Real Time Synchronization of Live Broadcast Streams with User Generated Content and Social Network Streams

Hans Stokking[1], Rianne Kaptein[2], Arjen Veenhuizen[3], Martijn Spitters[4], Omar Niamut[5]

[1-5]TNO, Delft, The Netherlands

E-mail: {[1]hans.stokking, [2]rianne.kaptein, [3]arjen.veenhuizen, [4]martijn.spitters, [5]omar.niamut }@tno.nl

Abstract: **This paper describes the work in the FP7 STEER project on augmenting a live broadcast with live user generated content. This user generated content consists of both video content, captured with mobile devices, and social network content, such as Facebook or Twitter messages. To enable multi-source synchronization, all content sources need to time stamp the content using synchronized clocks. For synchronized play out we use a synchronization server to control the output timing of the various content. Using social analytics, we determine the most relevant social network messages to present synchronously with the broadcast. A main part of our future research in the project will focus on determining the optimal timing of such social content with a live broadcast.**

Keywords: media synchronization, IDMS, social analytics, HAS, media streaming, second screen

1 INTRODUCTION

A recent trend in media consumption is consuming multiple different media at the same time, see e.g. [1]. More and more, people tend to multitask while watching TV, by being active on multiple social networks while at the same time consuming content from sites offering video content. In the living room, the second screen and the connected TV have changed the setting completely. Users no longer just watch a single television show. Instead, users use secondary screens such as tablets and smartphones, during their television usage. While much of this usage is unrelated to the broadcast content, i.e. people browsing the web and sending e-mails, the number of apps that provide context related to the broadcast is growing fast. Also, connected TVs not only allow for browsing the web or consuming over-the-top content, they also include new features such a communicating with friends while watching the same broadcasted show.

One of the two major use cases in the STEER project is the 'Live Augmented Broadcast' use case. Here, various people are visiting an event that is being broadcasted, for example a skiing event in Schladming, Austria. While visiting the event, these visitors can use a STEER mobile application to record (short) videos. Also, a lot of posts related to the event are being made on various social networks, e.g. on Twitter or Facebook. Even though their friends at home will not experience the event in the same way as visitors present at the event, their broadcast experience can be enriched with these various other streams. User generated video's supply alternative views of the event, showing views from the audience perspective and perhaps capturing certain events missed by the broadcast camera's. The streams from social networks provide additional information and discussion, making the user more 'part of it' by colouring the main broadcast stream.

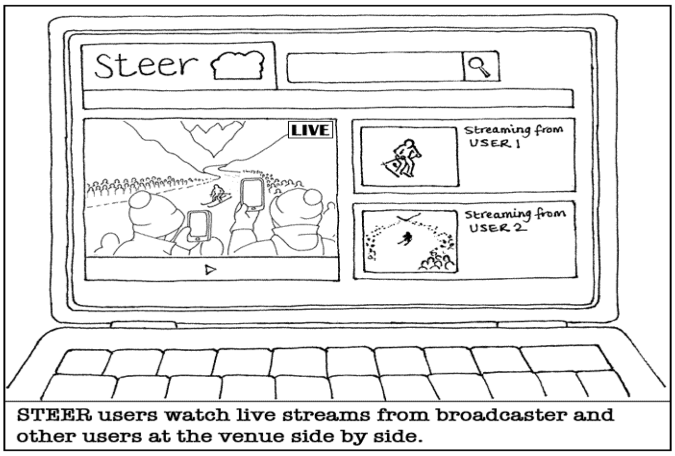
Steer
LIVE
Streaming from USER 1
Streaming from USER 2
STEER users watch live streams from broadcaster and other users at the venue side by side.

Figure 1 - STEER Live Augmented Broadcast use case.

The broadcast can be a professional broadcast, provided by the event organization through the regular television broadcast chain. The broadcast can also be a semi-professional capture provided over the Internet, either by the event organization or by one or more users visiting the event. The user generated content shown to a particular user can be content from friends or family visiting the event, but may also be content provided by an anonymous user at the event using the STEER mobile application. Twitter and Facebook messages can be provided both by users at the event and by users at home watching the broadcast. Figure 1 depicts a possible user interface for consuming this live augmented broadcast at home.

