Remote Temporal Couplers for Multiple Content Synchronization

Ricardo M. Costa Segundo
Informatics Department
Federal University of Espirito Santo - UFES
Vitória, ES - Brazil
rmcs87@gmail.com

Celso A. S. Santos
Informatics Department
Federal University of Espirito Santo - UFES
Vitória, ES - Brazil
saibel@inf.ufes.br

***Abstract*— Watching TV shows changed in both technology (3D, iTV, smartTV, ...) and source (P2P, IPTV, CDN, ...). Also the user can access digital content from diverse sources by using one or more devices for composing his presentation. This content can range from additional media objects to media streams transmitted in broadcast. The challenge is to make the clocks associated with the presentation of content from different sources (Extra Contents – EC) aligned under one particular clock (the Main Content one – MC) at each site (User Device – UD) where they will be presented. In other words, a MC from the same source is presented in different times for users in different locations, and the EC accessed by each UD will be synchronized with the MC being presented at that moment through a local clock that aligns extra and main content. The proposed model allows any content supplier to offer synchronized contents to a MC regardless any explicit synchronization information being provided by the MC provider. Two evaluations were applied to validate the proposal: a frame correlation and an accuracy test. They have shown that the model worked and achieved a fine-grained precision.**

Keywords—Synchronization, Multiple-Contents, Multiple-Sources, Temporal Couplers;

I. Introduction

Watching movies, TV series, sport events, news and other shows changed. Decades ago it was necessary to stay tuned on a TV or go to cinemas to watch movies and shows. As technology evolved, also did those options: colors appeared, analog content turned into digital, interactivity arisen (and died in many cases), 3D, high/ultra resolution and second screens are only part of this evolution. But while this evolution happened, new ways of watching were born: Web repositories, video streaming, IPTV, P2P players…, are some other ways that people watch their programs and shows.

Although these entire changes, one thing remains: the content (called here Main Content). It doesn't matter if you watch a movie in your TV, in cinema or through Internet: the movie will be the same. This content is used in this work to connect it with other contents regardless where people are watching it.

Besides, the user is no longer limited to consume a single content. He can access digital content from other sources by using one or more devices for composing his presentation environment. This content can range from additional media objects (statistics of a tennis match) to media streams transmitted in broadcast (listen to the radio narration of a soccer match shown on TV at the same time).

On the other hand, these contents sources aren't necessarily under the control of who generates the main content. Synchronization in the user's environment can be compromised because each provider delivers its own content independently and with no previously synchronization specification that binds them.

It is therefore necessary to introduce a model that allows the user to mashup his multimedia content in a synchronized way. The user can attach to the main content complementary services that are synchronously presented with the MC regardless any specification within the main content, because such specification is made with the user's choice and the extra content provider.

The approach proposed is based on three main entities related to content synchronization called Main Content (MC) Provider; Content Suppliers (CS) and User Devices (UD). In addition, it takes three general assumptions:

1. MC Provider offers no explicit timing information in its content (video, audio or metadata) to support the synchronization of content provided by other actors (the CS) with a MC presented in UD;
2. It is possible to generate temporal couplers related to the MC presentation by processing this one;
3. These temporal couplers support the synchronization of the content provided by CS during the MC presentation.

Following sections detail the proposal and its limitations and are organized like this: section 2 defines the study scenario; section 3 shows how to use the temporal couplers to provide multiple content synchronization; Section 4 presents a proof of concept to validate the proposal; followed by section 5 which presents some works related to the multiple content synchronization issue; finally, section 6 presents the final considerations.

II. Multiple Contents Integration

A traditional video presentation involves a single **User Device** (UD) that is able to decode and present this single content (the **Main Content** or MC) originated from a unique **source** (the **MC Provider**). In a non-traditional scenario [1], the **presentation environment** is composed of multiple **user devices** (TV, smart phones, tablets, etc.) able to present multiple **contents** delivered for multiple **sources**.

In this situation, the user accesses a mashup of digital contents that may have no explicit synchronization defined to orchestrate the combined presentation.

Mashups are applications generated by combining content, presentation or other applications functionalities from disparate sources. They aim to combine these sources to create useful new applications or services (the offer and consumption of data between two devices) to users [6]. From this point of view, a TV is a consumption device while a TV Station offers audiovisual content to these devices (service). New contents can be added form other service, such as interactive TV applications.

Following the TV scenario, a viewer may add other services and contents that could enhance his experience while watching a TV show broadcasted by a TV Station (MC Provider). Typical examples of services are subtitles and audio streams from different idioms related to a TV movie. Other sources besides the TV Station can act as MC Provider to the TV (CDN, P2P, USB, etc.). In fact, it is not an uncommon scenario if one considers VoD streaming services like Netflix, Hulu and Amazon as MC providers. In this situation the viewer should be able to combine different services and create a mashup of his contents: watch a movie provided by his streaming service; use native subtitles or import them from a Web server with new languages or inclusive services (like audio description [2] or sign language [3]).

