Architecting Social TV

Óscar Figuerola Salas, Hari Kalva
Florida Atlantic University, 777 Glades Road, Boca Raton, FL, USA 33431-0991

ABSTRACT

This paper provides an overview of technologies and guidelines that can be used for the development of Social TV applications. A prototype using two core technologies, WebRTC and WebSocket was developed following the ideas presented in the literature in order to demonstrate the feasibility of such applications in the context of modern browsers without the use of any extra plugins.

Keywords: Social TV, video synchronization, WebRTC, WebSocket

1. INTRODUCTION

Social TV is any technology that enables social interaction in the context of watching TV content. People tends to enjoy watching TV in company, furthermore, over 60% of the videos are preferred to be watched in a social setting[1]. However, nowadays it is not always possible to meet to watch TV together, and that is where the idea of synchronous Social TV comes in, trying to simulate a collocated watching experience while in different locations.

Some media content is not suitable for synchronous interaction, and it is very common to establish conversations with friends or colleagues about movies and TV shows after having watched them. This way of communication is replicated in Social TV by the concept of asynchronous interaction, allowing users to comment on video content through some kind of social network, or directly on the media itself by embedding the comments attached to specific timestamps.

Based on the work found in the literature, we distinguish three different layers in Social TV: infrastructure, application, and social. The infrastructure layer includes any low-level technologies and protocols used in Social TV platforms, such as RTP, DASH (Dynamic Adaptive Streaming over HTTP), etc. In a higher level of abstraction we find the application layer, which comprises all the applications built on top of the lower level tools offered by the infrastructure layer. Lastly, the social layer covers every aspect regarding social interaction, which takes into account different aspects of the user's social behavior in front of TV.

2. RELATED WORK

The concept of Social TV arises from the field of interactive TV, just by adding features centered on social interaction. For this reason, many of the concepts and conclusions from studies regarding interactive TV also apply to Social TV. An example of this can be found in [2], where they present a study on interactive TV applications with communication services, and show how they can act as community builders.

Research more specifically directed to the field of Social TV has been very extensive for quite a few years, with some work dating from as early as 2002, and being still active nowadays. The most basic concept of Social TV consists on creating shared watching experiences while being in different locations. Collocated experiences can be easily achieved by creating real-time communication links between users, which can be done in three different modalities: text, video, and voice. One of the earliest examples is Reality Instant Messenger[3], which tried to take advantage of the already existing instant messaging infrastructure in order to build interactive TV services by allowing users to communicate while watching the same content. Other prototypes that helped creating a more consolidated definition of what a Social TV platform should look like were AmigoTV[4], CollaboraTV[5], and Social TV 2[6].

More modern prototypes include NeXtream[7], and CloudMoV [8], both supporting mobile devices. Systems targeting more specific groups of people have arisen lately too. The application iNeighbour TV[9] tries to promote health care and social interaction among senior citizens. Gameinsam[10] on the other hand focuses on providing families with a way to interact in a playful way during the daily TV watching experience.

3. SOCIAL TV LAYERS

As can be seen in Figure 1, Social TV applications can be abstracted into three layers: infrastructure, application, and social. These layers separate the technology necessary to support this type of applications from the applications themselves, and from the social features.

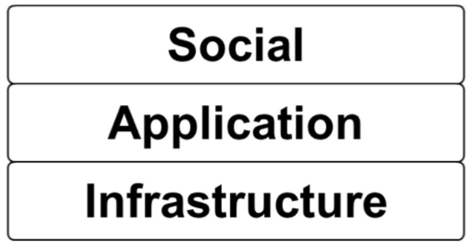

Figure 1. Social TV layers.

3.1 Social layer

The social layer has two different components: social interaction, and user behavior. When building Social TV applications, it is very important to understand both how the user behaves in front of TV, and how it interacts with other people while watching media content. This knowledge is the basis for the development of applications that are non-intrusive and useful to the user, at the same time that encourage social interaction between different users.

User behavior refers to how the user interacts with the TV itself. Watching TV has been traditionally seen as a lean-back experience[11], and probably most of the people like it that way. Following this idea, Social TV applications should support this use, providing efficient settings to the user so that almost every aspect of the application can be customized: privacy, types of social interaction enabled.

Social interaction can be synchronous or asynchronous, and it changes depending on the nature of the content. Synchronous interaction takes place while watching the media, like when you are commenting with friends about what is happening in a TV show. On the other hand, asynchronous interaction takes place either before or after watching the content, such as when you recommend a movie to a colleague, or when you talk about the movie that was aired on TV the previous day with a close relative. These types of interactions happen naturally everyday, without mediation of any extra tools, and one would expect it to change when altering the environment where it happens. However, it has been found that the way people interact regarding media content does not change significantly when it is done over the Internet[12]. An explanation for this in the case of synchronous interaction could be that when watching TV for example, the viewer is focused on the display that is showing the video, as opposed to on the people around it.