This paper deals with the synchronized delivery of broadcast media, user generated content and social network streams like Twitter and Facebook. Out of scope for this paper is the optimized delivery and caching of content, as well as the necessary group- and session management for delivering these social telemedia services. These topics are part of the STEER

Corresponding author: Hans Stokking, TNO, Brassersplein 2, Delft, The Netherlands, +31-(0)88-866278, hans.stokking@tno.nl

ISBN/EAN – 978-90-5986-439-9

project, but are not included in this paper. Instead, the focus of this paper is on the synchronization of the various content streams and the issues encountered therein. The STEER project is currently running, and no experiments have been performed so far. This paper is giving an overview of our design and our plans in the project.

This paper is structured as follows. In section 2 we discuss related work, both in the area of synchronization and in the area of social analytics. Next, various challenges in synchronization are discussed in section 3, explaining the main research goals of this part of the STEER project. We carry out this work based on the use case discussed in section 4. In section 5 we go into more detail on the social analytics aspects of our work. The content distribution architecture and related synchronization architecture and implementation are discussed in in section 6. Finally, we conclude with some remarks on the work that is still ahead of us in section 7.

2 RELATED WORK

For media synchronization, a good overview is given in [3], discussing intra-media synchronization, inter-media synchronization (i.e. lip-sync) and inter-destination media synchronization. Our paper focusses on both inter-media synchronization and inter-destination media synchronization, also discussed extensively in [4]. More recently, various synchronization use cases are presented in [5], showing synchronization requirements ranging from very high (<10 ms) to low (< 2 sec) for various IDMS use cases.

Earlier work on synchronization of media coming from various sources was presented at the Media Sync Workshop in 2012, showing synchronization in the HBB-TV context [7][8]. Synchronization of various sources is performed using (inserted) timestamps in the various media streams. Other relevant work is in the area of social TV, of which [9] gives a good overview, showing categories of social TV, including ‘community building’, defined as ‘commenting about a television program with a large community’.

Considering related work in the field of social analytics, the most popular social network is Twitter, due to its real time nature, and ease of access to publicly available streams of Twitter data. Second screen applications for broadcasted events using Twitter data have been developed commercially, e.g. Umami (http://www.umami.tv/), Zeebox (http://zeebox.com/) as well as for research [10],[11]. Twitter volumes around broadcasts are typically large, and therefore most of the second screen applications focus on giving overviews and statistics such as volume, aggregated sentiments, frequently used terms, etc. Also a common element in Twitter and second screen applications is a list of tweets, either from your own social network, a curated set of influential Twitter users, or a (random) selection of tweets.

The main social analytics techniques we intend to use in our set-up include adaptive event profiling and tracking, sentiment mining, and user authority measurement.

The simplest way to retrieve relevant tweets is to use the official hash tag of the broadcasted event. However, in order to improve recall and cope with the constant evolution of word and hash tag use during events, we need much more sophisticated techniques. A lot of research has already been conducted in this area, see e.g. [12],[13],[14],[15] and [16].

Sentiment mining is the problem of finding subjective expression of attitude or sentiment towards e.g. a certain product, brand, or topic in text. Sentiment mining has been extensively applied to social media [17][18], often for brand analysis and recommendation purposes. Improving retrieval and recommendation based on information from social networks has been investigated by a.o. [19] and [20].

3 CHALLENGES IN SYNCHRONIZATION

There are various challenges in providing a synchronized experience for the end user watching the broadcast at home. The major challenge we currently focus on, is how to provide synchronization of social network streams. At a first glance, this seems a rather easy task. Each message on the various social networks such as Facebook or Twitter is time stamped. Although the timestamp granularity is usually in whole seconds, in general this should provide the necessary accuracy for these kinds of semi-real time messages. However, the complexity comes from two issues encountered when synchronizing these message streams to a live broadcasted stream.

The first issue is due to uncertainty about the origin of the message, i.e. it is not known who is actually providing the message. One can distinguish here between a user at the actual event and a user at home. If a user at the event, he or she is able to respond much more quickly. That is, he has seen the event in real time, while users at home only see the event after it has been delayed by the entire broadcasting chain. Hence for any messages directly in response to something that happens at the event a real-world timestamp is only relevant for users present at the event. For users at home, the timestamp for the message should relate to the timestamp of the broadcast stream.