This combination of services and contents however brings the following issue: how synchronize these multiple extra contents for each user in its environment, since the contents are transmitted through different channels and from different sources that are not explicit synchronized among them? This paper aims to exactly tackle this issue.

III. Timeline Aligment

Multimedia synchronization is about providing coherent playout of orchestrated contents. In a broader view, coherence in a playout addresses temporal, spatial, semantic and sensorial constraints related to the combined content presentation, although this paper focus only on the temporal issue.

The problem addressed can be seen as a clock synchronization problem: one intends to provide a solution to synchronize contents from multiple sources for each user presentation environment. The solution for this problem in a centralized system is trivial using a centralized server that will impose a global clock for the whole system. In a distributed system the issue is more complex, but in a first look, there is no central component that controls all users and contents: each user is independent from the others, synchronizing contents with the main content in his environment.

The problem to be solved in our work is a bit different than previous ones. A global clock could not solve the problem, since each content will be presented in a different time at each user. This happens because the MC that groups them also arrives in each user at different times. The solution suggests an alignment of each content timeline (presentation) with the main content in each user, where a mashup of different contents is done.

The challenge is to make the clocks associated with the presentation of each content from different sources to align under one particular clock at each site where the mashup will be presented. In other words, a MC from the same source is presented in different times for users in different locations, and the extra content and services accessed by each user will be synchronized with the MC being presented at that moment through a local clock that aligns extra contents and MC.

The proposed model allows any content supplier to offer synchronized contents to a MC regardless any explicit synchronization information being provided by the MC Provider.

A. *Model Entities*

The proposed model for multiple sources content time synchronization is composed of three entities, as shown in Fig. 1.

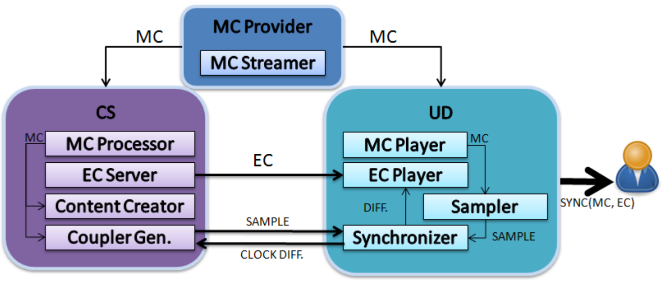

Fig. 1. Entities Composition for Content Synchronization

1) *The Main Content (MC) Provider*

The MC Provider component is responsible for streaming the MC, which acts as main reference to couple Extra Contents (EC) from CS in each UD. MC usually is a video stream transmitted over an accessible channel to UD and CS.

2) *The Content Supplier (CS)*

The CS component is responsible for providing EC related to a MC requested by UD. The MC Player plays or analyses the MC and is used as input to the Coupler Generator and Content Creator.

The EC (Extra Content) Server provides the content created by the CS to the users. These contents are treated as messages that are sent from the CS to the UD, containing all necessary information to play and synchronize extra and main contents. These messages are modeled as notifications, as detailed in section 3.2.

The Content Creator is responsible for generating the EC provided by CS to UD. EC can be generated: (i) manually by an operator that watches the same MC delivered to the user and derives new contents; (ii) automatically processing the contents of MC (ex.: subtitle generation for live events [3][11]); (iii) an application that connects existing contents to the MC, providing them as services, such as subtitles, audio tracks or accessibility services.

The Coupler Generator creates synchronization points that allow connecting the MC with the local clock (time counter for a particular device). In the proposed model, a coupler represents the temporal relationship between a matched scene in MC and the local clock of an entity

A matched scene is defined as an occurrence of a singular situation or condition on the MC during its presentation. An example of a particular situation is the occurrence of an advertisement break during a TV show. Multiple techniques can be used to identify when scenes occurs in the entity (CS). Some of these techniques depend on the MC Provider, such as audio/video watermarking or Presentation Time Stamps Extraction; meanwhile audio/video fingerprinting [10] and processing [12] are examples of mechanisms that can be used independently of that MC Provider.

3) The User Device (UD)

The UD component is responsible to control one or more services provided by CS and playing the MC.

The MC Player decodes and reproduces the MC delivered by the MC Provider. The EC Player is responsible for the reproduction of the EC provided by the CS.