The type of content does change the way people interact, including genre, duration, and plot structure. Although most of the videos are suitable for Social TV, not all of them are for the same type of applications. The complexity of the plot structure and duration of the content seem to be the most decisive factor at the time of choosing the type of interaction needed, and while some media requires the viewer to be fully focused, other allows for side conversations or even jokes at the same time. In some cases, the genre of the media is closely related to the plot structure, thus there are certain genres that are more suitable for synchronous interaction (see Table 1), like sports, reality TV, soaps, poor quality movies, and comedy shows[12] [1] [13]. More complex plot structures and genres like documentaries, movies, and news, and short clips like music videos are great for asynchronous interaction[14].

Besides how and when the social interaction takes place, another aspect to take into account is with who it is performed. An interesting case of study is the sports genre, because it is very common to see people in bars that do not know each other, but are able to interact while watching a game for example. In an attempt to enhance social interaction by taking it beyond the borders of the household, this leads to the concept of interaction at community level, where people is able to interact over the Internet even though they have never met, and it is surprising to find how people can feel comfortable with it.

Table 1. Video genres suitable for each type of social interaction.

Social interaction	Video genre
Synchronous	Sports, reality TV, soaps, poor quality movies, comedy shows
Asynchronous	Movies, documentaries, news, music videos

3.2 Application layer

The applications built for Social TV use the tools and APIs provided by the infrastructure layer to build systems that will support the social uses of TV. Following the guidelines from the literature[14] is especially recommended when developing Social TV applications. Key aspects like privacy, and avoiding distractions should be taken into account.

It is important to minimize distractions such as using overlaid controls that cover most of the content. In this area, the second display approach seems to be the main trend[15 7]. It consists on using an extra device to show related content or settings, which makes more sense as the adoption of smartphones and tablets is increasing nowadays, since people is already getting used to managing and multitasking with a secondary device. The study from [16] tries to support this idea by analyzing the social interaction in twitter generated when a TV show is aired, finding that this type of interaction through secondary devices is already happening by the audiences own initiative.

Most of the users will be concerned about their privacy when using Social TV applications, since there could always be situations in which a user would not want to be available for chatting, or have its watching status visible to others. In these cases it becomes essential to provide the user with the appropriate options in order to avoid unnecessary worries, such as a configurable presence status like AmigoTV[4] does.

3.3 Infrastructure layer

The infrastructure layers support the higher levels of Social TV applications, providing the means to make it happen, i.e. content delivery and communication between viewers. Synchronous applications enabling direct communication have strong real-time requirements, and it is vital to provide users with a seamless way to communicate if we want to make them feel together and successfully simulate a collocated watching experience.

Video content consumed by users can be broadly divided into live and on-demand content. Most of video services delivered over the Internet today use HTTP for video delivery. Live content on the Internet is actually near-live with typically 10-seconds of delay for prefetching segments before playback. Content delivered over cable TV and similar services is scheduled programming and is delivered in real-time. In this area, the standard DASH developed by MPEG (Moving Pictures Expert Group) is the first HTTP-based bitrate adaptive streaming standard for video that is being adopted internationally. Other options are Apple's HLS (HTTP Live Streaming), and Microsoft's Smooth Streaming. The segment caching approach used by such services makes content synchronization among viewers in a Social TV session simpler.

In order to provide users with a satisfactory Social TV experience, we must ensure a good QoE (Quality of Experience). The field study from [17] tries to measure the synchronization requirements when co-watching online videos, pointing out how the traditionally used threshold from the telecommunications field does not hold true for Social TV applications. They recommend a maximum threshold of 1 second based on when the viewers noticed the synchronization difference in the media being watched.

WebRTC is a web technology that enables web browsers with real-time communication (RTC) capabilities[18]. One of the key advantages over other technologies that provide the same functionality is its integration within the browser, and its ease of use through HTML5 and JavaScript APIs, which eliminate the need for a video plugin. The WebRTC standard is currently under development and is being developed by a joint effort between the IETF RTCWeb and W3C WebRTC working groups. The IETF standard is known as RTCWeb and focuses on the underlying protocols, while the W3C efforts focus on the JavaScript APIs for use within browsers.

