Multi-interface Extension to a Scalable Video Streaming Architecture

Tiia Sutinen, Helena Rivas
VTT Technical Research Centre of Finland, Espoo, Finland
Email: {tiia.sutinen, helena.rivas}@vtt.fi

Abstract—Video service adaptation capabilities are essential for the efficient utilization of the network resources in heterogeneous multi-access environments and ensuring sufficient perceived service quality for the end users. Multi-homing is seen as one important enabler for high-quality and reliable video streaming services and it is expected to be supported extensively in the future Internet. Already today, terminal devices are equipped with multiple network interfaces and able to use them simultaneously. In the future, mobile multimedia services such as video streaming can maximize the user experienced quality by utilizing the available network resources, concurrently. However, intelligent decision-making as well as dynamic mapping of traffic to network interfaces are required for optimal operation. In this paper, we introduce an overall architecture for adaptive video streaming exploiting the novel scalable video coding technology. We also propose a multi-interface streaming extension to the architecture to allow dynamic and concurrent usage of the available access networks in the video service delivery. We present a prototype implementation of the proposed multi-interface video streaming system and test its operation in the presence of network impairments in an experimental evaluation.

Index Terms—Heterogeneous networks, multi-homing, SVC, video QoS, adaptation.

I. INTRODUCTION

Heterogeneous multi-access network environments where several access networks and technologies (e.g. WLAN, WiMAX, and HSPA) are at the disposal of a mobile user simultaneously create both challenges and opportunities for service development. This is especially true for Quality of Service (QoS) sensitive multimedia services, such as video streaming, that require certain throughput and delay characteristics from the network for reliable and acceptable operation. Whereas the delay requirements are bound to the application (e.g. real-time video conferencing vs. streaming of stored content in a VoD service), the bitrate of the video stream may be adapted to match with the throughput characteristics of the network; thus increasing the number of access options that can be considered to be used for accessing the service. To ease video bitstream adaptation and increase the flexibility of video services targeted to heterogeneous network environments, novel coding mechanisms such as the Scalable Video Coding (SVC) extension of H.264/AVC [1] have appeared. Bitrate adaptation, nevertheless, comes with the cost of degraded visual quality, which can be considered as acceptable up to a certain degree depending on the content as well as the user’s context [2].

In order to fully utilize the potential of heterogeneous multi-access network environments, the services should be able to flexibly combine the resources of the available access networks in their data transmission. The simultaneous usage of multiple access networks is doable already with today’s terminal devices supporting multiple network interfaces, making the concurrent utilization of multiple access networks in carrying the user’s traffic a feasible solution for coping with resource shortage in individual networks. The term ‘multi-homing’ is widely used for referring to the ability of a terminal to be connected to and communicating over multiple IP networks at the same time. In this paper, we also use the term ‘multi-interface streaming’ to indicate the reception of a video stream through multiple network interfaces of the end-user terminal, simultaneously. Although there are no standard solutions for implementing both IP mobility and multi-homing today, extensions have been proposed to protocols such as Mobile IP (MIP) and Host Identity Protocol (HIP) to support the functionality. Therefore, it can be expected that multi-homing will become an important feature and enabler for the future Internet.

In this paper, we present an architecture for adaptive transmission of scalable video streams in heterogeneous network environments. The architecture was specified in the EUREKA/Celtic SCALNET project [3] and it defines the elements and functions required for scalable video stream adaptation in the different parts of the end-to-end transmission system, that is, the service provider’s servers, the transport and access networks, and the end-user terminals or home networks. The SCALNET architecture supports video streaming towards the user terminal essentially over one access network at a time. Therefore, the aim of this paper is to define the required extensions to the SCALNET system components and architecture to implement support for multi-interface scalable streaming. This paper is based on the results reported in [4] extending them to the context of the overall SCALNET architecture.

Our proposed approach for implementing the multi-interface streaming support into the SCALNET system
architecture includes also a mechanism for network interface selection. It allows the streaming client to select the appropriate network interfaces for the different SVC layers, dynamically. Certain level of dynamicity is required to ensure optimal routing of SVC layers over the available networks, of which availability or characteristics may change over time; for example, with host mobility or the activity of the other users. The dynamic network interface selection solution utilizes cross-layer information in its decision-making. In this paper, we present a prototype implementation of our solution and experimental results demonstrating its operation in response to different types of changes in the network conditions.