The Sampler extracts samples from the MC (frames, audio tracks, metadata, etc.) to be used in the synchronization mechanism. A sample is defined as a portion of MC that is used to uniquely identify the timing interval or instant of a particular scene during MC presentation. This sample will relate the MC presentation with the CS in the UD, matching the scene with the coupler generated in CS.

The Synchronizer uses the samples extracted from UD over the received MC to create a synchronized (or aligned) timelines with the CS Coupler Generator.

B. *Timeline Alignment*

The proposed synchronization approach considers that: (1) any CS can provide synchronized content to the user's final application without depending on a explicit time specification sent by the MC Provider; (2) the method can be implemented at application level or lower, not depending on specific middleware or platform; (3) multiple contents may be synchronized on the same local environment, without direct communication among providers; (4) all CS entities involved in the synchronization process have access to MC; (5) it is not possible to guarantee that Local Clocks have the same value(no entity is previously synchronized with any other one).

It is important to emphasize that this approach does not aim to synchronize both streams in the Global Clock; the objective is to present synchronous notifications generated by each CS in all UDs that requires contents generated by this CS. The synchronization is guaranteed in respect to each UD Local Clock that dictates the mashup presentation integrating MC and CS in this UD. The proposed solution is to align the timeline of the CS's with the UD one, using the MC as reference in order to present CS contents synchronized with MC in UD.

1) Dynamics of the time alignment

The Fig. 2 depicts how the timeline alignment works in a situation involving a content supplier (CS_1) and a user device (UD_2). Each entity has its own local timeline: instants prefixed with T_1 are related to the CS local timeline and T_2 to the UD one. S_X is a scene that belongs to the MC and is presented, in different instants, on CS_1 and on UD_2; T_{ij} is the j^{th} timestamp in the Local Timeline of an entity *i*. It indicates when a j^{th} notification is generated by CS_i or is delivered to the UD_i.

A notification N_i^{Sx} generated by a CS_i triggered by a scene S_X to a UD is formally defined by the tuple:

$$N_i^{Sx} = \{n_i, ts_i^{Sx}, act_i^{Sx}, ec_i^{Sx}\}$$

where: n_i identifies the n^{th} notification generated by CS_i when a scene S_X in found in its *MC*; ts_i^{Sx} is the timestamp when S_X occurs at CS_i based on its local time; act_i^{Sx} identifies the notification type sent to the application running in UD (play a video, update a statistic, etc.) and; ec_i^{Sx} is an extra content used as a parameter to the act_i^{Sx} (it can be a specific data for the application or a URI to a digital content or media stream).

Considering again the scenario in Fig.2, in the beginning, MC is delivered to both entities, but a same scene S_A that belongs to MC arrives in CS_1 and UD_2 at different instants

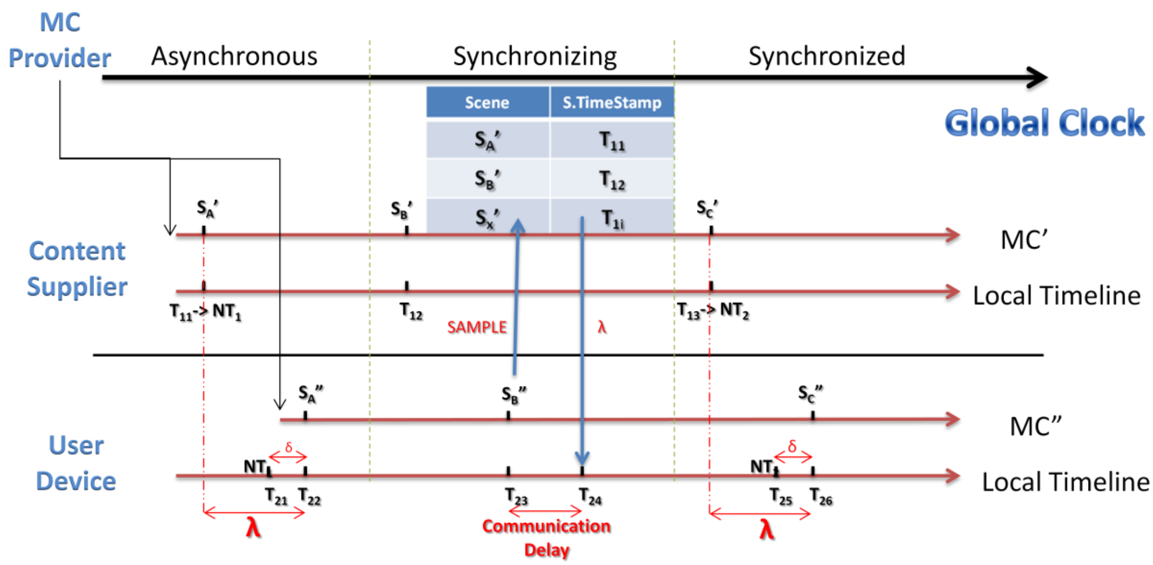

Fig. 2. Synchronization Scheme: (left) – period without synchronization of MC and EC in UD; (center) – notifications exchange for user timeline alignment; (right) – with its timeline aligned, the UD can present both contents synchronously.

