Closing the Gap: Human Factors in Cross-device Media Synchronization

Mu Mu*, Lyndon Fawcett†, Jamie Bird†, Jamie Jellicoe†, Steven Simpson†, Hans Stokking‡ and Nicholas Race†
*The University of Northampton, Northampton, UK
mu.mu@northampton.ac.uk
†School of Computing and Communications, Lancaster University, Lancaster, UK
{l.fawcett1, j.bird1, j.jellicoe, s.simpson, n.race}@lancaster.ac.uk
‡TNO, The Hague, The Netherlands
hans.stokking@tno.nl

***Abstract*—The continuing growth in the mobile phone arena, particularly in terms of device capabilities and ownership is having a transformational impact on media consumption. It is now possible to consider orchestrated multi-stream experiences delivered across many devices, rather than the playback of content from a single device. However, there are significant challenges in realising such a vision, particularly around the management of synchronicity between associated media streams. This is compounded by the heterogeneous nature of user devices, the networks upon which they operate, and the perceptions of users. This paper describes IMSync, an open inter-stream synchronisation framework that is QoE-aware. IMSync adopts efficient monitoring and control mechanisms, alongside a QoE perception model that has been derived from a series of subjective user experiments. Based on an observation of lag, IMSync is able to use this model of impact to determine an appropriate strategy to catch-up with playback whilst minimising the potential detrimental impacts on a users QoE. The impact model adopts a balanced approach: trading off the potential impact on QoE of initiating a re-synchronisation process compared with retaining the current levels of non-synchronicity, in order to maintain high levels of QoE. A series of experiments demonstrate the potential of the framework as a basis for enabling new, immersive media experiences.**

I. Introduction

Over the last five years we have witnessed an increasing trend towards coordinated media experiences. This has been driven by growth in mobile phone and tablet ownership, leading to the development of applications that provide second screen experiences [10], [4], designed to act as a companion to content being viewed on a primary screen. Mobile devices are, themselves, also used to deliver an increasing amount of streamed media content, with research highlighting that 40 %–65 % of tablet devices are used to stream movie and TV programmes at least once a day [8]. People are spending far more time watching video content on their mobile devices; in the U.S. this is estimated to be 59 % more than compared with 5 years ago [24]. The role of mobile devices as 'second screens' is also evolving, with early examples focusing around the provision of supplementary companion information in non-real time, to more recent examples demonstrating the potential of orchestrated media experiences. Examples include semantic video applications that adapt a single-screen application into a multi-screen environment based on either the author or user preferences [31] and systems that offer multi-screen orchestration linking television programmes with a "social sense", making use of QR codes on the TV screen and the camera of second screens to connect these experiences [13]. A number of projects are also looking to widen the experience beyond that of a single television image. Microsoft's IllumiRoom project augments the area around a television using projection, with visualizations designed to enhance gaming experiences [14] and a similar concept has been demonstrated by the BBC who produced a short film to demonstrate the potential of their Surround Video technology within a domestic, living-room environment [36]. The potential psychological impact of these additional screens has also been studied, with investigations into attention split, cognitive load and perceived comfort in order to determine an appropriate number of screens that could be viewed simultaneously [37], [3]. At the forefront of spatial audio research, transaural audio, ambisonics, and wave field synthesis have been actively researched to enhance audience experience using specialized equipment and proprietary designs [32].

This paper addresses the underlying challenges associated with orchestrating new and immersive media experiences across multiple end-user devices. Its specific contributions are based around the design and implementation of an open synchronisation framework that uses modern web technologies together with an adaptive synchronisation model in order to maintain high levels of QoE. Notably, the model has been derived from a detailed analysis of the relevant human factors, and provides a balanced approach to resolving issues with the synchronicity of content. The paper is structured as follows. Section II provides a context for our work, highlighting a use case scenario that we use to establish our design objectives. We also discuss related work closely associated with media synchronization. In Section III, we introduce the synchronization framework and provide a detailed description of the framework components, the devices types and their operation. Section IV describes a series of experiments that were used to measure user perceptions of re-synchronization (involving a potential change in playback speed) and the non-synchronicity of multiple streams played in a shared location. Section V provides a summary of the framework implementation and analysis of results. Finally, Section VI concludes the paper.

II. Use case and related work

Our use case scenario considers a shared environment, such as a public house or sports bar, with a group of individuals that are drawn to video highlights of a football championship. After browsing the video library on a tablet device one of the group initiates playback of their content in order to showcase the event. The experience, however, is underwhelming and limited by the capabilities of the individual device - notably, the sound emanating from the tablet seems flat. In order to improve the experience, three of the group use their own smartphones and join the application. These additional devices contribute multiple background soundtracks capturing sound from audiences and team benches, along with ambient light and vibrations, that help to recreate the immersive experience of the sport event. This scenario serves to highlight the objective of this work: creating vibrant and immersive orchestrated media experience across a series of heterogeneous user devices, connected via a range of networks (such as WiFi and LTE), through the formation of a "device cloud" (Figure 1).

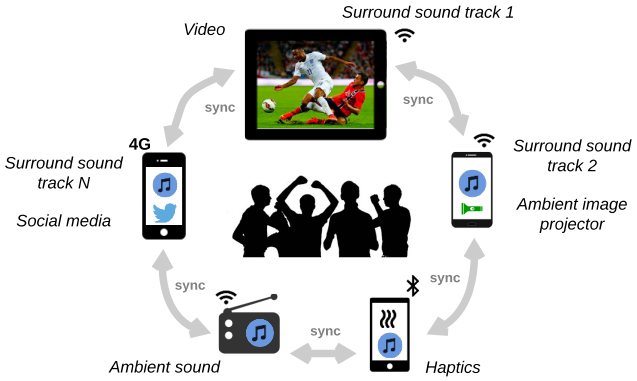


Fig. 1. Enabling immersive media experiences across multiple devices

Similar use cases have also been explored recently where mobile device clouds are constructed to offer advanced sound features such as multi-channel surround sound [17], directional sound [5] and noise cancellation [11]. More such examples are also seen in the field of Internet of Things (IoT) [7]. The main challenge of ensuring the quality of user experience in immersive and interactive multi-device applications is the real-time measurement, QoE evaluation, and control of the synchronicity between media objects in an ensemble of user devices over heterogeneous networks. Even a small degree of media non-synchronicity can be detrimental to the user experience. Many external and internal factors, such as clock drift and intermittent CPU overload at user devices, or explicit user interactions (such as pause and skip) will often cause linked media objects to fall out of sync. It is therefore essential to determine the optimal re-synchronization strategy that has minimal impact on the user experience.

Research on the topic of media synchronization is conventionally categorized into *intra-stream synchronization*, *inter-stream synchronization* and *inter-destination synchronization* (IDMS). Intra-stream synchronization addresses the fidelity of media playback with respect to temporal relationships between adjacent media units (MUs) within the same stream. Inter-stream synchronization refers to the preservation of temporal dependencies between the playout processes of correlated media streams [22], [23]; a common example of this is lip-sync [33], [6]. With the increasing demand of simultaneous media streaming to geographically distributed end systems, the level of synchronicity between media streams has become a deterministic factor in assuring both quality of user experience and fairness. Recently, Rainer et al. introduced a self-organizing control scheme with temporal distortion metrics based on the buffer level for peer synchronization in an IDMS session [29], [25]. Montagud et al. extensively reviewed 19 emerging media applications that require inter-destination synchronization from the level of "very high" (10 µs–10 ms) to "low" (500 ms–2000 ms) [22]. In a recent study on perceived synchronization of multi-sensory media, Yuan et al. concluded that users may tolerate haptic and air-flow media being one to three seconds behind corresponding video content [39]. A game-with-a-purpose (GWAP) approach was also taken to measure the lower asynchronism (non-synchronicity) as 400 ms for a social TV scenario [26]. The impact of rate changes on audio and video content are studied in [28]. However the work does not cover the cumulative impact of rate change over time, which is essential to balance the duration of non-synchronicity and the impact of rate change. Belda et al. demonstrate the synchronized playback of video and social media within one user interface using web technologies [2]. While we share some underlying development principles, our work focus on the user experience of closely coupled and continuous media, which is different from how human perceive social media in principle. Most existing studies focus on the perception of synchronicity in granular thresholds and do not systematically model the quantitative combined impact of non-synchronicity and re-synchronization in this emerging scenario, where multi-stream synchronization occurs at the same physical location. As a result, media objects cannot be orchestrated appropriately to ensure the best user experience.

Kim et al. studied the multi-device user experience of "commodity mobile devices" from an acoustic perspective. Rather than considering network impact, the work focuses on the specifications of the loudspeakers on user devices and their distance to listeners [17]. Examples of multi-device media applications have also emerged from industry, with the innovations in multi-room wireless audio such as Sonos being a particularly recent example. Most of the products in this area make use of customized chipsets or a proprietary mesh-network for synchronicity. Our work aims to provide an open, portable, and QoE-aware application-level synchronization framework, which can be enabled on different types of user devices with minimal configuration and user intervention. The key enabler of the framework is a novel user perception model purpose-built to capture inter-stream synchronicity with respect to the delivery of immersive media experiences across connected user devices. Although its QoE impact functions are tailored for audio-visual media streams, the framework can expand its support on multi-sensory media types based on the same synchronization mechanisms.

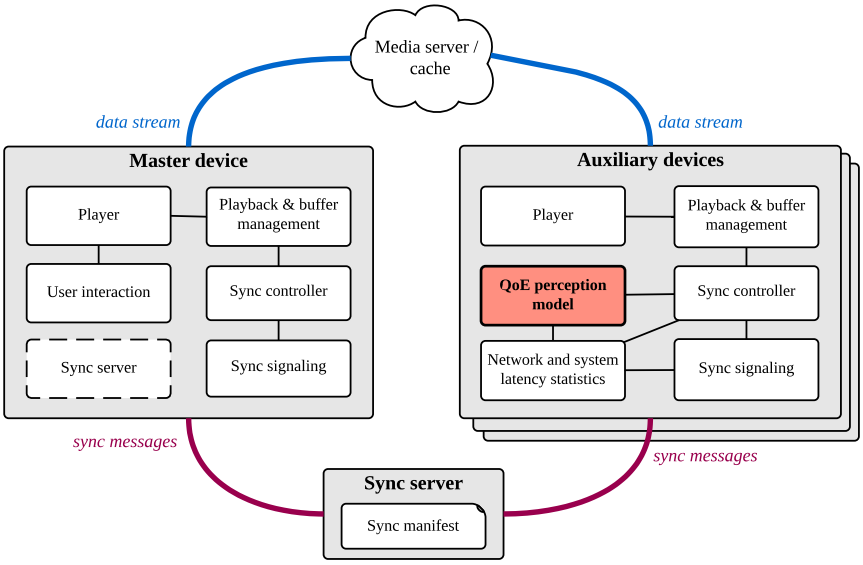


Fig. 2. Inter-stream synchronization framework and testing environment

III. Inter-stream media synchronization framework

The purpose of the inter-stream media synchronization (IMSync) framework is to enable the development, evaluation, and operation of a QoE perception model, allowing QoE-aware orchestration of media streams across multiple devices. The framework, shown in Figure 2, was designed to be portable, lightweight and operative with minimal requirements on user devices, allowing heterogeneous devices to participate the delivery of immersive experience easily and without requiring additional applications. IMSync is not dependent on any media distribution mechanisms such as MPEG DASH since it coordinates directly with media players for high-level media playback status and control functions.

