NORTHERN RESEARCH INSTITUTE

REPORT

15/2014
ISBN 978-82-7492-295-2
ISSN 1890-5226

DISTRIBUTED SYNCHRONIZATION OF HTML5 MEDIA

Author(s): Njål T. Borch, Ingar M. Arntzen

P.O. BOX 6434, TROMSØ SCIENCE PARK
N-9294 TROMSØ, NORWAY
TEL: +47 77 62 94 00, FAX: +47 77 62 94 01
post@norut.no
www.norut.no

PROJECT NAME: **MediaScape** Project No.: 531

CONTRACTING: Contracting ref.:

Document No.: 15/2014 Document Type: Report Status: Open

ISBN: 978-82-7492-295-2 ISSN: **1890-5226** No. of Pages:11

Projectleader: Njål T. Borch Date: 12.12.2014

AUTHOR (S): Njål T. Borch, Ingar M. Arntzen

TITLE: **Distributed Synchronization of HTML5 Media**

Summary:
In this report we analyze the quality of synchronization we can expect when synchronizing HTML5 audio and video on multiple devices using Shared Motion. We demonstrate that the concept of Shared Motion enables sub-frame synchronization for video, and near perfect synchronization for audio. Experiments are conducted in real world scenarios.

Keywords: multi-device, synchronization, distributed, HTML5, video, audio, shared motion

Notices:

PUBLISHER: **Norut**

Distributed Synchronization of HTML5 Media

By Njål Borch and Ingar Arntzen, Norut & Motion Corporation

Abstract

People and households have a growing number of devices, and multiple devices are increasingly used simultaneously. The presence of multiple devices opens for interesting multi-device services and applications, but many of these require a relatively tight synchronization between the devices. Norut has created the concept of Shared Motion to address this problem, providing a generic and high quality timing mechanism for web based applications over the Internet. Motion Corporation was founded to provide Shared Motion as a commercial service. Motion Corporation offers *InMotion,* the first production quality implementation of Shared Motion, and the first hosted service of this kind.

In this report we analyze the quality of synchronization we can expect when synchronizing HTML5 audio and video on multiple devices using Shared Motion. We demonstrate that the concept of Shared Motion enables sub-frame synchronization for video, and near perfect synchronization for audio. Experiments are conducted in real world scenarios. See the Demo section at the end of this document to test it for yourself.

1 Shared Motion

The traditional approach to synchronization of media on different devices is based on either sharing of data (multicast/broadcast) or sharing of clock pulses (e.g., gstreamer, MIDI). The first only provides coarse synchronization, and the second is limited to local network environments where latency is low. However, the essential property needed for synchronisation is simply the media offset, or more precisely, how the media offset changes in time. *Shared Motion [MSV] encapsulates this perfectly. Shared Motion* is a novel approach to distributed media synchronization, providing a mechanism for multi-device synchronization on the Web, designed specifically to allow precise and highly scalable synchronization across the Internet. Shared Motion compensates for variances in latency, and can thus synchronize devices globally, independent of network carrier. Shared Motion is pure Web technology and works in any modern Web-browser, no plugins required. Still, even though Shared Motion is designed for the Web it is in no way restricted to the Web. As such, Shared Motion provides a general mechanism for distributed timing and coordination, applicable to all connected agents, from web-browsers on laptop computers to native applications on embedded devices.

For our experiments, Shared motion was provided by Motion Corporation [MCORP] via their public service InMotion. Client libraries are available as Open Source software.

2 Synchronizing media playback

Synchronized playback is one of the more challenging tasks in multi-device media. This is particularly true when viewers can observe more than one device, and use their keen senses to spot inaccuracy. So in order to please human observers, one should strive to synchronize videos to be on the same video frame at all times. In a traditional video, each frame lasts for 40ms (25fps) or 33.3ms (30fps). For higher frame rates this decreases to 16.6ms (60fps). If two devices playing back video are within these limits, they will appear to be perfectly synchronized. Synchronizing audio to video have similar demands to reach “lip sync”. For television applications, audio should be ahead of video by no more than 15 milliseconds and audio should lag behind video by no more than 45 milliseconds [IS-191]. For film, acceptable lip sync is considered to be no more than 22 milliseconds in either direction [KUDRLE].

The most difficult task however, is synchronizing a single audio signal from multiple sources. This is where human senses are most sensitive. For example, humans may recognize echo if identical/similar audio signals are separated in time by about 10 ms. This means that two perfectly synchronized loudspeakers will produce echo if the listener is situated 3.3m closer to one of the loudspeakers. This also corresponds to echo effects in a small office. A synchronization error of 1ms can be corrected by shifting one of your desktop loudspeakers 33cm in the right direction. A large auditorium echoes at about 100ms. In fact, for a 10 meter wide stage, the sound takes about 30ms to traverse the stage from one end to the other.

