Design and Evaluation of a Second Screen Interactive Digital Media Solution using MPEG-DASH

Jomar Brudeli
Department of Electronic Systems, IE
NTNU
Oslo, Norway
jomar.brudeli@gmail.com

Shafaq Irshad
Department of Electronic Systems, IE
NTNU
Trondheim, Norway
shafaq.irshad@ntnu.no

Andrew Perkis
Department of Electronic Systems, IE
NTNU
Trondheim, Norway
andrew.perkis@ntnu.no

***Abstract*—This research work designs and evaluates synchronized second screen interactive digital media using Moving Picture Experts Group Dynamic Adaptive Streaming over HTTP (MPEG-DASH). Second screen technology involves the use of a computing device, such as a smartphone or tablet, to provide an enhanced viewing experience for content on another device. The study design uses a structured format for defining MPEG-DASH inline events to trigger the second screen events. Furthermore, the ability for users to make choices that impact the story was implemented by generating several manifests and enabling the trigger manifest to refresh after each question. By performing a formal subjective user experiment, this study assesses the Quality of Experience (QoE) performance of the prototype. The results were positive and strongly indicate a high QoE for the entire system. A secondary result of our work is enhancing the QoE methodology for the assessment of second screen applications. The implemented system can easily be incorporated in the MPEG-DASH standard, allowing support for using the inline event for second screen triggering.**

***Index Terms*—Interactive media; Streaming; MPEG-DASH; Entertainment; QoE; Second screen**

I. INTRODUCTION

The world of entertainment and media consumption is constantly changing. It has evolved from stories around the bonfire and painting in caves to cinemas, black and white televisions and finally ended up as the endless amounts of content available on online streaming services today. New mixed media allow a combination of new technologies, such as Second Screen with Interactive storytelling [1].

There has not been a lack of attempts at combining the traditional forms of entertainment with the new technologies [2]–[6], where results have usually just turned out to be some gimmick, adding little to nothing to the overall end-user experience. In the context of 3D movies, however, the recent popularity of Netflix's blockbuster interactive movie "Black Mirror: Bandersnatch" has shown that the audience might be ready for something more [7]. This has led to two of the most prominent players in the industry, i.e., Netflix and YouTube, announcing that they have plans to increase their investments in creating interactive digital media content [8].

The second screen can be a tool for storytelling, interactivity, and increase immersion. A standardized way to utilize all these secondary screens in video streaming technologies is to use Moving Picture Experts Group (MPEG) - Dynamic Adaptive Streaming over HTTP (DASH) [9] [10]. MPEG and 3rd Generation Partnership Project (3GPP) have developed MPEG-DASH to enable interoperability in the industry [11] [12].

One of the typical applications of DASH is audiovisual streaming service due to its high bit-rate. Several studies have been performed to evaluate the Quality of Service (QoS) of audiovisual services [13], where they suggest empirical quality models of audiovisual content [14]. However, there has been very little work on the Quality of Experience (QoE) assessment of such video streaming technologies. We believe it is essential to evaluate the perceived QoE of second screen streaming solutions for finding the optimal balance between available network resources and quality. In this paper

1) We studied a solution for streaming audiovisual content for second screen applications using the MPEG-DASH standard. The goal was to support the viewers by extending and enhancing broadcast TV without disrupting the viewers' attention.
2) We studied using the MPEG-DASH standard to stream audiovisual content for the second screen by designing a second screen streaming solution using MPEG-DASH. The application aimed to support the viewers by extending and enhancing broadcast TV without disrupting the viewers' attention.
3) We presented a solution for synchronized second-device media content that could be used to create a more immersive experience for users. A methodology for QoE testing of interactive second screen prototype is also described. A subjective user experience evaluation was conducted to evaluate the prototype and assess its quality.
4) For the subjective testing, four different event types were used in four versions of the video (in the second screen

 ISBN: 978-1-61208-839-6

prototype) to ensure consistency in the results.

5) Results of the subjective testing were evaluated using two approaches. First, the results of version V1 (event types: none) were compared with V3 (Event type: vibration). In the following approach, V2 (event type: vibration, sound, video, and web) was compared with v4 (event type: sound, video, and web). Mean Opinion Score (MOS) was used to test the subjective performance of the system, and a t-test was used to compare results between the different versions.
6) We studied whether the means of the two distributions are equal. Furthermore, if there is a significant change in the QoE between different versions. (i.e. V1 vs V3 and V2 vs V4)
7) The validation provided us with insights into whether it is possible to develop a method for synchronized second screen content based on existing international standards for video codecs or streaming technology or not.