The framework defines three reference device types: *master device*, *auxiliary device*, and *sync server* with each representing a specific role within a multi-device environment. The media playback session on the master device is the temporal reference point of all auxiliary devices. Auxiliary devices may join at any point to enhance the media experience while maintaining their synchronicity to the master. The sync server is a central point where measurements related to playhead position and player statistics are gathered and dispatched. The playback statistics of all devices are monitored and logged by the sync server using sync signalling. When the sync server detects a noticeable gap in playhead position (PP) between any auxiliary device and the master device, a sync message will be sent to the corresponding auxiliary device with additional timing information. Using such information, the QoE perception model at the auxiliary device then studies the degree of non-synchronicity and determines the optimal solution for re-synchronization based on impact to user experience and capacity of relevant devices. The role of a device is managed solely by the sync server and determined by the type of sync messages received by a user device. Therefore, any participating device can be elected as the master device during the course of the application. The election of master device follows three principles: 1) The first user device to start a media application becomes the master device; 2) If the master device fails to maintain its connection with the sync server for more than 10 seconds, the sync server will name an auxiliary device as the new master. When the replaced master device re-establishes the connection with the sync server, it will be treated as an auxiliary device by receiving sync messages for maintaining its synchronicity with the new master; 3) Whenever a user interacts with a device such as a skip operation to move the content forward, that device becomes the master and all other associated devices will use the latest user instruction as the reference and make adjustments accordingly. The sync server also maintains a sync manifest, through which the roles and media information of participating devices in the application are determined.

A. Master and auxiliary devices

Master and auxiliary devices share three functional modules: *playback and buffer management*, *sync signaling*, and *sync controller*. On the auxiliary devices, a *QoE perception model* is also active to measure the perceived experience of non-synchronicity from the master and to instruct the sync controller with optimal playout adjustments. Since such calculations are only conducted on auxiliary devices for their own synchronicity with the master, the framework allows the number of connected devices to scale up without additional workload on the master. As the sole reference point for synchronization, the master device does not require a QoE perception model to adjust its playout.

1) Playback and buffer management: The playback and buffer management module directly interacts with media applications on the same device. The module also intervenes in the activities of the player including playout rate adjustments and pre-emptive buffering according to decisions made by the sync controller. Modern web browsers provide detailed

runtime statistics and control interfaces of the native audio and video playback engine. Monitoring the buffer level also provides insights into buffering delay, which is one of the main causes of non-synchronicity between user devices.

2) Sync signaling: In order to measure the discrepancy between the playhead positions of media streams, sync messages carrying information such as the playback statistics from participating devices are exchanged periodically and efficiently by the sync signaling module via the sync server. This imposes two challenges in the framework design: 1) to define the means of time reference for sync messages, and 2) to mitigate the impact of network QoS and device capability on the performance of the framework in operation.

The most straightforward means of referencing time is to exploit the absolute time provided by the internal clocks on user devices, and use it to timestamp each event (e.g., "Device 1 is playing frame number 326 at (local) time 15:56:12.240"). To ensure the synchronicity of the clocks, the Network Time Protocol (NTP) is commonly used to adjust a device's clock using a time broadcast by an NTP server, with transmission time compensated for by a one-way-delay (OWD) metric. Clocks may also drift after NTP synchronization, hence, some time-critical applications require the clock synchronization process to take place periodically. The NTP-based clock synchronization requires additional ports and connections at user devices. It should also be recognized that clock synchronization requires, by its nature, long periods to maintain accurate timekeeping. Periods of hours or days and tens or hundreds of comparisons are required for the convergence to maintain local time to within a few tens of milliseconds [20]. Further, NTP clients are not available or enabled on all user devices.

The IMSync framework departs from the conventional designs with dependencies on the Network Time Protocol (NTP) and employs web technologies such as WebSockets to enable efficient full-duplex communication channels for the exchange of timing information directly and synchronously. All devices must establish a socket connection with the sync server, which uses a "heartbeat" mechanism to send out periodic (every second) "keep-alive" messages to the clients to see if they are still online. The clients subsequently respond with an *ACK* acknowledgement message with standard player data attached to it.

Using the statistics gathered through ACK messages, the sync server maintains the play-head-position information on all connected devices. It also has specially designed mechanisms to compensate for any signalling or playback delay. In a typical digital TV scenario, media servers process all media content and interleave time bases as part of the media transport streams in order to measure and control media synchronicity at the client side [9], [41]. Such an approach is not feasible for distributed user-generated content. IMSync capitalizes on its web-based interactive design and uses the playhead position (offset by sync manifest) reported by media players as reference. In practice, sync messages carrying the current playhead position of the master device p_{master} may take the time of Δt_0 to arrive at an auxiliary device, by which time the master has already a new playhead position of $p_{\text{master}} + \Delta t_0$. When a media player is instructed to adjust its playhead position (e.g., by seeking forward to a specific point in the media stream), the instruction will be delayed by Δt_0 in transmission.

Moreover, when the sync controller and the playback management function instruct the media player to re-synchronize by adjusting its playhead position, the media player must request a new data range from the content server and wait for the player buffer to be filled to a certain level before the playback can resume. This process often leads to an additional buffering delay of Δt_1 determined by the available bandwidth and the buffer size/buffering strategy at the end device. Without the help of a synchronization framework, the streams at auxiliary devices may lag behind the master for $\Delta t_0 + \Delta t_1$, which could be in the scale of hundreds of milliseconds to tens of seconds. To mitigate the impact of such a delay, the sync signaling module monitors the round trip time of the sync messages exchanged between user devices and the sync server and estimates the network delay Δt_0. This is similar to the design principle behind NTP but executed and maintained natively. Because the measurement is conducted on sync messages directly (rather than using separate NTP probing messages), the mechanism is more efficient for interactive media applications, having very little overhead. Together with the playback and buffer management module, the sync signaling function also maintains a statistical measurement of Δt_1, the delay between an order being sent from the playback management function and the media player completing the execution.

3) Sync controller: The sync controller monitors the level of playback non-synchronicity with the master device, and derives from that the timing and strategy for the re-synchronization process. We consider re-synchronization as a process of taking the master media stream as the reference and adjusting the auxiliary streams to a point where the non-synchronicity is imperceptible by the user. The sync controller currently employs two re-synchronization approaches, namely Adaptive Media Playout (AMP) and Predictive Playhead Projection (PPP), which are selectively enabled for the best results as perceived by humans. Table I defines the metrics and functions used by the sync controller. Given a current lag, the controller chooses to increase playback speed temporarily (AMP), and balances the choice of speed against the duration of the adjustment, such that the QoE impact is minimized. Only under extreme conditions does it perform a discrete jump (PPP) to perform the bulk of the work, with AMP for a final correction.

The impact of non-synchronicity is denoted as $I_{\text{non-sync}}$, a function of the non-synchronicity s measured by $p_{\text{master}} + \Delta t_0 - p_{\text{aux}}$. To reduce s by Δs, the AMP approach temporarily changes the original playback rate v of the auxiliary media stream to a new v'. The change of playback rate G is defined as $\frac{v'}{v}$. It would take the duration of $T = \frac{\Delta s}{|G-1|}$ for the auxiliary media stream to be perceptually in-sync with the master stream. Given Δs and v, T is inversely proportional to $|G-1|$. Therefore, a more radical change in playback rate (i.e., a higher value of $|G-1|$) could reduce the non-synchronicity quicker and therefore result in lower cumulative impact (i.e.,

Symbol	Description
s	Non-synchronicity between an auxiliary device and the master device.
S_L	The level of s when the non-synchronicity becomes perceivable by human.
S_H	The level of s when the non-synchronicity is too severe for the AMP approach to rectify without taking too much time or causing highly detrimental distortions.
Δs	The amount of non-synchronicity to alleviate. By default, it is set equal to s to completely remove non-synchronicity ($s - 0$). Alternatively, if the objective of a system is to reduce the non-synchronicity to a (statistically) imperceivable level, then Δs can be configured as $s - S_L$.
v	The original (native) playback rate.
v'	The adjusted playback rate during AMP.
G	The gain of the playback rate. $G = \frac{v'}{v}$.
T	The duration of the AMP re-synchronization process with G in effect. $T = \frac{\Delta s}{\lvert G-1 \rvert}$.
G_{limit}	The maximum playback gain that can be supported by the device and network.
T_{limit}	The maximum use of time for re-synchronization.
$I_{\text{non-sync}}$	The perceptual impact of non-synchronicity.
$C_{\text{non-sync}}$	The cumulative impact of non-synchronicity.
$I_{\text{re-sync}}$	The perceptual impact of re-sync process.
$C_{\text{re-sync}}$	The cumulative impact of re-sync process.
J	The overall impact of non-synchronicity and re-synchronization to the user.

TABLE I
SYNC METRICS AND FUNCTIONS

$C_{\text{non-sync}}$) to the user experience. However, the change made on the playback rate can be noticeable or even annoying to the user. The cumulative re-synchronization impact $C_{\text{re-sync}}$ is contributed by G and T. Given Δs, different combinations of G and T can be selected. Applying $G = 1.2$ for $T = 8$ s and $G = 1.8$ for $T = 2$ s would both help in reducing the non-synchronicity by 1.6 seconds though their QoE impact can be significantly different. Finding an optimal solution for a given Δs requires quantitative modelling of the impact from G and T which are believed to be non-linear in psychological scales. In practice, there might also be constraints on G. The execution of G by the user device is determined by the buffer occupancy and the network bandwidth. G_{limit} defines the upper limit of G that the device can possibly perform. The sync controller passes all relevant measurements to the *QoE perception model*, which returns a re-synchronization strategy that leads to minimal total impact between $C_{\text{non-sync}}$ and $C_{\text{re-sync}}$ (denoted as J).

While the AMP approach can be exploited to smoothly re-synchronize media streams, it might not be suitable when Δs reaches a certain threshold. Predictive playhead projection (PPP) is an approach that directly manipulates the playhead position of auxiliary streams (i.e., skipping). Although frame skipping is perceptually detrimental to the user experience, it is more efficient to rectify severe Δs. When a media player skips to a non-buffered point in the media stream, a further buffering delay Δt_1 is introduced. This will then cause the non-synchronicity of Δt_1 following the skipping. By monitoring the available bandwidth and buffer status, PPP estimates the Δt_1 (as $E(\Delta t_1)$) and pre-emptively appends this as an additional adjustment to the new playhead position. Any small residual (the difference between the observed and expected Δt_1 (i.e., $|O(\Delta t_1) - E(\Delta t_1)|$) will be subsequently corrected by AMP.

Figure 3 depicts the IMSync re-synchronization algorithm.

