

Mobile and Interactive Social Television – A Virtual TV Room

Francesco Cricri Tampere University of Technology Tampere, Finland francesco.cricri@yahoo .it	Sujeet Mate Nokia Research Center Tampere, Finland sujeet.mate@nokia.com	Igor D. D. Curcio Nokia Research Center Tampere, Finland Igor.curcio@nokia.com	Moncef Gabbouj Tampere University of Technology Tampere, Finland moncef.gabbouj@tut.fi
--	---	---	--

Abstract

Smart phones are becoming more and more powerful. Services that were traditionally designed for a static environment can now be implemented into mobile devices. One such service is Interactive Social TV, which allows geographically dispersed people to meet in a virtual shared space and watch TV while being able to interact with each other. This paper presents two novel architectures of a Mobile and Interactive Social TV system. In both of the architectures, the interaction is represented by a rich audio-visual media, allowing users to hear and see each other. In the first architecture, the mixing of the TV content with the interaction media is performed at the server side. In the second architecture, the mixing is performed in each client device. The issues of decoding and rendering simultaneous content and interaction media streams on a mobile device are discussed, and the related implementation is presented.

1. Introduction

TV watching is not only a pure individual action, but it has had a social dimension since its first introduction on the market. People often prefer to watch TV programs, such as a football game or a movie, together with other people, in order to *consume* the content as a *social experience* [1]. As stated in [2], the content very often represents a “medium for social interaction between people”, since it provides common interests; in such situations in which people choose to watch TV together, “socializing around the content might be more important than the content itself” [2]. In the usual case, people wishing to share the same content must be located in a relatively restricted

geographical area, in order to be able to meet each other at home or some other mutually convenient location. If these people, for some reason, are either in different geographical locations, or do not have time for meeting others in a common location, then shared watching experience is not feasible. All of these observations represent the main motivation behind the concept of *Mobile and Interactive Social TV* [1] [2] [3] [4]. In such a system, there is a possibility to use mobile devices for watching TV or video content not only with people who may be far apart, but also with those who are on the move, as if they were in the same physical room. The new ingredient, if compared to typical recent mobile TV applications and systems, is that here rich interaction possibilities between participants watching a common content together facilitate a higher feeling of virtual presence. Synchronized playback of TV content amongst participants such that all of them watch the same event with minimal time difference is very important. Rich interaction coupled with synchronized content playback ensures that all participants are watching the same content and interactions nearly at the same instant of time, and thus have a *common shared context* of the viewing experience. This shared context creates a feeling of watching together.

This paper is organized as follows. Section 2 presents different modalities of interaction and selected prior art on interactive social TV. Section 3 presents the novel implementation issues for mobile social TV systems. Section 4 describes the actual implementation of the proof of concept mobile social TV system; while Section 5 presents the user feedback for the implemented. Section 6 gives an overview of the potential future work; and finally Section 7 concludes the paper.

2. Modalities of interaction and prior art

Interaction should ideally satisfy the requirement of creating a virtual presence of the participants, which is as close as possible to the real presence, without distracting the attention of the viewers from the shown content (at least not more than in real world social TV viewing experience). In a social TV system, we could consider mainly the following four modalities of interaction: text chat, emoticons, audio conferencing and video conferencing (the latter includes both audio/speech and video). The real time property of the interaction is fundamental for creating a virtual presence of people. For low-bandwidth and low computational complexity solutions, text, chat and emoticons would be preferable. For high-bandwidth networks and high-end mobile devices, audio and video conferencing would be the best possible options, since they provide a stronger feeling of social presence to participants.

AmigoTV [3] is one example of already existing social TV systems in the research community, and it has also led to a commercial application. The interactions used in this implementation are audio conferencing, graphic symbols and avatars. 2BeOn [4] is another project in which TV watchers would use a Set Top Box (STB) that provides them with several functionalities, such as user tracking, communication services and collaboration with content tracking; in particular, communication services consist of audio/video conferencing and instant messaging, allowing users to communicate and interact with each other while watching TV content. ConnecTV [5] is an interactive social TV system that aims to study and analyze the behavioural changes of users with respect to the traditional way of watching television. Living@room [6] gives the possibility to users to share not only TV content but also local contents that one of the participants owns, such as holidays videos. ClipSync [7] is a web based social TV which allows users to synchronously play videos offered by several providers on the Internet such as YouTube, while interacting via text chat and animations that are displayed on the screen. In Lycos Cinema [8] text chat is used as means of interaction among participants who are watching the same movie or TV show.