The paper is organized as follows. Section II defines some literature and theories that are needed to keep up with the method. Section III describes the design methodology in detail, including essential terminologies and specifications of the system. Section IV gives a systematic overview of subjective user evaluation. Sections V and VI discuss the results, and Section VII is a conclusion to the paper summing up the essential points and findings of our work.

II. Background and Related Work

A. Second Screen Technology

Second screen interactive digital media platforms are used nowadays for detecting and dynamically synchronizing media content (e.g., television programs or movies) that a viewer can watch while providing related content on a second screen for enhancing the viewer experience [15].

Using second screen interaction to enhance media consumption experience is not a new idea, and several attempts have been made to enhance the viewers' experience with second screens [16]. In sports, Tour de France developed an app that gave viewers real-time information synced to the live race [4]. In the gaming industry, PlayStation released ”Hidden Agenda”, a game where friends get together and play the game together on their mobile phones [5]. The dutch movie APP [3] from 2014 came with a supplementary mobile app that used audio to sync to the movie, showing other videos and information.

The latest attempt of creating an interactive cinematic is the movie Late Shift [2] [6] giving the audiences the possibility of changing the course of the movie by making choices and other interactive elements on a synced application. Late Shift even offered these options in the cinema, letting moviegoers vote for the choices that the characters in the movie should make. Current research is evolving around methods and systems to display content playing on the second screen device.

B. Streaming Technology

Dynamic Adaptive Streaming (DASH) over Hypertext Transfer Protocol (HTTP), ISO/IEC 23009-1 [9], commonly-known as MPEG-DASH, is an ISO standardized streaming protocol that enables adaptive bit-rate video streaming over HTTP. This is similar to Apple's HTTP Live Streaming (HLS) and Adobes HTTP Dynamic Streaming (HDS), but not open to all and not bound by any company licensing and limitations. For an extended period, HLS has been the preferred solution for video streaming. However, now we see that the transition to DASH is getting along well with many of the major streaming companies, including Google and Netflix [10].

All modern browsers that support HTML5 and Media Source Extension (MSE) now support MPEG-DASH. The only significant exception from this is Apple's Safari browser, which does not support MSE. There are no reports yet regarding when that might happen.

To offer an adaptive bit-rate for the stream, the content needs to be encoded in different resolutions and then divided into small segments in time. In addition to all these small segments in different bit-rates, the stream would also usually include subtitles and audio in different languages and other metadata. To keep control over all these different parts, MPEG-DASH uses a Media Presentation Description (MPD), also known as a manifest. This is an XML document that follows a specific structure that tells the client where to find all the different parts and modify them. In Figure 1, the high-level representation of an MPD is showed.

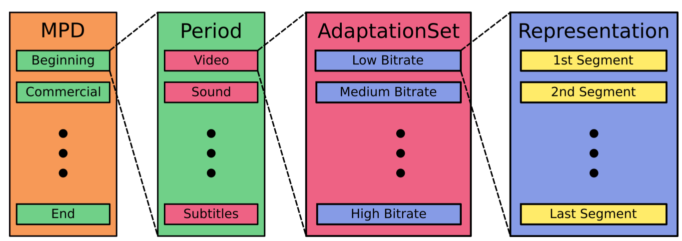

Fig. 1. High level illustration of how a simple MPD could be structured. Figure adapted from [10]

The MPEG-DASH standard ISO/IEC 23009-1 [9] defines that events may be signaled in the MPD on **Period** level. The EventStream element contains two different attributes, @schemeIdUri and @value. The @schemeIdUri provides a Uniform Resource Identifier (URI) to determine which scheme the event follows. The @value attribute is optional and used to distinguish between different types of events in a scheme. In the standard, an MPD validity expiration event is specified as an inline event. The event has the following URI:

`urn:mpeg:dash:event:2012`

This research paper uses existing international media standards for the video to develop second screen applications that enhance users' experience. Embedding and extracting the information from and into the media stream is an important aspect. The second screen's content should be triggered synchronized with the original media presentation at specific predefined points or frames.

III. Design method

The method used in this study is based on a user-centered design, and the use case is based on inserting trigger points in a Sci-fi movie developed by the mixed-media production company KapOow [17]. The scene used for the test and validation required trigger points for specific events, for example, opening a breaking news radio segment about 10 seconds into the clip. To investigate the QoE of the interactive second screen, a prototype was developed for testing purposes where different trigger points were implemented.