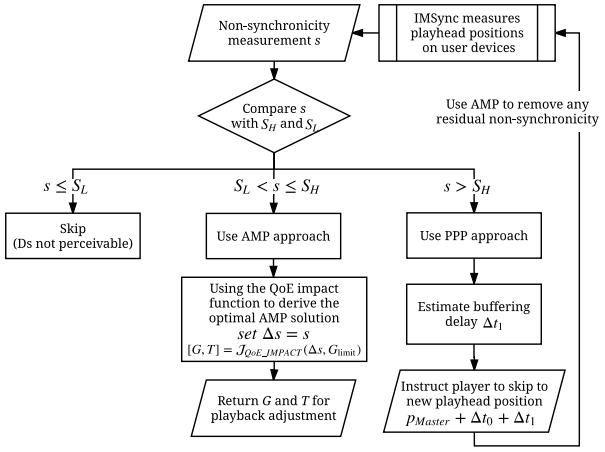


Fig. 3. IMSync re-synchronization flowchart

B. QoE perception model

The QoE perception model is ultimately the decision maker that assesses the perceivable non-synchronicity impact and assists auxiliary devices to adjust their playback status in order to be actively in-sync with the master (reference) device. Internally, the model incorporates multiple mathematical models to: 1) correlate the cumulative measurements of non-synchronicity and re-synchronization with subjective user opinion, and 2) derive the overall impact of the two. The output of the model is a re-synchronization solution that diminishes the non-synchronicity between any two media streams to a level not perceivable to human users with minimal overall impact. The subsequent execution of the solution is conducted by the sync controller. Section IV and Section V give the details of the modelling and evaluation of the perception model.

C. Sync server

The sync server bridges the connected user devices so that application configurations and sync timing information can be efficiently exchanged. An alternative design is to use a self-organizing overlay to carry the function of a sync server [29]. We recognize the distinctive benefits of each design and favour the presence of a sync server function because of its relatively minimal network- and application-level run-time overheads and software requirements at user clients. The sync server also receives users' participation preferences (e.g., media type and position) from devices, and uses a manifest describing the media in the form of URIs to map these preferences to separate components of the media to be dispersed across devices. Timing information is carried by the periodic messages (as part of the sync signaling) initiated by the sync server and forwarded to the sync controllers of all connected devices using the most efficient connection type possible. For instance, persistent full-duplex socket connections are often used to establish sync signaling channels with low overheads. The framework also allows the sync server to run on a user device so private device clouds can be established in a local environment.

IV. Modelling the Human Factor

The change in playback speed yields two perceived effects: re-synchronization (change of speed) and non-synchronicity. This section introduces experiments that serve to measure these two effects independently, and allow us to produce a combined model to capture the overall human perception.

A. Test environment

A test environment is designed to model the human factor and derive the perception model for the IMSync framework (Figure 4). The test environment is designed with network impact and device capability in mind, integrating controllable *network emulators*, and a bespoke *full-reference (FR) sync measurement* device. The network emulators allow us to evaluate the effectiveness of the framework in the context of best-effort delivery networks. The FR objective measurement device directly samples and comparatively measures the rendered outputs from media players in order to accurately evaluate the level of non-synchronicity between devices. The FR device is only used for building the perception model and evaluating the framework.

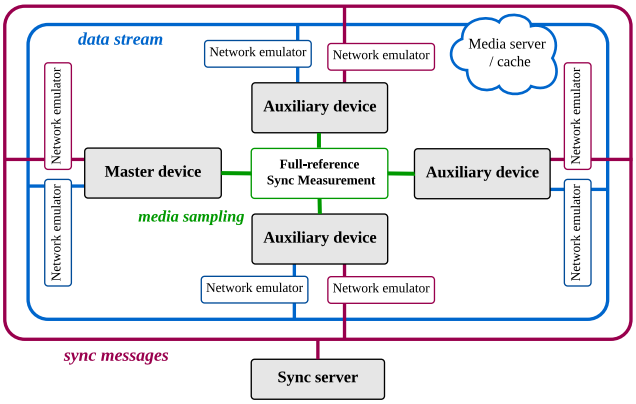


Fig. 4. IMSync test environment

1) Full-reference sync measurement: One of the challenges of designing and evaluating the QoE perception model is to accurately measure the absolute non-synchronicity between media streams under the influence of network latency, bandwidth constraint, and device capacity. We chose audio as the reference signal and use a full-reference (FR) measurement device to simultaneously capture the rendered audio outputs from two user devices using wired audio connections and then measure their non-synchronicity level. The FR measurement is only used to assist the model design and evaluation. The IMSync framework does not require such measurement to operate. We take the audio sampling in the rate of 10 000 samples per second from both sources and measure the cross-correlation between samples. Conventionally, cross-correlation is calculated based on the entire range of data from the sampling process, and the time offset from 0 that gives the peak of cross-correlation defines the inter-stream "lag" (i.e., non-synchronicity). However, the granularity of the results from such measurements is too coarse to capture the change of non-synchronicity influenced by the re-sychronization methods. Hence, we designed an *expandable moving slice* algorithm to better capture the intensity and variation of non-synchronicity (Figure 5).

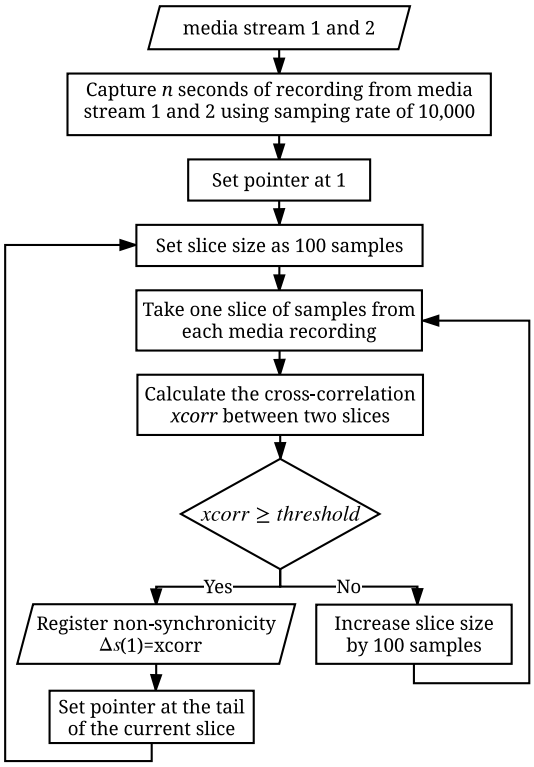


Fig. 5. Full-reference non-synchronicity measurement

The algorithm begins by taking a slice in the size of 100 samples from both audio sources to calculate cross-correlation. A small slice size gives finer measurement, but no matches between two slices can be found if the slice size is smaller than the non-synchronicity. With 10 000 samples per second and a slice size of 100 samples, each slice covers a duration of 10 ms. Therefore, an analysis based on 100-sample slices will detect non-synchronicity below 10 ms. If a correlation over a pre-defined threshold is found, a measurement is registered and the calculation will move on to the next slice. Otherwise, we increase the slice size by 100 samples to expand the search range, until a result is found.

2) Network emulator: We use a network emulator, an independent network device, to emulate network impairments such as latency, packet loss, packet corruption, and jitter in real networks. The emulator can be applied to any device in the experiment on its data stream or/and sync messages. It is also possible to apply an automation script so that the network status fluctuates over time during a experiment.

B. Perception of non-synchronicity

In practice, keeping media streams on multiple devices at the exact playhead position is very difficult. Even for two speakers that are directly connected to a playback device, the length of the audio cables and the location of listeners to each speaker can cause differences in reception. Fortunately, human ears and visual systems are able to tolerate such differences

to an extent. Existing studies on non-synchronicity focus on the measurements between tracks of a single media stream or the lag between media streams at different locations [22]. We focus on studying synchronously played multiple streams at a shared location, which reflects our use-case scenario. The modeling of human perception helps us determine the optimal timing and strategy of the re-synchronization process. The ultimate means to construct the impact model is through subjective user experiments.

Our non-sychronicity user experiments took place in an unused office, which was configured with a single display to play the video content accompanied by two audio sources (Figure 6). Both audio sources have nearly identical distance to the test participants, therefore any latency caused by the speed of sound is negligible.

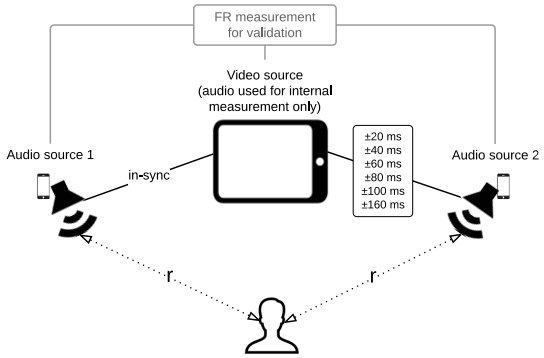
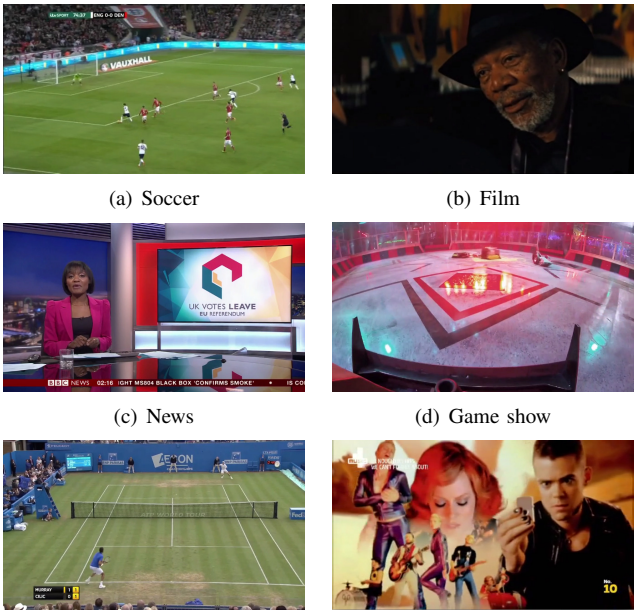


Fig. 6. Non-synchronicity test environment

We selected six representative 20-second video clips for the experiment (Figure 7). The *soccer* clip is taken from a *FIFA Worldcup 2014* match with audio commentary. The *news* clip shows a short news item on *BBC NEWS*. The *film* clip is a scene from the film *Now You See Me* with two characters engaged in a conversation. The *game show* clip is part of a round of the game show *Robot Wars* with multiple robots battling in an arena. The *tennis* clip is a 2016 game of tennis between Andy Murray and Marin Cilic. The *music* clip is the video for the track *I Don't Feel Like Dancin'* by the *Scissor Sisters*, and contains multiple people singing and dancing. We prepared test materials from all six clips with audio source 2 lagging behind audio source 1 (which is in perfect sync with the video source). Over the different tests, audio source 1 was lagged by: 20 ms, 40 ms, 60 ms, 80 ms, 100 ms and 160 ms.