The most important features of mobile devices are *location-independence* (allowing people to use mobiles wherever they are) and *time-independence* (at any moment of the day) [9]. The social TV service should be accessible from mobile devices to make the service available to the maximum number of users in addition to enabling people to join a social TV session from any place and at any moment (see Figure 1).

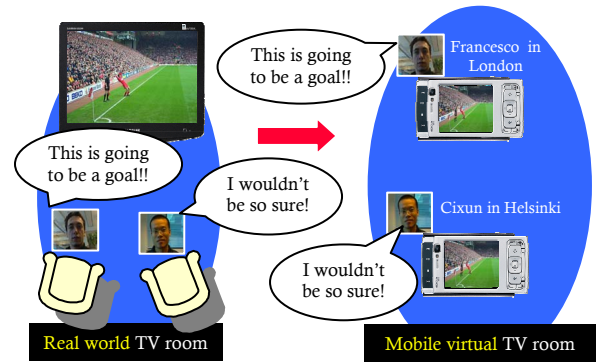


Figure 1. Mobile social TV

Thus, by jointly exploiting the communication and multimedia features that modern smart phones offer, mobile social TV is the next natural development step of the general social TV concept. One prior art mobile social TV service is AMUSE [10], in which video content is broadcasted to mobile devices using DVB-H, and a text chat represents the only interaction mean among the participants.

Implementing audio or even video conferencing as a means of interaction on mobile devices is a very challenging task, and represents the novelty of the problem being studied and solved in this paper.

3. Proposed mobile social TV architectures

3.1. General Issues

Mobile devices have several known limitations, mainly regarding computational performance, memory capabilities, power capabilities, screen size, and usability. Screen sizes in such devices are relatively small, and this could represent a limitation for a comfortable TV watching experience. Screen dimensions and hardware capabilities of mobile phones may pose a great challenge for achieving video playback for a satisfactory consumer experience.

Modern mobile phones, on the other hand, come with larger screen resolutions, making watching experience at least acceptable. Thus, mobile social TV service could be implemented only in high-end and multimedia oriented devices that have large screens and high computational performances. Usability of mobile phones is also one of the aspects to take into account. In fact, mobiles are usually held in one hand, making it difficult for a user to watch long video content. Also, as stated in [11], other people could see the video content being displayed on the screen if they are close to the user, such as in a bus, thus reducing the privacy of that user. Another usability issue is represented by the background noise that could be too

loud especially in an outdoor environment, thus impairing the audio playback experience, and eventually also the interaction, if an audio or video conference is being adopted. Hands-free can be used in noisy places to work around this problem, although this has other disadvantages like reducing situational awareness to some extent.

3.2. Real time AV as interaction mechanism

The main goal of interaction in a social TV system is to allow participants to do what they would usually do when they are watching TV content together in the same real world TV room, such as commenting the events being shown on the screen (common especially when the content is news), looking at the faces of the others and at the gestures others do (this is more likely to happen if the content is a movie or a sport event). A multiparty videoconference system coupled with synchronized content streaming may provide the *virtual presence* of the participants in a mobile TV room. This is the concept under the focus of this paper.

Using real time audio and video for interaction in such a system represents a challenge because the bit rates needed for video streaming are high, and the available bandwidths in mobile network is usually relatively low. Also occasional packet losses are quite common in such a communication environment. Therefore, picture quality and latency are the biggest issues to tackle when designing such a system. Also, computational complexity of video processing is higher than for other types of data, and this could be an important issue especially in mobile devices that have limited computational capabilities, due to the requirements of having small size and low power consumption. Audio streaming is also very sensitive to latency. This makes the use in mobiles of real time audio and video, as means of interaction, a much more demanding task than, for example, a text chat or an audio-only conference.