The prototype was evaluated by a target group, chosen from Sci-fi interested candidates with a keen interest in using mobile technology. The validation criteria were to optimize user QoE. Our definition of QoE is according to the Qualinet White Paper on Definitions of QoE [18]. The definition is: "The degree of delight or annoyance of the user of an application or service. It results from the fulfillment of users' expectations concerning the utility and enjoyment of the application or service in the light of the user's personality and current state."

Using techniques like in-depth interviews and user experiments to get feedback from users regularly during the development of the product, one can make sure that the final solution ends up being something the end-users want to use. Here, User-Centered Design (UCD) is used, a design methodology that focuses on involving end-users in developing a product. This research also investigates how to assess the QoE for end-users of second screen applications by using the prototypes in a subjective evaluation experiment, focusing on the effect of triggering synchronized vibrations on the second screen.

The development of the Android Application and the database structure is not within the main focus of this research paper; however, the detailed documentation and source code can be found at the public GitHub repository [19]. Important terms used are described in Table I.

TABLE I
Table describing important terms

Trigger-event:	Event that occurs on second screen (e.g. video, sound)
Trigger-point:	Timestamp for trigger-event (HH:MM:SS)
Trigger-message:	Defines the trigger-event (e.g. video::filename.mp4)
Trigger-information:	Information containing trigger-points and trigger-message for all trigger-events that are defined for a video.

The next section describes the specifications and a step-by-step procedure of development.

A. Specifications

Trigger information

- solution based on the MPEG-DASH standard (See Fig 3).

The master screen contains

- a custom streaming server.
- a webpage integrating the Dash.js player, also providing login and user identification support.
- some form of login and user identification.

We will also explore the possibility of interactivity (path-choice) using existing technology in the MPEG-DASH standard.

The second screen contains

- functionality for playing sound and video.
- functionality for vibrations.
- a solution to log on to a session based on scanning a QR-code shown on the master screen.
- support for interactivity and voting with several users in the same session.

B. Development

Figure 2 shows the setup for the prototype.

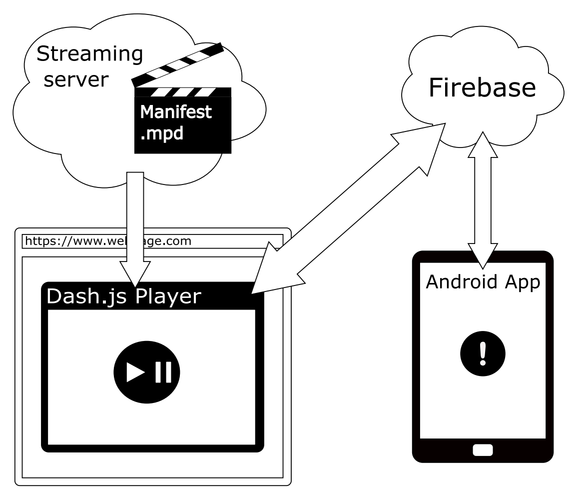

Fig. 2. Simple sketch of the setup used in the prototype

1) Encoding video files: To ensure the best possible experience for streaming, the desired video was encoded into different bit-rates (resolutions) to let the player jump between them depending on available bandwidth, i.e., adaptive bit-rate. This is a common way of doing video streaming today and made the setup as realistic as possible. The test video was already h.264 encoded with a framerate of 25 fps and placed in an MP4 container. Table II shows the five different resolutions and corresponding bit-rates used in the prototype.

The interval between keyframes was set to 25. With a framerate of 25 fps for the original video, this will guarantee

 ISBN: 978-1-61208-839-6

TABLE II
TABLE SHOWING THE RESOLUTIONS AND CORRESPONDING BIT-RATES CHOSEN FOR THE VIDEO. ALL IS IN THE 16:9 ASPECT RATIO.

Resolution[p]	Bit-rate [kbps]
426x240	260
640x360	600
848x480	1060
1280x720	2400
1920x1080	5300

that one keyframe will be located in each segment when the segment length is set to 1 second, after encoding the videos as put into an MP4 container, using MP4Box. The H.264/AVC encoder x264 was used to do the encoding. The details of parameters used can be found in the bash script written for the task, *generate_dash.sh*, found in Appendix A.