Before the study started, each participant was given an explanation of the experiment and shown two demonstration videos, one normal and another where non-synchronicity had been introduced. Each participant, on their own, then watched half of all of the aforementioned test cases in a random order (18/36 videos). The participants rated each test case in the form of ACR-HR (absolute category rating with hidden reference) [1] using the ITU 5-point rating in the impairment scale (*5 - Imperceptible*; *4 - Perceptible but not annoying*; *3 - Slightly annoying*; *2 - Annoying*; *1 - Very annoying*). A total of 32 participants completed the study; 25 males and 7 females, with 2 aged between 18-30, 22 between 31-40 and 8 older than 40. Study participants were offered a £10 voucher for completing the experiment.



(a) Soccer (b) Film (c) News (d) Game show (e) Tennis (f) Music

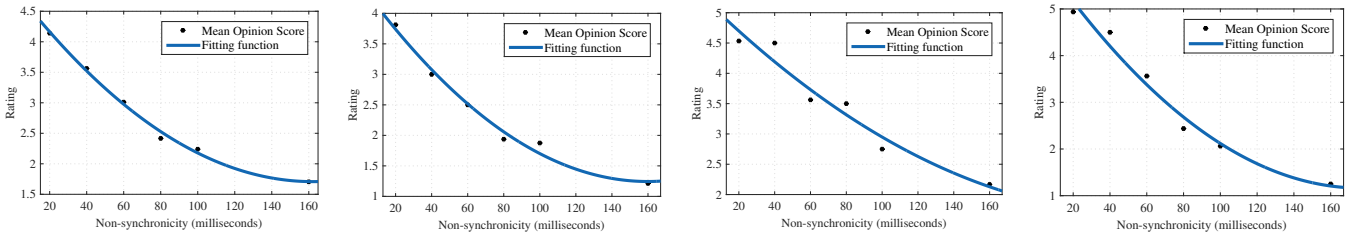
Fig. 7. Audio-visual clips for user experiments

The scores given by each participant were re-scaled to the range [1, 5]. Figures 8(b), 8(c) and 8(d) show the mean opinion score (MOS) for all six source videos. While they all exhibit a polynomial-like distribution, the non-synchronicity on the game show clip seems to be far more tolerable compared with the same test conditions applied to the film and news clips. The MOS of the experiments on the game show clip does not drop below 2 ("annoying") even when audio source 2 manifests a 160 ms lag, which is considered as "very annoying" by many participants on other clips. From short post-experiment user interviews, we learned that the echo-like effect caused by the non-synchronicity between multiple audio sources resonates with the experience in a large stadium or gaming arena. Because of this specific context, non-synchronicity on the game show clip is considered to be more acceptable. Some users also suggested that their attention was drawn to the actions of the robots in the game show rather than the sound effects. Accommodating the influence of content characteristics in modeling is an interesting research topic to be part of our future work, though the feasibility of the model could be affected by increased run-time complexity. Judging from the overall experimental results in Figure 8(a), a non-synchronicity of 20 ms is barely noticeable. When the level reaches 40 ms, it is perceivable by some users though not considered as annoying. From 60 ms, the non-synchronicity becomes annoying.

To generalize our findings, we derived the overall fitting function from user scores of all six video sources as below:

$$U_{\text{non-sync}}(s) = a_U\, s^2 - b_U\, s + c_U \tag{1}$$

With $a_U = 0.00012$, $b_U = -0.0394$, and $c_U = 0.0965$, the fitting has a goodness-of-fit of $R^2 = 0.9952$ and RMSE of



(a) Overall (b) News (c) Gameshow (d) Music

Fig. 8. Aggregated ratings on non-synchronicity

0.081 to the observed data.

The corresponding impact function (how much the user scores deviate from $5 - Imperceptible$) is given below with $a_I = -0.00012$, $b_I = 0.03941$, and $c_I = -0.09655$

$$I_{\text{non-sync}}(s) = 5 - U_{\text{non-sync}}(s) = a_I s^2 - b_I s + c_I \tag{2}$$

In practice, when s reaches a certain level s_0 that is perceivable by the user, re-synchronization mechanisms reduce non-synchronicity to a level s_1 that is unnoticeable by the user. We define the amount to catch up as $\Delta s = s_0 - s_1$:

Assuming the catch-up process will linearly reduce the non-synchronicity and it takes a certain amount of time T for the process to complete, the instantaneous non-synchronicity during the catch-up from time $t = 0$ to $t = T$ is $s(t) = s_0 - \frac{t}{T}\Delta s$. First, we expand (2) by substituting $s(t)$:

$$\begin{aligned} I_{\text{non-sync}}(t) &= a_I \left(s_0 - \frac{t}{T}\Delta s\right)^2 + b_I \left(s_0 - \frac{t}{T}\Delta s\right) + c_I \\ &= a_I s_0^2 - 2a_I s_0 \frac{\Delta s}{T} t + a_I \left(\frac{\Delta s}{T}\right)^2 t^2 + b_I s_0 - b_I \frac{\Delta s}{T} t + c_I \\ &= a_I \left(\frac{\Delta s}{T}\right)^2 t^2 - \left(2as_0 \frac{\Delta s}{T} + b_I \frac{\Delta s}{T}\right) t + a_I s_0^2 + b_I s_0 + c_I \end{aligned}$$

The non-synchronicity experienced by the user is then a cumulative effect of $I_{\text{non-sync}}(t)$ characterized by s_0, s_1, and T. We consider the accumulation linear in time scale and yield the cumulative impact factor $C_{\text{non-sync}}$.

$$\begin{aligned} &\int I_{\text{non-sync}}(t)\mathrm{d}t = a_I \left(\frac{\Delta s}{T}\right)^2 \frac{t^3}{3} \\ &- \left(2a_I s_0 \frac{\Delta s}{T} + b_I \frac{\Delta s}{T}\right) \frac{t^2}{2} + (a_I s_0^2 + b_I s_0 + c_I)t + K \end{aligned} \tag{3}$$

We now integrate $I_{\text{non-sync}}(t)$ with our specified limits:

$$C_{\text{non-sync}} = \int_0^T I_{\text{non-sync}}(t)\mathrm{d}t \tag{4}$$

$$= a_I (\frac{\Delta s}{T})^2 \frac{T^3}{3} - (2a_I s_0 \frac{\Delta s}{T} + b_I \frac{\Delta s}{T}) \frac{T^2}{2} + (a_I s_0^2 + b_I s_0 + c_I)T \tag{5}$$

$$= \frac{\left[2a_I \Delta s^3 - 6a_I s_0 \Delta s^2 - 3b_I \Delta s^2 + 6a_I s_0^2 \Delta s + 6b_I s_0 \Delta s + 6c_I \Delta s\right]}{6|G-1|} \tag{6}$$

With s_0 and s_1 defined, $C_{\text{non-sync}}$ is directly proportional to T which suggests that the quicker we bring media streams back in sync, the less perceivable non-sychronicity impact there will be to the user. However the processes of re-synchronization can also lead to new distortions to the media, which sometimes can be more detrimental than the non-synchronicity itself.

C. Perception of re-synchronization

AMP is a rate control mechanism that has been widely used to achieve smooth media playback or to harmonize buffer level via the dynamic adjustments of the media playout rate to mitigate the perceptual impact of network impairments. Li et al. defined multiple thresholds for the playout controller to start playback and dynamically adjust the playout rate based on the "buffer fullness" [18]. Learned from "informal tests", Kalman et al. concludes that the change of playback rate of up to 25 % is often unnoticeable and a change of up to 50 % is sometimes acceptable [16]. The threshold of 25 % has been adopted by a number of previous works as the guidance for the maximum playback rate variation [21], [34]. Li et al. uses a "simple linear function" to model the "slowdown cost" due to playing slower than the original playback rate [19]. A number of studies (e.g., [35]) also exploit a quadratic impact function initially proposed in [15], though the function does not seem to have been derived from subjective experimentation. It is then uncertain whether the values given by the impact function are in psychological scales for QoE optimization. Rainer et al. [27] and Li et al. [19] also recognized the influence of content characteristics (visual and acoustic features) on the perception of AMP. Rainer et al. evaluated the impact of playout variations on the QoE by adopting a crowdsourcing approach [30].

There are three main issues with such abstract rules and functions found in existing work. Firstly, they do not quantitatively capture the impact of AMP as perceived by users. Hence the re-sync process would not be able to optimize for the user experience. Secondly, the modeling on the impact of the duration of AMP, which is unlikely to be linear, is missing. A 30 % increment in playback rate for 1 second can be imperceptible, while it may simply take users a bit longer to start noticing the playback distortion or even find it annoying. Finally, there is a lack of systematic study on the joint perceptual impact of non-synchronicity and re-synchronization to optimize the balance between the two.

To fill the gap in this research field, we carried out further user experiments to quantitatively model the impact of AMP-

based re-synchronization by the change of playback rate $G = \frac{v'}{v}$ and the effective duration T of AMP. This effectively contours the operational range of AMP. We asked the same 32 participants that took part in the non-sychronicity experiments to review a second set of test videos, again generated using the six representative clips in Figure 7. Each test video had one test condition applied to it which is a combination of G and T. The selection of G is 1.1, 1.2, 1.4, 1.8 and 2.0 which maps to the playback rate v' of 33, 36, 42, 54 and 60 fps for our test videos with the same native rate v of 30 fps. The durations T of 1, 2, 4 and 8 seconds are selected. The test conditions are then applied to the reference videos. A total of 120 test videos were generated, and each participant watched 60 of them in a random order. The video playback starts at its native rate v; switches to v' at t_0; and finally goes back to v at t_1. We avoid the first 5 seconds of the clip, so $t_0 > 5$. The videos were assessed by the participants using the same rating system as used for the non-sychronicity experiments.

$$\begin{aligned} C_{\text{re-sync}}(G,T) =& p_{00} + p_{10}T + p_{01}G + p_{20}T^2 + p_{11}TG + \\ & p_{02}G^2 + p_{21}T^2G + p_{12}TG^2 + p_{03}G^3 \end{aligned} \tag{7}$$

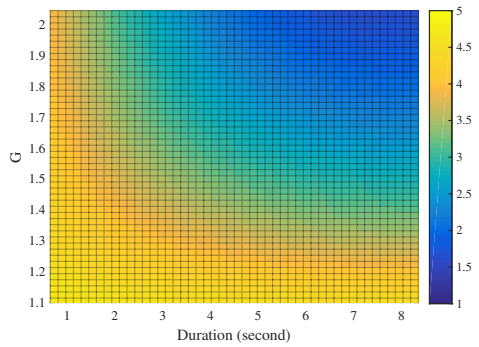
$$\begin{aligned} C_{\text{re-sync}}(G) =& p_{00} + p_{10}\frac{\Delta s}{G-1} + p_{01}\frac{\Delta s}{G-1} + p_{20}(\frac{\Delta s}{G-1})^2 \\ & + p_{11}\frac{\Delta s}{G-1}G + p_{02}G^2 + p_{21}(\frac{\Delta s}{G-1})^2G + p_{12}\frac{\Delta s}{G-1}G^2 \\ & + p_{03}G^3 \end{aligned} \tag{8}$$

The impact metric derived from user scores is modelled using a two-variable polynomial function (Equation 7). We use a second-order fitting option for the duration T and a third-order fitting option for the gain G to achieve the optimal balance between the performance and the complexity of the fitting function. We also investigated models with higher order coefficients. However they prove to be overly complex and generally cause over-fitting. Since $T = \frac{\Delta s}{G-1}$, the function can be simplified into a single-variable polynomial in G (Equation 8). The fitted coefficients are shown in Table II. Overall, function $C_{\text{re-sync}}(G)$ exhibits the goodness-of-fit of $R^2 = 0.988$ and $RMSE = 0.1294$. The fitting process is also carried out on test results of three clips separately which exhibit very similar measures of the goodness-of-fit.