The rest of the paper is organized as follows: First, we discuss the related work in Section II. Then in Section III, we present the overall SCALNET system architecture that is used as the basis for the multi-interface streaming system. The modifications to the SCALNET architecture and components required for implementing the multi-interface streaming support are discussed in Section IV. In Section V, we present the first-pass prototype implementation of the multi-interface scalable video streaming system. The prototype’s experimental evaluation and the obtained results are presented in Section VI. Section VII finally concludes the paper with some insights regarding the future work.

II. RELATED WORK

Today, several algorithms and architectures exist for implementing adaptive video services, ranging from traditional sender-based mechanisms such as the TCP Friendly Rate Control [5] to more complex architectures supporting adaptation on multiple protocol layers [6]. Most of them, however, consider video adaptation over a single network path. The benefit of heterogeneous multi-access environments is that they create a potential for a terminal with multiple network interfaces to remain connected to the used networked services via the best access anytime and anywhere (i.e. to stay ’always best connected’ [7]). Thus, in situations where a single access network is not able to provide the required QoS for the used services and the desired Quality of Experience (QoE) for the user, simultaneous usage of two or more networks for service delivery may be useful in order to optimize the QoE.

Various solutions for implementing multi-homing with mobility support for IP hosts have been proposed in the literature. In [8], multi-homing and simultaneous multi-access (SIMA) is enabled with a HIP-based solution, whereas the paper [9] utilizes simultaneous transmission over two network paths to improve handover performance. With mobility we mean the ability of Internet hosts to change their point of attachment to the Internet (i.e. IP address) without disrupting ongoing communications; and with multi-homing, the situation where a host equipped with multiple network interfaces may communicate using several IP addresses, simultaneously.

Prior works studying the advanced streaming capabilities of SVC mainly focus in the protocol-level implementation of the mechanism in order to solve the problem of splitting SVC layers to separate RTP (Real-time Transport Protocol) sessions and combining them at the receiver. The basic idea of multisession transmission (MST) of SVC is presented in [10]. The paper [11], on the other hand, addresses solutions for decoding order recovery after multi-session transmission. Some experimentation on the concept is also presented in [12] including a simulation study on a solution for splitting an SVC stream over multiple network connections depending on their capacity.

In heterogeneous networks and especially in a mobile context, cross-layer information that reflects the overall system status can provide significant performance improvements to both mobility management and video streaming systems. This has led to the development of several cross-layer signalling frameworks such as the IEEE 802.21 Media Independent Handover (MIH) Services [13] and the triggering framework [14]. These frameworks enable the collection and distribution of extensive cross-layer information within a terminal device as well as between remote network entities allowing the development of context-aware network elements and applications. However, not many solutions that implement cross-layer enhanced decision-making for routing SVC layers optimally in heterogeneous multi-access environments exist today. For instance, the solutions presented in [8] and [12] consider only limited information regarding the network status (i.e. application type -based mapping of flows to available networks or end-to-end available bandwidth estimates) in their decision-making. In [9], a more complete quality oriented mobility management framework for multimedia applications is presented but its validation is limited to simulations and no consideration for SVC streams was included.

In this paper, we present the SCALNET system architecture for adaptive scalable video streaming, targeted especially to real deployments with currently available technology. We also discuss the extension of the architecture to support multi-interface video streaming with cross-layer enhanced decision-making for optimizing SVC transmission in heterogeneous networks. Moreover, we present a first-pass prototype implementation of the system with related performance evaluation in the context of multi-interface streaming.

III. SCALABLE VIDEO STREAMING ARCHITECTURE

The overall SCALNET system architecture is depicted in Fig. 1. The SCALNET architecture provides for adaptive scalable video streaming across heterogeneous networks. The architecture consists of four network nodes, namely the streaming server, the user terminal running the streaming client, the Media Independent Handover (MIH) inner and the management server. In this section, we introduce the individual system components while the rest of this paper focuses in discussing the implementation of multi-interface streaming on top of the SCALNET system architecture. For a more detailed description of
the SCALNET architecture as well as for other implementation approaches and use cases, an interested reader is advised to refer to [3].