(T_{11} and T_{22}, respectively) due the variation of transmission time or the local MC starting time. Before timeline alignment takes place (identified in the figure as the asynchronous phase) without information about transmission delays and without previous synchronization of CS_1 and UD_2, it is not possible to calculate the difference time (λ) between presentation of S_A' and S_A" (the reproduction of the scene A in CS_1 and UD_2, respectively). Neither in T_{21} is possible to calculate how long will take (δ) to achieve the presentation time of S_A", this way there is no guarantee that the notification NT_1 will be performed synchronously with the MC on UD_2, as specified by CS_1. However the presentation time of S_A' in CS_1 is known: T_{11}. It's also known that T_{22} occurs at the same presentation time (time relative to MC) of S_A" in UD_2 and that $T_{11} \neq T_{22}$ due transmission delay variation related to the Global Clock. The solution aims in finding the difference δ between presentations of same scenes on different entities.

The timeline alignment is accomplished as follows: for each scene S_X presented, CS creates a coupler of type {S_X, *Scene Timestamp*} that identifies the local time (timestamp) when the scene occurred in its local timeline. A sample from a scene S_B" is taken in UD. It is sent to the CS that matches this sample with its couplers, returning to UD the time when the correspondent sample's scene occurred in CS timeline. UD then updates its local timeline, T_{UD}, using (1). In the equation, *curTime* is the local clock when the sample was extracted and sent; *comTime* is the delay between sending the sample and receiving the response; and *respTime* is the time contained in the response (scene timestamp of the CS table). *curTime – respTime* gives the difference (λ) between the scene presentation in CS and UD. The *comTime* takes place in equation to remove the delay of messages communication.

As result, the local clock of UD may be updated with the timeline value. This new value of T_{UD} implies that the local timestamp when a scene S_X occurs in CS will be the same when the scene S_X occurs in UD.

$$T_{UD} = T_{UD} + (curTime - respTime - comTime) \qquad (1)$$

Hence in Fig.2, in synchronizing phase, is calculated the local timestamp in which a scene S_B" occurs in UD_2 and CS1 has the local timestamp when S_B' occurred locally. With these values it is possible to align the timelines, finding the difference of time between S_B" and S_B'. UD_2 and CS_1 perform the synchronization and UD_2 updates its timeline. After the alignment of the timelines T_1 and T_2, if a scene S_X' occurs in T_{1i} in CS_1, the corresponding scene S_X" will occur in $T_{2J} = T_{1i}$ in UD_2. This implies that it is possible to know how long a notification containing the extra content produced by CS_1 is delayed or early to be synchronized with the MC presented in UD_2.

Once the alignment is ensured, when a notification NT_i arrives, it is possible to calculate how long it is in advance or late and present it synchronously with the MC in UD. The time is the result of (2), where: *ts* of NT_K is the time when it occurs in the local timeline of the CS_x, meaning that it should be performed at this same time in the UD local timeline; $rcptTime_i$ is the local timestamp when the notification NT_K arrived in the User. Then, δ_i is the delay or the advance of the notification NT_K in the UD local timeline.

$$\delta_i = NT_K.ts - rcptTime_K \qquad (2)$$

If $\delta_i \geq 0$, NT_K arrival is in advance to the MC and should wait δ_i to be executed, else if $\delta_i < 0$, NT_K is late and should be discarded, presented right away or the MC presentation should be delayed in UD, delaying local timeline and giving time to notifications arrival and presentation, such that future δ_i have positive values.

In synchronized phase of Fig. 2, when a scene S_C is found at T_{13} by the *Coupler Generator* of the CS_1, a notification N_1^{Sc} related with this scene in MC_1 will be send to the UD_2 with the timestamp T_{13}^{Sc}, and necessary action and parameters, such as:

$$N_1^{Sc} = \{3, T_{13}^{Sc}, play^{Sc}, youttube.vrxs4.fr\}$$

2) *Multiple alignment*

When an UD requires multiple extra contents, it will make a timeline alignment with each necessary CS, creating for each one a new local timeline that will be responsible to present this CS corresponding notifications with the full presentation.

