Scalable Media Coding Enabling Content-Aware Networking

Michael Grafl, Christian Timmerer, and Hermann Hellwagner
Alpen-Adria-Universität, Austria

George Xilouris and Georgios Gardikis
National Centre for Scientific Research Demokritos, Greece

Daniele Renzi and Stefano Battista
bSoft, Italy

Eugen Borcoci
University Politehnica of Bucharest, Romania

Daniel Negru
University of Bordeaux, France

Content-aware network processing will allow the underlying infrastructure to identify, process, and manipulate media streams and objects in real time to maximize quality of service and experience.

The Future Internet (FI) development has raised a rich set of research issues given the huge, global impact of this technology and new societal needs for media services.[1] The term FI encompasses a range of activities to improve the architecture of the current Internet—an Internet characterized by many ad hoc solutions and technologies that were designed for purposes different from their actual deployment. Future developments must address long-term goals toward the Internet's full potential. We have witnessed a significant trend toward information-centric services, and consequently, new challenges are emerging. In particular, significant changes in communications and networking have been 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 toward the FI and the basics of scalable media formats. The new conceptions are generally divided into revolutionary and evolutionary approaches. The *revolutionary* (or clean-slate) approaches are often referred to as *information-centric networking* (ICN), which is an umbrella term for related concepts such as content-oriented networking (CON) and content-centric networking (CCN).[2,3] *Evolutionary* (or incremental) approaches, on the other hand, such as content-aware networking (CAN), attempt to build on existing Internet infrastructures.

In this article, we explain the role of CAN for multimedia services in more detail. We present four media streaming use cases that characterize different requirements with respect to content-aware network processing and highlight the utility of scalable media formats.

Alicante Project

Clean-slate ICN approaches are promising,[1,3] but they raise a long list of research challenges, including the degree of preservation of the classic transport layering principles (such as TCP/IP), 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 (QoS) and quality of experience (QoE), and media flow adaptation. In additional, new business models are needed for users, content producers, consumers, and service/network providers, and deployment issues such as scalability, privacy, and compatibility with existing equipment become crucial.

In parallel, evolutionary approaches that will help us move toward the FI such as CAN are being proposed[4] and are being developed within the Alicante (Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments) Project (http://ict-alicante.eu). The goal of this work is to enable efficient routing and forwarding of content based on given content and context characteristics and also to enable content adaptation. Alicante deploys content- and context-aware strategies at the network edges.[5] A main challenge of evolutionary approaches is obviously overcoming the limitations of the current Internet.[1]

1070-986X/13/$31.00 © 2013 IEEE

Published by the IEEE Computer Society

The Alicante CAN environment attempts to optimize network resource utilization while maintaining the expected QoS and QoE:

- It establishes virtual networks on top of the physical infrastructure that feature inherent content awareness, for example, by dynamically providing network resources appropriate for different content types.
- It provides in-network media caching and real-time adaptation, exploiting scalable media coding formats, such as scalable video coding (SVC), which are vital components of this objective thanks to their compression efficiency and flexibility.[5]

Both of these functions are provided by enhanced network nodes, or *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 to achieve in-network media processing. SVC is an extension of MPEG-4 Advanced Video Coding (AVC) and requires a moderate compression overhead of approximately 10 percent over single-layer coding (AVC).[6] In SVC, the video bitstreams are encoded using a layered approach that consists of an AVC-compliant base layer providing the basic quality (such as frame rate, spatial resolution, and signal-to-noise ratio [SNR]) and one or more incrementally added enhancement layers. For example, the base layer provides the content quality needed for legacy or mobile devices (for example, 1,280 × 720 pixels). Then, high-definition quality (such as 1,920 × 1,080 pixels) and beyond can be reached with additional enhancement layers. Currently, the next generation of SVC is being developed within MPEG based on the High Efficiency Video Coding (HEVC) technology.[7]

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 enable dynamic adaptation toward available bandwidth.

Use Cases

We can begin by illustrating four use cases—unicast, multicast, peer-to-peer (P2P), and adaptive HTTP streaming—that highlight the benefits of using SVC in CAN.