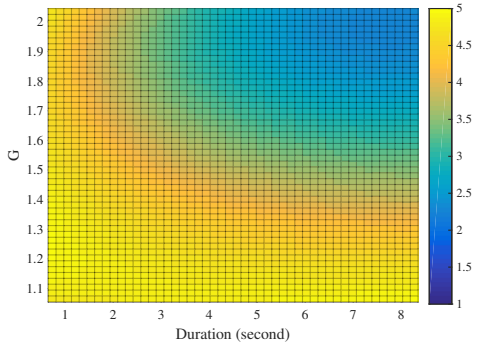
TABLE II
FITTED VALUES OF COEFFICIENTS

Coefficient	Fitted value	Coefficient	Fitted value
p_{00}	-2.781	p_{02}	-2.282
p_{10}	-1.073	p_{21}	-0.0442
p_{01}	4.838	p_{12}	-0.183
p_{20}	0.04283	p_{03}	0.364
p_{11}	1.251		

We also plotted the colormap to demonstrate the user opinion scores of AMP-based re-synchronization with respect to the combinations of G and T (Figure 9(a)). Note that both the intensity and the duration of the playback rate adjustment have a non-linear impact to the perception of re-synchronization. Overall, when G is below 1.2, users are unlikely to notice any anomaly even when the duration of it is as high as 8 seconds. In fact, the combination of 1.2 gain and 8 seconds duration results in 48 additional frames being played for a 30 fps content, allowing any auxiliary stream to catch up by 1.6 seconds of playback time. Using a higher G such as 1.8 could also yield the same results, though its impact starts to become annoying when the duration T exceeds 2 seconds. The re-synchronization impact on the game show clip is shown in Figure 9(b) as a comparison to the figure based on all experimental results. We learn that content characteristics do influence the perceptual impact of AMP re-synchronization. Temporal change of playback rate is less noticeable on the game show than on other clips. Users find a 2-second long doubling of playback rate "perceivable but not annoying". The user interviews suggest that the high motion and complexity of some test scenes can lead to a "masking effect", which affects the perception of the playback rate change.



(a) Overall

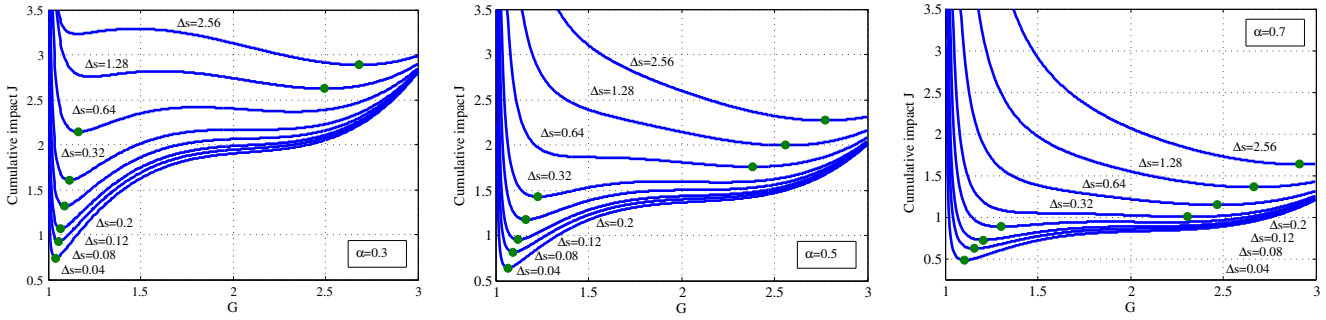


(b) Game show

Fig. 9. Colormap of user scores

D. Balancing the perceptual impacts

The modelling of the non-synchronicity impact $C_{\text{non-sync}}$ (Equation 6) and the re-synchronization impact $C_{\text{re-sync}}$ (Equation 7) enables us to identify the optimal solutions to adjust playback rate with the minimal overall impact J to user experience. We normalize and rescale both impact functions into $[0, 4]$ before combining them using the weighted-sum method for the global model combination (Equation 9). The weight coefficient α defines the balance between non-synchronicity impact and resynchronization impact when searching for the optimal solution using function J. The IMSync framework is flexible in tuning the AMP solution for applications/users that are more affected by non-synchronicity (with $\alpha > 0.5$) or more



(a) $\alpha = 0.3$ (b) $\alpha = 0.5$ (c) $\alpha = 0.7$

Fig. 10. Impact curve J configured using different values of weight coefficient

susceptible to the change of playback rate (with $\alpha < 0.5$). Given Δs, G_{limit}, and α, J is a function of G.

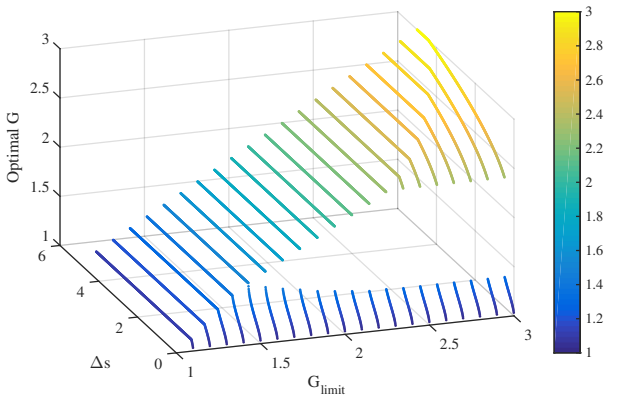
$$J = \alpha\, C'_{\text{non-sync}} + (1-\alpha)\, C'_{\text{re-sync}}, with\ 0 \le \alpha \le 1 \qquad (9)$$

Figure 10 shows the overall impact functions of AMP re-synchronization for different levels of non-synchronicity and device/network capabilities G_{limit}. Figure 10(b) represents the case when $C_{\text{non-sync}}$ and $C_{\text{re-sync}}$ are valued equally ($\alpha = 0.5$). The figure clearly manifests the joint impact of $C_{\text{non-sync}}$ and $C_{\text{re-sync}}$. When the non-synchronicity is relatively low (e.g., below 0.2 seconds), the best solution with minimum total cumulative perceptual impact can be found using a small playback gain G allowing a mild Δs to be rectified without causing high re-synchronization distortion to the application. However, when there is a high degree of Δs (e.g., above 0.6 second), $C_{\text{non-sync}}$ may accumulate a large impact over time. In this case, a more intensive adjustment to the playback rate is required to greatly reduce the non-synchronicity quickly with a small cost in re-synchronization distortion.

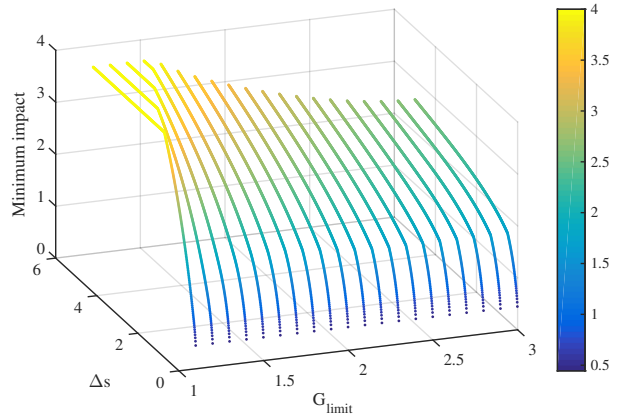
The weight coefficient α has great influence on the impact function J as well as the optimal configurations for the AMP re-synchronization process. As depicted in Figure 10(a), with more weight on the re-synchronization impact ($\alpha = 0.3$), the framework favours mild playback gain G until the non-synchronicity to catch up reaches the level of 1.28 seconds (compared with 0.32 seconds when $\alpha = 0.5$). For applications that are more prone to the level of non-synchronicity than the change of playback rate, α can be set above 0.5 to trigger the framework to use more radical approach. Figure 10(c) gives an example of α being set to 0.7 where the framework favours higher G to mitigate non-synchronicity.

In order to automate the process to derive an ideal AMP solution to balance the two impacts for a given Δs and G_{limit}, the sync controller dynamically calculates the value of G that minimizes Equation 9. The mathematical approaches to search for the minimal value on our impact function are not limited by the capabilities of the playback device. In production environments, the optimal Gs can be pre-computed based on intervals of Δs and G_{limit}. This would greatly improve the run-time efficiency of the synchronization process.

The optimal Gs on the impact curve J for different Δs and a G_{limit} are marked in Figure 10. We also take samples of Δs in the range of (0,3] and G_{limit} in the range of (1,3] to study the performance of the framework when the non-synchronicity and device/network limit varies. Figure 11(a) gives the optimal G while their corresponding total impact is shown in Figure 11(b). The visible leap around $\Delta s = 0.5$ when $G_{\text{limit}} > 2$ in Figure 11(a) reflects the shift of minimal impact point in Figure 10. When $G_{\text{limit}} < 2$, IMSync favours a lower G which leads to higher impact. Figure 11(b) gives an overview of the effective range of the AMP re-synchronization. In general, AMP is most suitable for non-synchronicity of low degree when the overall impact is below 2 (at which point



(a) Optimal G



(b) Impact of the optimal G

Fig. 11. Optimal G for given Δs and G_{limit}

the users find it "perceptible" or "slightly annoying"). This is also determined by the user device and the network. Re-synchronization can be less detrimental to user experience on devices connected via broadband networks. Figure 11(b) also suggests the points when the AMP-based approach is comparable to the more straightforward PPP-based re-synchronization. For instance, when $\Delta s = 2$ and $G_{\text{limit}} = 1.5$, skipping may be preferable, compared with a 4-second long annoying AMP.