If a particular CS (CS_1) is not able to generate couplers, it can partially act as a UD, aligning its own local timeline with other CS (CS_2). In this case it can import the CS_2's coupler table or forward a User's (UD_1) synchronization request to CS_2 and after synchronization it starts sending to UD_1its (CS_1) contents. Since scene S_1 is presented with the same local timestamp in each entity (CS_1, CS_2 and UD_1), it is possible to present CS_1 contents synchronized in UD_1through the sync information provided by CS_2.Thus the timeline of CS_2 has been propagated to CS_1 and UD_1in order to achieve synchronization. It implies that if an entity Y has aligned its timeline with X, and Z aligns with Y, it will result that Z is also aligned with X.

At this point, a question arises: what happens if the presentation of the MC is delayed (stopped to buffer) after it started being presented? Wouldn't the timeline be misaligned? Yes, it would, and all synchronization of the notifications would be lost. There are two solutions for such problem: (i) resynchronize; (ii) the MC Player can take the delay value and then add it to the current timeline value.

IV. PROOF OF CONCEPT IMPLEMENTATION

This section presents a Proof of Concept (PoC) of the approach to demonstrate its feasibility and to evaluate if its concepts can be used with real situations involving multimedia content synchronization. The PoC aims to simulate a real world scenario: the MC corresponds to a movie and the EC corresponds to subtitles and optional sound tracks added on demand by the user and provided by different sources in perfect synchronization with the movie he is watching.

The application was developed using HTML5 specification for both UD and CS. Video streaming (MC Provider) was represented by a video streaming server that provided HTTP video stream in OGG (Theora + Vorbis). This configuration allowed compatibility with multiple platforms (PC, Smartphone and Tablets).

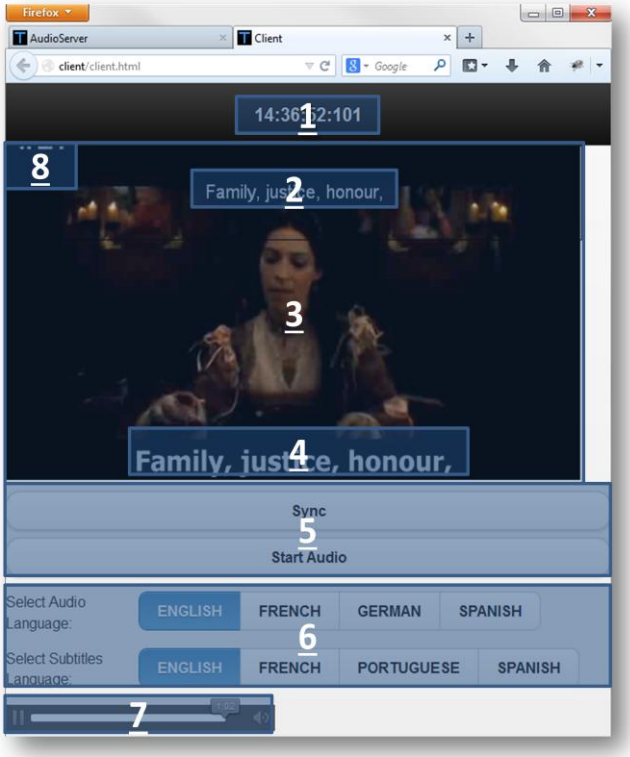

Fig. 3. Content Integrated User Interface

Two services were offered to the client application that executes in the UD: multiple language audio tracks (mp3 streams from the CS for each connected user) and subtitles (notifications containing the text to be presented on the screen). Each client could choose subtitles and languages to accompany the presentation of the MC. In this case each EC is provided by a different CS, this way the UD manages two internal timelines, one for each CS.

Fig. 3 shows the implemented user interface. It is divided in eight sections: 1) value of the local timestamp during MC presentation. The adopted format is [h: min: s: ms]. It is used in synchronization tests; 2) a canvas to draw the service subtitles; 3) MC presentation area (HTML5 video player); 4) subtitles burned on the video for means of evaluation; 5) synchronization buttons that allows the client to synchronize or not the extra contents. This allows analyzing the application before and after the synchronization process; 6) language selection for audio and subtitles; 7) extra audio tracks player (HTML5 audio player); 8) video sampling area.

The model requires the extraction of a sample from the MC to accomplish the alignment. As described, this can be done in multiple forms, like audio/video analyses, time information and both watermarking and fingerprinting techniques. In the PoC implementation, a video watermarking was the best choice. The watermarking added to MC frames corresponds to a binary code that can be easily processed by the Coupler Generator in CS. In a real world application, this is far from being the best solution because has direct impact on the frames and changes the video. However for the PoC using the binary watermarking provides a simple manner to validate the model and brings an interesting borderline effect too: with the added watermarks, it is possible to visually analyze the synchronization process from a recording of the experiment (as will be seen in next sections).