Real time audio and video interaction feed may be created by sending the locally captured interaction media from a mobile to a central node (the videoconference server) which combines them into a single video stream which is then sent back to each of the clients. An alternative approach consists of avoiding the videoconference server and making the clients send their local interaction media to each other; thus, each client device has to combine these media locally. The second option requires the mobile devices to be able to receive and decode multiple multimedia streams, for example three, four, or even five streams. After some testing, this option was considered not feasible by using high-end mobile phones currently

available on the market, and thus the first option has been adopted.

The videoconference system can be fused with the TV content streaming in different ways. In this paper two approaches are considered: performing the mixing at the server side (*centralized* mixing) or at the client side (*end-point* mixing). As previously mentioned, video processing on mobile devices is a demanding task. Therefore, centralized mixing would be the best choice because each client could receive only one video stream in which TV content and videoconference feed are already combined. In the end-point mixing approach each client device receives two streams: the TV content stream from a remote content provider, and the videoconference stream from the videoconference server. This second approach requires each client to receive, decode and display two incoming video streams simultaneously, thus being a more challenging solution than the first approach. Modern video encoding techniques, such as H.264, are able to deliver very high quality/bitrates, at the expense of a higher computational complexity in both the encoding and the decoding steps, other than causing longer delays and requiring bigger buffer sizes. Therefore, the most demanding processes for an end user device in a video streaming scenario is represented by the encoding and the decoding that have to be performed in real time in order not to miss any frames. This means that in such a context the time within which each frame has to be decoded is very limited. This could lead to huge power consumption. Therefore, it becomes fundamental to make a trade-off among the following factors: low-bandwidth streaming, real time processing and low battery power consumption.

4. Implementation

As already mentioned, in this paper two novel approaches for fusing TV content with the videoconference feed and the relative architectures are proposed. These architectures are described in the next sub-sections.

4.1. Centralized mixing architecture

The architecture of the mobile social TV with centralized media mixing [11] has three main actors, as illustrated in Figure 2: the *Content Provider*, the *Interaction Server* – or *videoconference server* - and the *clients*. The Content Provider sends the video content (such as a movie) to the Interaction Server. Each client captures the interaction media using the embedded front-camera and the microphone, encodes

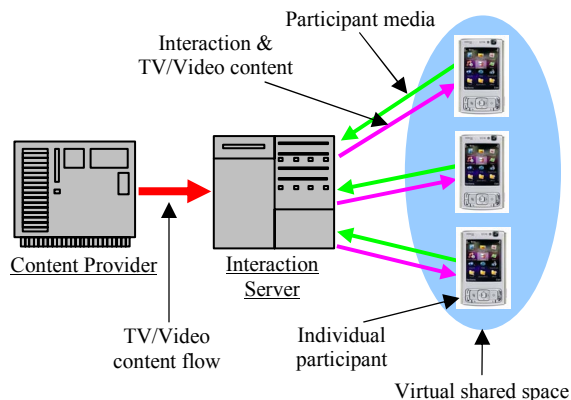


Figure 2. Centralized mixing architecture

the video and audio streams and sends them to the Interaction Server. This server creates a new media stream by combining the interaction media received from every client wishing to participate to the same *virtual shared space* with the TV content stream received from the Content Provider. This new stream is sent to all the clients. Each client has to receive the combined stream, decode it and play it out.

4.2. Establishing a social TV session

All participants wishing to use the mobile social TV service need to register their mobile devices to a SIP registrar server that is embedded in the Interaction Server. The actual session establishing procedure starts when one client sends an invite request (through HTTP) to the server, specifying the SIP addresses of the participants to be invited. Therefore, any of the clients need to have the SIP URI (for example, sip:myname.mysurname@example.com) of the desired invitees. The server sends the invite to each of these participants, specifying which client sent the invite request. By accepting the invitation request, the participants enter that particular social TV session and start interacting with the other invited participants who have accepted the invitation request.

If a participant refuses the invitation request, it is still possible for him/her to join in the social TV watching session later. In this case, the user checks whether there are any ongoing sessions and chooses which of them to join. In case there is no prior permission for a particular session, the session initiator should approve the joining request. However, for the sake of simplicity in proof of concept system implementation, the ongoing sessions are assumed to be public.