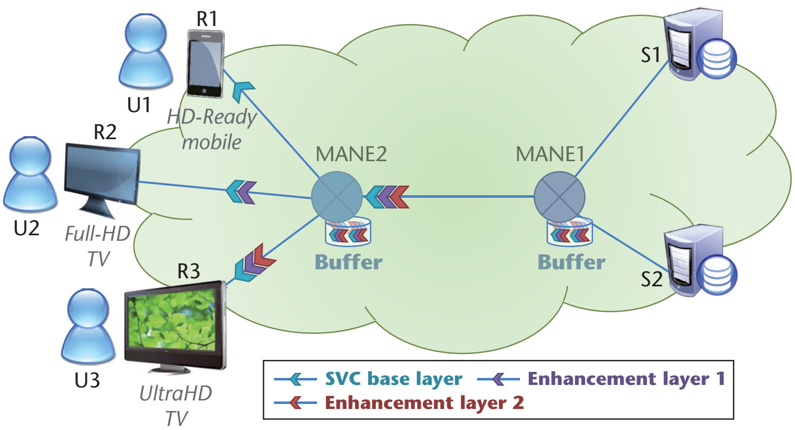


Figure 1. High-level system overview. The simplified system consists of two senders (S1, S2), two MANEs (MANE1, MANE2), and three receivers (R1, R2, R3) with different terminal and potentially different network capabilities. U1, U2, and U3 represent three connected end users.

Figure 1 depicts a simplified and generic high-level system overview for the use cases in question. The system consists of these entities: two senders (S1, S2), two MANEs (MANE1, MANE2), and three receivers (R1, R2, R3) with different terminal and potentially different network capabilities, to which three end users (U1, U2, U3) are connected. Our use case discussion addresses streaming of previously recorded content (such as video on demand [VoD]), unless noted otherwise. In more complex scenarios, more senders, more receivers, and additional MANEs distributed over multiple autonomous network domains may be deployed.

Unicast Streaming

For the unicast use case, we have only one sender (S1) that streams the scalable video content to a single receiver (R3), as in a traditional VoD 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, the Real-Time Transfer Protocol (RTP) and Real-Time Session Protocol (RTSP) are typically used. In the unicast use case, the SVC stream is typically sent via single-session transmission mode over RTP—that is, all SVC layers are packed into one RTP session.

Multicast Streaming

The second use case is multicast streaming, which is characterized by a single sender

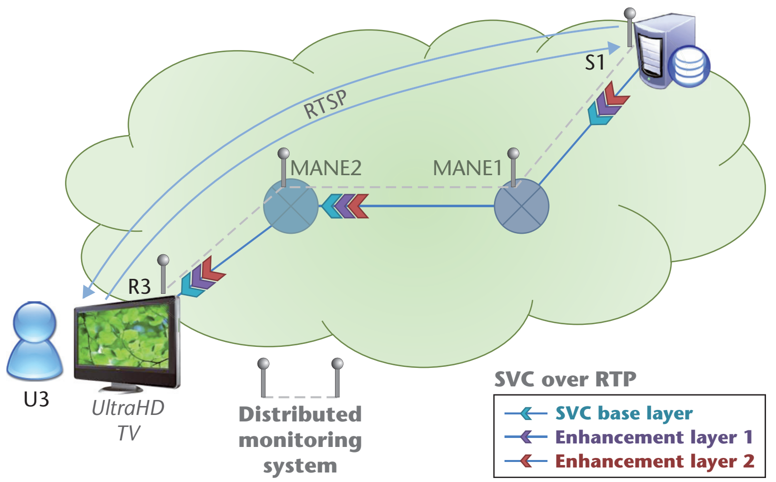


Figure 2. Unicast streaming in content-aware networks. In this example, a MANE can react to changing network conditions by dropping enhancement layers of the SVC stream.

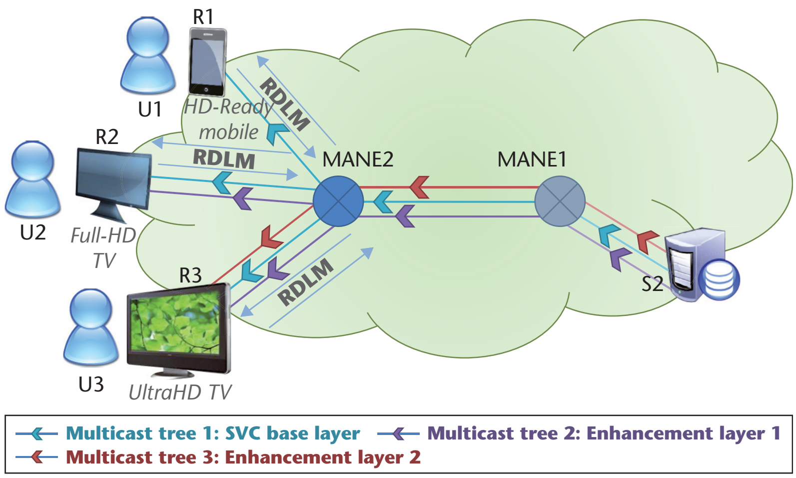


Figure 3. Multicast streaming in content-aware networks. Here, a single sender is providing the same content to multiple receivers, although they may be receiving different levels of content (from HD-ready mobile to UltraHD TV).