Figure 1. The SCALNET architecture.

A. Streaming Server

The SCALNET streaming server facilitates adaptive streaming of pre-encoded SVC videos over the UDP, RTP, and RTSP (Real Time Streaming Protocol) protocols. In addition, Session Description Protocol (SDP) is used for communicating the video stream characteristics to the client in the beginning of the session.

SVC is the scalability extension of the H.264/AVC standard (Annex G.) [1], and it may be used for easy implementation of adaptive video services that meet the diverse user, device or network requirements in heterogeneous network environments. In SVC, video data is encoded into layers with hierarchical dependencies. That is, only the SVC base layer is needed for decoding and the enhancement layers can be used to increase the spatial resolution, frame rate or fidelity of the video presentation. Moreover, the base layer of an SVC-encoded bitstream is backward compatible with H.264/AVC.

In the SCALNET system, the adaptation of SVC streams is incorporated into the SCALNET streaming server through specific modules, namely the Adaptation Decision Taking Engine (ADTE) and the SVC Filter. The ADTE implements different decision algorithms and makes decisions regarding the video streaming parameters (e.g. maximum resolution or bitrate for the SVC bitstream) according to client feedback. The ADTE’s decision is conveyed to the SVC Filter, which then adapt the scalable video bitstream accordingly (i.e. adds or drops enhancement layers). In case of legacy H.264/AVC clients, the ADTE’s decisions can be used to guide an SVC-to-AVC transcoder to provide a compatible and maximum quality stream to them.

The streaming server interfaces with the User and Context Management (UCM) to collect context information from the clients. Additional context information is also received through RTSP and RTCP (Real-time Transport Control Protocol).

B. Media-aware Network Element

As a deployment option, the SCALNET system architecture supports also Media-aware Network Elements for handling scalable video stream adaptation or transcoding in the network. The SCALNET MANE is a reduced version of the streaming server supporting only the most important protocols and modules for SVC bitstream adaptation. The MANE is implemented as a RTSP proxy and intended to be deployed within an access or a home network (e.g. integrated into a WLAN access point or a set-top-box). The MANE may be used for fast adaptation of SVC streams to local access network conditions.

C. Streaming Client

The SCALNET streaming client handles the de-packaging, decoding, and displaying of the video data received from the server using the UDP, RTP, and RTSP protocols. Different video player implementations can be considered to be used [3] as long as their network interfaces support the RTP payload format of SVC [15]. The choice for a decoder also ranges from freely available ones (e.g. JSVM and Open SVC) to commercial implementations. The streaming client provides RTCP feedback of the streaming performance to the server and interfaces with the User and Context Management to offer more extensive context information for the server and other interested entities (e.g. a mobility manager).

The client terminal device may be connected to the server through heterogeneous access technologies, including for instance, broadcast networks (DVB), wireless access (e.g. WLAN, WiMAX or HSPA), or fixed access (e.g. xDSL, cable or PLC). The user is also able to change devices in between a streaming session thanks to the manual session handover functionality supported by the system [3].

D. User and Context Management

For the collection, maintenance, and distribution of various context information reflecting the system status, the SCALNET architecture includes a User and Context Management Module (UCMM) in both the streaming server and the client machines. The context information may originate from various server and client side context information sources (CISs), such as the user (via the video player’s GUI), the video player, the video streaming session (RTCP and RTSP), a QoS monitoring tool, etc. The UCMMs store and process the received context information for a predetermined time and supply it to subscribed system entities, which may be located either on the different layers of the local protocol stack or on distinct nodes. The context information is provided in a defined format (e.g. XML) and through a generic interface.

The context information may be used within the system to facilitate different kinds of decision-making regarding a video streaming session. Such decisions include, for
example, those related to scalable video bitstream adaptation and handovers. For instance, the context information transfer between the UCM Server Module and the Administration and Management Server (AMS) allows the user to continue a paused stream on a different device after a successful login (i.e. the manual session handover functionality in [3]).