2) Generating Manifest: When all the videos have been prepared, they must be divided into segments, and then a manifest must be generated. Using the multimedia packager MP4Box, the manifest can be automatically generated for us by adding the correct parameters. In this project, the following parameters were used:

```
MP4Box -dash 1000 -rap video_1080.mp4
... video_240.mp4 video.mp4#audio
```

The `-dash` option specifies the duration of the segments in milliseconds. A segment duration of 1 second is considered a good value. The `-rap` specifies that every segment should start with a random access point. Finally, the `#audio` modifier extracts the audio track from the source file.

3) Adding Events: For this project, the dash.js [20] reference client implementation for MPEG-DASH playback via JavaScript was used.

To embed the event information in the media stream, it was decided to use inline events. Inline events are declared inside the manifest on the Period level, as shown in Figure 3.

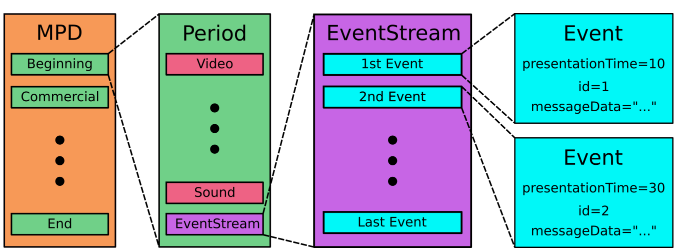

Fig. 3. High-level model showing how EventStreams and Events are placed in a MPD.

Figure 4 shows an example of how to write a functioning EventStream inside the MPD. There are a few important things to notice. First, the `id` of each of the Events must be unique; otherwise, they will not function properly. Secondly, the `presentationTime` property sets the time in second for when an event is triggered. Finally, the `schemeIdUri` property of EventStreams is important for identifying what event this is.

```
<EventStream schemeIdUri="urn:mpeg:dash:event:trigger:2019 value="1">
  <Event presentationTime="10" id="1" messageData="video::test_video.mp4" />
  <Event presentationTime="30" id="2" messageData="sound::test_sound.wav" />
  <Event presentationTime="60" id="3" messageData="vibration::1000,255" />
  <Event presentationTime="90" id="4" messageData="web::https://www.ntnu.no" />
</EventStream>
```

Fig. 4. EventStream that follows our suggested format for declaring trigger-events. All four supported event-types are declared in the example at trigger points 10 sec, 30 sec, 60 sec and 90 sec.

In the MPEG-DASH standard, two Uniform Resource Identifiers (URI) were specified, each with a specified set of rules for what the client should do in response. Dash.js handles the events internally and does not expose them to the developers using the client. This makes it impossible for us to detect an event from outside the client. Also, the standard specifies that events should be deleted after they are triggered, making it impossible to rewind in a video and trigger the event several times.To overcome these complications, the following modifications were made:

- Define a new URI for second screen events.
- Make the client expose the trigger instead of handling it internally.
- Do not remove events after they have been triggered.

To implement our suggested changes, the dash.js had to be modified to suit our needs. Specifically the *Eventcontroller.js*, found in Appendix A.

The standard states that all users may define a schemeIdUri for their respective application, as long as it does not conflict with any other URI in the standard. We suggest the following URI for second screen events.

```
schemeIdUri="uri:mpeg:dash:event:
trigger:2019"
```

Another property worth mentioning is the `value` parameter of the `EventStream` objects. This could be used to distinguish between different EventStreams, e.g., to tailor events to users in different countries or with different devices. For this project, the default value of 1 was used.

4) Path Choice: To let the viewer impact the story they are watching, path choices were implemented using MPEG-DASH. The following trigger-message was suggested.

```
choice::<dur>,<opt_1>,<branch_1>,
     <opt_2>,<branch_2>
```

