connection speeds, server bandwidth can be further reduced. To validate this assumption we performed simulations with the same setup as [16], except that MANEs acting as peers had symmetric connection speeds (15 peers with 16 Mbps and 25 peers with 8 Mbps). Like in the original evaluation, the maximum bitrate of the content was set to 1,400 kbps. The simulation results of server bandwidth requirements over time are shown in Figure 6. Original server bandwidth for asymmetric connection speeds of peers is labeled *Peer Assisted*, server bandwidth for symmetric connection speeds is labeled *Peer Assisted (MANE)*. MANEs acting as peers in this HTTP streaming scenario were able to reduce server bandwidth by up to 29.5%. It should be noted that the simulation did not consider frequent updates of the manifest file, which contains the current list of peers. Updating the manifest file every 60 or 120 seconds would bring further performance gains.

The deployment of SVC in HTTP streaming also brings benefits to caching and buffering mechanisms. While HTTP streaming of non-layered media formats requires switching between different content representations (e.g., frame-rate, resolution, quality) for adaptation, SVC-based adaptation is performed by adding/removing enhancement layers. Thus, the MANE only has to cache one SVC stream instead of multiple streams for different representations. This both reduces storage requirements and increases cache performance. Simulations conducted by Sánchez et al. compared the combination of SVC-based HTTP streaming and a streaming-optimized caching strategy to AVC-based streaming under Least Recently Used (LRU) strategy [15]. Their results show that the cache hit ratio can be increased by up to 11.5 percentage points for congestion in the cache feeder link (i.e., the link between the sender and the cache) and by up to 25.7 percentage points for congestion in the access links.

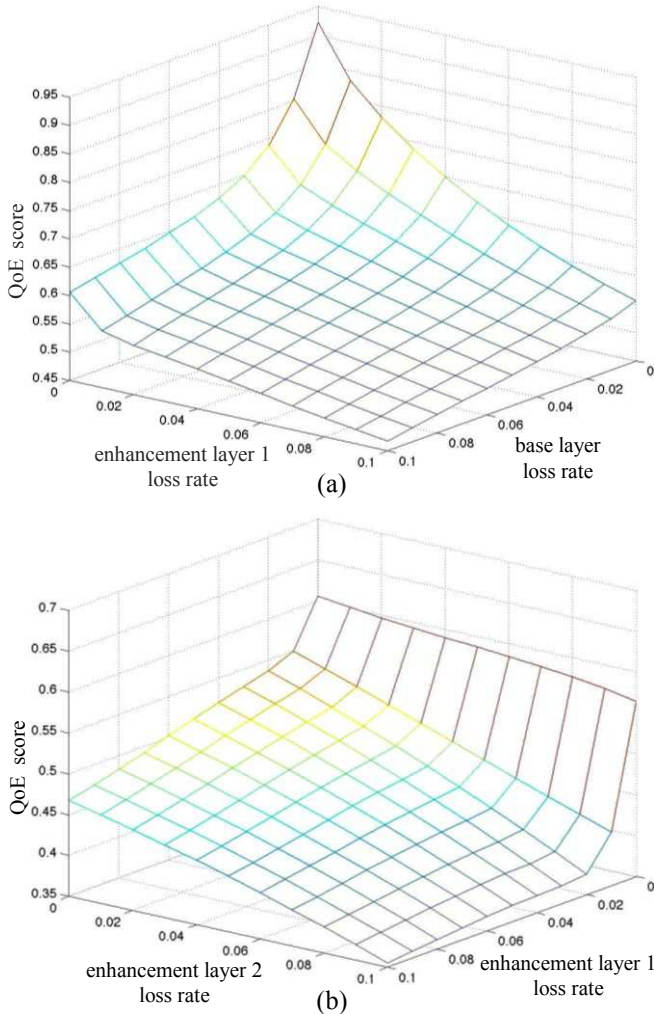


Figure 7. QoE scores vs. (a) loss rate at SVC base layer and enhancement layer 1, and (b) loss rate at enhancement layer 1 and enhancement layer 2 with base layer loss rate of 10%.

C. QoS/QoE Management

A primary goal of content-aware networking is to manage and optimize the QoS and consequently QoE at the application level.