Related to this is the issue of first messages in relation to follow-up messages. In a message stream, people respond to each other as well as to events that occur. If someone responds to another message, the thread should be maintained. If the message is a first message, e.g. in response to the event, the timing of the message should be in correlation with the timing of the event, regardless of the location of the user. To be able to distinguish between the two is not obvious and hence constitutes a part of our research.

The second issue is caused by the delay in message consumption. Because of the delays of the various broadcast chains, it is quite possible that Twitter messages posted by users at the actual event arrive earlier than the broadcast stream. So, a message like ‘Awww, that fall must have hurt!’ can be quite spoiling the experience, if you have not yet seen the skier actually fall. To solve this, we aim to synchronize the messages to the broadcast stream. However, since people do take time to respond to an event, it might be the case that messages arrive too late and hence are no longer relevant. The delay in the broadcast chain gives us the chance to display messages from users at the event, earlier with respect to the broadcast, as we can time the play out of these messages. In the STEER project, we aim to investigate the optimal timing of social network streams in relation to a broadcasted event.

Also, in our experiment we will analyse the social streams to detect the most relevant messages to display during the broadcast. This analysis will take time, due to processing of the various messages and the construction of the overall view of what is relevant. This will cause an additional delay, and will determine the lower bound on the delay which we will have to take into account in our augmented live broadcast implementation.

Besides these particular social-network related synchronization questions, more typical media synchronization issues arise. To be able to synchronize multiple streams coming from different sources, one needs three things:

1. One needs the different sources to use synchronized clocks. Each source will provide timestamps in the media streams. Unless the clocks on which these timestamps are based are synchronized clocks, it is not possible to synchronize the related streams using these timestamps. Clock synchronization itself is well-known, and negotiating on synchronized clocks is under standardization in the IETF [21]. One related issue here is the access to clocks and synchronized clocks on consumer devices, as these often do not allow modification of such settings in regular apps, i.e. root access may be needed. As discussed later, we foresee using (S)NTP as a well-supported clock synchronization mechanism, but which has some downsides as well.
2. One needs the different sources to insert timestamps based on these synchronized clocks in the media streams. Even when a synchronized clock is available, one needs to ensure that the proper timestamps are inserted into the various streams. This is often not the case, e.g. no timestamps at all are inserted, timestamps are starting at 0, or random offsets are chosen. With a (semi-professional) content ingest platform this will not be a major problem, but including proper timestamps in an off-the-shelf consumer device (i.e. in an app) can become complicated.
3. One needs the timestamps to remain untouched during the entire broadcast chain or other content delivery chains. If transcoding is performed within the delivery chain, this often causes timestamps to be changed. Transcoding is often part of such a chain, e.g. to optimize video for a certain distribution network or to provide multi-bitrate encodings to allow switching between bitrates with changing network conditions. In such cases, either the timestamp should be independent of the encoding, or the transcoder should insert the original timeline in the transcoded version or the transcoder should be able to match in- and outgoing timestamps.

These issues are further discussed in the implementation section.

4 SOCIAL ANALYTICS ARCHITECTURE

In Figure 3 an overview of the proposed architecture for social analytics within STEER is given. In this section we discuss each module in the architecture. We start with the adaptive event profiler, which generates and incrementally adapts an event query which is sent to the social media scraper. The *adaptive event profiler* receives a basic description of the event through the STEER mobile application based on the available event broadcast meta information. This initial profile is extended and adapted based on information retrieved from external sources such as Wikipedia and news sites. Also, the profile is continuously adapted based on the media objects retrieved by the *social media scraper*. A media object is any item that can be retrieved from a social network, e.g. tweets, YouTube videos, status updates and photos posted on Facebook, etc. The adaptive event profiler generates a number of key terms and relevant social network user identifiers related to the event which can be used by the social media scraper to search and filter social media streams. The social media scraper uses a combination of publicly available search and streaming APIs to retrieve the media objects. Media objects and their relevant attributes are retrieved by the social media scraper within a certain relevant time window, and are stored in a central database. The collected media objects are used as feedback to keep the event profile up to date with the latest developments related to the event.