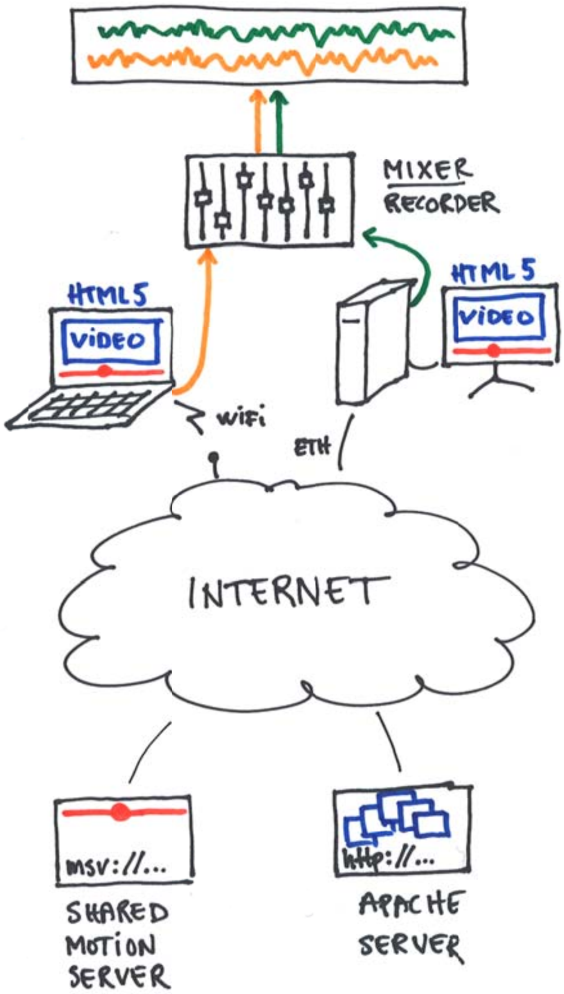
When synchronizing closer than 10ms, various audio effects might become apparent, like failing to hear one of the audio sources, interpreting false directionality, or even locating the audio source to be in the middle of your head [DSPGUIDE]. For these reasons, we regard 10ms to be a reasonable target for multi-device audio synchronization.

3 Synchronizing HTML5 Media Elements

As the HTML5 audio and video elements currently lack support for external timing, we have created a simple media synchronization wrapper in JavaScript to synchronise elements using periodical adjustments to playback rate. We also have a more primitive variant that simply adjusts playback offset by seekTo operations. However, altering playback rates gives the best result, both in terms of user experience and precision. The effect is typically one where the devices first play with an echo, which then disappears within a few seconds. The current implementation is simplistic, so results will only improve over those presented in this report.

4 Experiments

In order to measure errors in synchronization, we need a method for comparing the media output from two computers playing the same media. Due to limitations in video frame rates, we opted for a pure audio measurement.



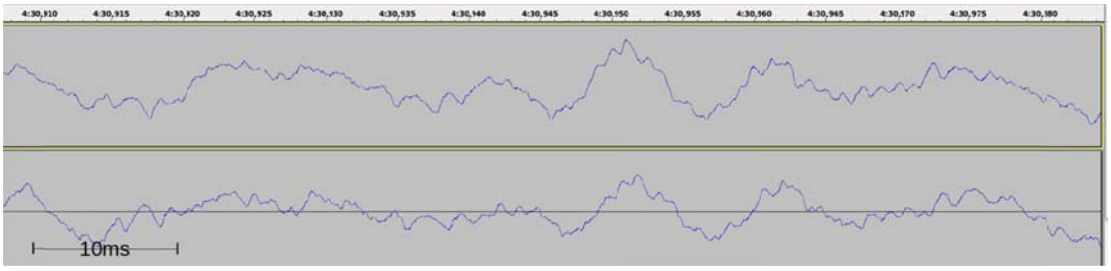
As shown in figure 1, two separate machines were set up, one stationary machine (Intel i5) on a gigabit ethernet connection and a laptop (Intel i7) on a 54Mbit WLAN. The video tests were performed using an external USB soundcard for the laptop, the audio test used internal soundcards on both. This did however not seem to affect synchronization. Both machines were running Linux during this test, but we have observed that browser type is more significant than operating system in this regard. From these machines, we connected the right channel from each machine to a sound mixer/recorder, which we used for recording the audio output. We ran both a pure audio test, as well as a video test. The Shared Motions were hosted by Motion Corporation, all files, including media files, were fetched from an Apache server. Both servers run on virtual computers in a Europe site at Google. No plugins were added to any of the browsers.