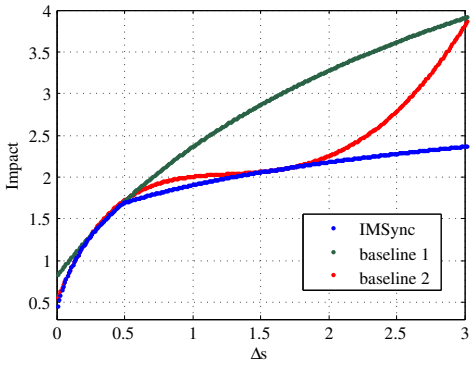


Fig. 12. Comparison of model outcomes with different Δs

We also compare our QoE-aware adaptive synchronization model with two baseline models. *Baseline1* uses a fixed playback gain G of 1.25 while *Baseline2* uses a fixed catchup duration T of 1 second for AMP despite the degree of non-synchronicity. Baseline1 and Baseline2 are the two typical models adopted in related work [21], [34]. Figure 12 compares the overall impact J of the synchronization process instructed by different models when the non-synchronicity level ranges from 0.01 to 3 seconds. IMSync's QoE model outperforms the two baseline models, leading to the smallest overall impact. Impact led by Baseline1 becomes increasingly higher than other models due to the fact that a fixed low playback gain leads to a long and annoying catchup period when the initial non-synchronicity becomes relatively high (i.e., greater than 0.5 second). There is also a sudden increase of impact on Baseline2 when the non-synchronicity goes beyond 2 seconds. This is caused by a detrimental high playback gain while the catchup period is fixed at 1 second. The IMSync model balances impact of catchup period and playback gain for the most optimal solutions.

V. Implementation and Experiments

The IMSync framework has been implemented using open web technologies such as JavaScript. Any user device that supports Javascript can participate in the delivery of immersive media without additional plug-ins. A customized Node.js server operates as the sync server handling device discovery and sync signalling as specified in the framework design.

A. Implementation of sync messaging

1) Heartbeat mechanism: Information about each client's player is piggybacked onto the heartbeat acknowledgement back to the sync server, shown in JSON Listing 1 below. This is mainly data referring to the media currently being played, directly from the HTML5 player (e.g. currentTime, ended, muted) but also contains some framework state (e.g., use_framework) to propagate settings from the master to all clients. Once received at the server, it is timestamped and stored.

Listing 1. Heartbeat sent from client

```
"frame":630,
"buffered":{},
"currentTime":25.230594,
"ended":false,
"muted":false,
"networkState":1,
"paused":false,
"playbackRate":1,
"played":{},
"readyState":4,
"seekable":{},
"duration":102.656,
"use_QoE_model":true
```

The heartbeat messages are also used to calculate the current network delay between all of the clients and the server, using a simple timestamp on sending and receiving. This round trip time data is stored alongside the player data for each client. The player data, framework state and network measurements data then serve as a central reference to be used when creating or maintaining synchronicity.

2) Sync message: As an up-to-date record of the player data is maintained; when the sync server identifies a detrimental degree of non-synchronicity from a user client, it will internally register such a sync request and log its socket connection ID shown in JSON Listing 2.

Listing 2. Sync log

```
target:iydGdPh-uNlDKcpRLbkU,
action:sync
```

The sync server can then collate all of the required data (i.e., master player data, master network measurements, and auxiliary device network measurements) necessary for the auxiliary device to make the precise QoE calculations for resynchronisation as shown in JSON Listing 3.

Listing 3. Sync message sent to client

```
"action":"sync",
"master_pd":{
        "frame":1230,
        "buffered":{},
        "currentTime":49.217563,
        "ended":false,
        "muted":false,
        "networkState":1,
        "paused":false,
        "playbackRate":1,
        "played":{},
        "readyState":4,
        "seekable":{},
        "duration":102.656,
```

```
        "use_QoE_model":true
},
"master_rtt":{
        "last_hb":1473024748184,
        "rtt":112,
        "rtt_sum":6853,
        "rtt_count":42
},
"server_time":1473024748723,
"aux_rtt":{
        "last_hb":1473024748183,
        "rtt":112,
        "rtt_sum":1405,
        "rtt_count":8
}
```

3) Sync calculation: Once the client receives the sync action, an arrival timestamp is added and an immediate decision is made based on the difference between the master and auxiliary players' current playback positions. If this difference is greater than a defined threshold (e.g., 3 seconds) the PPP approach is used, otherwise only AMP is used.

The true current playback time of the master p'_{master} can be calculated by accounting for the time taken for the master's playhead position to traverse the network (half the two RTTs), and for its time spent at the sync server:

$$\Delta t_0 = \frac{R_{\text{aux}} + R_{\text{master}}}{2} + (n_{\text{sync}} - H_{\text{master}}) \tag{10}$$

$$p'_{\text{master}} = p_{\text{master}} + \Delta t_0 \tag{11}$$

n_{sync} and H_{master} are server_time and master_rtt:last_hb from the sync message. Other symbols are defined in Table III.

Depending on the previous decision whether to begin with a PPP or not, there are two possible calculations. Using only an AMP, the correction Δs is:

$$\Delta s = p'_{\text{master}} - p_{\text{aux}} \tag{12}$$

Using PPP followed by AMP needs to take into consideration Δt_1, the latency between sending the seek command to the player and the player actually playing (due to buffering). The player jumps forward to p_{master} (an increase of $p_{\text{master}} - p_{\text{aux}}$), and then AMP is applied to correct by the remainder.

In the demonstrated scenario which followed the PPP-then-AMP approach, a total correction of 0.668 seconds was calculated.Using the QoE model, this produces an optimal $T = 0.481\,36$ s and $G = 2.3878$. Table III lists the timing calculation results.

TABLE III
TIMING CALCULATION

p_{master}	master_pd:currentTime	49.2175 s
p'_{master}	True master playback time (after PPP)	49.8856 s
p_{aux}	player.currentTime (after PPP)	49.2202 s
Δt_1	Player latency	17.065 ms
R_{master}	master_rtt:rtt	112 ms
R_{aux}	aux_rtt:rtt	112 ms
Δs	Correction	0.668 s
T	Catchup period	0.481 36 s
G	Play rate adjustment	2.3878

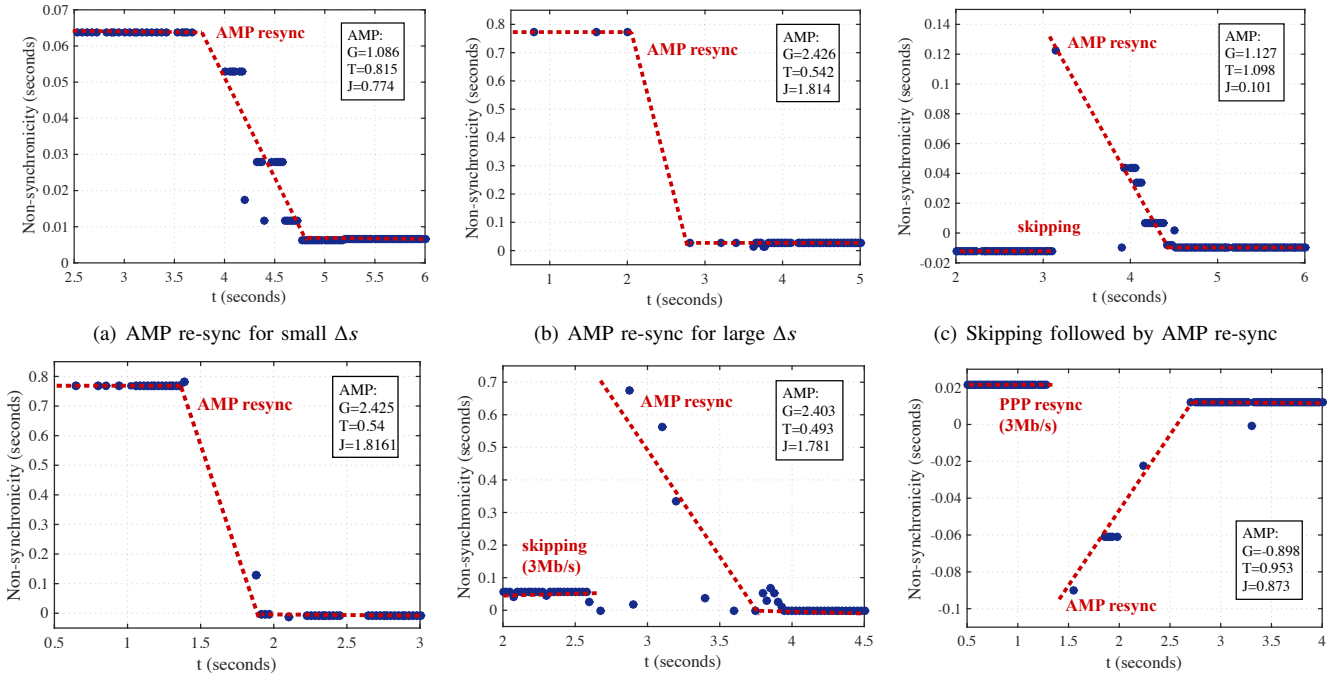
B. Performance evaluation

In order to evaluate the framework, we set up a testbed environment with multiple user devices, a sync server that hosts the JavaScript libraries, a media server which serves media content, a full-reference sync measurement device, and emulators for networks of different properties. Audio-visual content is distributed as native HTML5 content over HTTP. We also use an admin web interface to monitor sync messages exchanged between devices and their player status (such as playhead position, playback rate and buffer level). The interface provides real-time measurements of network statistics on all devices and control interfaces for experimentation. The framework is configured to weigh the impact of non-synchronicity and re-synchronization equally ($\alpha = 0.5$).

We used the full-reference sync measurement device to capture the operations of the framework (Figure 13). Every marker presents a point of valid measurement. A positive value of non-synchronicity denotes the auxiliary stream being behind the master stream. We also use dash lines to plot the trends of the measurements. Due to the nature of the sampling method, the measurement tool will yield fewer results when the non-synchronicity is high, though the accuracy of measurements is not affected. The synchronicity during the change of playback rate and skipping is very difficult to capture. The results given during these transition periods are, however, still valuable in understanding the operations of the IMSync framework. Based on the user study results shown in Figure 8, we define the threshold of 30 ms (just below the display time of one video frame for a 30 fps video content) as the measure of whether a pair of media streams in the same location are "in-sync".

The first group of tests are performed with no network emulation. The available bandwidth is 100 Mbit/s and the round trip time between user devices and the sync server is less than 10 ms. This resembles the scenario when an application is running locally with all devices joining a local network. We start the playback of a media stream on all devices with the synchronization framework turned off and control the playhead positions of the auxiliary stream to be around 65 ms behind the master stream. We then activate the framework which immediately detects the non-synchronicity on the auxiliary device and uses the AMP method to re-synchronize the media streams by slightly increasing the playback rate ($G = 1.086$) for 0.815 seconds. With the impact J of just 0.774, users are very unlikely to notice any distortion from the time the framework is enabled. Figure 13(a) suggests that the media streams are around 5 ms apart after the process.