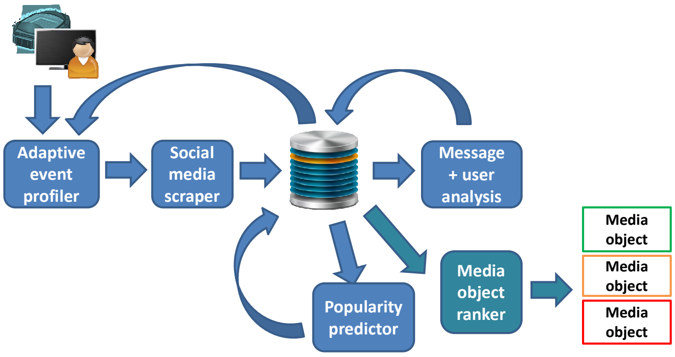

Figure 2 – Proposed architecture for Social Analytics.

In the module for *message and user analysis* the textual contents and the user information contained in the media objects are analysed. From the textual contents, entities such as people, locations and sub events are extracted. In case of the skiing event the participants will be frequently mentioned people in the buzz around the event. Sub-events are for example the men’s giant slalom, or the women’s downhill. In addition to entities, sentiment and sentiment around entities is determined. Sentiment can be extracted from the original textual contents of the media objects itself, as well as from comments and the likes on YouTube videos or Facebook posts, or replies on tweets. Furthermore, the users that have generated the media objects are ranked on authority within the event space. Authority is measured based on a combination of number of friends and followers, centrality in the social network graph that is created for the event, number of event related (re)tweets, etc. This extra information about the textual contents and the users is added to the media objects in the database.

The *media object ranker* generates the top ranked media objects that will be shown in the STEER application. The media ranker is personalized, i.e. content from people in your own social networks is considered more relevant. Other factors that the media object ranker takes into account include sentiment, popularity, relevance weights, and time. Entities can be used to group media objects around persons, locations or sub events. Finally, an important analysis for synchronization is *popularity prediction*. The popularity predictor predicts the popularity of entities and media objects by looking at early sentiments, number of times they are linked, authority of users involved, etc. The estimation of popularity potential can be used to handle synchronization and distribution issues due to high demands for a (video) media object.

5 SYNCHRONIZATION ARCHITECTURE & IMPLEMENTATION

5.1 Live Augmented Broadcast Architecture

Figure 3 shows the current high-level STEER architecture for the live augmented broadcast chain. On the right side, it shows multiple capture sources, which can be (semi-) professional camera's for the broadcast stream and a variety of consumer devices running the STEER application for user generated content. These streams are inserted into the content ingest platform. From there, STEER will use a P2P overlay network to transport the streams to the various homes where users watch the augmented broadcast. This P2P overlay network is capable of adapting to the network conditions, but is as such not the focus of this paper. Once the streams enter the home gateway, they can be distributed to various different end-user devices, such as a set-top box connected to a large flat screen or to secondary screens such as tablets or smartphones. The home gateway also supports content transcoding, to deal with fluctuating home network conditions such as a tablet temporarily having a bad Wi-Fi connection and thus having less available bandwidth. Specifically, it can provide multi-bitrate encodings using HTTP Adaptive Streaming such as HLS [22] or DASH [23].

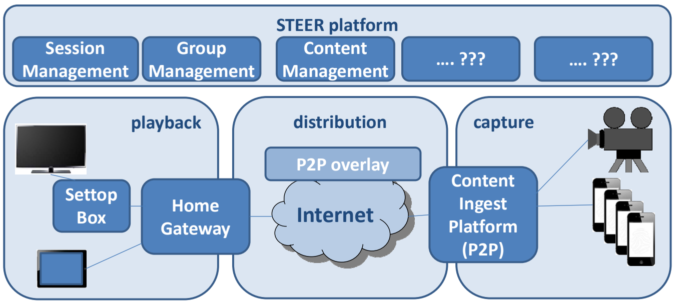

Figure 3 - High-level STEER live augmented broadcast chain architecture.