Two evaluations were developed to evidence the correct behavior of the proposed synchronization mechanism: (1) a frame correlation to evaluate if the timeline alignment locally synchronized each client with the CS timeline; and (2) an accuracy test to evaluate how accurate the subtitles service was compared to the subtitles that were present in the video.

A. *Frame Correlation Test*

The first evaluation was performed with two clients on two different devices (a Galaxy Tab 2 7.0’ and one 10.1’ running the Mozilla FireFox Browser for Android). They were recorded with a camera at a 60fps rate during the transmission of a movie from the streaming server. The experiment starts with no synchronization at beginning. Thus, the clients would not have synchronization at Global Clock neither in the relative MC local timeline. In this state, both clients would present different local timestamps for a same scene S_X.

Some time later the synchronization mechanism is applied. As result both clients will present the same timestamp for a scene S_X.

Fig. 4 and Fig. 5 show frames from a testing record. In the test both User Devices ($User_1$ and $User_2$) are recorded in parallel (both are captured simultaneously: $User_1$ on top of $User_2$). Each figure presents two frames: the frame on left (a) happen milliseconds (α) before the frame on the right (b). α is the time that separates the presentation of a scene S_X on $User_1$ and the same S_X on $User_2$. The scenes can be identified due the watermarking. The same watermarking indicates that it is the same scene. The watermarking is highlighted with within the red rectangle. The red ellipses highlight the timestamp in each case.

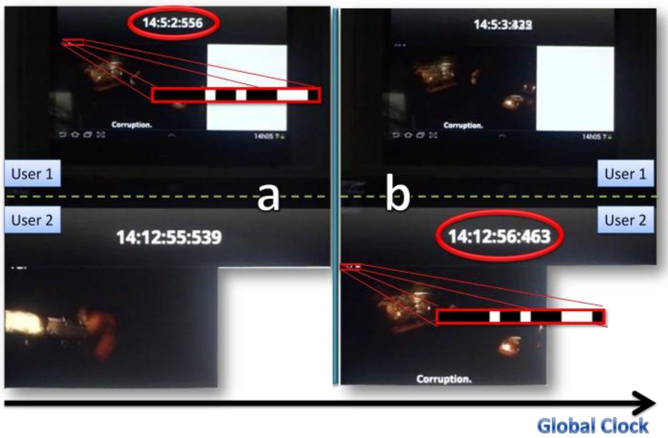

Fig. 4. User Devices in pre-sync phase

Fig. 4 shows the frames in pre-sync phase. On left (a) both devices are presented. Due to different initialization times, both screens present the MC in different times. As they present different scenes on frame (a), it is not possible to analyze if their timelines are aligned or not. This conclusion is taken after looking frame (b), where $User_2$ presents the same scene of

$User_1$ in the first frame. The local timestamp for the $User_1$ in the scene S_{1166} (value of the decoded watermark) is 14:05:02:556. In (b) the value of local timestamp for $User_2$ 14:12:56:463.The difference of 07:53:907 implies that their timelines are not aligned to the CS (remember transitivity: if U_1 and U_2 aligns with CS, U_1 is also aligned with U_2) and that if they receive an extra content to be displayed, they will be not presented in the correct instant (i.e. not synchronized) with the MC for each device. If they were aligned, they should present the same value for the same scene.

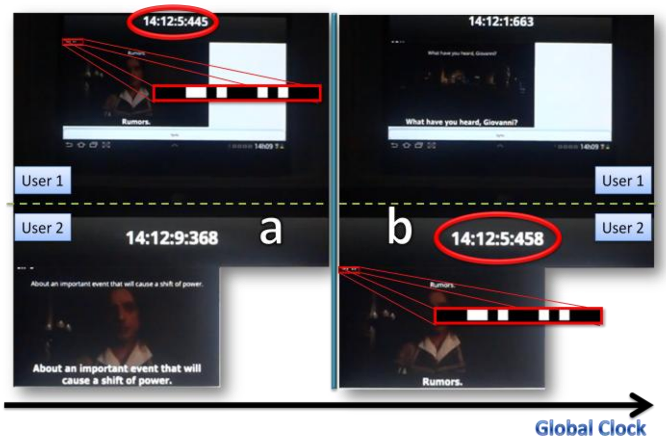

Fig. 5. User Devices after sync phase

[redacted] Fig. 5 presents the same scenario of Fig. 4 after the synchronization. In Fig. 5, a visual inspection of the main content at each user show that both devices continue asynchronous regarding the Global Clock (IDMS). But now, analyzing the local timestamp of each user when they present the same scene (S_{1550}) the values are: 14:05:05:445 for $User_1$ and 14:12:05:458 for $User_2$. The difference between $User_1$ and $User_2$ for the same frame (S_{1550}) becomes 13ms.