4.1 VIDEO SYNCH EXPERIMENT

We used a simple web page with a HTMLVideoElement. The video source points to an online video (Sintel 720p). Playback is controlled by Shared Motion and our JavaScript synchronization wrapper (using variable PlaybackRate). The web page was also hosted on the Motion Corporation home page, as it's one of their demonstration applications. (See Demo section below). The page was opened on both test machines manually, with no particular focus on synchronized loading as this does not affect the results. We disregarded the video (i.e., visual output) for this trial even though it was displayed by both devices. As the video is only 30fps, we also don't have any way to determine sub-frame accuracy for video.

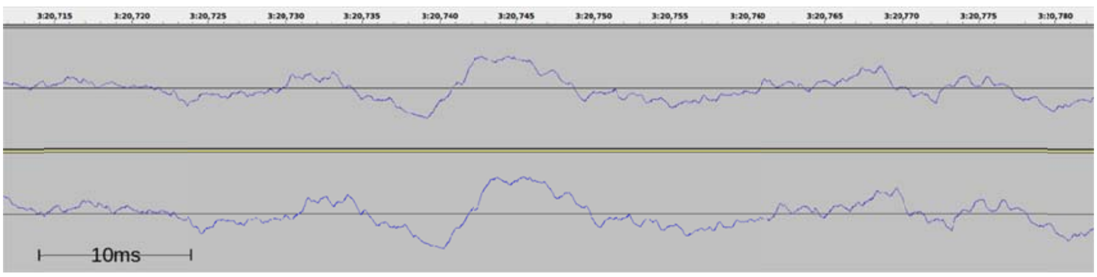
We performed three tests, one with Chromium, one with Firefox and one with both.

4.1.1 TEST 1 - FIREFOX & CHROMIUM



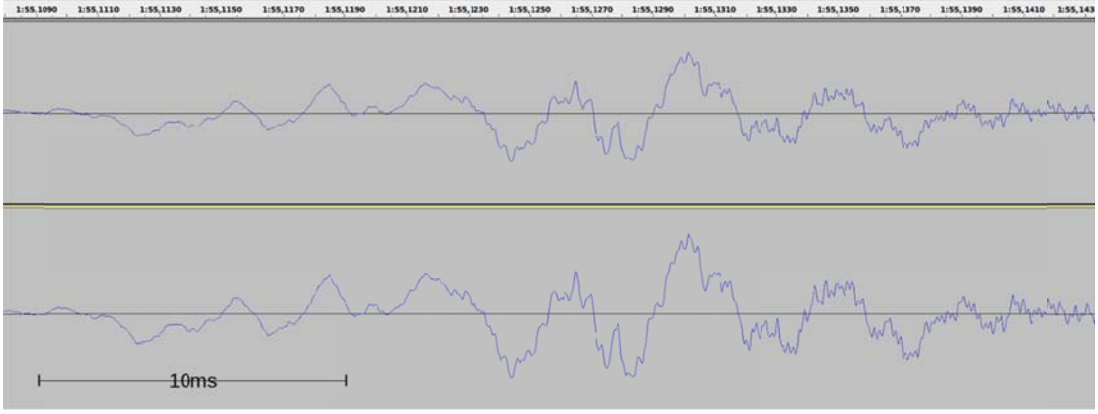
In figure 2, one device is running Firefox, the other running Chromium. We've added **7ms** skew to one channel in order to align them. The graph spans 76ms.

4.1.2 TEST 2 - FIREFOX & FIREFOX



In figure 3, both devices run Firefox. Again, we've shifted one channel **7ms** to align them. The graph depicts a total interval of 70ms.

4.1.3 TEST 3 - CHROMIUM & CHROMIUM



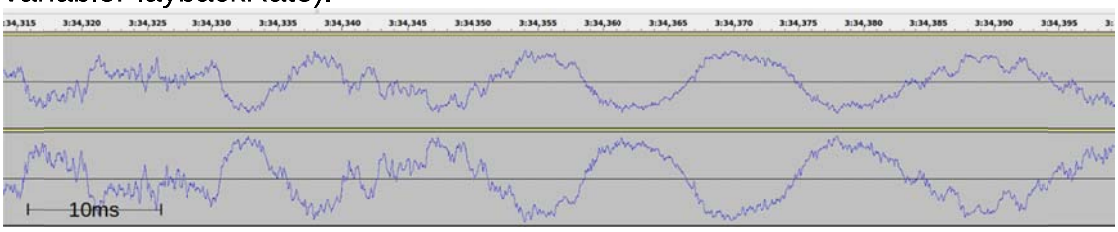
In figure 4 both devices run the Chromium browser. **2.5ms** of skew was added to align the tracks. The graph spans only 36ms in this case, approximately one video frame.