5.2 Sync Architecture

Figure 4 shows a more detailed functional overview of this augmented broadcast chain. These functions are now discussed, with a special focus on the functions that deal with or have to support the synchronization of the various streams when presented at the various client devices.

The capture device main functions include capturing a stream, usually consisting of video and audio, and uploading this stream to the content ingest platform. For synchronization purposes, it has to support the functionalities mentioned in the synchronization challenges section: it has to have its clock synchronized to the various other capture sources, and during capture it has to insert timestamps into the streams based on this synchronized clock.

The content ingest platform is mostly agnostic to the content, using P2P transparently for carrying the streams to their various destinations. The exception to this occurs when the P2P overlay network wants to downscale because of network conditions, i.e. when limited bandwidth is available in the overlay network. In such a case, the ingest platform will perform transcoding, and will have to keep timestamps intact or report on the combination of incoming and outgoing timestamps. The ingest platform also contains a content segmentation function. This function is used to supplement the capture device, taking care of transforming streams to HAS segments to be used on tablet devices and the like. The P2P network can carry these HAS segments transparently as well, delivering HAS segments instead of streaming media.

The home gateway functions as a server in the home network domain, hosting various STEER functions. It receives the P2P segments and recombines these either in an outgoing stream, e.g. to a set-top box, or collects the HAS segments and stores these on the internal HAS server. The home gateway also contains a transcoder function to be able to deal with fluctuating network conditions inside the home network. The home gateway can either monitor these conditions itself through a bandwidth monitoring function (also not further discussed in this paper) or rely on e.g. HAS clients to request lower-bitrate versions of segments once the available bandwidth has dropped. The home gateway also contains a synchronization server. This server collects play out timings from the various clients and client modules, calculates synchronization settings and thus effectively controls the play out timing of the various streams on the various end-user devices.

Finally, the client devices contains the functions for playback of the various media streams. To achieve this, we have split up that functionality in a number of modules. We have a separate STEER session client, which enables users to join specific augmented broadcast sessions, e.g. join with their friends who are present at the actual event broadcasted. For video playback, we have a video client and for the presentation of social network streams we have a social network module. Both of these report on their play out timing to the synchronization server located at the home gateway, and are under control of this synchronization server to achieve synchronization. Note that to display various streams, the client device will normally have multiple instances of the video client running at the same time. All stream presentations are bundled together in a single interface in the user interface module. Finally, to enable synchronization, the client device also needs a synchronized clock.

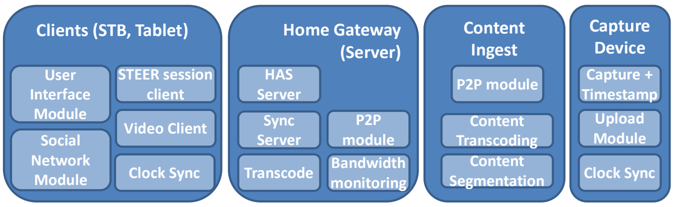

Figure 4 - Detailed functional overview of the live augmented broadcast chain.

5.3 Synchronization Implementation

5.3.1 Introduction

Our implementation of synchronization of hybrid media builds on the technology developed within the EU FP7 Next Generation Hybrid Broadcast Broadband, or HBB-Next, project [24]. This project aims to develop technical solutions that allow for innovative new services to be delivered over hybrid-broadcast-broadband platforms. Topics being researched in HBB-Next include group recommendation systems, synchronization of heterogeneous media on a single or multiple devices and deployment of device and platform independent applications.

Within STEER, we build upon the achievements of this on-going project with respect to (frame-accurate) synchronization of hybrid media on one or multiple end-user devices. Frame accurate synchronization can be useful in case of, for example, multi-angle video where one camera angle is broadcasted via for example DVB, while another angle is send over-the-top to a mobile device. As discussed earlier, we focus on synchronizing media streams of different sources (audio, video, chat messages, etc.) in a live case. In contrast to the goals of HBB-Next, it has been decided that the accuracy of synchronization is limited to the order of one second. Frame accurate synchronization is deemed out of scope since the content creation (in contrast to HBB-Next) is not limited to a closely managed environment (e.g. everyone can contribute within STEER).