E. Service Administration

Finally, to support real deployment scenarios, the SCALNET architecture defines also service administration mechanisms that may be used to implement any required service provider control and management functions (e.g. access control and billing). The architectural entity dedicated for this is the AMS, which is contacted every time a user wants to start a video streaming session. The user credentials are provided to the AMS through a web portal, which also provides links to the available video files for the user. Thus, every video streaming session in the SCALNET system is initiated using the web portal. When the user selects a video file, the video client application is launched and it then contacts the streaming server to initiate the streaming session. Moreover, the AMS keeps a record of the static context information regarding the users, video content, devices, and networks. The AMS makes the static context information available to the video adaptation and session management purposes through the UCMM.

IV. Multi-Interface Extension

As today’s user terminals are typically equipped with multiple network interfaces that may be used also simultaneously for accessing networked services, a natural extension to the SCALNET system architecture would be the multi-interface streaming capability. Even more so, thanks to the inherent support of SVC for the streaming of individual layers in their own RTP sessions, the client is able to stop and restart individual SVC layers interface.

A. Modifications to the Video Streaming Components

In the multi-interface streaming scenario, the SCALNET system still implements SVC streaming over RTSP and RTP between the streaming server and the client terminal. However, in this case, the user terminal is equipped with multiple network interfaces; and both the SVC streaming server and the SVC streaming client are able to handle scalable video delivery over multiple RTP sessions. As the SVC layers are transported in their own RTP sessions, they can be easily routed through the different access networks the user terminal is connected to. In accordance to the current definition in [16], the SVC base layer should be sent using the H264 RTP payload format and the enhancement layers using the H264-SVC format.

The routing of the SVC layers or substreams can be enforced via a multi-homing protocol (e.g. MIP or HIP extended with multi-homing capabilities) or with simple routing table adjustments as suggested in [17]. In the latter, also the server has several (possibly virtual) network interfaces (i.e. IP addresses), over which it offers the different SVC layers. The client then adjusts its routing table entries accordingly so that requests and feedback to the server for a specific SVC layer go through the desired interface.

An example RTSP signalling sequence between the user terminal and the server in the case of a two-layer SVC stream is illustrated in Fig. 3. Before the streaming is started, the client obtains information about the video stream characteristics in the session description, which is provided through SDP by the server as a response to the RTSP DESCRIBE message. As shown in Fig. 4, the SDP contains information on a per SVC layer basis (i.e. SVC layer bitrate, framerate, dependencies, locations, etc.) as defined in [18]. Based on the obtained information, the client sends RTSP SETUP messages to the server regarding the SVC layers it wishes to receive. Once the streaming sessions have been set up, the client initiates the streaming by sending RTSP PLAY messages to the server for each SVC layer. During the streaming, the client is able to stop and restart individual SVC layers using RTSP requests.

Finally, RTCP is used for feedback information transfer between the client and server regarding the streaming performance. Each RTP flow should be associated with a RTCP feedback channel. The server may use the RTCP feedbacks in detecting problems in SVC layer transmission.

B. Dynamic Network Interface Management

A heterogeneous network environment with multiple options for access networks is very diverse and dynamic operation environment for video streaming services. Thus, the services will need to be able to adapt to changes in access network availability and QoS. For this, specific cross-layer assisted network interface management support is needed for the SCALNET streaming client to allow it to be able to make informed decisions regarding

![Figure 2. Multi-interface streaming of SVC.](image-url)
the mapping of the SVC layers to the available network interfaces as well as its subscription to individual layers. Therefore, we introduce a new element into the SCALNET SVC streaming client, depicted in Fig. 5, namely the Application QoS Policy Engine (AQPE). Clearly, the video stream adaptation approach is different in the multi-interface case from that of the basic SCALNET system operation; here the decision to adapt is taken in the client and not in the server or MANE as in the case of the ADTE.

The AQPE is part of the video streaming application and it controls the RTSP streaming client based on system status information. The client-side instance of the UCMM allows the AQPE to gather context information originating from various sources like network interface cards, QoS/QoE measurement tools, bandwidth estimation tools, mobility managers, etc. In addition, the AQPE itself is able to generate context information and distribute it to other entities in the system through the UCMM.