A simplified list of the protocols used to establish the peer connection and to send voice data using WebRTC can be seen in Figure 1. The media data is carried using the Secure Real-time Transport Protocol (SRTP), and the RTP Control Protocol (RTCP) is used to monitor transmission statistics related to the media stream. The architecture of WebRTC allows the media data to flow directly between browsers, but before any media data can be sent, the communication needs to be coordinated. The application exchanges media configuration information using the Session Description Protocol (SDP). This process is called signaling, and it is important to note that WebRTC does not specify the signaling mechanism, leaving it to the application layer. As opposed to the peer-to-peer media path, the signaling path goes through an external server that can modify, translate, and manage the exchanged messages. The STUN (Session Traversal Utilities for NAT) protocol and its extension TURN (Traversal Using Relay NAT) are used to find out the public IP address and port, and traverse NAT boxes and firewalls so that clients can receive UDP-based media streams.

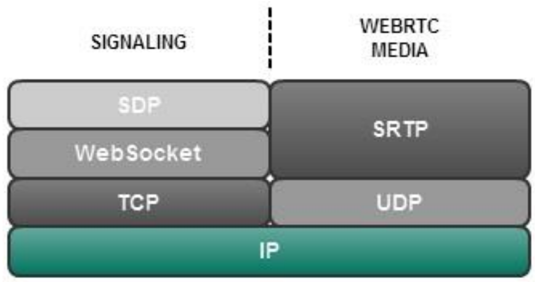

Figure 2. Protocol stack for WebRTC peer-to-peer communication.

At its current implementation status, WebRTC allows to use the MediaStreams API to get audio and video from a webcam and a microphone, the PeerConnection API to establish a peer-to-peer link for voice and video, the DataChannels API to share data through a PeerConnection, and the Stats API provides useful metrics such as the number of packets lost of the streams being exchanged. There are two main implementations right now, Chrome and Firefox, whose developers collaborate in order to maintain interoperability between them.

On the other hand, WebSocket[19] is a signaling web technology that provides bi-directional communication channels over a single TCP connection. WebSocket is composed of an API being developed by W3C, and a protocol that has been standardized by IETF. This technology complements WebRTC (see Figure 2), allowing the exchange of signaling messages needed to establish a peer-to-peer connection, and it can also be used as well to exchange messages between a browser client and a server, and vice versa. Application signaling is useful to synchronize media playback across users in a group. Such synchronization includes playback control and media play out time synchronization.

4. SOCIAL TV PROTOTYPE

Following the ideas and using the technologies discussed previously in this paper, we created a simple Social TV platform that shows how easy has become nowadays to develop this type of applications. The application is web-based and allows multiple viewers to watch a video synchronously while sharing a voice link. WebRTC takes care of the real-time requirements of the voice link, while WebSocket is used to initiate a peer connection between viewers and to exchange control messages so that the media can be synchronized and controlled remotely. The messages exchanged using WebSocket are distributed through a central server.

The viewers are allowed to execute media events at any point, i.e. play, pause, and seek, which will be executed by all the viewers. The application uses the concept of local lag described in [20], i.e. events are not executed right away, but at the pace of the client with the largest delay, trying to make viewers execute the events approximately at the same time.

The media synchronization is controlled by comparing the play-out time of a master viewer, which could be the one initiating the session for example, to the play-out times of the rest of the viewers (see Figure 3). The master viewer shares its play-out time and a local timestamp with a central server periodically. In order to calculate the play-out time difference, the local time of each viewer is shared with the central server, who adapts it to the local times of each viewer. As described previously, following the recommendations in [17], in the cases in which the difference is greater than 1 second, the play-out time of the viewer is readjusted to match the one of the master viewer.

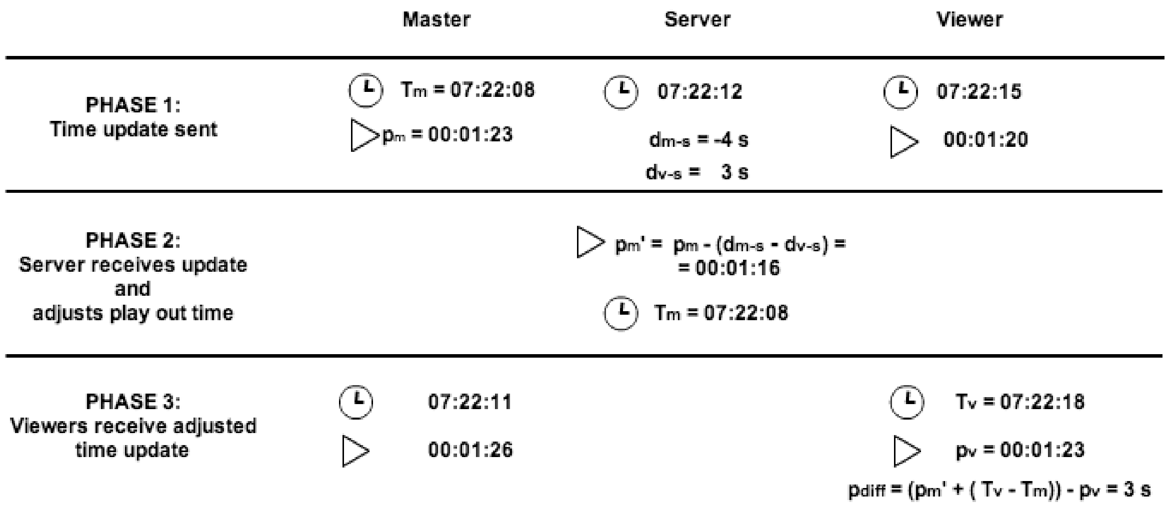

Figure 3. Time updates mechanism for media synchronization.