providing the same content to multiple receivers. In this case, one sender (for example, S2 in Figure 1) is streaming the content to heterogeneous trees of MANEs and subsequently to multiple receivers (such as R1, R2, and R3). The term *heterogeneous trees* denotes a set of trees allocated to different SVC layers. All the trees have the same root (for example, S2 in Figure 3) but different leaves, depending on the transported SVC layer. That is, Figure 3 shows that the SVC base layer is delivered to all receivers, while the highest SVC layer is received only by R3.

Scalable media formats enable the realization of this use case via receiver-driven layered multicast (RDLM),[8] and with SVC, this approach is becoming efficient enough to surpass simulcast.[5] With RDLM, different layers are transmitted over separate multicast groups. RTP realizes this via the multi-session transmission mode. SVC layers are separated into multiple RTP sessions at the sender side and rearranged to the proper SVC bitstream at the receiver side. Each receiver subscribes only 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 adaptation operations. MANEs can transparently neglect the video header information because the mapping of SVC layers to multicast groups is realized at a lower level, simplifying the content-adaptation process. In other words, a MANE simply adjusts the number of subscribed RTP sessions without having to inspect each and every RTP packet header.

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 more receivers consume the content. A scalable media format enables each receiver to request only the layers that are supported by its media player.

Compared with conventional P2P content distribution, P2P streaming has a timing constraint where every piece must arrive before its play-out deadline expires. P2P streaming systems typically use a sliding window of pieces that are currently relevant for the receivers. Within this sliding window, a piece-picking algorithm at the receiver side manages the downloading of pieces that provide the highest quality to the end user. That 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.[9]

Although a P2P system is traditionally organized as an overlay network that is transparent to the core network, a CAN will allow MANEs to participate in the streaming process in several ways. Figure 4 shows an 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, which we describe later.

Adaptive HTTP Streaming

The previous use cases have shown streaming scenarios with various numbers of senders and receivers. To overcome common shortcomings of RTP-based streaming, such as network address translation (NAT) and firewall issues, this use case introduces adaptive HTTP streaming in the context of CAN. With HTTP streaming, the content is typically fragmented into segments that are downloaded by the receiver via individual HTTP (partial) `GET` requests. This approach allows for a stateless sender and enables caching at the MANEs and dynamic content adaptation at the client at the same time. Based on several industry solutions, MPEG has recently standardized Dynamic Adaptive Streaming over HTTP (MPEG-DASH).[10]

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 manifest file of the content that describes the structure of the media segments and the available media representations. A media representation denotes a particular encoding configuration of the content, such as bit rate or resolution. For layered coding formats such as SVC, those representations can define either the individual layers or even subsets of bitstream layers. The receiver selects the appropriate representation based on its processing and 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 Figure 5 shows.

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 the sender and receivers cache the content segments for subsequent requests by other receivers, the result will be a multicast-like tree.

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. The receiver may select any of them to download the segments, thus balancing the load among the senders.

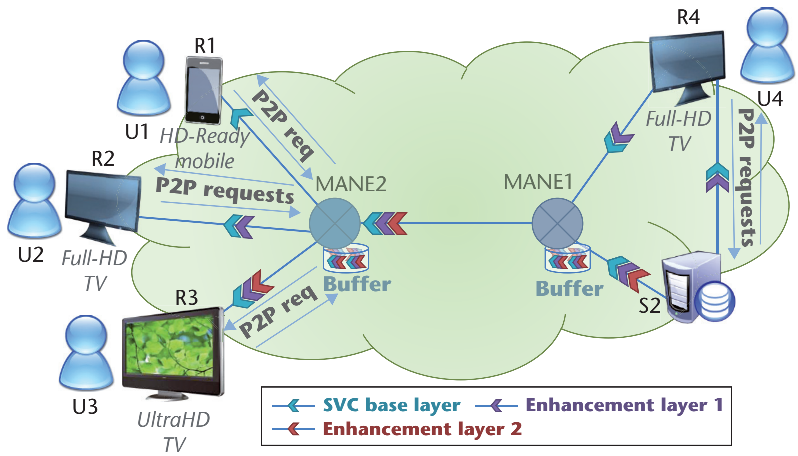


Figure 4. Peer-to-peer (P2P) streaming in content-aware networks. In this scenario, a MANE can cache pieces in a content-aware manner or act as a peer itself.