In the second test, we greatly increase the initial non-synchronicity to around 800 ms. Using the impact function, the framework instructs a short surge ($T = 0.542$) of high playback gain ($G = 2.426$) which brings the non-synchronicity back to around 10 ms (Figure 13(b)). The impact of operation is increased to $J = 1.814$, which suggests that statistically users will experience a very short but not annoying distortion. The third test studies how the IMSync framework reacts to media events. We start the test with all streams in-sync ($s \cong 18$ ms),



(a) AMP re-sync for small Δs

(b) AMP re-sync for large Δs

(c) Skipping followed by AMP re-sync

(d) AMP re-sync with added RTT

(e) Skipping with bandwidth limit

(f) PPP followed by AMP re-sync

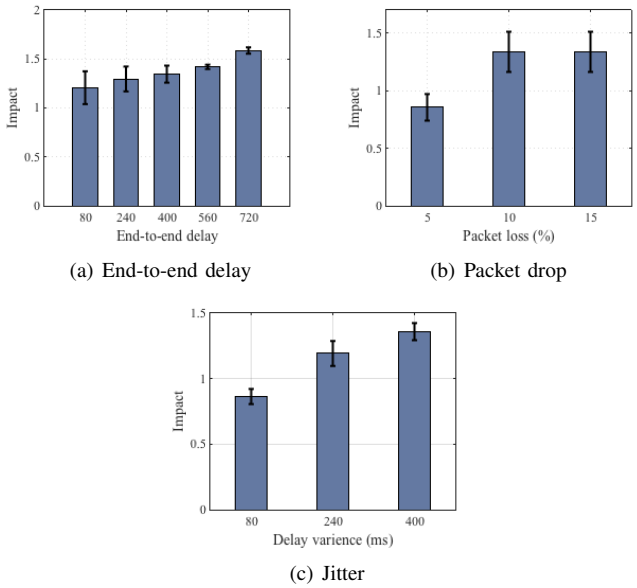
Fig. 13. Experimental results

then commit a skip operation to a point around 15 minutes further into the video on the master stream (which is a common user operation). Because the non-synchronicity (around 15 minutes) is beyond the range of AMP, IMSync instructs the auxiliary stream to skip (Figure 13(c)) whilst factoring in the signaling delay. Due to the small buffering delay, the auxiliary stream becomes over 120 ms behind the master stream. This is immediately followed by AMP which closes up the gap with minimal impact in one second (Figure 13(c)). The measurements in negative values imply that an auxiliary device is ahead of the master device.

The second group of tests evaluate the framework's performance when network delays and bandwidth affect sync signaling and media buffering. We apply a 100 ms round-trip delay to the link of the auxiliary device and enable the AMP on 800 ms of non-synchronicity. The results demonstrate that the framework detects the additional network latency and adjusts the playback rate change to close up the lag between media streams. We then limit the available bandwidth of the auxiliary device to 3 Mbit/s and repeat the skipping test. As a result, the limit on the buffering throughput increases the non-synchronicity after the skipping tenfold to around 700 ms. It then takes AMP to apply a high playback gain of $G = 2.403$ with impact of $J = 1.781$ to adjust the media stream (Figure 13(e)). The PPP approach is brought in to estimate the buffer delay based on 1) the moving average of previous skip events, and 2) out-of-band bandwidth monitoring using probing packets. The estimated buffer delay is then employed to skip the auxiliary stream to a projected playhead position further into the future so that the playback deficit can be greatly reduced when the skip event completes. Figure 13(f) gives an example of how the bandwidth/buffering delay measurement could improve the synchronization. With the same set-up used for Figure 13(e), the PPP-based approach takes the measurement of around 500 ms of buffering delay based on statistics from previous events, and reduces the non-synchronicity after the skip to just under 100 ms, which has much less impact ($J = 0.873$) to catch up further by AMP.

We also investigate the robustness of IMSync in dealing with a range of different network impairments such as delay, jitter and packet loss. We introduce network impairments on the link of the auxiliary device, and randomly change the playhead position on the master device. The test then measures the overall impact J of the synchronization process instructed by IMSync. Each test is repeated 20 times. Figure 14(a) compares the mean and 95 % confidence interval of the overall impact when the round trip delay of the network is 80 ms, 240 ms, 400 ms, 560 ms and 720 ms. Higher network delay results in a higher degree of initial non-synchronicity, and hence costs a higher impact to rectify. Assisted by its QoE model, IMSync adapts AMP strategies according to the runtime measurements and retains its overall performance.

We maintain the round-trip delay in the network as 80 ms, and repeat the experiment by randomly discarding packets with the drop rate of 5 %, 10 % and 15 %, which affects both the content distribution and the sync signalling. As Figure 14(b) illustrates, there is a positive correlation between the drop rate and the overall synchronization impact with respect to the mean and the standard deviation measures. This is caused by the increased buffering time and the retransmission of sync messages. When the drop rate reaches 10 %, some sync messages may be retransmitted a few times before success which explains the high level of standard deviation of the overall impact. We also investigate the impact of jitter at



(a) End-to-end delay (b) Packet drop

(c) Jitter

Fig. 14. Overall perceptual impact influenced by network impairments

TABLE IV
SURROUND SOUND DEVICE CONFIGURATION

Position	Device	Bandwidth
Video	Macbook Pro	20Mbit/s
Center speaker	Android phone A	2Mbit/s
Left speaker	Android phone B	Unlimited
Right speaker	Android tablet	3Mbit/s
Rear left speaker	Raspberry Pi A	4Mbit/s
Rear right speaker	Raspberry Pi B	3Mbit/s
Low frequency speaker	Rasberry Pi C	6Mbit/s

TABLE V
SUBJECTIVE EVALUATION TEST CONFIGURATIONS

Test	Abbr	Description
A	1.0	Two devices: video and center audio
B	3.0	Four devices: all devices from A plus two devices for front left and front right audio
C	5.1	Seven devices: all devices from B plus two devices for rear left and right audio, and one additional device for low frequency effects
D	5.1x	Seven devices: all devices from C but without IMSync's QoE-aware synchronization capabilities

80 ms, 240 ms and 400 ms. The results suggest that large jitter does have more of an impact on the synchronisation process. IMSync is able to cope with such severe network impairments while keeping the overall impact relatively low (Figure 14(c)).

Overall, IMSync outperforms the two baseline models using its effective and QoE-aware synchronization model. In summary, the results from experiments demonstrate the effectiveness of the IMSync model in performing re-synchronization in different scales with minimal QoE impact. The QoE modelling and the adaptive algorithm are proven to be particularly beneficial when linked user devices such as smartphones and tablet computers are connected via best-effort wireless and mobile networks.

C. Subjective evaluation

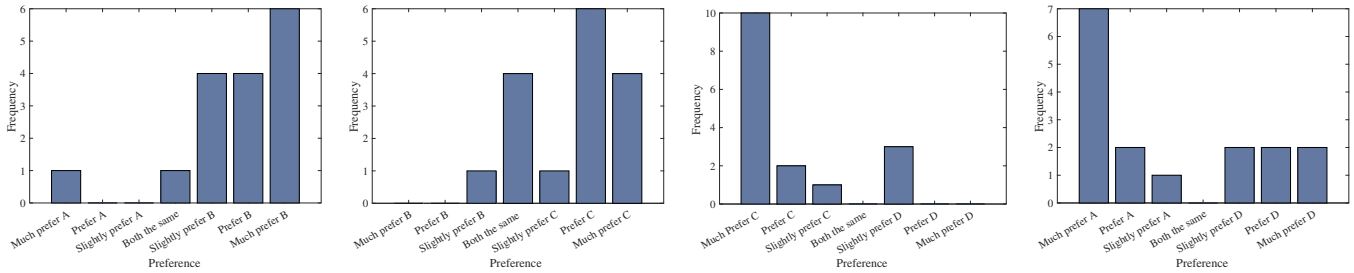
With a complete IMSync system, we wanted to test its effectiveness in a realistic environment using a range of typical end user devices. We therefore conducted a subjective evaluation to investigate, i) Would the coordinated delivery of associated media across multiple user devices greatly enhance the user experience?, ii) Whether the user experience is in any way correlated to the number of participating devices?, and iii) Does the QoE-aware synchronization capabilities of the IMSync framework improve the overall user experience in a multi device configuration?

The evaluation was established in an unused office and comprised of various devices that were connected via the framework, including Android phones, an Android tablet, a MacBook Pro, and Raspberry Pis with speakers attached. To individually modify the networking characteristics of each connected device we used the *netem* emulator. With netem operating between the devices and media source, we applied bandwidth restrictions as shown in Table IV.

Our evaluation entailed showing participants a 5.1 surround sound *Star Wars Rogue One* trailer. The audio channels were encoded into separate AAC files (plus a single video file) from the original lossless (FLAC) audio source. With the information from the IMSync manifest, devices were able to request individual elements of the trailer (e.g. the audio for the front right speaker). A total of 4 test conditions were generated (identified as A-D), 3 with different audio configurations and a final test case that used a simple sync/skip message with no IMSync adjustments. Details of the test configurations are shown in Table V.

A paired comparison was conducted between each subsequent test case by each participant using a comparison scale such as *Much prefer A, prefer A, slightly prefer A, Both the same, slightly prefer B, prefer B, much prefer B*. After this, each participant underwent a small interview, and were asked to describe their experience of viewing each test case. In the interview we asked each participant three questions 1) "Do you have any comments about what made any particular test case annoying to watch?" 2) "Given the scenario of watching a video on somebody's device; would you be willing to contribute your own device (e.g. mobile, tablet, wearable) to enhance the overall user experience?" 3) "Can you think of any other ways in which your day-to-day devices could create a more immersive experience when viewing content?". A total of 16 participants completed this study; 11 males, 4 females and one preferred not to say, with 7 aged between 18-30, 2 between 31-40 and 4 older than 40.

In the first test, when comparing 1.0 with 3.0, the vast majority (87.5%) said they preferred 3.0 over 1.0, with this scenario being our most obvious preference, see Figure 15(a). This clear preference can be attributed to the limited quality and volume of the mobile device speaker which in test case B was boosted through additional devices. This is supported by our participant responses to question 1, which included "*[Disliked 1.0] Only one sound source*" and "*Video A [(1.0)] did not have great sound quality.*". These negative aspects towards test case A could be addressed by using a device with a better quality speaker but this comparison does confirm that the user



(a) 1.0 (A) compared with 3.0 (B) (b) 3.0 (B) compared with 5.1 (C) (c) 5.1 (C) compared with 5.1x (D) (d) 1.0 (A) compared with 5.1x (D)

Fig. 15. Subjective paired comparison results

experience is enhanced using multiple coordinated devices over a single mobile device. Comparing 3.0 and 5.1; 68.75% of participants said they preferred 5.1, see Figure 15(b). The experience is once again enhanced with the addition of 3 more devices, creating a positive correlation of improvement. The preference is less significant than the 1.0 to 3.0 comparison, but the additional devices in this case were providing a low-frequency speaker and two speakers for the rears, which might be considered to be less significant than the front speakers in a surround sound configuration. One participant was not in favour of this configuration, stating that "*Video C [(5.1)] had too much going on*" suggesting that a minority of people may be uncomfortable with this setup due to sensory overload.

When comparing the 5.1 test cases where IMSync QoE-aware synchronisation is either enabled or disabled, 81.25% of the participants said that they preferred the 5.1 configuration with QoE-aware synchronisation capabilities enabled, see Figure 15(c). This would indicate that QoE-aware capabilities do improve the overall user experience and are required for a successful synchronisation framework. One participant affirms this view during the interview, stating "*Lip sync with video D was particularly problematic - and thus particularly annoying*" highlighting that without sychronisation the levels of non-synchronicity are clearly perceivable, particularly during periods of dialogue within the trailer.