The term *QoS* describes properties of the network that influence the transport of media flows. Metrics like delay, packet loss, and jitter are used to measure QoS. The more recently coined term *QoE* targets the degree of delight or annoyance of the user about an application or service. Besides QoS parameters, also user-related factors (e.g., expectations) as well as terminal capability and performance play a role in QoE. QoE is typically measured as Mean Opinion Score (MOS) based on user ratings. More information on QoS and QoE can be found in [17].

QoS/QoE optimization can be achieved through context-aware mechanisms both at the end-user side and within the (core) network. At the end-user side, several aspects of the usage environment (such as terminal capabilities) can be taken into account during content request and consumption.

Other aspects such as user preferences and the current status of the end-user terminal may even dynamically affect the configuration of the requested SVC stream.

Within the (core) network, context-awareness relates to the current condition of the network. Network monitoring enables MANEs to react to network fluctuations by performing in-network adaptation of SVC content. Monitoring information is used locally and is aggregated at the CAN level for managing the network behavior and establishing long-term adaptation policies [4].

One important aspect is the appropriate media encoding configuration. The ALICANTE project is working on encoding guidelines for SVC that facilitate distributed adaptation. Those guidelines will comprise a description of typical resolutions, which and how many bitrates to use for each resolution, appropriate scalability modes (temporal, spatial, or SNR), how to combine these modes, differences among use cases, and more. On the other end of the media delivery chain, the project investigates the video quality at the client when there have been packet losses in any of the SVC layers. Evaluations are performed using a no-reference QoE tool called ALICANTE Pseudo Subjective Quality Assessment (A_PSQA) [18], which uses a continuous QoE score ranging from 1 (excellent) to 0 (bad) to estimate video quality based on packet loss characteristics. The SVC streams used in the experimental setup comprised three layers. Figure 7 shows how the quality of a video degrades for packet loss at any of these layers.

The QoE scores are subsequently used for triggering the adaptation and enhancing the granularity by which the system reacts to context variations. Thus, QoE evaluations are a vital part of advanced adaptive media delivery systems.

As already mentioned, SVC enables a fine-grained control over the QoE at the network level. A non-scalable media format will suffer from severe QoE degradation if not all packets of the stream are transmitted. With SVC, lower layers can be prioritized, maintaining smooth and undistorted playout with controlled QoE degradation. SVC can also be conveniently combined with error recovery techniques at the decoding side, in order to further enhance the QoE perceived by the user.

As a conclusion, Table 1 summarizes the discussed CAN-related challenges for each of the described use cases. Note that for *QoS/QoE management* we make no explicit distinction between the use cases.

IV. CONCLUSIONS

Scalable media coding formats (such as SVC) in combination with in-network adaptation – and, as a consequence, its capabilities in terms of flow processing, caching/buffering, and QoS/QoE – are becoming promising concepts towards enabling content-awareness within the (core) network. This concept is referred to as Content-Aware Networking (CAN) and provides an elegant, powerful, and flexible tool to accommodate existing and imminent challenges for a variety of traditional and emerging use case scenarios in the context of multimedia delivery within the Future Internet.

Table 1. Summary of CAN-related challenges addressed by the presented use cases.

Use case	CAN challenge		
	<i>Flow processing</i>	<i>Caching & buffering</i>	<i>QoS/QoE management</i>
<i>Unicast</i>	<ul style="list-style-type: none"> explicit adaptation signaling adaptation decision to sender 	<ul style="list-style-type: none"> SVC-based prefix caching for low start-up delay 	<ul style="list-style-type: none"> local & aggregated monitoring of network conditions smooth, undistorted playout via SVC
<i>Multicast</i>	<ul style="list-style-type: none"> implicit adaptation multicast bridges for hybrid multicast differentiated forwarding of SVC layers 	<ul style="list-style-type: none"> SVC-based prefix caching for low start-up delay no buffering 	
<i>P2P streaming</i>	<ul style="list-style-type: none"> explicit adaptation peer for CAN P2P overlay network 	<ul style="list-style-type: none"> aggregating requests content-aware buffering of pieces within sliding window 	
<i>Adaptive HTTP streaming</i>	<ul style="list-style-type: none"> implicit adaptation transparent proxy cache 	<ul style="list-style-type: none"> SVC-based prefix caching <i>multicast</i>: buffering within sliding window <i>multisource streaming</i>: aggregating requests & content-aware buffering within sliding window 	

We have argued that sender-driven use cases such as unicast and multicast streaming greatly benefit from content-awareness for routing and forwarding. In P2P streaming, the combination of enhanced forwarding and buffering techniques may allow MANEs to collaborate with receivers within the P2P network. Content- and context-aware caching/buffering are furthermore important aspects in the adaptive HTTP streaming use case.