Figure 3. Client view after a TV room session establishment - Multiparty videoconferencing

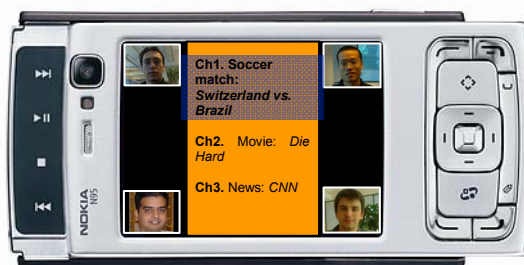


Figure 4. Client view. Users can go through a list of channels to choose the TV content

4.3. Functionalities

Figure 3 displays the client's screen just after four participants have entered the social TV session. The session starts as a usual multiparty videoconference session. The participants can talk and discuss about which TV content they are going to watch together by going through a list of available channels, as shown in Figure 4. Once agreed, they can start watching the same content in a virtual TV room, thus being still able to see and hear the other participants. Figure 5 shows an illustration of this use case. A useful limited privacy functionality is added in the system in such a way that if a participant does not want to show his/her face, just pressing the camera button on the phone will make his/her personal video disappear from the screens of the other participants.

As mentioned earlier, the screen size on mobile phones is quite limited, thus a feature that makes the watching experience more comfortable has been introduced. This technique adapts the size of the individual interaction videos according to the voice activity of the respective users, so that the videos of those users who are not talking are kept small, thus covering a minimal area of the TV content. Figure 5 also shows an example of this feature: the interaction video on the top-left corner is larger than the others



Figure 5. Client view. TV content and interaction media overlay

because the user is speaking. There are also other variations possible to manage the real estate on the display, like rendering interaction video feed only when the user is speaking.

As can be seen in the same figure, in addition to audio and video conferencing, it is possible to have also a text chat among the participants that can be utilized for example during the TV content playback in order not to disturb the other users from comfortably following the audio and speech part of the TV content.

4.4. Implementation of server, client and communication system

The mobile social TV service presented in this paper is targeted to Symbian S60 platform. The clients were implemented on Nokia N95. For the transport of the multimedia data (video and audio), RTP (Real time Transport Protocol) [12] over UDP is used in order to ensure a real time communication, necessary for an interactive application as the mobile social TV. For encoding the video streams, both the content and the interaction, the H.264/AVC standard [13] has been used. The protocol used for exchanging session initiation related information between clients and the Interaction Server is SIP [14]. HTTP [15] is used as the protocol for the control channel, thus enabling the clients to choose among different view modes (only videoconference, only TV content, TV content combined with videoconference, zooming on only one user of the videoconference) and to cooperatively decide what content or channel they are going to watch. HTTP is used also for sending the information about ongoing sessions from the server to the clients, and for requesting the server to invite the participants present in a list. SDP [16] is used for the session description.

For this proof of concept system, TV content is made available via a video content file that is locally stored in the Interaction Server, instead of streaming it from a content server. Thus, from now on we will refer

to the “server” as the Interaction Server. A typical star network topology has been used, where the server is the central node and the clients are the end nodes. The server side has been implemented on a Linux machine. There are mainly three types of protocol flows between the server and the clients: HTTP, SIP and RTP. Therefore, the Interaction Server includes an HTTP server component, a SIP user agent and a media handling/mixing component. A Symbian HTTP client is used in order to support handling of the HTTP traffic at the client side. For the SIP infrastructure side, a standard SIP registrar and a proxy server are used. For implementing the SIP server, the Sofia SIP User-Agent library [17] has been used. A native standards compliant implementation of the RTP stack is used. For the media handling and processing, the implementation of the S60 client uses extensively the *MultiMedia Framework* (MMF) of Symbian. For the video side, it uses the low-level API *DevVideo* for both the encoding (*DevVideoRecord*) and the decoding (*DevVideoPlay*) processes. Since real time processing is very important for this application, in order to speed the rendering, the *Direct Screen Access* (DSA) method (allowing direct access of the screen buffer and bypassing the *Window-Server*) is exploited.

