Wersync: A WEB-BASED PLATFORM FOR DISTRIBUTED MEDIA SYNCHRONIZATION AND SOCIAL INTERACTION

Jordi Belda, Mario Montagud, Fernando Boronat, Marc Martinez, Javier Pastor
Universitat Politècnica de València
Grau de Gandia (SPAIN)
{jorbelva@epsg., mamontor@, fboronat@dcom., marmarc3@upvnet., fjpastor@dib.}upv.es

ABSTRACT
This paper presents Wersync, which is an adaptive and accurate web-based platform that enables distributed media synchronization and social interaction across remote users. By using Wersync, users can create or join on-going sessions for concurrently consuming the same media content with other remote users in a synchronized manner. Besides, social interaction is provided by sharing the navigation control commands and by integrating synchronized text chat channels. Additionally, two social presence mechanisms have been added to stimulate the participation of external users in on-going sessions on Wersync. By exclusively relying on standard web-based technologies, this platform can guarantee cross-network, cross-platform and cross-device support, which is a key point in the current heterogeneous media delivery ecosystem.

Author Keywords
IDMS, Interactive Media, Synchronization, Web.

ACM Classification Keywords
C.2.4 [Communication Networks]: Distributed Systems.

INTRODUCTION
This paper presents a first release of Wersync, which is an adaptive and accurate web-based platform that enables distributed media synchronization and social interaction across remote users. A key component of Wersync to support shared media experiences is the concurrent and independent synchronization of the media playout for different groups of users. This process is commonly known as Inter-Destination Media Synchronization (IDMS) and is essential to guarantee coherent interactions between the involved users [1]. Additionally, Wersync enables social interaction by synchronizing the execution of navigation control commands and by providing synchronized text chat channels. Moreover, two social presence mechanisms have been added to stimulate the participation of external users in on-going sessions on Wersync.

Our goal was not to develop just another web-based IDMS-enabled platform, but we believe Wersync is an outstanding platform compared to other existing ones (e.g., [2] and [3]) and can have an impact on the current media consumption paradigm. A key advantage of Wersync is the exclusive reliance on standard web-based technologies, such as HTML5 and Javascript. On the one hand, this guarantees cross-network, cross-platform and cross-device support. On the other hand, the use of HTML5 for media consumption allows selecting the most proper format (e.g., codec, resolution…) of the media file, according to the device capabilities and/or the network conditions. A parallel motivation for developing Wersync was to constitute a flexible and accurate testbed to objectively and subjectively assess the impact of several technological components and strategies on both the synchronization performance and the perceived Quality of Experience (QoE), respectively. Moreover, Wersync can be used to assess the tolerable asynchrony limits in different scenarios. We believe this platform is especially suitable for this last objective, as it can be deployed in large-scale, cross-device and cross-platform domestic scenarios, rather than in small-scale and artificial lab settings, as done in previous related works.

Although the design and development of Wersync are still in progress, the key technological components to achieve the targeted functionalities are already available, and presented in this demo paper. The audience will be able to interact and experience with Wersync, from whom we expect valuable feedback about its applicability, usability, design aspects and future functionalities. At least 4 consumption devices (PCs, laptops, tablets and smartphones) will be available and placed in separate locations (within the conference building) to allow the attendees testing the functionalities and the (interactivity and synchronization) performance of Wersync. Moreover, they can also use their own devices, as only a web browser is needed for that, without requiring the installation of any (third-party) hardware and/or software.

WERSYNC PLATFORM
Wersync has been developed by relying on mainly four technological components. The first one is the HTML5 video element, which allows embedding full-fledged media players into webpages. The second one is *Node.js*, which is an open-source, cross-platform runtime environment, written in

Javascript, for server-side and networking web-based applications. The third one is *Socket.IO*, which is a lightweight Javascript library that enables real-time bidirectional communication between web clients and a (*Node.js*) web server. By using *Socket.IO*, different types of messages (with different data types) can be sent via a single communication channel, by leveraging of its event-driven behavior. The fourth component is the clock synchronization between all the involved entities (e.g., by using Network Time Protocol or NTP) to ensure a coherent notion of time in the shared session. By combining these components, Wersync provides the targeted synchronization, interaction, presence and privacy features, which are briefly described in the next sub-sections.