Interesting challenges remain, such as the integration of on-the-fly QoE evaluation of SVC content for adaptive media streaming or the further improvements to the involvement of MANEs into P2P streaming. As future trends indicate more advanced video compression technologies targeting resolutions beyond 1080p, (e.g., a new scalable extension for HEVC), efficient and reliable buffering at MANEs becomes increasingly important in order to reduce overall network loads. Furthermore, adaptive HTTP streaming becomes increasingly popular due to its relatively easy deployment. Therefore, our future work will focus on how MANEs can further improve the existing HTTP infrastructure.

ACKNOWLEDGMENT

This work was supported in part by the EC in the context of the ALICANTE project (FP7-ICT-248652). The authors thank Stefan Lederer for his help with simulation results.

REFERENCES

[1] J. Pan, S. Paul, R. Jain, "A survey of the research on future internet architectures", *IEEE Communications Magazine*, vol.49, no.7, pp.26-36, July 2011.

[2] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, R. Braynard, "Networking named content", in *Proceedings of ACM CoNEXT 2009*, Rome, Italy, December 2009.

[3] J. Choi, J. Han, E. Cho, T. Kwon, Y. Choi, "A survey on content-oriented networking for efficient content delivery", *IEEE Communications Magazine*, vol.49, no.3, pp.121-127, March 2011.

[4] H. Koumaras, D. Negru, E. Borcoci, V. Koumaras, C. Troulos, Y. Lapid, E. Pallis, M. Sidibé, A. Pinto, G. Gardikis, G. Xilouris, C. Timmerer, "Media Ecosystems: A Novel Approach for Content-Awareness in Future Networks," *Future Internet: Achievements and Promising Technology*, Springer Verlag, pp. 369-380, May 2011.

[5] ALICANTE Web site, <http://ict-alicante.eu/>. Accessed Nov. 11, 2012.

[6] M. Graf, et al., "Scalable Video Coding in Content-Aware Networks: Research Challenges and Open Issues," in: N. Blefari-Melazzi, G. Bianchi, and L. Salgarelli (eds.), *Trustworthy Internet*, Springer, pp. 349-358, June 2011.

[7] M. Wien, H. Schwarz, and T. Oelbaum, "Performance Analysis of SVC," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1194-1203, September 2007.

[8] MPEG output document N12957, "Joint Call for Proposals on Scalable Video Coding Extensions of High Efficiency Video Coding (HEVC)," *ISO/IEC JTC 1/SC 29/WG 11/N12957*, Stockholm, Sweden, July 2012. Available online "http://mpeg.chiariglione.org/working_documents/mpeg-h/hevc/HEVC_ext_CfP.zip", accessed Nov. 11, 2012.

[9] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Conference Proceedings on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Palo Alto, California, United States, pp. 117-130, October 1996.

[10] M. Eberhard, T. Szkaliczki, H. Hellwagner, L. Szobonya, C. Timmerer, "An Evaluation of Piece-Picking Algorithms for Layered Content in Bittorrent-based Peer-to-Peer Systems," *Proc. IEEE Multimedia and Expo (ICME'11)*, Barcelona, Spain, July 2011.

[11] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet", *IEEE Multimedia*, vol. 18, no. 4, pp. 62-67, October-December 2011.

[12] Z. C. Zhang and V. O. Li, "Router-assisted layered multicast," in *Proceedings of IEEE International Conference on Communications (ICC)*, vol. 4, pp. 2657-2661, 2002.

[13] IETF RFC 3973, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", IETF Request for Comments, Jan. 2005.