REFERENCES

[1] Dezfuli, N., Khalilbeigi, M., Mühlhäuser, M., and Geerts, D., "A study on interpersonal relationships for social interactive television," in Proceedings of the 9th international interactive conference on Interactive television, 21–24 (2011).

[2] Quico, C., "Are communication services the killer application for interactive TV? Or:'I left my wife because I am in love with the TV set'," in Proceedings of the 1st European Conference on Interactive Television: from Viewers to Actors, 99–107 (2003).

[3] Chuah, M., "Reality instant messenger: The promise of iTV delivered today" (2002).

[4] Coppens, T., Trappeniers, L., and Godon, M., "AmigoTV: towards a social TV experience," in Proceedings from the Second European Conference on Interactive Television" Enhancing the experience", University of Brighton 36 (2004).

[5] Nathan, M., Harrison, C., Yarosh, S., Terveen, L., Stead, L., and Amento, B., "CollaboraTV: making television viewing social again," in Proceedings of the 1st international conference on Designing interactive user experiences for TV and video, 85–94 (2008).

[6] Harboe, G., Metcalf, C.J., Bentley, F., Tullio, J., Massey, N., and Romano, G., "Ambient social tv: drawing people into a shared experience," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1–10 (2008).

[7] Martin, R., Santos, A.L., Shafran, M., Holtzman, H., and Montpetit, M.-J., "neXtream: A Multi-Device, Social Approach to Video Content Consumption," in Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, 1 –5 (2010).

[8] Wu, Y., Zhang, Z., Wu, C., Li, Z., and Lau, F.C.M., "CloudMoV: Cloud-based Mobile Social TV," IEEE Transactions on Multimedia Early Access Online, (2013).

[9] Almeida, J.A.P., Monteiro, N., Oliveira, B.T.N., and Pereira, L., "Results from the iNeighbour TV Field Trial."

[10] Herrmanny, K., Budweg, S., Klauser, M., and Kötteritzsch, A., "Gameinsam - A playful application fostering distributed family interaction on TV," in EuroITV 2012 Adjunct Proceedings, 21–22 (2012).

[11] Lu, K.Y., "Interaction Design Principles for Interactive Television," Thesis, Georgia Institute of Technology (2005).

[12] Ducheneaut, N., Moore, R.J., Oehlberg, L., Thornton, J.D., and Nickell, E., "Social TV: Designing for Distributed, Sociable Television Viewing," International Journal of Human-Computer Interaction 24(2), 136–154 (2008).

[13] Geerts, D., Cesar, P., and Bulterman, D., "The implications of program genres for the design of social television systems," in Proceedings of the 1st international conference on Designing interactive user experiences for TV and video, 71–80 (2008).

[14] Geerts, D., and De Grooff, D., "Supporting the social uses of television: sociability heuristics for social tv," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 595–604 (2009).
[15] Basapur, S., Harboe, G., Mandalia, H., Novak, A., Vuong, V., and Metcalf, C., "Field trial of a dual device user experience for iTV," in Proceddings of the 9th international interactive conference on Interactive television, 127–136 (2011).
[16] Lochrie, M., and Coulton, P., "Tweeting with the telly on!," in 2012 IEEE Consumer Communications and Networking Conference (CCNC), 729–731 (2012).
[17] Geerts, D., Vaishnavi, I., Mekuria, R., van Deventer, O., and Cesar, P., "Are we in sync?: synchronization requirements for watching online video together.," in Proceedings of the 2011 annual conference on Human factors in computing systems, 311–314 (2011).
[18] Bergkvist, A., Burnett, D.C., Jennings, C., and Narayanan, A., "WebRTC 1.0: Real- Time Communication between Browsers," W3C working draft 21 (2012).
[19] Hickson, I., "The WebSocket API," W3C Editor's Draft 9 (2013).
[20] Vaishnavi, I., Cesar, P., Bulterman, D., Friedrich, O., Gunkel, S., and Geerts, D., "From IPTV to synchronous shared experiences challenges in design: Distributed media synchronization," Signal Processing: Image Communication 26(7), 370–377 (2011).