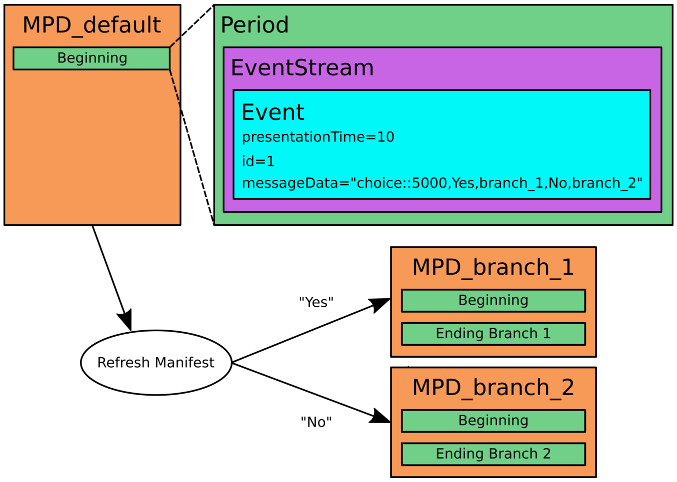

Fig. 5. High-level implementation of path choice using MPEG-DASH standard

Three manifests are uploaded to the webserver (Firebase),

- VIDEO_default.mpd,
- VIDEO_branch_1.mpd
- VIDEO_branch_2.mpd

First, the default manifest is loaded, then the video is initialized. Our examples consist of only one Period that contains the media for the beginning of the video. When a choice-event is triggered on the second screen, two buttons appear, making the user chose between two options, opt_1 and opt_2. A timer is also counting down the time. When the timer has reached zero, a refresh manifest is triggered. If the viewer chose opt_1 then when the player requests a refresh of the manifest, it receives VIDEO_branch_1.mpd and the opposite for opt_2. If none of the options is chosen by the user before the time runs out, a random option is picked. The logic handling the manifest refresh is handled with cloud functions in the webserver. Figure 5 show an example of how this is implemented.

All the functions described for the prototype were verified in the lab, and QoE testing was performed.

IV. Evaluation - Quality of Experience Assessment

A subjective user experiment was conducted to evaluate the prototype and to assess the QoE of the system. The central hypothesis to be tested was that the second screen vibration event positively affected the quality of experience.

KapOow contributed to the video that was used during the experiment. Four different manifests, V1, V2, V3, and V4, were prepared, all with different events included. Table III shows which events were included in which versions.

TABLE III
Table showing the second screen events enabled in each of the four versions of the video used in the subjective user experiment

Version	Events types
V1	None
V2	Vibration, Sound, Video and Web
V3	Vibration
V4	Sound, Video and web

To ensure that there were no systematic errors due to the order in which the participant view the different versions, each participant viewed them in random order.

1) Statistical analysis: To evaluate the effect of vibration on the overall quality of experience, two different approaches are chosen. The first is to compare the version results without any event occurring on the second screen (V1) and the version with only the vibrations events occurring (V3). Next, the version with all events (V2) is compared to V4 (everything except vibration).

The MOS is commonly used in telecommunication to evaluate the subjective performance of a system (e.g. sound quality). The MOS for a specific question is defined as

$$MOS = \frac{\sum_{i=0}^{n} R_i}{n} \quad (1)$$

where R_i is the score from participant i, and n is the total number of participants.

To determine if the MOS for each question in the subjective questionnaire significantly increases when vibration was added. Paired observations Student's t-Test between the score of questions with and without vibration was used. The differences $d_1, d_2, ..., d_n$ between the scores for each of the n questions are used as a basis for the hypothesis test. They are assumed to be drawn from the stochastic variables $D_1, D_2, ..., D_n$, each with mean $\mu_D = \mu_2 - \mu_1$, and variance σ_D. The hypothesis to be tested is given by

$$\begin{aligned} H_0 &: \mu_d = 0 \\ H_1 &: \mu_d \neq 0 \end{aligned} \quad (2)$$

Where H_0 states that the means of the two distributions are equal, and therefore no significant change has occurred. If we can successfully reject H_0 on the other hand, based on our results, we can successfully say that we have an increased QoE.

V. Result and Analysis

Twenty users between 18 and 26 years of age were chosen to participate. As the media provided by KapOow to use in the experiment was in a particular genre (Sci-Fi/ horror/comedy), the participants were chosen within the assumed target audience. 65% of the users were male, and 35% were female. However, even if this genre might be more appealing to a male audience, the gender distribution ended up being quite well distributed.

 ISBN: 978-1-61208-839-6

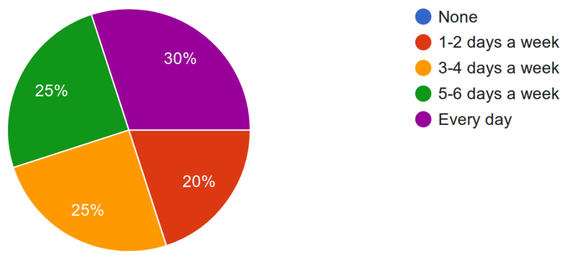

Fig. 6. Figure demonstrating the days of week participants used streaming services (Netflix, HBO, Amazon Prime, etc.) to watch movies and/or series

The streaming habits of the participants were also considered an important parameter worth measuring. While all of them used streaming services weekly, the amount of use was well distributed, as Figure 6 shows.