Regarding the decision-making, the AQPE can have different roles depending on the multi-homing or mobility solution used. We present here three options envisioned for the implementation.

1) If the terminal supports a lower layer multi-homing and SIMA implementation (e.g. similar to the one presented in [8]) with centralized policy management, the AQPE only needs to provide the stream requirements obtained through SDP to a centralized Mobility Manager (MM) entity via UCMM. The MM then makes the actual routing decisions based on the available context information. The decision is fed back to the AQPE so that it can initiate the streaming of the supported layers accordingly. In this case, the AQPE can remain agnostic of the underlying network and routing structures and operate based on the resources allocated to it.

2) In the case of the proprietary multi-homing solution presented in [17], the AQPE assumes also the responsibility of making the routing decisions as there is no separate MM entity in the system. For this, the AQPE collects context information from the UCMM in order to know the characteristics and status of the available networks. Based on the routing decision, the AQPE makes the needed routing table adjustments and instructs the RTSP streaming client to setup the selected SVC layers.
and to start their playout. Whenever there is a change in the context information, the AQPE needs to re-evaluate its decision and act accordingly (e.g. trigger SVC layer rerouting). However, rerouting of the base layer in this case requires additional mobility support from the underlying network to ensure session continuity.

3) Yet another option would be to extend the UCMM to handle decision-making regarding the SVC layer routing. Since the UCM collects the measurement and context information from different sources, it can be used also for making the decisions or providing the hints for other distributed decision making entities and other UCM components in the networked system. This would be a more generic approach for implementing the decision-making and it could also be linked to a distributed decision-making architecture like the one proposed in [19]. In this case, the role of the AQPE would be similar to the first case. Also the mobility management related decision-making would be ported from the MM to the UCMM.

Different decision-making algorithms can be used for mapping the SVC layers into the available network interfaces. Game theory, fuzzy logic, and genetic algorithms are examples of applicable technologies. For instance, one game theory modelling solution is presented in [20] to solve a network selection problem. A similar approach may be used also for our SVC-layer-to-network-interface mapping problem. In our case, the players are the available interfaces and the resources the SVC layers. Another game theory approach can be found in [21]. Here, the decisions are based not only on link layer information but also on the user’s preferences for a connection. In our case, we could set the preferences for the network interface we prefer for a certain SVC layer. In addition, more studies related to network selection problem can be found in [22] for a fuzzy logic approach and in [23] for combining both fuzzy logic and genetic algorithms.

In the experimental evaluation presented in this paper, we consider only a very simple decision-making logic for the SVC layer mapping. More advanced algorithms like the ones briefly discussed in this section will be included in this first prototype. The prototype is used to handle decision-making regarding the SVC layer routing decisions, etc.), and this signalling can be handled through the UCMM to reduce the number of signalling interfaces required for the entities.

V. PROTOTYPE IMPLEMENTATION

We have developed a first-pass prototype implementation of the proposed multi-interface streaming system for scalable video. From the SCALNET system architecture presented in Section III, the prototype includes the streaming server, the streaming client and player, the UCM Client Module. In addition, it includes two context information sources for dynamic network interface management, namely the L2CIS and the QoSCIS. For the video streaming components, we used our own implementations to allow fast experimentation with the system. In the future work, we plan to migrate to using the components used in the SCALNET system implementation in [3], that is, the Darwin Streaming Server, the MPlayer, and the Open SVC decoder. In this section, we present the implementation of the different components included in this first prototype. The prototype is used for the experimentation of the dynamic network interface selection functionality in Section VI.

A. Video Streaming Components

The streaming server used in the prototype is our own implementation and supports streaming of SVC encoded videos over multiple RTP sessions. The use of our own implementation made it possible for us to experiment with the multi-interface streaming of SVC and to obtain first-pass results validating the system operation quicker than what would have been possible with open source components.