5.3.2 Synchronized content creation and play out

In general, media synchronization has two major hurdles to overcome: 1) aligning the timing information of the different media sources and 2) synchronizing the play out of different media streams on one or multiple devices. In a way, timing information needs to be shared between content creators and content renderers. Sharing a common clock can be achieved in numerous ways and has been abundantly discussed in literature, see e.g. [25],[26]. One way to achieve this common clock is to alter each media stream, adding (additional) time information which has a relation between streams. Another approach utilizes a central time-keeping unit, with each device reporting their time information and adhering to the time information which is shared between the different content creators. Yet another method assumes that all devices are in sync with each other by using a common clock (e.g. NTP, Network Time Protocol). Within STEER, we aim to utilize the latter approach, making use of the NTP infrastructure which contains a precise clock signal. This time information is then used to precisely timestamp generated multimedia content. Also, Facebook and Twitter messages are time stamped using (presumably) synchronized clocks, which allows us to display these messages in synchrony with the other content. Part of the STEER project is to analyse the clock usage of Facebook and Twitter, to see if our assumption holds up.

With regard to synchronized play out, the same NTP synchronized clock could be used. Assuming that content distributed over the renderers has related timing information, sharing that information is sufficient to obtain synchronized media play out. Within STEER, this sharing of timing information is performed using a synchronization server provisioned at the home gateway. A detailed description of this approach is given in [27].

NTP may not always give accurate clock synchronization. Support for NTP on various devices fluctuates, as some devices only support the less accurate SNTP (Simple NTP). Also, different NTP servers may have clocks running (slightly) differently, so using various NTP servers can also lead to inaccuracies. Finally, NTP assumes symmetric networks. Asymmetric networks and network congestion may lead to some inaccuracy in the clock synchronization. Still, NTP, or at least SNTP, is widely available and the alternative on mobile devices (GPS) is less available. Because of this last point, within STEER we have chosen (S)NTP as a clock synchronization mechanism, also since our use case is not depending on exact frame-accurate synchronization.

5.3.3 HBB-Next synchronization platform

The HBB-Next synchronization platform allows for frame-accurate synchronization of hybrid media on multiple devices. In its current implementation it is able to synchronize DVB-S, HTTP Live Streaming (HLS) [22], MPEG-DASH [23] and regular media files across devices, given that the media streams share a common clock. [7] gives an in depth overview of this framework. The framework is primarily focused on synchronizing audio and video content, but can be extended to support other media types as well. The framework builds upon the open source GStreamer platform [28]. This platform allows for accurate and high performance multimedia play out and supports numerous (streaming) media formats. The GStreamer framework, running on for example a laptop, is specifically useful for rendering media on a regular television and as such, basically acts as an easy to access set-top-box. Furthermore, the GStreamer platform is capable of communicating with other devices using regular (TCP) sockets and/or Web Sockets . This allows for inter-device communication. The mDNS [29] (also known as Bonjour) protocol is supported in order to announce a service, such as a synchronization service, on a network to for example media renderers, further simplifying device discovery and inter-device synchronization.

5.3.4 Synchronization within STEER

Within the STEER project, we aim to synchronize user generated media content which is not limited to just video and audio, but includes messaging services like Twitter and Facebook as well. For content creation, the use of a common clock is proposed (e.g. NTP). Multimedia requires appropriate time stamping (e.g. internal time codes likes Presentation Time Stamp (PTS) need to relate to the global clock). This way, no additional clock information needs to be inserted into the media stream, resulting in full compliance to all the applicable standards and further simplifying the design and implementation.

Assuming that the media streams are properly aligned to each other, the next step is to render them synchronously. To this purpose, the use of a centralized sync server (see Figure 5) is proposed to which each client can subscribe. This node distributes a common clock to the connected clients, taking propagation delays into account (for further reading, see D4.3.1). Each node should adhere to this timing information.