Distributed Media Consumption and Synchronization

IDMS is provided by periodically exchanging messages between the involved entities, including their current playout times. These messages are also assigned a timestamp to compensate for the effect of network delays. Moreover, Wersync allows sharing and synchronizing the execution of the navigation control or VCR-like commands (e.g., *play, pause, seek*…). Only the master client (it will be initially the creator of the session) will have enabled such commands, but the other (slave) clients can request to become the new master if they want to take control of the shared session.

Social Interaction Channels

A synchronized text chat tool has been developed by using *Socket.IO* and the common timeline across the involved entities. Each of the chat messages is assigned a timestamp, which will be time aligned with the associated position of the video file in each of the remote clients (i.e., inter-stream sync). We also considered using Twitter as the text chat tool. However, the use of an ad-hoc chat tool provides: i) better interactivity (i.e., lower delays); ii) higher flexibility for adding and interpreting timestamps (i.e., for achieving sync); iii) "private" chat rooms for each group, instead of having a "public" chat room when using Twitter

Presence Mechanisms

Wersync targets social and real-time interactions between remote users. Therefore, it is strongly recommended to provide the potential users proper mechanisms to notify about the active members and the media being consumed at the platform at any time. Two mechanisms have been included to contribute to this. First, the currently active sessions, their members, and a brief description of the media being consumed, can be checked through an internal menu with drop-down lists. Second, an external presence mechanism has been also developed by integrating our platform with Twitter, by using its Javascript API. This way, every time a new user creates or joins a shared session, if he/she is logged in on Twitter, a new tweet will be posted informing about that (if desired). This tweet will include appropriate hashtags to uniquely identify the platform and session identifiers (e.g., *#Wersync* and *#session_id*), a brief description of the video being watched, and a URL to the session (see Fig. 1). This second mechanism allows external users to be aware of the activities of their Twitter contacts on Wersync, which will undoubtedly contribute to encourage their participation in on-going shared sessions.

Privacy Concerns

Despite the notifications via Twitter, the adopted session-based approach allows restricting the members who can join each session on Wersync. Once a new user requests to join a session, a new message will be sent to the master of this session, who can accept or reject that request. Likewise, the use of a "dedicated" chat room for each session also contributes to guarantee privacy in Wersync. Moreover, the chat messages can be encrypted (if desired).

Demo videos showing the capabilities of Wersync can be watched at goo.gl/6NjDRf.

FUTURE WORK

First, we plan to extend the capabilities of Wersync to enable the concurrent synchronization of external media files, from which users will provide their URLs. Second, we plan to integrate interactive and synchronized audiovisual chat channels. Third, we want to improve the User Interface Design. Finally, we also want to synchronize live services.

ACKNOWLEDGMENTS

This work has been funded, partially, by the "Fondo Europeo de Desarrollo Regional" (FEDER) and the Spanish Ministry of Economy and Competitiveness (MINECO), under its R&D&I Support Program in project with reference TEC2013-45492-R. We also thank "Samsung Electronics Iberia, S.A." for the support in the testing stage.

REFERENCES

1. M. Montagud, et al. Inter-Destination Multimedia Synchronization; Schemes, Use Cases and Standardization, MMSJ, 18(6), 459-482, Nov. 2012.
2. I. Vaishnavi, et al., From IPTV to synchronous shared experiences challenges in design: Distributed media synchronization, Signal Processing: Image Communication, 26(7), 370-377, August 2011
3. M. Wijnants, et al., "synchronous MediaSharing: social and communal media consumption for geographically dispersed users", MMSys'12, North Carolina (USA), Feb. 2012.

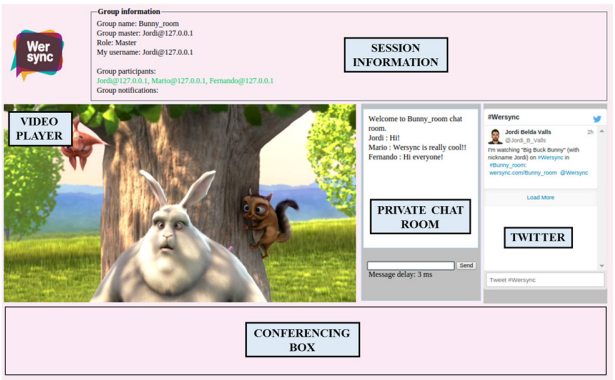

Figure 1. Components and Structure of Wersync.