4.5. End-point mixing architecture

The mobile social TV system with end-point mixing [18] has a similar architecture as for the centralized mixing approach when it comes to handling the interaction among participating users. In fact, the main actors are still the *Content Provider*, the *Interaction Server* and the *clients*. The difference lies in the way these actors are inter-connected. In the end-point mixing architecture, as illustrated in Figure 6, the TV Content Provider is connected to the clients and not anymore to the Interaction Server. The Content Provider sends the video content (such as a movie) to the clients. Each client sends the captured and encoded interaction media to the Interaction Server. This server combines the media received from each client wishing to participate to the same social TV session and creates a new interaction feed – the multiparty videoconference stream – which is sent to all the clients. Each client combines the received interaction feed and the movie before it plays them out.

4.6. Implementation of end-point mixing architecture

The focus of the work for the end-point mixing architecture is on video decoding and rendering of multiple video streams. After receiving two video streams – the movie and the videoconference – each

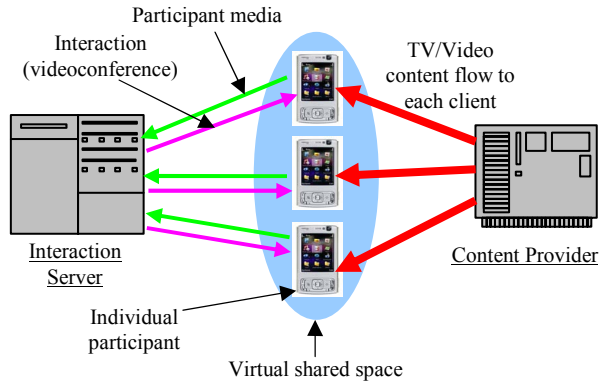


Figure 6. End-point mixing architecture

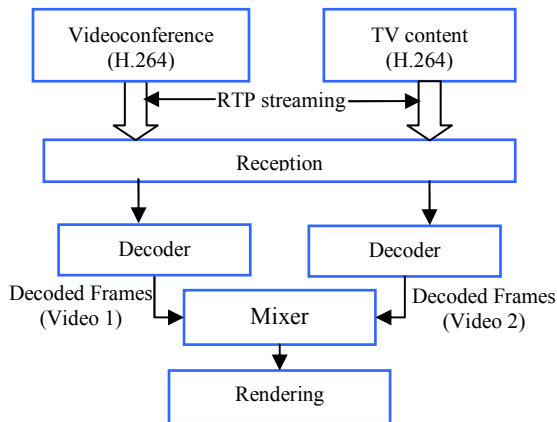


Figure 7. Video data flow in the S60 client. Target implementation

client needs to display them on the same screen. The current MMF implementation does not allow S60 clients to use two DSA processes. Therefore each device needs to perform a mixing step just after having decoded the two videos and before DSA. The decoded pictures being mixed are in YUV raw (uncompressed) data format and can occupy a huge amount of memory. Care needs to be taken to minimize copy operations on such buffers within the blocks of the processing and rendering pipeline.

The target video data flow for the S60 client is shown in Figure 7. The current implementation presents two differences with respect to this figure: the video content is a pre-downloaded file, whereas in the target implementation it is received through an RTP stream, and it is currently encoded using the H.263 video compression standard [19] instead of H.264. Therefore, the end-point mixing architecture presented in this paper is intended to give a proof of concept for the multiple video decoding and rendering on S60 clients. The resolution used in the current implementation for both the videoconference feed and the pre-downloaded file is QCIF (176 x 144 pixels).



Figure 8. Client view. End-point media mixing

The choice of the H.263 codec and QCIF resolution in the implementation is influenced by the need to maintain spare processing capacity in the mobile device for other tasks. This means that the resolution of the video content is smaller than the screen size of the client devices used for experimentations; thus, the watching experience is not optimal. A view of the S60 client is illustrated in Figure 8. As can be seen, not the whole screen size is used for the video rendering.

4.7. Client CPU load

An analysis of the CPU load for the S60 clients in the end-point mixing architecture is presented below. During the analyzed time interval, the mobile device was performing the following operations simultaneously: capturing the live media (video and audio from the embedded front-camera and microphone), encoding the video into H.264/AVC and the audio into AMR (Adaptive Multi-Rate), sending the interaction media to the Interaction Server, receiving and decoding the videoconference media (H.264/AVC video and AMR audio), decoding the local video file (H.263-encoded), mixing the decoded pictures of the two videos in the YUV format, and rendering the resulting combined video using DSA.