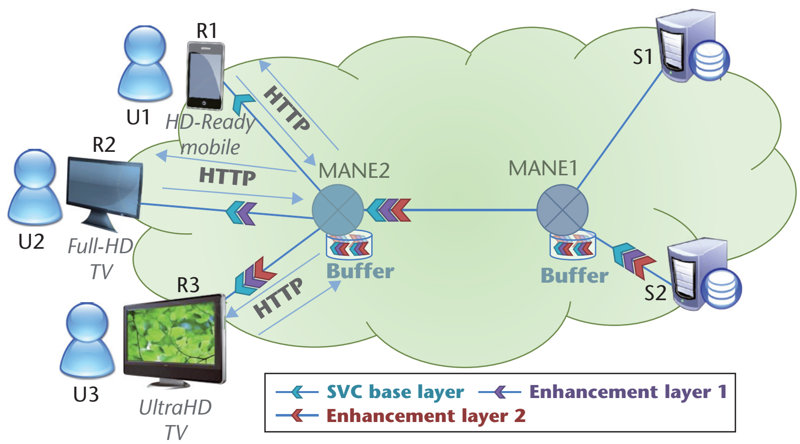


Figure 5. Adaptive HTTP streaming in content-aware networks. MANEs along the network path can act as caches or as content delivery network (CDN) nodes.

Use-Case Analysis

We can apply CANs to each of the multimedia streaming use cases we have described here. This section provides an analysis concerning CAN operations, such as flow processing, caching and buffering, and QoS/QoE management for the use cases in question and presents some recent scientific advances.

Flow Processing

In the unicast use case, the use of scalable media formats such as SVC in a CAN brings three main advantages.

> **A MANE can perform efficient in-network adaptation of the content in reaction to network fluctuations.**

First, the sender can easily adapt the content to the receiver's capabilities by sending only those layers that the receiver supports (for example, 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 media stream, assuring continuous play-out 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 play-out to either stall or show too many visual artifacts as a result of a high packet-loss rate. As soon as the network conditions return to normal, the MANE can return the initial 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 of which SVC layers to drop or forward is solved by an *adaptation decision-taking engine* (ADTE). The ADTE is not specific to the SVC adaptation but is used to steer 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 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. Although in-network adaptation is a good solution for mitigating short-term network fluctuations, it wastes bandwidth between the sender and the MANE during long 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 roundtrip 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.[8] That is, the receivers control the subscriptions to multicast groups. Hence, in-network adaptation is achieved implicitly because the receiver controls it through subscription to appropriate SVC layers. MANEs aggregate and combine subscriptions from downstream entities—both receivers and MANEs—using them to subscribe to appropriate SVC layers upstream. Alicante adopts and extends the RDLM approach for video content distribution in multicast-based scenarios.

There are two ways for MANEs to assist with the network-aware adaptation of multicast streaming: Downstream forwarding of one or more SVC layers can be temporarily truncated in case of congestion at an outgoing link,[11] or a MANE can control multicast group subscriptions by sending prune or graft messages to upstream neighbors as defined in RFC 3973.[12]

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 in Alicante. Furthermore, Alicante supports traffic engineering as well as content and service classification and differentiation mechanisms (such as DiffServ and MPLS) that enable selective treatment of SVC layers, for example, by increasing priority and the robustness of the base layer.

For the P2P streaming use case, a MANE may act as a peer, autonomously requesting pieces that it deems relevant for any connected receivers. Running a P2P engine on a MANE increases **the MANE's** processing requirements, but it also offers a flexible, 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.

The aforementioned flow-processing policies are also applicable to adaptive HTTP streaming with some noticeable differences. TCP uses reliable transmission that is unsuitable for in-network adaptation achieved through enhancement-layer dropping. If a MANE simply dropped 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 play-out deadline.

Thus, for HTTP streaming, a MANE acts as a (transparent) proxy cache in combination with CDN functionality. Because 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 the receiver supports. 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, such adaptation decisions (the selection of which SVC layers to forward) are 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 those adaptation decisions is still an open research challenge.[5]

Intelligent buffering at MANEs along the network path between sender and receivers constructs a bandwidth-efficient multicast tree.