Now, a notification that should be performed with scene S_{1560} of MC, will be correctly presented in both devices, each one at its own time, but synchronously with MC.

B. Frame Accuracy Test

The second evaluation, an accuracy one, aims to test the accuracy of the method, comparing the delivery of extra content of a CS with contents burned in the video of one UD. For this purpose, the subtitles are used. The CS provides subtitles based on the same reference times used with the ones burned into the video frames. Also, comparing the subtitles is possible due to its visual characteristics, so a frame by frame visual analysis reveals the difference time between burned and CS subtitles.

This evaluation was performed in a Windows 7 i5 PC, to facilitate the record of the screen at a higher frame rate. In this experiment the desktop recorder FRAPS (fraps.com) tool was used to obtain 100 frames per second of capture rate. The subtitles were added to the MC using the FFMPEG software. VirtualDub software enabled the frame by frame analyses of recorded video.

Fig.6 – Frame by Frame Analysis of accuracy recording

Fig. 6 shows the experiment frame by frame view. On left neither CS subtitles nor the burned ones on video are shown. However the following recorded frame presents both subtitles. The capture frequency was 100 fps, indicating that the error of synchronization can be of 10ms at maximum considering the recording. But the MC frame rate of 30 fps means that the recording is faster than the MC.

This implies that the MC subtitles update rate is approximately of 33ms. In this case the lower rate must be considered as the accuracy. Considering the worst case, where the subtitle should be shown right after the MC video update and the subtitle from CS is shown right in the start of the following frame, the error between their presentations is 33ms (video frame rate). Thus the accuracy of this experiment is ≥ 33ms.

This fine-grained precision implies that the subtitles service would be considered synchronized by the user once the QoS required to such relation between video and text is ±250ms[8]. This also means that any other relation with a requirement ≥ 33ms could be supplied, including lip-sync (±80ms).

A final note about the PoC considers the question where the MC time can be changed during the presentation. As this implementation uses a video streaming, delays in transmission can happen and the video may be paused to buffer. Or the user in the player can pause or even advance video presentation. In these cases, the synchrony will not be affected. When the MC suffers any interruption, a video player controller is able to identify what happened to the video: if it was paused; advanced or backward; and also how long. This difference then is added to the timeline.

V. Related Works

Related works in the literature have contributions and definitions that converge with the ones presented in this one. The following paragraphs briefly describe some of them.

Matsumura and Evans [9] describe the study to develop a system to personalize broadcasting content using the Internet. In the system, the broadcaster provides additional content for personalization via the Internet, which is synchronized with the broadcasting program on air. The receiver combines both types of content to generate a personalized presentation. Both contents are produced and distributed by the broadcaster and the synchronization between subtitles, audio and video is achieved using timestamps. Therefore they use a design to buffer broadcast content for a certain amount of time at the receiver before presentation to keep both contents synchronized.

Howson et. al. [10] propose a system, for the deployment of second screen personalized TV services, which enables the rendering of content components, delivered independently over broadband and broadcast networks, to be accurately synchronized in user devices. The solution is based on the addition of an auxiliary component timeline associated with each group of media components delivered over the broadcast and the broadband networks. In the solution both content provider and broadcaster needs to add an additional timeline to their streams.

Soursos and Doulamis[11] propose a framework for creating a hybrid solution, where end users can enrich broadcast content with Internet-based enhancements so they can enjoy improved and personalized viewing experience. This allows for content composition, by merging the broadcast content with Internet content. To synchronize content they require incorporating network synchronization strategies for the internet flows for a just-in-time reception with the broadcast streams at the receiving end.

Fink et. al. [12] describes a framework for combining mass media with a highly personalized Web-based experience. Using the ambient audio originating from the television, personalized applications are available. They did tests of the proposed applications with controlled conversational interference and with “living-room” evaluations.

Mendes and Santos [14] present a systematic review which aims to survey existing research on synchronization video MC with other content. It classifies works in regard to synchronization type, form of transportation, channels used, control schemes, location and information about contents. Works related to time relations consider that multiple contents must be presented in a limited time interval to be synchronized. They are directly related to the scope of this work, being 69% of the works of the review. Highlights of the review are: most of papers (85%) did not present the accuracy of synchronization; 53% uses synchronization points within the main content, while only 16% use an additional channel such as the Internet itself; articles consider four points to perform the synchronization process: on the server (40 percent), customer (43%), with a pre-synchronization (13%) and by other means (4 percent).