The server implementation includes basic support for the RTP and RTSP protocols to be able to communicate with the client. The server machine is Windows-based and has two Ethernet interfaces and two IP addresses: the first is used for receiving RTSP requests from the client and sending the base layer stream, and the second is used for sending the enhancement layer stream. That is, only one control channel (i.e. RTSP and RTCP signalling) is
information is formatted as triggers. Each trigger contains the following fields: an identifier (ID), a type, and a value, and they are formatted as TLV or XML message structures depending on the interface implementation. The IDs are allocated so that each entity generating triggers (i.e. source) will have a specific ID (or a set of IDs). The type field may then be used for distinguishing between triggers generated with a single ID. The value field carries the actual context information (e.g. end-to-end delay) and any other additional information required for identifying the trigger source or for using the information carried in the trigger. In the prototype, the UCMM is listening for context information messages originating from the QoSCIS and L2CIS, and passes them to the AQPE.

The QoSCIS is implemented as a module that connects to a QoS measurement tool called QoSMeT [24] for QoS measurement data. It can be configured to provide triggers containing the QoS measurements (i.e. delay, loss, and throughput) to the UCMM either periodically or whenever some value exceeds a predefined threshold. For the experimentation conducted for this paper, we configure the QoSCIS to monitor the packet losses detected by QoSMeT in the received video stream. The QoSCIS averages the packet loss measurements with a sliding window taking account three previous values and sends a context information message to the UCMM whenever a specific, decoder-dependent threshold (e.g. 4\%) is exceeded. The minimum interval between the triggers is set to 0.5 seconds.

The L2CIS, on the other hand, is implemented as a loop constantly checking the available network interfaces in the client terminal as well as the entries in the routing table. It reads the output from the following commands: `ipconfig` and `route print`, and then keeps a list of the available network interfaces, their types, the IP addresses assigned, and whether there is a route to the server from that address in the routing table. The L2CIS sends a context information message to the UCMM whenever it detects that a network interface or a route is removed or added to the list. The L2CIS version used in the prototype is implemented in Windows, because of our Windows-only video player. In Linux, the L2CIS is able to collect also additional information like WLAN received signal strength indication (RSSI).

C. Dynamic Network Interface Selection Solution

The AQPE is integrated into the MUPlayer. Its current version supports a very simple mapping algorithm to be used in our tests. In the future, we plan to implement more advanced decision-making algorithms for the AQPE but they are not included in this first prototype version. Moreover, we do not have mobility support included in the prototype, but the routing decisions are implemented in the AQPE. Also due to the lack of an IP mobility mechanism, we do not consider the handover of the base layer in our tests.

In the current implementation, the AQPE has three operational states: INIT (initial state), BASE (only receiving
base layer), and ENH (receiving both layers). In the INIT state, the mapping of SVC layers to network interfaces is enforced based on the init file. Different events may trigger the state transitions like discussed in the following.

For example, whenever the two SVC layers are being routed through different networks due to a QoS mismatch etc. (i.e. AQPE is in INIT or ENH state), the reception of a context information message indicating that a network interface has been assigned with an IP address triggers the AQPE to check, if the address is 0.0.0.0 (i.e. link down). If so and if it is the enhancement layer network interface that was lost, the AQPE signals the RTSP streaming client to drop the enhancement layer. The AQPE then updates the routing tables by removing the route from that interface to the server, and moves to the BASE state. In the experiments presented in this paper, we do not consider the loss of the base layer interface as it would require IP mobility support (e.g. MIP) for the interface to ensure session continuity. When the AQPE receives a new context information message indicating that a network interface has been assigned with a valid address (i.e. link up), it updates the routing table accordingly and restarts the enhancement layer stream over the new interface.

The context information produced by the QoS CIS can also cause state transitions. For instance, when the both SVC layers are initially routed over the same interface (INIT) and congestion is detected based on QoS CIS information, the AQPE signals the streaming client to drop the enhancement layer (INIT→BASE). It then checks, if there is a secondary network interface available, and if so, it restarts the enhancement layer stream through that interface (BASE→ENH). That is, this procedure performs the re-routing of the enhancement layer stream to achieve the required level of QoS for the whole video stream and optimal QoE to the user.