Caching and Buffering

MANEs can buffer previously requested content and may even act as CDN caches by 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 content, the MANE starts streaming from its cache while requesting the content's suffix from the sender.[13]

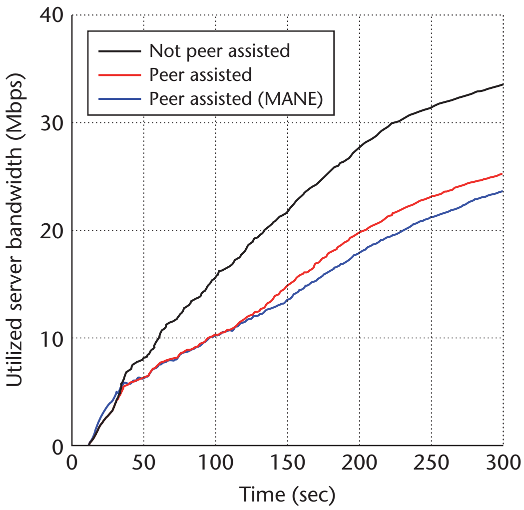
The use of SVC offers a trade-off between quality and availability to the MANE. The prefix cache may contain only the base layer for less popular content. Thus, the end user starts receiving only the base layer, but with a low start-up delay, and later the enhancement layers from the sender are added.

Proactive caching can also be used in the multicast use case to mainly reduce 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 the receivers share the same time window for the content; thus, each piece will be highly popular for a short time. By buffering a piece during this timeframe, the MANE can reduce network utilization and latency even with a limited buffer size. In most cases, such behavior is transparent to the peers within a 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; we call this *content-aware buffering*. Unlike conventional buffering, the MANE may intercept requests and transmit a buffered piece instead of forwarding the requests. This approach would constitute an evolutionary implementation of the CCN functionality.[2] A small drawback of

Figure 6. Simulation of peer-assisted HTTP streaming with MANEs as peers. The original server bandwidth for asymmetric connection speeds of peers is labeled peer assisted, *and the server bandwidth for symmetric connection speeds is labeled* peer assisted (MANE).



this approach is that the peer selection of the first receiver might not always be the optimal selection. However, 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 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 earlier. In contrast to RTP-based streaming, HTTP streaming immediately benefits from existing HTTP caching infrastructures[14] that may be deployed on top of CANs. The multicast mode relies on buffering and request aggregation at the MANE for bandwidth-efficient streaming. As we mentioned earlier, intelligent buffering at MANEs along the network path between sender and receivers constructs a bandwidth-efficient multicast tree. 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 is typically the case for live streaming services.[10] On the other hand, the MANE must be aware of the streaming session to buffer the segments accordingly. The straightforward solution is for the MANE to parse the manifest file and retrieve such information from there. An alternative solution would be that the MANE learns 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 streaming session's sliding window. An open research challenge is the impact of the discussed request aggregation on the load-balancing strategies between the senders.

In a recent study, Stefan Lederer and his colleagues proposed a peer-assisted HTTP streaming architecture compliant with MPEG-DASH.[15] 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 if clients have asymmetric Internet connections with significantly lower uplink bandwidth than downlink bandwidth, this solution reduces server bandwidth by up to 25 percent.

Although that method focuses on conventional client peers,[15] MANEs can act as peers just as well. Because 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 Lederer and his colleagues,[15] except that the 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 content's maximum bit rate was set to 1,400 kbps.

Figure 6 shows the simulation results of the server bandwidth requirements over time. The original server bandwidth for asymmetric connection speeds of peers is labeled *peer assisted*, and the 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 percent. This 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.

Deploying SVC in HTTP streaming also brings benefits to caching and buffering mechanisms. Although HTTP streaming of nonlayered media formats requires switching between different content representations (such as frame rate, resolution, and quality) for adaptation, SVC-based adaptation is performed by adding and 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. An earlier set of simulations compared the combination of SVC-based HTTP streaming and a streaming-optimized caching strategy to AVC-based streaming under the least-recently used (LRU) strategy.[14] Those results show that the cache hit ratio can be increased by up to 11.5 percentage points for congestion in the cache feeder link (the link between the sender and the cache) and by up to 25.7 percentage points for congestion in the access links.

Adaptive HTTP streaming will become increasingly popular due to its relatively easy deployment.

QoS/QoE Management

A primary goal of CAN is to manage and optimize both the QoS and QoE at the application level. The term QoS describes network properties that influence the transport of media. Metrics such as delay, packet loss, and jitter help measure QoS. The more recently coined term QoE targets the user's degree of delight or annoyance with an application or service. Besides QoS parameters, user-related factors (expectations) as well as terminal capability and performance play a role in QoE. QoE is typically measured as a mean opinion score (MOS) based on user ratings. (More information on QoS and QoE is available elsewhere.[16])

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 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 aggregated at the CAN level to manage the network behavior and establish 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 include a description of typical resolutions, which (and how many) bit rates to use for each resolution, appropriate scalability modes (temporal, spatial, or SNR), how to combine these modes, and the differences among use cases.