The MOS for all ten questions was calculated for all four versions. Figures 8 and 9 show how the MOS evolved over the questions. We can observe that the version without any events (V1) occurring on the second screen generally performed worse than the rest. It also shows that the version that showcased all the different types of second screen events performed best (V2).

With only 20 participants in the test, it is limited how much we can deduct from the MOS, at least not without considering the scores' spread. Therefore the MOS was calculated within a 95% confidence interval as well. Figure 8 shows how the score for questions 1 to 5 behaved, and Figure 9 shows the same for questions 6 to 10. The versions are grouped two-and-two such that it is easy to compare versions that differ only regarding if the vibration was present or not.

To statistically prove that vibration has a statistically significant positive effect on the MOS for each question, a Paired Student's t-test was performed on each question. The first test, T1, compared the version with no events (V1) with the version with only vibration (V3).

The second test, T2, compared the version with all event types (V2) with everything except vibrations (V4). The initial null hypothesis is that the MOS of the compared versions is equal. A P-value below the typical value of 0.05 indicated that the difference is significant and that the null hypothesis, therefore, can be rejected.

The results from the t-Tests can be seen in the Tables below. Table IV a) shows the results from the first t-Test and Table IV b) shows the results from the second. The P-values below the chosen threshold of 0.05 are marked with a *. Of all the 10 questions, 6 of them passes the t-Test in both T1 and T2, and only question number 4 fails in both tests.

Some of the participants made a few additional comments, which are listed below.

- "In the end, a bit much vibration in my opinion".
- "Notification/commercial in the middle of experience was like breaking the 3rd wall and pulled me out of the immersion. Sound/vibration and a second screen for news did enhance the experience."

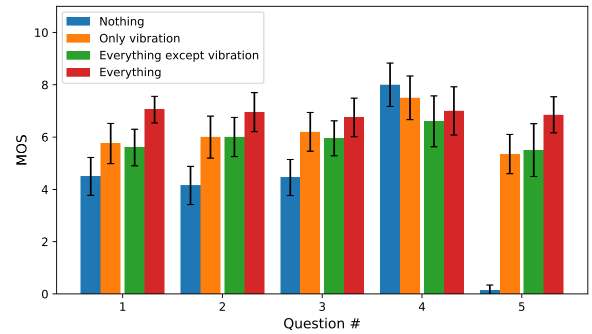

Fig. 8. MOS for questions 1 to 5, with a 95% confidence interval

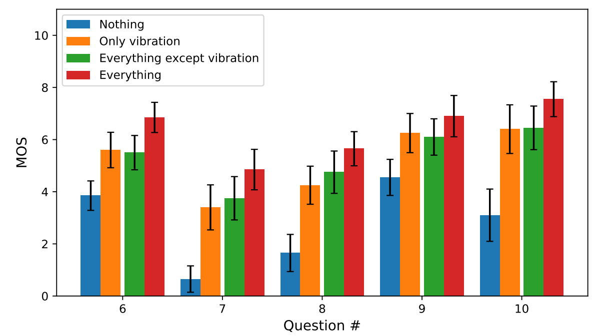

Fig. 9. MOS for questions 6 to 10, with a 95% confidence interval

- "Lack of sync between vibration and sound. Vibrations should be modulated with the sound."
- "Notifications for ads that paused the movie broke the immersion quite a bit. vibration for the garage door lasted a bit long. Other than that: had a very nice experience"
- "Enjoyed the vibration and sound from the synchronized second screen, but the video did not feel natural and paused the story without adding something essential to it. All in all a great experience."
- "The screen in itself seems somewhat distracting, but a media for receiving makes the experiment considerably richer, at least for the horror elements. Could maybe work just as well with a handheld device without a screen (e.g. gaming controller). The light from the second screen was somewhat distracting."
- "It was cool!"