VI. EXPERIMENTAL EVALUATION

In this section, we discuss the results obtained from testing the prototype under network impairments. The dynamic network interface selection is to ensure optimal throughput performance for the streaming under different network conditions. In the experimental evaluation, we tested the prototype operation under changes in network availability as well as congestion. The evaluation was done in a network environment where the user terminal is connected to Ethernet and WLAN networks. Ethernet was chosen to test sudden access link disconnection by simply unplugging the cable. It should be noted that we did not consider strictly the appropriate mapping of SVC layers to available networks but instead tested the adaptability of our system in the presence of network problems.

A. Test Setup and Scenarios

In the tests, we considered two scenarios: a cable unplugged scenario and a limited bandwidth scenario. In both scenarios, a two-layer SVC video stream was transmitted from a Windows XP laptop running the SVC streaming server to a high-end Windows Vista laptop running the MUPlayer. In addition, a Linux laptop running Netem software was used to introduce network impairments to the video streaming. In the tests, the QoSMeT measurement tool was utilized to measure the streaming performance. The QoSMeT was set up to measure separately the throughput of the SVC base and the enhancement layer streams.

The video used for the testing (‘Earth from above’) was encoded using the JSVM reference encoder to have two quality layers with fixed resolution and frame rate of 640x352 and 15 fps, respectively. The difference in the SVC layer throughputs is significant: in average, about 100 Kbps for the SVC base layer and about 500 Kbps for the quality enhancement, producing a clear visual difference between the base quality and the enhanced one. The user thus needs to receive the whole stream for good visual quality.

1) Cable Unplugged Scenario: In this scenario, depicted in Fig. 6, the SVC base layer stream is routed through the client’s WLAN interface and the enhancement layer stream through the Ethernet interface. The dynamic network interface selection is tested by unplugging the Ethernet cable in the middle of the streaming. When the AQPE detects that the cable has been unplugged, it signals the RTSP streaming client to send a RTSP PAUSE message to stop the enhancement layer stream. This message can be sent since we have only one control channel going through the base layer interface between the server and the client. When the cable is plugged in again, the enhancement layer stream is restarted over the Ethernet interface by sending a RTSP PLAY request.

2) Limited Bandwidth Scenario: In this scenario, shown in Fig. 7, both SVC layers are first sent to the client’s Ethernet interface. We wait a few seconds and
then introduce some congestion to the wired link with Netem. When three consecutive packet loss reports with values higher than 4% are received in 5 seconds’ time, the AQPE assumes that there is not enough bandwidth for both the layers in the wired link and decides to drop the enhancement layer (RTSP PAUSE). We opted to trigger the enhancement layer dropping based on persistent congestion, hence the 5 seconds’ monitoring period. A generic packet loss threshold of 4% was selected for the testing as it can be seen as a high packet loss for a good viewing experience even with a robust decoder. Since the client has also a WLAN network available to it, the enhancement layer stream is rerouted through the WLAN interface.

B. Results

The results for the cable unplugged scenario are shown in Fig. 8. At 24 seconds, the Ethernet cable is unplugged from the client device causing the throughput to drop. At 34 seconds, when the cable is plugged in again, the enhancement layer is restarted and the throughput increases. The results thus demonstrate the correct operation of our dynamic network interface selection solution. The client monitors the available network interfaces and whenever one is lost or a new one becomes available, the client issues requests for the enhancement layer stream, accordingly.

The results from the second scenario are shown in Figs. 9 and 10. On the left side of Fig. 9, the measured throughput of the whole SVC stream in the beginning of the limited bandwidth scenario is shown. At 30.65 seconds (indicated with a vertical line in the figure), the RTSP client is signalled to re-establish the enhancement layer session over WLAN by the AQPE after observing three consecutive >4% packet loss indicators in 5 seconds’ time. After a while, we can see the throughput to increase on the wireless link caused by the enhancement layer rerouting. The base layer continues to be transmitted over the Ethernet interface. In Fig. 10, we observe the reduced performance (i.e. increased loss) when both streams are sent over the wired link, and the performance increase (i.e. zero loss) after the rerouting of the enhancement layer stream. The >4% packet loss triggers received from QoSCIS at 28.66, 29.16, and 30.65 seconds before the AQPE’s rerouting decision are indicated with vertical lines in the figure. Also in this scenario, we can see the benefits of the multi-interface streaming capability combined with our dynamic network interface selection solution.