On the other end of the media delivery chain, the project is investigating 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),[17] which uses a continuous QoE score ranging from 1 (excellent) to 0 (bad) to estimate video quality based on packet-loss characteristics. The experimental setup uses SVC streams with 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 to trigger adaptation and enhance the granularity by which the system reacts to context variations. Thus, QoE evaluations are a vital part of advanced adaptive media delivery systems.

As we already mentioned, SVC enables a fine-grained control over the QoE at the network level. A nonscalable media format will suffer from severe QoE degradation if not all

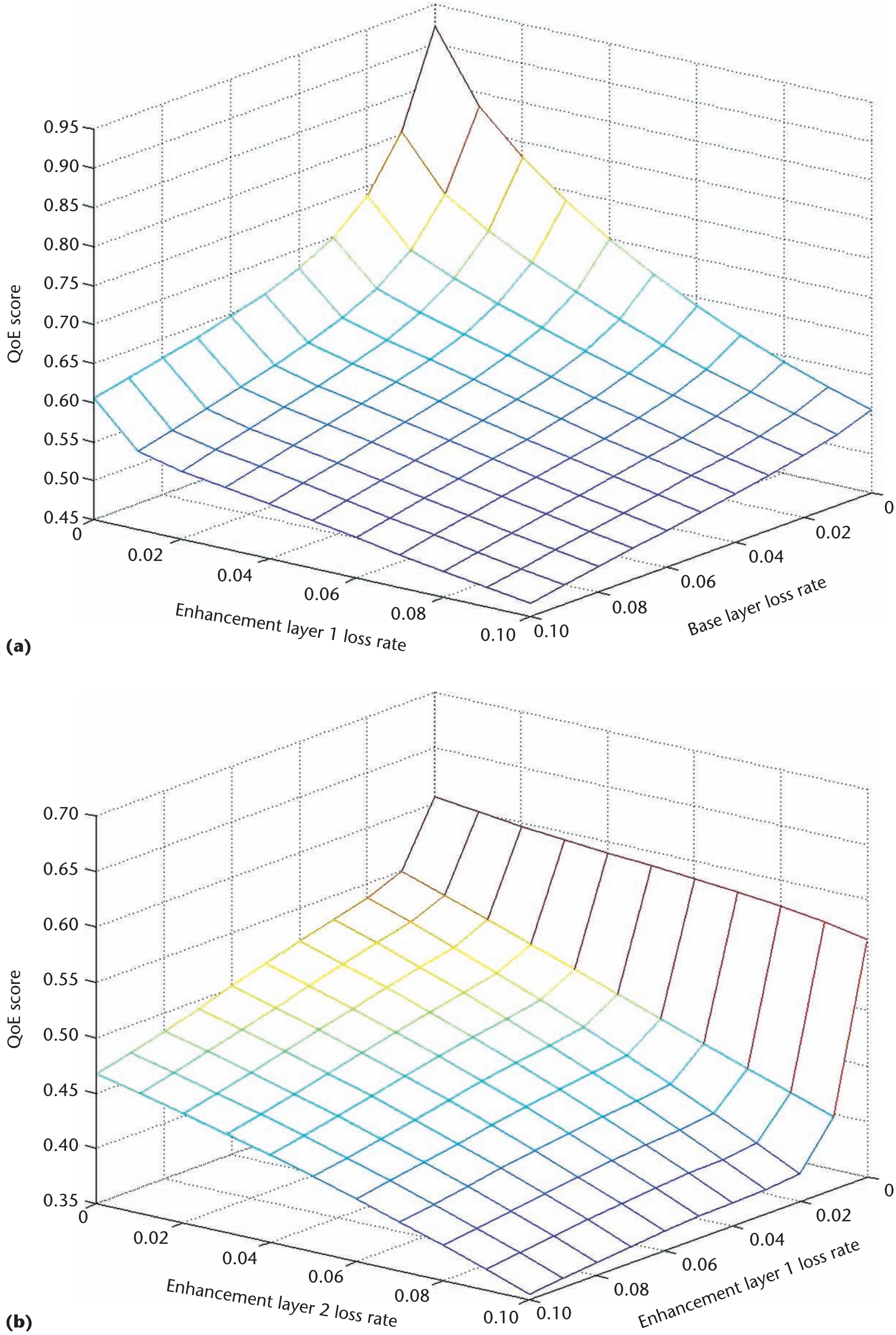


Figure 7. Quality of experience (QoE) scores versus (a) loss rate at the SVC base layer and enhancement layer 1 and (b) loss rate at the enhancement layers 1 and 2 with a base layer loss rate of 10 percent.