It was observed that the CPU load caused by the social TV client application is taking most of the resource usage. The media decoding and playback operations (DL - downlink) have an impact of approximately 55% over the total CPU computational capabilities, whereas the interaction uplink (UL) causes a load of approximately 25%. The H.263 local video file decoding does not appear in the CPU load because it is decoded by the HW-accelerated H.263 decoder.

4.8. Comparison of the two architectures

We presented in this paper two different approaches for fusing the TV content consumption with interaction in the mobile social TV system. The two architectures have different advantages and disadvantages. The first approach consists of performing a centralized mixing

inside the Interaction Server. In the second approach the mixing is done at the clients. The main advantage of the first architecture over the second one is that each client device has to receive and decode only one video stream. This is very important especially because we are targeting mobile platforms, which is a resource constrained platform with limitations on computational processing power and battery consumption requirements. The advantages of the second architecture over the first one consist in allowing an easier accountability of the TV content (performed by the content providers), wider availability of contents since the Interaction Server owner does not need to deal with all the content providers whose content the social TV participants wish to watch. Each individual client or group of clients can connect to the desired content provider. In case of centralized mixing architecture approach the choice of available TV or video content is limited to set of content providers that have prior arrangement with the Interaction service provider. It needs to be noted that we have not dwelt into the legal and copyright aspects of using content in the above described architectures in a commercial or public service. The terms for doing this are likely to vary depending on the content type and country specific legislations. In case of end-point mixing architecture approach the Interaction service needs to have additional mechanism to maintain stream synchronization between content streams and interaction streams of users participating in the social TV session. Both the architecture approaches would require synchronization of content playback between multiple users. The need for such synchronization mechanism becomes more necessary as disparity between access networks' characteristics (for example, network bandwidth, latency, etc) used by each client in the social TV session becomes larger [18].

5. User feedback

The architecture with centralized mixing has been tested in both WLAN and 3.5G environments. Considering first WLAN communication, the general user feedback is positive. The response time of user actions such as selecting the movie or switching between different view-modes is short. The limited bit rate and the packet losses make the quality of videos with high motion content (such as sport events) degrade significantly with respect to a local video playback, whereas the watching experience is much better if more low motion contents are considered.

Considering now the 3.5G network technology, some tests have been performed by having client devices in different countries. The mobile social TV system over 3.5G has proved to be stable. Because of

further bandwidth limitations that 3.5G networks have compared to WLAN, it was necessary to lower the total bit rate, thus obtaining a lower quality than in WLAN environment. Even on 3.5G network, the quality of the video playback has been considered to be acceptable. Also, the user input responsiveness has been acceptable, thus giving confidence that this service works not only in WLAN environments, but could also be used in cellular networks. The architecture with end-point mixing has not been subjected to testing on cellular 3G networks. One relevant feedback has been that the resolution of the video content needs to be higher for obtaining a more pleasant watching experience. Also, the frame rate needs improvement.

6. Future work

Future work includes a full implementation of the *Mobile and Interactive Social TV* with both *centralized* and *end-point* media mixing. Moreover, support for streamed reception of the TV content being watched by the participants from content providers should be added. This would replace the playback from local video files stored in the Interaction Server (for the centralized mixing architecture) or in the client devices (for the end-point mixing architecture). Furthermore, H.264/SVC (Scalable Video Coding) could be used instead of H.264/AVC, so that devices with varying computational capabilities and screen resolutions would be able to decode only the necessary data, thus providing optimal playback quality without wasting resources. Further user study is required in order to better understand how the current features could be fine tuned and what could be the features to add to the current system. A more comprehensive study is needed to understand the effects of interactions on media consumption and what kind of video content would better fit such a service. We need to understand what other modalities of interactions could be adopted for making virtual presence of the participants more realistic.

7. Conclusions

In this paper, a *Mobile and Interactive Social TV* system has been presented. In such a system, the real world TV room is simulated by streaming a video content to mobile clients and interaction among the participants is provided in order to emulate their social presence in a social TV viewing session. This allows people to virtually meet for watching TV content together, even if they are in different locations and on the move. As means of interaction, real time audio-video and also text messaging has been adopted in this

work. This choice represents a challenge, since mobile devices have limited computational, memory, screen size and power capabilities.