4.1.4 VIDEO RESULTS

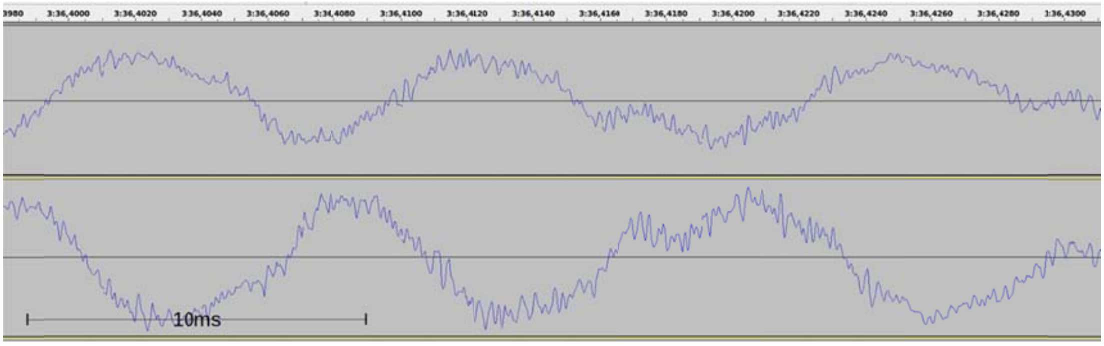
From these graphs we can see that Shared Motion combined with our relatively simplistic synchronization approach provides excellents results far within the necessary bounds for multi device media synchronization. In fact, even using different browsers the results are within audio synchronization bounds, meaning co-presented video will not produce a noticeable echo.

4.2 AUDIO SYNCH EXPERIMENT

We also performed an audio test, similarly using a test application provided by the Motion Corporation. It is a simple web page with an HTMLAudioElement. The media source is an online MP3 file, and with playback controlled by SharedMotion and our JavaScript synchronization wrapper (using variablePlaybackRate).



In figure 5, both devices play an MP3 file. 1ms skew was added to align the tracks. The graph spans 85ms. Apparently, the internal soundcard of our laptop is phase inverted.



In figure 6 we display from **the same experiment** as figure 5, **without** alignment. The graph spans only 33ms.

4.2.1 AUDIO RESULTS

It is clear that audio synchronization is very close to perfect according to our measurement systems. During this tests, several audio phenomena appears when using ear phones, such as stereo sounds, mono sounds, one speaker "disappearing" etc. This is likely due to our sensory system being challenged by getting the same sound from different places without echos being returned from walls and objects in the room.

As noted above, the 1 millisecond skew is equivalent of 33cm distance (at sea level). You can experience what this sounds like by placing yourself 33cm further from one speaker than the other.

5 Conclusion

We created a test with far from an ideal setup, being close to what we expect in a real world scenario. The networks are different, the machines are different and we even tested with different browsers and sound cards. Still, we are able to demonstrate that it is absolutely feasible to synchronize multiple standard web browsers supporting variable playback rates to well below frame accuracy, both for audio and video.

Crucially, this was achieved without any kind of direct communication between the two computers. The only communication is in-direct, cross-Internet via MotionCorporation servers. Furthermore, we were using different network technologies and unmodified web browsers.

As Shared Motion is basis for audio synchronisation, the underlying Shared Motion likely synchronizes even better. However, discovering this limit is within the scope of this report. By improving our synchronization wrapper, or even better, getting native support in browsers for external timing for audio and video players, web-based media synchronization will likely improve even further.

6 Demo

Shared Motion works right out of the box in most browsers, so testing it yourself is only a click away.

- Note that demos require login via Google.
- Note also that browser support for variablePlaybackRate provides the best results, so we recommend Chrome or Firefox for these demos.

Video link: http://www.mcorp.no/examples/sintel/
Video controls: http://www.mcorp.no/examples/sintel/ctrl.html

Open the video link in two browser windows. Control playback from the controls page (perhaps from your smart phone?)

- Note that Chrome may have issues with presenting video in multiple tabs concurrently. Open new Chrome Window or complement with other browser.

To better appreciate synchronized audio, make sure to open the video link on different computers. You may have to fiddle with audio volume levels to convince yourself that they are both outputting audio.

References

IS-191: *Relative Timing of Sound and Vision for Broadcast Operations, ATSC, 2003-06-26*

KUDRLE: *Sara Kudrle et al. (July 2011). "Fingerprinting for Solving A/V Synchronization Issues within Broadcast Environments". Motion Imaging Journal (SMPTE)*

DSPGUIDE: *http://www.dspguide.com/ch22/1.htm*

MCORP: *http://mcorp.no*

MSV: *Ingar M. Arntzen, Njål T. Borch and Christopher P. Needham. 2013. "The Media State Vector: A unifying concept for multi-device media navigation". In Proceedings of the 5th Workshop on Mobile Video (MoVid '13). ACM, New York, NY, USA, 61-66.*