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

	CAN challenge		
Use case	**Flow processing**	**Caching and buffering**	**QoS/QoE management**
Unicast	Explicit adaptation Signaling adaptation decision to sender	SVC-based prefix caching for low start-up delay	Local and aggregated monitoring of network conditions Smooth, undistorted play-out via SVC
Multicast	Implicit adaptation Multicast bridges for hybrid multicast Differentiated forwarding of SVC layers	SVC-based prefix caching for low start-up delay No buffering	
P2P streaming	Explicit adaptation Peer for CAN P2P overlay network	Aggregating requests Content-aware buffering of pieces within sliding window	
Adaptive HTTP streaming	Implicit adaptation Transparent proxy cache	SVC-based prefix caching Multicast: buffering within sliding window Multisource streaming: aggregating requests and content-aware buffering within sliding window	

the packets in the stream are transmitted. With SVC, lower layers can be prioritized, maintaining smooth and undistorted play-out with controlled QoE degradation. SVC can also be conveniently combined with error-recovery techniques at the decoding side to further enhance the user's QoE.

Table 1 summarizes CAN-related challenges we have discussed here for each of the described use cases. For QoS/QoE management, we make no explicit distinction between the use cases.

Conclusions

Interesting challenges remain in this area, such as the integration of on-the-fly QoE evaluations of SVC content for adaptive media streaming and further improvements to the involvement of MANEs in P2P streaming. Future trends indicate more advanced video compression technologies targeting resolutions beyond 1,920 × 1,080 pixels (such as a new scalable extension for HEVC), so efficient and reliable buffering at MANEs will become increasingly important to reduce overall network loads. Furthermore, adaptive HTTP streaming will become increasingly popular due to its relatively easy deployment. Therefore, our future work will focus on how MANEs can further improve the existing HTTP infrastructure. **MM**

Acknowledgments

This work was supported in part by the European Commission's Alicante Project (FP7-ICT-248652). We thank Stefan Lederer for his help with simulation results.

References

1. J. Pan, S. Paul, and R. Jain, ''A Survey of the Research on Future Internet Architectures,'' *IEEE Comm. Magazine,* vol. 49, no. 7, 2011, pp. 26–36.
2. V. Jacobson et al., ''Networking Named Content,'' *Proc. 5th Int'l Conf. Emerging Networking Experiments and Technologies* (CoNEXT), ACM, 2009, pp. 1–12.
3. J. Choi et al., ''A Survey on Content-Oriented Networking for Efficient Content Delivery,'' *IEEE Comm. Magazine,* vol. 49, no. 3, 2011, pp. 121–127.
4. H. Koumaras et al., ''Media Ecosystems: A Novel Approach for Content-Awareness in Future Networks,'' *Future Internet: Achievements and Promising Technology,* Springer Verlag, 2011, pp. 369–380.
5. M. Grafl et al., ''Scalable Video Coding in Content-Aware Networks: Research Challenges and Open Issues,'' *Trustworthy Internet,* N. Blefari-Melazzi, G. Bianchi, and L. Salgarelli, eds., Springer, 2011, pp. 349–358.
6. M. Wien, H. Schwarz, and T. Oelbaum, ''Performance Analysis of SVC,'' *IEEE Trans. Circuits and*