- [14] A. Kayssi, H. Karaki, and W. Abu-Khrybeh, "RTP-based caching of streaming media," in *Proceedings of Eighth IEEE International Symposium on Computers and Communication (ISCC)*, pp. 1067-1071 vol.2, July 2003.
- [15] Y. Sánchez de la Fuente et al., "iDASH: improved dynamic adaptive streaming over HTTP using scalable video coding," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, New York, NY, USA, pp. 257-264, February 2011.
- [16] S. Lederer, C. Müller, and C. Timmerer, "Towards peer-assisted dynamic adaptive streaming over HTTP," in *Proceedings of 19th International Packet Video Workshop (IPV)*, pp. 161-166, May 2012.
- [17] P. Le Callet, S. Möller, and A. Perkis (eds.), "Qualinet White Paper on Definitions of Quality of Experience," European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Lausanne, Switzerland, June 2012.
- [18] W. Cherif, A. Ksentini, D. Negru, M. Sidibe, "A_PSQA: PESQ-like non-intrusive tool for QoE prediction in VoIP services," in *Proceedings of IEEE International Conference on Communications (ICC)*, Ottawa, Canada, June 2012.

BIOGRAPHIES

Michael GRAFL (michael.grafl@itec.aau.at) is a research assistant at the Institute of Information Technology, Alpen-Adria-Universität Klagenfurt, Austria. He is currently working towards a Ph.D. degree. He works in the scope of the ICT FP7 project ALICANTE and participates in standardization activities as an ISO/IEC MPEG committee member. His research interests include distributed multimedia adaptation, SVC tunneling, and multimedia service platform technologies.

Christian TIMMERER (christian.timmerer@itec.aau.at) is an assistant professor in the Multimedia Communication Group at Alpen-Adria-Universität Klagenfurt, Austria. His research interests include immersive multimedia communication, streaming/adaptation in heterogeneous environments, and Quality of Experience. He was the general chair of WIAMIS'08 and QoMEX'13. He participated in several EC-funded projects, notably ALICANTE, and in ISO/MPEG, notably MPEG-21, MPEG-V, and MPEG-DASH.

Hermann HELIWAGNER (hermann.hellwagner@itec.aau.at) is a full professor of Informatics in the Institute of Information Technology, Alpen-Adria-Universität Klagenfurt, Austria, leading the Multimedia Communications group. His research areas are distributed multimedia systems, multimedia communications, and quality of service. He is a senior member of the IEEE, member of the ACM, GI, OCG, and of the Scientific Board of the Austrian Science Fund.

George XILOURIS (xilouris@iit.demokritos.gr) was born in Athens, Greece in 1976. He received his BSc in Physics from University of Ioannina in 1999 and his MSc in Automation Control Systems from National Technical University of Athens in 2001. His current research activities include Digital Broadcasting, Media Networking, Network Management, and IT services.

Georgios GARDIKIS (gardikis@iit.demokritos.gr) received his BSc and PhD in Electrical and Computer Engineering from the National Technical University of Athens in 2000 and 2004 respectively. His expertise lies in the fields of digital broadcasting, distribution networks for multimedia services, QoE assessment and application/network coupling. He is currently Researcher at NCSR "Demokritos", having more than 50 publications in international journals and conferences.

Daniele RENZI (daniele@bsoft.net) received the MSc in electronic engineering from University Politecnica delle Marche, Italy, in 2003. Between 2003 and 2005 he was a system engineer with GEM Electronics, Italy. From 2005 to 2012 he was a software engineer with bSoft, Italy. Currently he's scientific assistant at EPFL, Switzerland. His main research topics are video coding, Quality of Experience and multimedia streaming.

Stefano BATTISTA (bautz@bsoft.net) received the M.S. degree in electronic engineering from the University of Ancona, Italy, in 1990. In 1999 he founded bSoft, Macerata, Italy, where he is a system engineer and project manager. His main activities are on video and audio coding, and software for multimedia systems. He actively contributed to the standardization activities of ISO/IEC MPEG since 1994.

Eugen BORCOCI (eugen.borcoci@elcom.pub.ro) is a professor at Electronics, Telecommunications and Information Technology Faculty of University "Politehnica" of Bucharest (UPB). His areas of interest are in FI architecture, network and services management, protocols, QoS, CON/CCN. He participated to many FP5, FP6, FP7 European research projects and has authored or co-authored 5 books and more than 130 scientific papers/ reports in the above areas.

Daniel NEGRU (daniel.negru@labri.fr) received his PhD in 2006 from the University of Versailles, France, in the field of Broadcast and Internet convergence solutions at the network and service levels. From 2007, he joined the University of Bordeaux as an associate professor and conducted research in media-aware networks, including content-, context- and network-awareness. He is coordinating the ICT FP7 ALICANTE project.