VII. CONCLUSIONS

Diverse adaptation capabilities are an essential requirement for video streaming services targeted to heterogeneous network environments. In this paper, we introduced a full end-to-end architecture for the adaptive transmission of SVC-encoded video streams. The architecture and its components were also extended to support multi-interface video streaming, which may be used for optimizing the throughput and therefore also the user perceived quality of video streaming. The multi-interface streaming support utilizes the flexible transmission capabilities of SVC and implements a dynamic network interface selection solution, which allows for dynamic management of the SVC layer routing in a changing network environment. The dynamic network interface selection solution makes
use of a cross-layer signalling framework which is used for context information collection and delivery within the client terminals.

In order to evaluate the multi-interface streaming concept, we have developed a first-pass prototype implementation of our solution, which was also presented in this paper. In addition, we performed a case study where we evaluated the performance of the dynamic network interface selection solution in two scenarios. The results from the study were presented in this paper and they showed that the multi-interface streaming capability for SVC combined with dynamic network interface selection allows maximizing video streaming performance (i.e., session continuity and throughput) under network impairments. The solution can thus be used for optimizing the user experience of video streaming in heterogeneous multi-access networks.

As future work, we plan to develop our multi-interface prototype as well as the network interface selection solution further. For the prototype, we plan to migrate to using open source components, which are also used in the SCALNET system implementation, including the Darwin Streaming Server, the MPlayer, and the Open SVC decoder. To enhance the dynamic network interface selection, we aim to implement more advanced decision-making algorithms, fine-tune the thresholds, and consider new performance metrics (e.g., estimates of available network bandwidth or QoE) in the decision-making. In addition, we will study different mechanisms that can be used for supporting flow mobility for the base layer in the prototype. Moreover, to improve the signalling performance under congestion, we will study the possibilities to utilize IP-level QoS mechanisms. We also aim at obtaining more advanced metrics (e.g., QoE) for evaluating the system performance and for this we will need access to a more robust SVC decoder such as the Open SVC decoder, as the reference one performs poorly under packet losses. Finally, we plan to investigate the incorporation of other SCALNET system components to the multi-interface streaming scenario that were left out from this study. In particular, the MANE can be used to create interesting test scenarios. For instance, if we map more than one SVC enhancement layers into a RTP session, instead of just one, we can use the adaptation capabilities of the MANE and its integrated ADTE and SVC Filter components to filter excess layers from the substream in the access network whenever needed.

Acknowledgements

The work reported in this paper was conducted in the EUREKA/Celtic Scalable Video Coding impact on Networks (SCALNET) project, which was partially funded by the Finnish Funding Agency for Technology and Innovation (Tekes) in Finland; the EU FP7 UniverSelf project; and the Intelligent Telecommunication Systems with Enhanced Cognitive Processing (ITSE) project of VTT Technical Research Centre of Finland. The authors would like to thank their colleagues who have participated in these three projects and especially those who have contributed in the development and implementation of the prototype. A special thanks goes also to our colleague Jyrki Huusko for his valuable comments during the paper writing process.

References


Tia Sutinen has a M.Sc. (Eng.) degree, which she received from the University of Oulu, Finland, in 2004. Currently, she is working as a senior scientist at VTT Technical Research Centre of Finland. She has gained experience of several years in national and international research projects. The main focus in her research work is in cross-layer assisted adaptive transmission of scalable video in heterogeneous multi-access networks. In specific, her interests lie in application-aware mobility management and application adaptation in different network scenarios, including different types of handovers and link-level mechanisms to improve video delivery over wireless links. Under these topics, she is pursuing a Ph.D. degree and has also published her work in international scientific forums.

Helena Rivas has a Computer Science Engineering degree, which she received from School of Engineering, Universidad de la República, Uruguay, in 2005. She is currently a research scientist at VTT Technical Research Centre of Finland. The main focus of her research work is in decision-making mechanisms for self-organized networks.