Systems for Video Technology, vol. 17, no. 9, 2007, pp. 1194–1203.
7. "Report of Results of the Joint Call for Proposals on Scalable High Efficiency Video Coding (SHVC)," MPEG output document N13158, ISO/IEC JTC 1/SC 29/WG 11/N13158, Oct. 2012; http://mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding/report-results-joint-call-proposals-scalable-high.
8. S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-Driven Layered Multicast," *Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm.* (SIGCOMM), ACM, 1996, pp. 117–130.
9. M. Eberhard et al., "An Evaluation of Piece-Picking Algorithms for Layered Content in Bittorrent-Based Peer-to-Peer Systems," *Proc. IEEE Multimedia and Expo* (ICME 11), IEEE CS, 2011, pp. 1–6.
10. I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE Multimedia,* vol. 18, no. 4, pp. 62–67, October-December 2011.
11. Z.C. Zhang and V.O. Li, "Router-assisted Layered Multicast," *Proc. IEEE Int'l Conf. Comm.* (ICC), vol. 4, 2002, pp. 2657–2661.
12. A. Adams, J. Nicholas, and W. Siadak, *Protocol Independent Multicast—Dense Mode (PIM-DM): Protocol Specification* (*Revised*), IETF RFC 3973, Jan. 2005.
13. A. Kayssi, H. Karaki, and W. Abu-Khraybeh, "RTP-Based Caching of Streaming Media," *Proc. 8th IEEE Int'l Symp. Computers and Comm.* (ISCC), vol. 2, IEEE CS, 2003, pp. 1067–1071.
14. Y. Sánchez de la Fuente et al., "iDASH: Improved Dynamic Adaptive Streaming over HTTP Using Scalable Video Coding," *Proc. 2nd Ann. ACM Conf. Multimedia Systems,* ACM, 2011, pp. 257–264.
15. S. Lederer, C. Müller, and C. Timmerer, "Towards Peer-Assisted Dynamic Adaptive Streaming over HTTP," *Proc. 19th Int'l Packet Video Workshop* (PV), IEEE, 2012, pp. 161–166.
16. P. Le Callet, S. Möller, and A. Perkis, eds., "Qualinet White Paper on Definitions of Quality of Experience," white paper, European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), 2012.
17. W. Cherif et al., "A_PSQA: PESQ-Like Non-intrusive Tool for QoE Prediction in VoIP Services," *Proc. IEEE Int'l Conf. Comm.* (ICC), IEEE CS, 2012, pp. 2124–2128.

Michael Grafl is a research assistant in the Multimedia Communication Group at Alpen-Adria-Universität (AAU) Klagenfurt, Austria, and a doctoral student at AAU. His research interests include distributed multimedia adaptation, SVC tunneling, and multimedia service platform technologies. Grafl has an MSc in computer science from AAU. Contact him at michael.grafl@itec.aau.at.

Christian Timmerer is an assistant professor in the Multimedia Communication Group at Alpen-Adria-Universität (AAU) Klagenfurt, Austria. His research interests include immersive multimedia communication, streaming/adaptation in heterogeneous environments, and quality of experience. Timmerer has a PhD in computer science from AAU. Contact him at christian.timmerer@itec.aau.at.

Hermann Hellwagner is a professor of Informatics in the Institute of Information Technology, Alpen-Adria-Universität Klagenfurt, Austria, where he leads the Multimedia Communication Group. His research interests include distributed multimedia systems, multimedia communications, and quality of service. Hellwagner has a PhD in Informatics from the University of Linz, Austria. He is a senior member of IEEE and a member of the ACM and the Scientific Board of the Austrian Science Fund (FWF). Contact him at hermann.hellwagner@itec.aau.at.

George Xilouris is an associate researcher at the National Centre for Scientific Research Demokritos. His research interests include digital broadcasting, media networking, network management, and IT services. Xilouris has an MSc in automation control systems from the National Technical University of Athens. Contact him at xilouris@iit.demokritos.gr.

Georgios Gardikis is a researcher at the National Centre for Scientific Research Demokritos. His research interests include digital broadcasting, distribution networks for multimedia services, QoE assessment, and application/network coupling. Gardikis has a PhD in electrical and computer engineering from the National Technical University of Athens. Contact him at gardikis@iit.demokritos.gr.

Daniele Renzi is a scientific assistant at EPFL, Switzerland. His research interests include video coding, quality of experience, and multimedia streaming. Renzi has an MSc in electronic engineering from University Politecnica delle Marche, Italy. Contact him at daniele@bsoft.net.

Stefano Battista is the founder, system engineer, and project manager of bSoft, Italy. His research interests include video and audio coding and software for multimedia systems. Battista has an MS in electronic engineering from the University of Ancona. Contact him at bautz@bsoft.net.

Eugen Borcoci is a professor with the Electronics, Telecommunications and Information Technology Faculty at the University ''Politehnica'' of Bucharest (UPB). His research interests include Future Internet architecture, network and services management, protocols, QoS, and CON/CCN. Borcoci has a PhD in telecommunications from UPB. Contact him at eugen.borcoci@elcom.pub.ro.

Daniel Negru is an associate professor at the University of Bordeaux. His research interests include media-aware networks, including content-, context- and network-awareness. Negru has a PhD in convergence of networks at management and service levels from the University of Versailles, France. He is coordinating the ICT FP7 ALICANTE Project. Contact him at daniel.negru@labri.fr.