Two different approaches for fusing the interaction media feed with the TV content have been considered and implemented. The first approach (*centralized mixing*) consists of mixing the two media streams at the server side. During a session, participants watch the same TV content and are able to see each other's faces and talk to each other. Also text chat can be used to communicate with the participants in the social TV session collectively. A preliminary user feedback has shown that such a system provides an acceptable viewing experience.

The second approach considers an *end-point* media mixing architecture performed at the client side. This paper focused on the decoding, mixing and rendering of two video streams on S60 devices. An analysis of the CPU load of client devices has been performed. A comparison between the two mixing approaches has been presented, highlighting the main advantages and disadvantages of each approach. The centralized mixing approach has lower resource usage requirements for the clients, but requires a complex back-end infrastructure. The end-point mixing architecture allows more flexibility for the users to consume the desired content, while requiring higher resource demands from the clients. Finally, some potential future extensions to the current mobile social TV room system are briefly introduced.

8. Acknowledgments

We would like to thank Markus Pulkkinen for his contribution in software implementation.

9. References

- [1] Lee B., Lee R.S., "How and Why People Watch TV: Implications for the Future of Interactive Television", *Journal of Advertising Research*, Vol. 35, No. 6, 1995, pp. 9-18.
- [2] Schatz R., Wagner S., Egger S., Jordan N., "Mobile TV becomes Social – Integrating Content with Communications", *Proceedings ITI'07*, June 2007, pp. 263-270.
- [3] Coppens T., Handekyn K., Vanparijs F., "AmigoTV: A Social TV Experience through Triple-Play Convergence", (Alcatel Technology white paper).
- [4] Abreu J., Almeida P., Branco V., "2BeOn – Interactive television supporting interpersonal communication", *Proceedings of the sixth Eurographics workshop on Multimedia 2001*, 2002, pp. 199-208.
- [5] Boertjes E., "ConnecTV: Share the experience", *Proceedings of EuroITV 2007*, Amsterdam, May 2007, pp. 139-140.
- [6] Ghittino A., Iatrino A., Modeo S., Ricchiuti F., "Living@room: a Support for Direct Sociability through Interactive TV", *Adjunct Proceedings of the 5th EuroITV Conference*, Amsterdam, May 2007, pp.131-132.
- [7] ClipSync. <http://www.clipsync.com>. Accessed on 24.9.2007.
- [8] Lycos Cinema. <http://cinema.lycos.com>. Accessed on 2.8.2007.
- [9] Schatz R., Wagner S., Jordan N., "Mobile Social TV: Extending DVB-H Services with P2P-Interaction", *Proceedings of the Second International Conference on Digital Telecommunications*, ICDT '07, 2007, p. 14.
- [10] AMUSE 2.0, <http://amuse.ftw.at>. Accessed on 5.12.2008.
- [11] Pulkkinen M., Mobile Virtual TV Room, *M.Sc. Thesis, Tampere University of Technology*, 2007.
- [12] RFC 3550 – RTP: A Transport Protocol for Real-Time Applications, <http://www.ietf.org/rfc/rfc3550.txt>
- [13] H.264: Advanced video coding for generic audiovisual services, <http://www.itu.int/rec/T-REC-H.264>
- [14] RFC 3261 – SIP: Session Initiation Protocol, <http://www.ietf.org/rfc/rfc3261.txt>
- [15] RFC 2616 - HyperText Transfer Protocol, <http://www.ietf.org/rfc/rfc2616.txt>
- [16] RFC 4566 – SDP: Session Description Protocol, <http://tools.ietf.org/html/rfc4566>
- [17] Sofia-SIP, <http://opensource.nokia.com/projects/sofia-sip/>
- [18] Cricri F., Media Mixing and Inter-Client Synchronization for Mobile Virtual TV-Room, *M.Sc. Thesis, Tampere University of Technology*, 2008.
- [19] H.263: Video coding for low bit rate communication, <http://www.itu.int/rec/T-REC-H.263/>