A final comparison was made between 1.0 and 5.1x (without QoE-aware synchronisation capabilities) to query whether users would rather have a single device experience over a multi device experience without complete synchronicity, see Figure 15(d). Although 62% of participants preferred a single device, there remains a significant minority that would favour multiple devices, despite them being slightly out of sync. We consider this comparison therefore somewhat less conclusive, and will ultimately depend on an individuals personal preference of sacrificing quality over synchronicity.

When participants were asked during the interviews whether they would be willing to contribute their own devices in order to receive an enhanced experience, the majority responded positively. However, there were concerns raised over the security associated with connecting to other people's devices and the potential implications for battery consumption. One participant said "*Yes, probably. Depends on the access method; would trust APIs in iOS more than a third-party magic app!*" while another suggested "*Yes, if I trust this person that I wouldn't get a virus from his device. Not if there is secret data on my device (like company data)*". Finally, the interviews gave an opportunity for participants to suggest additional devices that could contribute towards a more immersive media experience. A multi-view scenario and the use of Internet-of-Things devices were amongst some of the suggestions, "*Imagine if, watching Star Wars in the same room, one viewer could see things from the Imperial perspective and the other viewer could see the Rebel perspective...*", and having additional devices "*...integrated into a chair or something...*" to create a synchronised multi-sensory experience.

VI. Conclusions and Future Work

Orchestrating multiple media streams across heterogeneous user devices in order to deliver new, immersive media experiences is a very challenging task. The paper contributes to this topic with the design and implementation of an open inter-stream synchronization framework, IMSync. The framework is unique in providing optimized re-sychronization strategies that have minimal perceptual impact to the user using a comprehensive QoE perception model, while incorporating an efficient sync-signaling mechanism and functional modules to interact with media engines. We implement the framework using web technologies and evaluate its performance using a tailor-made testbed. Whilst IMSync is able to achieve absolute inter-stream synchronicity, its role extends further than this by providing a foundation for new media applications and user experiences by enabling the temporal attributes of associated media objects over multiple devices to be specified. IMSync also represents a crucial step forward in supporting novel spatial audio and video designs using non-specialized equipment such as smartphones and tablets that can be used across heterogeneous networks. The change of playback rate is executed by IMSync using standard APIs such as HTML5 audio and video controls to minimize complexity on user devices. We conducted a series of experiments to evaluate the performance of IMSync in delivering enhanced user experience using a number of synchronized user devices.

Future work will investigate further ways in which the impact of re-synchronization can be reduced, including techniques used in the professional audio industry that rely on real-time audio processing equipment to preserve the pitch of the signal (e.g. WSOLA [12], [38]). We currently use audio and video as the reference media to model the human perception of media synchronization. Humans have a very high sensitivity to audio asynchrony in the degree of tens of milliseconds,

therefore audio-visual content provides an ideal reference media to evaluate the performance of IMSync. Related work suggests that synchronization between sensorial effects and multimedia content is important to the user experience and the perception of synchronicity can be different across haptic, air, and olfaction [40]. We will also further investigate the impact of sensual overload observed on a small number of participants in our tests. IMSync's models and experimentation platform lay the groundwork for future work in the synchronized delivery of multi-sensory media.

REFERENCES

[1] Subjective video quality assessment methods for multimedia applications. *ITU-T Recommendation P.910*, 1996. ITU.

[2] J. Belda, M. Montagud, F. Boronat, M. Martinez, and J. Pastor. Wersync: A Web-based Platform for Distributed Media Synchronization and Social Interaction. *ACM TVX*, 2015.

[3] A. Brown, M. Evans, C. Jay, M. Glancy, R. Jones, and S. Harper. Hci over multiple screens. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 665–674. ACM, 2014.

[4] P. Cesar, D. C. Bulterman, and A. Jansen. Usages of the secondary screen in an interactive television environment: Control, enrich, share, and transfer television content. In *Changing television environments*, pages 168–177. Springer, 2008.

[5] J. Cheer, S. J. Elliott, Y. Kim, and J.-W. Choi. Practical implementation of personal audio in a mobile device. *Journal of the Audio Engineering Society*, 61(5):290–300, 2013.

[6] M. Chen. A low-latency lip-synchronized videoconferencing system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 465–471, New York, NY, USA, 2003. ACM.

[7] J. Chmielewski. Device-independent architecture for ubiquitous applications. *Personal and Ubiquitous Computing*, 18(2):481–488, 2014.

[8] Exacttarget. 2014 mobile behavior report - combining mobile device tracking and consumer survey data to build a powerful mobile strategy. 2014.

[9] M. Ferreira Moreno, R. Monteiro de Resende Costa, and L. F. Gomes Soares. Interleaved time bases in hypermedia synchronization. *MultiMedia, IEEE*, 22(4):68–78, 2015.

[10] D. Geerts, R. Leenheer, and D. De Grooff. In front of and behind the second screen: Viewer and producer perspectives on a companion app. In *Proceedings of the ACM International Conference on Interactive Experience of Television and Online Video (TVX2014)*, 2014.

[11] N. V. George and G. Panda. Advances in active noise control: A survey, with emphasis on recent nonlinear techniques. *Signal processing*, 93(2):363–377, 2013.

[12] S. Grofit and Y. Lavner. Time-scale modification of audio signals using enhanced wsola with management of transients. *IEEE transactions on audio, speech, and language processing*, 16(1):106–115, 2008.

[13] H. Hu, J. Huang, H. Zhao, Y. Wen, C. W. Chen, and T.-S. Chua. Social tv analytics: a novel paradigm to transform tv watching experience. In *Proceedings of the 5th ACM Multimedia Systems Conference*, pages 172–175. ACM, 2014.

[14] B. R. Jones, H. Benko, E. Ofek, and A. D. Wilson. Illumiroom: peripheral projected illusions for interactive experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 869–878. ACM, 2013.

[15] M. Kalman, E. Steinbach, and B. Girod. Rate-distortion optimized video streaming with adaptive playout. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages III–189. IEEE, 2002.

[16] M. Kalman, E. Steinbach, and B. Girod. Adaptive media playout for low-delay video streaming over error-prone channels. 14(6):841–851, 2004.

[17] H. Kim, S. Lee, J.-W. Choi, H. Bae, J. Lee, J. Song, and I. Shin. Mobile maestro: enabling immersive multi-speaker audio applications on commodity mobile devices. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 277–288. ACM, 2014.

[18] M. Li, T.-W. Lin, and S.-H. Cheng. Arrival process-controlled adaptive media playout with multiple thresholds for video streaming. *Multimedia systems*, 18(5):391–407, 2012.

[19] Y. Li, A. Markopoulou, J. Apostolopoulos, and N. Bambos. Content-aware playout and packet scheduling for video streaming over wireless links. 10(5):885–895, 2008.

[20] D. L. Mills. Internet time synchronization: the network time protocol. *Communications, IEEE Transactions on*, 39(10):1482–1493, 1991.

[21] M. Montagud, F. Boronat, and H. Stokking. Design and simulation of a distributed control scheme for inter-destination media synchronization. In *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, pages 937–944, 2013.

[22] M. Montagud, F. Boronat, H. Stokking, and R. van Brandenburg. Inter-destination multimedia synchronization: schemes, use cases and standardization. *Multimedia Systems*, 2012.

[23] M. Mu, M. Broadbent, N. Hart, A. Farshad, N. Race, D. Hutchison, and Q. Ni. A Scalable User Fairness Model for Adaptive Video Streaming over Future Networks. *IEEE Journal on Selected Areas in Communications*, 34(8), Aug 2016.

[24] Nielsen. The u.s. digital consumer report. *http://www.nielsen.com/us/en/insights/reports/2014/the-us-digital-consumer-report.html*.

[25] B. Rainer, S. Petscharnig, and C. Timmerer. Merge and forward: self-organized inter-destination multimedia synchronization. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 77–80. ACM, 2015.

[26] B. Rainer, S. Petscharnig, C. Timmerer, and H. Hellwagner. Is one second enough? evaluating qoe for inter-destination multimedia synchronization using human computation and crowdsourcing. In *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on*, pages 1–6. IEEE, 2015.

[27] B. Rainer and C. Timmerer. Adaptive media playout for inter-destination media synchronization. In *Quality of Multimedia Experience (QoMEX), 2013 Fifth International Workshop on*, pages 44–45. IEEE, 2013.

[28] B. Rainer and C. Timmerer. A quality of experience model for adaptive media playout. In *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, pages 177–182. IEEE, 2014.

[29] B. Rainer and C. Timmerer. Self-Organized Inter-Destination Multimedia Synchronization for Adaptive Media Streaming. In *Proceedings of the ACM International Conference on Multimedia*, pages 327–336. ACM, 2014.

[30] B. Rainer and C. Timmerer. A subjective evaluation using crowdsourcing of adaptive media playout utilizing audio-visual content features. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–7. IEEE, 2014.

[31] M. Sarkis, C. Concolato, and J.-C. Dufourd. The virtual splitter: refactoring web applications for themultiscreen environment. In *Proceedings of the 2014 ACM symposium on Document engineering*, pages 139–142. ACM, 2014.

[32] L. Savioja, A. Ando, R. Duraiswami, E. A. Habets, and S. Spors. Introduction to the Issue on Spatial Audio. *Selected Topics in Signal Processing, IEEE Journal of*, 9(5):767–769, 2015.

[33] R. Steinmetz. Human perception of jitter and media synchronization. 14(1):61–72, 1996.

[34] Y.-F. Su, Y.-H. Yang, M.-T. Lu, and H.-H. Chen. Smooth control of adaptive media playout for video streaming. *Multimedia, IEEE Transactions on*, 11(7):1331–1339, 2009.

[35] E. Tan and C. T. Chou. A frame rate optimization framework for improving continuity in video streaming. *Multimedia, IEEE Transactions on*, 14(3):910–922, 2012.

[36] G. Thomas, P. Mills, P. Debenham, and A. Sheikh. Surround Video. *White Paper WHP 208, http://downloads.bbc.co.uk/rd/pubs/whp/whp-pdf-files/WHP208.pdf*.

[37] R.-D. Vatavu and M. Mancas. Visual attention measures for multi-screen tv. In *Proceedings of the 2014 ACM international conference on Interactive experiences for TV and online video*. ACM, 2014.

[38] W. Verhelst and M. Roelands. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 2, pages 554–557. IEEE, 1993.

[39] Z. Yuan, T. Bi, G.-M. Muntean, and G. Ghinea. Perceived synchronization of mulsemedia services. *Multimedia, IEEE Transactions on*, 17(7):957–966, 2015.

[40] Z. Yuan, G. Ghinea, and G.-M. Muntean. Beyond multimedia adaptation: Quality of experience-aware multi-sensorial media delivery. *IEEE Transactions on Multimedia*, 17(1):104–117, 2015.

[41] L. B. Yuste, F. Boronat, M. Montagud, and H. Melvin. Understanding timelines within mpeg standards. *IEEE Communications Surveys & Tutorials*, 18(1):368–400, 2016.