6 THE STEER PROJECT

The work presented in this paper is on-going research, conducted as part of the European FP7 project STEER [1]. In STEER (a Social Telemedia Environment for Experimental Research), we focus on so-called “Social Telemedia”, a cross-breeding of social networks and networked media. In the project, various combinations of social analytics and content delivery are being studied. Within the STEER project, we use the term social analytics to refer to techniques and algorithms for intelligent analysis of social media data streams and networks, whereas content delivery represents the communication medium through which social informatics are exchanged through a bundle of synchronized and heterogeneous flows. Social analytics involves intelligent automated analysis of social connections, opinions, topics and entities. Such information can be used to optimize the transport of content, to optimize caching of content, and to find related content or to give recommendations for content. STEER envision various interworking aspects of social analytics and media distribution aspects, as depicted in Figure 5.

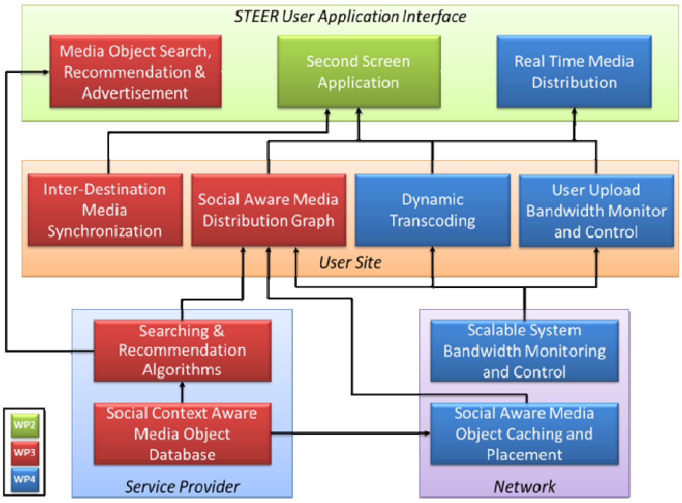
STEER User Application Interface
Media Object Search, Recommendation & Advertisement
Second Screen Application
Real Time Media Distribution
Inter-Destination Media Synchronization
Social Aware Media Distribution Graph
Dynamic Transcoding
User Upload Bandwidth Monitor and Control
User Site
Searching & Recommendation Algorithms
Social Context Aware Media Object Database
Service Provider
Scalable System Bandwidth Monitoring and Control
Social Aware Media Object Caching and Placement
Network
WP2
WP3
WP4

Figure 5 - The STEER Reference Architecture.

7 FUTURE WORK

As the STEER project has only recently started, this paper describes the current status of the on-going research carried out in this project. We are now in the process of implementing the mentioned architectures, with the goal of carrying out experiments with our implementation in 2014.

The main topic on which our synchronization experiments will focus is the level of required synchronization between social media steams and a live broadcast stream. As discussed in the challenges section, we will be able to achieve various synchronization levels on a technical level. A main purpose of doing our experiments will be to discover users’ appreciation for the various levels of synchronization.

On a technical level, the experimental environment builds on earlier synchronization work from projects like FP7 HBB-NEXT, and extends on these projects with synchronization of social network streams and a separate synchronization server. By continuing our technical work on synchronization, we expect to have new insights in how synchronization can be achieved best. Also, this will further our experience with and insight in challenges in both media synchronization and clock synchronization as a main requirement to achieve media synchronization.

Acknowledgment

This work is partially funded by the European Commission's 7th Framework Programme under grant agreement no. 318343 (STEER project).

References

[1] Nielsen, Action Figures: How Second Screens are Transforming TV Viewing, http://www.nielsen.com/us/en/newswire/2013/action-figures--how-second-screens-are-transforming-tv-viewing.html, 17 june 2013.

[2] Denazis, Spyros and Efthymiopoulos, Nikos and Koufopavlou, Odysseas and Den Hartog, Frank and Niamut, Omar and Hutchison, David and Race, Nicholas and Mu, Mu and Taal, Jacco and Mori, Luigi and Boniface, Michael and C Phillips, Stephen and Wu, Felix (2012) STEER: A Social Telemedia Environment for Experimental Research. In: FIRE engineering workshop, 2012-11-07, Ghent, also found at http://www.fp7-steer.eu/

[3] Ud Din, Shahab, and Dick Bulterman. "Synchronization Techniques in Distributed Multimedia Presentation." MMEDIA 2012, The Fourth International Conferences on Advances in Multimedia. 2012.