Although these works influence in the design and implementation of this work, each one has their own unique contributions and limitations. Both in papers and systematic review, third-party content synchronization is little explored and most work has focused on specific platforms and technologies. However, the approach presented here brings the possibility to sync content from multiple sources without dependence on a particular technology.

VI. Conclusions

Television, Web Repositories, P2P, video on demand, IPTV, and many others are examples of how people watch their programs and shows nowadays. A same show or even movie can be watched in different platforms by multiple users. These platforms have different protocols, formats and characteristics. But at least one thing remains: the movies itself, in other words the content (the Main Content).

Enhancing it with alternative services, can generate personalized presentations based on each user will. This enhancement can be featured by the use of interactive applications with the videos, inclusive tools, and many others.

It is in this scenario that this paper proposed a model that considers the MC sent by those platforms as a waypoint to add other services provided by other entities.

This paper proposes the use of temporal couplers that synchronize each of the content accessed by the user, with the Main Content. The alignment of timelines allows synchronizing multiple contents with each user in his environment. At the end of synchronization, each customer can receive extra content synchronized with the main content that is being presented. Even though each client receives the main content in a different Global time (due to the delay of transmission, or delayed access), the extra content will adapt to each user environment ensuring a local sync.

The next efforts aim to work on two vital points not addressed in this paper scope: the discovery and publication of CSs; and identification of MC scenes. Various studies already approach similar problems ([4], [5], [7], [10], [11], [12], [13] and [15]), not becoming the initial focus of this work.

Acknowledgment

The authors acknowledge the financial support provided by FAPES.

References

[1] Zixia Huang, Klara Nahrstedt, and Ralf Steinmetz. 2013. Evolution of temporal multimedia synchronization principles: A historical viewpoint. ACM Trans. Multimedia Comput. Commun. Appl. 9, 1s, Article 34 (October 2013), 23 pages. DOI=10.1145/2490821 http://doi.acm.org/10.1145/2490821.

[2] Benoît Encelle, Magali Ollagnier-Beldame, Stéphanie Pouchot, and Yannick Prié. 2011. Annotation-based video enrichment for blind people: a pilot study on the use of earcons and speech synthesis. In The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '11). ACM, New York, NY, USA, 123-130. DOI=10.1145/2049536.2049560.

[3] T. M. U. de Araújo et al.An approach to generate and embed sign language video tracks into multimedia contents, Information Sciences, Volume 281, 10 Oct. 2014, pp. 762-780, ISSN 0020-0255, http://dx.doi.org/10.1016/j.ins.2014.04.008.

[4] P. Shrestha, M. Barbieri, and H. Weda. Synchronization of multi-camera video recordings based on audio. In Proc of ACM International Conference on Multimedia, pages 545–548, Augsburg, Germany, Sep. 2007.

[5] R. Malhotra. Hybrid Broadcast Broadband TV: The Way Forward for Connected TVs. IEEE Consumer Electronics Magazine,v.2, n.3, p.10–16, July 2013.

[6] J. Yu et al. Understanding Mashup Development. IEEE Internet Comp., v.12, n.5, p.44–52, Sep.–Oct. 2008.

[7] G. Blakowski and R. Steinmetz. A media synchronization survey: Reference model, specification, and case studies. IEEE J. Sel. Areas Commun., v.14, n.1,pp.5–35, 1996.

[8] R. Steinmetz. Human perception of jitter and media synchronization. IEEE J. Sel. Areas Commun., v.14, n.1, pp. 61–72, 1996.

[9] K. Matsumura and M. Evans. Personalization of broadcast programs using synchronized internet content, In: IEEE ICCE'10, pp. 145–146, 2010.

[10] C. Howson and E. Gautier. Second screen TV synchronization. In: IEEE ICCE'11, pp. 361–365, Sep. 2011.

[11] S. Soursos and N. Doulamis. Connected TV and beyond. In: IEEE Consumer Communications and Networking Conf., pp. 582–586, 2012.

[12] M. Fink et al. Social-and interactive-television applications based on real-time ambient-audio identification. In: EuroITV'06, p.10, 2006.

[13] M. De Castro and D. Carrero. Real-time subtitle synchronization in live television programs. In: IEEE BSMB'11, p.1–6, 2011.

[14] Segundo, R.M.C.; Santos, C. A. S. .Systematic Review of Multiple Contents Synchronization in Interactive Television Scenario. ISRN Communications and Networking, v. 2014, p. 1-17, 2014.

[15] KlaraNahrstedt, Bin Yu, et al, "Hourglass Multimedia Content and Service Composition Framework for Smart Room Environments", Elsevier Pervasive and Mobile Computing J., Vol. 1, No. 1, Mar. 2005, pp. 43-75