VI. Discussion

For second screen applications to be a successful extension to multimedia content, at least three critical factors should be in place. Technology, content, and audience. In this paper, we have focused on the technology aspect and the audience experience through new ways of consuming stories. The subjective user evaluation experiment results indicated that the technical solution suggested in this work has great potential and that the

TABLE IV
TABLE SHOWS MOS FOR ALL FOUR VERSIONS. THE TABLE ALSO INCLUDES P-VALUES CALCULATED FROM BOTH PAIRED STUDENT'S T-TESTS, T1 AND T2. (* P-VALUE LARGER THAN 0.05. NOT SIGNIFICANT POSITIVE CHANGE IN MOS)

(a)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
MOS V1	4.500	4.150	4.450	8.000	0.150	3.850	0.650	1.650	4.550	3.100
MOS V3	5.750	6.000	6.200	7.500	5.350	5.600	3.400	4.250	6.250	6.400
P-value T1	0.056*	0.006	0.008	0.144*	0.000	0.003	0.000	0.000	0.0023	0.000

(b)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
MOS V2	7.050	6.950	6.750	7.000	6.850	6.850	4.850	5.650	6.900	7.550
MOS V4	5.600	6.000	5.950	6.600	5.500	5.500	3.750	4.750	6.100	6.450
P-value T2	0.001	0.031	0.149	0.408*	0.056*	0.002	0.032	0.046	0.104*	0.025

QoE for the system is promising. There are, however, a few remarks that need to be emphasized.

The synchronization is the weakest part of the technical solution. It does not consider the variation in delay for the application to detect a change in the database. During the lab validation, it did not appear to be a problem as long as it took the setting of the trigger times for the events into account. However, a few participants commented on slight delays, indicating that the variation between sessions is too significant to be ignored. This is probably the biggest reason why this technology's most popular commercial implementations have used watermarked audio. However, we think that our solution has several advances over this technology, The most important one being that the user can watch the video with their headsets. It reduces the device's processing power demands, and the users are not dependent on good audio conditions. The synchronization issue should be investigated further, but a simple solution might be implementing time correction/synchronization protocols between the client player and the second screen application. It should also be mentioned that not all event types have the same requirements for synchronization, but vibration and sound most certainly do.

Moreover, is it essential to embed the trigger information in the media file or stream? It might seem like a complicated solution to a simple problem. For this solution, we used the .trig file to store the information before it gets uploaded and parsed into the database and embedding the inline event from MPEG-DASH. By using already implemented functionality from an international standard, the process gets much more manageable. DASH compatible web players already have support for listening after inline events. The only thing developers need to do is specify what event they are waiting for in their JavaScript environment, keeping the complexity as low as possible.

We have suggested and implemented a few different second screen events for this project, including sound, video, path-choice, and web notifications. These are just suggestions developed in collaboration with content creators. There is no limit to what might be triggered on the second screen, as long as the MPEG-DASH client detects the events and the surrounding application implements them. This is an important aspect. Instead, the technology should not be in focus, work as a toolbox for creative minds to utilize for creating exciting and involving content. Not create content and then throw some second screen content on it afterward, which would make the technology just a gimmick.

This is one of the significant problems with the evaluation of the system. The content was created for regular linear storytelling and was just converted into a second screen experience by an engineer. This is not at all ideal and would probably affect the QoE significantly. Throwing some new tech at some old content and just expecting the audiences to embrace it is naive. Netflix's success with "Black Mirror: Bandersnatch" is an excellent example of this. The story and interactivity were tailor-made to fit each other, enhancing the experience. Later attempts have not reached the same popularity. The fact that the content is considered an influential factor is one of the essential differences between QoS and QoE. Even if the system and the technology work perfectly, the users' experience will be damaged if they do not feel that the technology contributes to the story in a good way. Take question 5 from the evaluation experiment, "Do you feel the second screen contributed to the story?". This was the question that showed the most significant jump in MOS between having no events (V1) and having just vibration events(V3), see Table IV. But still only reached a score of 5.35 on the scale "Not at all" (0p) to "Very much" (10p), reaching barely over the middle of the scale.

VII. CONCLUSION

The most important discoveries from this work include using MPEG-DASH inline events to trigger the second screen events. Also, a structured format for defining trigger-information has been suggested. Finally, users' ability to make choices that impact the story was implemented by generating several manifests. The solution was made possible with existing standards and infrastructure and minor changes to MPEG-DASH clients. Evaluation of the prototype was done with a subjective user experiment, and the results were positive and strongly indicated a high QoE for the entire system. The subjective user experiment was centered around the hypothesis that synchronized vibration events on the second screen would positively affect the QoE. In 6 out of 10 questions, we observed a significant increase in MOS compared to versions without vibration. From this, we can conclude that the hypothesis is, to a certain degree, confirmed. The results can be used in the ongoing development of the DASH standard. With few modifications to the current standard, using the inline event for second screen triggering could be implemented.