[4] Boronat, Fernando, Jaime Lloret, and Miguel García. "Multimedia group and inter-stream synchronization techniques: A comparative study." Information Systems 34.1 (2009): 108-131.

[5] Montagud, Mario, et al. "Inter-destination multimedia synchronization: schemes, use cases and standardization." Multimedia Systems 18.6 (2012): 459-482.

[6] Stokking, Hans, Ray van Brandenburg, Fernando Boronat, Mario Montagud. "Standardization of Inter-Destination Media Synchronization." Proceedings of the Media Synchronization Workshop 2012.

[7] Veenhuizen, Arjen, Ray van Brandenburg. "Frame accurate media synchronization of heterogeneous media sources in an HBB context.", Proceedings of the Media Synchronization Workshop 2012."

[8] Yuste, Lourdes Beloqui, Padraig O Flaithearta, Hugh Melvin. "Interactive Multi-source Media Synchronisation for HbbTV." Proceedings of the Media Synchronization Workshop 2012.

[9] Cesar, Pablo, and David Geerts. "Past, present, and future of social TV: A categorization." Consumer Communications and Networking Conference (CCNC), 2011 IEEE. IEEE, 2011.

[10] Shamma, David A., Lyndon Kennedy, and Elizabeth F. Churchill. "Summarizing media through short-messaging services." Proceedings of the ACM conference on computer supported cooperative work. 2010.

[11] Dork, Marian, et al. "A visual backchannel for large-scale events." Visualization and Computer Graphics, IEEE Transactions on 16.6 (2010): 1129-1138.

[12] Lin, Jimmy, Rion Snow, and William Morgan. "Smoothing techniques for adaptive online language models: Topic tracking in tweet streams." Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.

[13] Becker, Hila, Mor Naaman, and Luis Gravano. "Beyond trending topics: Real-world event identification on twitter." Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM'11). 2011.

[14] Osborne, Miles, et al. "Bieber no more: First story detection using Twitter and Wikipedia." SIGIR 2012 Workshop on Time-aware Information Access. 2012.

[15] Efron, Miles. "Hashtag retrieval in a microblogging environment." Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM, 2010.

[16] Sakaki, Takeshi, Makoto Okazaki, and Yutaka Matsuo. "Earthquake shakes Twitter users: real-time event detection by social sensors." Proceedings of the 19th international conference on World wide web. ACM, 2010.

[17] Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A.: Twitter power: Tweets as electronic word of mouth. In JASIST 60, 2009, 2169—2188.

[18] Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment in Twitter events Journal of the American Society for Information Science and Technology Volume 62, Issue 2, pages 406–418, 2011.

[19] Walter, F., Battiston, S., Schweitzer, F., A Model of a Trust-based Recommendation System on a Social Network, Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), 2007

[20] Shang, S. et al, Wisdom of the Crowd: Incorporating Social Influence in Recommendation Models`, IEEE 17th conference on Parallel and Distributed Systems (ICPADS), 2011

[21] Williams, A.M., K. Gross, R. van Brandenburg, and H. Stokking. "RTP Clock Source Signalling draft-ietf-avtcore-clksrc-04." (2013).

[22] Pantos version 11, Active Internet Draft, http://datatracker.ietf.org/doc/draft-pantos-http-live-streaming/, last accessed 1 July 2013

[23] Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats, ISO/IEC 23009-1:2012, 2012-04-03.

[24] http://www.hbb-next.eu/index.php, accessed on 1 July 2013.

[25] L. Ehley; B. Furht; M. Ilyas, "Evaluation of Multimedia Synchronization Techniques", ICMCS'94, pp 514-519, 1994

[26] Y. Ishibashi; S. Tasaka, "A Comparative Survey of Synchronization Algorithms for Continuous Media in Network Environments", IEEE Conference on Local Computer Networks (LCN) 2000, Nov. 2000.

[27] Brandenburg, Ray van, Arjen Veenhuizen, Immersive second-screen experiences using hybrid media synchronization, Media Sync Workshop 2013, Nantes, 2013 - To Be Published

[28] http://gstreamer.freedesktop.org/, accessed on 1 July 2013.

[29] Multicast DNS, RFC 6762, ISSN: 2070-1721