Acknowledgment

A big thank you goes to Patrick Velasco for developing both the prototypes.

References

[1] H. Simon, E. Comunello, and A. Von Wangenheim, "Enrichment of interactive digital tv using second screen," *International Journal of Computer Applications*, vol. 64, no. 22, pp. 58–64, 2013.
[2] (2016) Late shift - the world's first cinematic interactive movie. [retrieved: March, 2021]. [Online]. Available: http://lateshift-movie.com/
[3] (2014) Review: 'app' is the first movie designed for you to use your phone while you watch, and it's not bad. [retrieved: March, 2021]. [Online]. Available: https://www.indiewire.com
[4] (2018) Letourdefrance: iphone, ipad and android official apps. [retrieved: March, 2021]. [Online]. Available: https://www.letour.fr/en/mobiles-apps
[5] (2019) Hidden agenda — ps4 games — playstation. [retrieved: May, 2019]. [Online]. Available: https://www.playstation.com/en-au/games/hidden-agenda-ps4/
[6] (2019) Ctrlmovie: An audience-collaborative cinematic experience. [Online]. Available: http://www.ctrlmovie.com/
[7] E. Pajorová and L. Hluchý, "Augmented reality as a higher education form for students with delimited ability," in *Smart Education and e-Learning 2019*, V. L. Uskov, R. J. Howlett, and L. C. Jain, Eds. Singapore: Springer Singapore, 2019, pp. 461–469.
[8] L. Shaw. (2019) Youtube is developing choose-your-own-adventure programs. [retrieved: March, 2021]. [Online]. Available: https://www.bloomberg.com/news/articles/2019-04-09/youtube-is-developing-choose-your-own-adventure-programs
[9] "Information technology – dynamic adaptive streaming over http (dash) – part 1: Media presentation description and segment formats," International Organization for Standardization, Standard, 2014.
[10] (2019) Mpeg-dash (dynamic adaptive streaming over http, iso/iec 23009-1). [retrieved: March, 2021]. [Online]. Available: https://bitmovin.com/dynamic-adaptive-streaming-http-mpeg-dash/
[11] T. Stockhammer, "Dynamic adaptive streaming over http–: standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 133–144.
[12] T. C. Thang, Q.-D. Ho, J. W. Kang, and A. T. Pham, "Adaptive streaming of audiovisual content using mpeg dash," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 78–85, 2012.
[13] S. Takenaka, K. Kanai, and J. Katto, "Qos improvement of mobile 4k video by using radio quality map," in *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2015, pp. 541–542.
[14] T. C. Thang, J. W. Kang, and Y. M. Ro, "Graph-based perceptual quality model for audiovisual contents," in *2007 IEEE International Conference on Multimedia and Expo*. IEEE, 2007, pp. 312–315.
[15] B. Slavin, S. Rosenberg, and A. Glennon, "Second screen interactive platform," May 2 2013, uS Patent App. 13/621,277.
[16] K. Blumenstein and et al., "Visualizing spatial and time-oriented data in a second screen application," in *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '17. New York, NY, USA: ACM, 2017, pp. 84:1–84:8. [Online]. Available: http://doi.acm.org/10.1145/3098279.3122127
[17] (2019) The next generation mixed-media production company. [retrieved: March, 2021]. [Online]. Available: https://kapoow.no/
[18] K. Brunnström, S. A. Beker, K. De Moor, A. Dooms, S. Egger, and e. a. Garcia, "Qualinet white paper on definitions of quality of experience," 2013.
[19] J. Brudeli. (2019) Github repository, they came. [Online]. Available: https://github.com/jbrudeli/they-came-colab
[20] (2019) dash.js. [retrieved: March, 2021]. [Online]. Available: https://www.videolan.org/vlc/index.html

Appendix A
Appendix (source code)

A. GitHub repository

https://github.com/jbrudeli/they-came-colab

B. Statistical analysis

https://www.dropbox.com/s/blicxhdz9d48zxi/analysis.py?dl=0

C. Generate DASH Manuscript, bash script

https://www.dropbox.com/s/i9p8qlfon1t6j9q/generate$_dash.sh?dl=$
0

D. Modified dash.js Eventcontroller

https://www.dropbox.com/s/9hkv67qf7yaoyeo/EventController.js?dl=0