Network Performance in HTML5 Video Connections

Sandra Sendra[1,2], Ana Isabel Túnez Murcia[1], Jaime Lloret[2], José Miguel Jiménez[2,3]

[1]Dept. of Signal Theory, Telematics and Communications Department (TSTC),
Universidad de Granada, Granada, Spain

C/ Periodista Daniel Saucedo Aranda, s/n. 18071 (Spain)

[2]Instituto de Investigación para la Gestión Integrada de zonas Costeras, Universitat Politècnica de València.

C/ Paranimf nº 1, Grao de Gandía – Gandia, Valencia (Spain)

[3]University of Haute Alsace, France,

2 rue des Frères Lumière, 68 093 Mulhouse Cedex

E-mail: ssendra@ugr.es, wirnemc@gmail.com, jlloret@dcom.upv.es, jojiher@dcom.upv.es

Received: July 19, 2018 Accepted: October 30, 2018 Published: October 31, 2018

DOI: 10.5296/npa.v10i3.13933 URL: https://doi.org/10.5296/npa.v10i3.13933

Abstract

Currently, most of remote education systems use video streaming as the main basis to support teaching. These emissions can be seen in devices with different hardware features such as personal computers, tablets or smartphones through networks with different capacities. The use of different web browsers and coding options can also influence the network performance. Therefore, the quality of the video displayed may be different. This work presents a practical study to establish the best combination of web browsers and containers to encode multimedia files for videos streaming in personal computers running Windows 7 and Windows 10 operating systems. For this, a video encoded with different codecs and compressed with different containers have been transmitted through a 1000BaseT network. Finally, the results are analyzed and compared to determine which would be the most efficient combination of parameters according to the resolution of the transmitted video.

Keywords: HTML5, e-Learning, Containers, Operating System, Bandwidth, Delay, Bitrate, Throughput, Error Rate.

1. Introduction

The evolution of Information and Telecommunications Technology (ICT) and the Internet has made possible that society gradually assumes the exchange of information through electronic devices as an essential part of their day-to-day [1]. Currently, we daily use technology to perform any task in home environments, the workplace or in the academic field [2] where access to teaching resources from the internet greatly facilitates the training of qualified personnel. The remote access to teaching resources is currently is known as virtual education or electronic learning (e-Learning) [3].

The streaming of digital video and audio over Internet with teaching purposes [4] is part of the audiovisual trends [5]. This sort of service involves the requirement of knowing the equipment the users have. It is also important to know the kind of content is being transmitted as well as the most appropriate protocols in order to adequate the transmission parameters to reach good values of quality of service (QoS) and quality of experience (QoE) [6]. Problems in network capacity can limit the amount of data the users can receive and thus the quality of video [7]. So, it is imperative the use the most adequate codec and container to transmit the video. According to Cisco annual report, the current traffic on Internet is mainly composed by video streaming, data, http content and VoIP, being platforms such as YouTube and Netflix the main sources of video content [8].

E-learning [9] is a type of online teaching that allows user interaction with teaching resources through the use of computer tools. This teaching can be focused on students of all ages. According to the previous definition, we can list a series of basic characteristics of e-learning:

- Simplicity of use.
- Multimedia system (text, audio, video, image).
- The distances between teachers/professors and students disappear.
- It is economical for the students.
- It is interactive.
- It is accessible.

Historically, these teaching platforms were designed by using Adobe Flash to distribute content to a greater number of devices. However, many mobile devices do not support this technology. HTML5 [10] technology is supported on a wide range of platforms and browsers which allows a greater number of devices to access the content that these platforms can offer. Compared to Flash-based platforms, HTML5 is more stable and flexible. Users do not have to download any additional applications since HTML5 is based on browsers and it can be accessed directly from them. It is also easy to integrate audio and video elements without using any type of additional plugins. In addition, HTML5 allows offline storage of content, as well as reduces CPU and battery usage which makes it much more efficient compared to Flash-based technologies. HTML5 is presented as the ideal solution to distribute multimedia content on all platforms as it efficiently and stably supports any video and audio execution option, without generating errors or including flash code. Therefore, it is not strange to ask ourselves in what way we can optimize the sending of video on a platform such as e-Learning.

Taking into account all these arguments, this paper studies how the format of video served affects the ratio of data transfer rate, transmission delays, transmission errors and throughput when distributing content over Windows 7 and Windows 10 using different browsers, containers and codecs on an online platform based on HTML5 The browsers used in this study are Google Chrome, Firefox and Opera. To do this, a video encoded with different codecs and containers is used to be transmitted on a 1000BaseT network.

The rest of this paper is structures as follows. Section 2 presents some interesting tests related to practical studies on how HTML5 can be used to optimize the video streaming. Section 3 summarizes the main features of HTML5 and the different containers and codecs we are going to use in our test bench. The scenario, tools and the videos used to carry out or test bench are presented in Section 4. Section 5 presents the obtained results. Finally, Section 6 shows the conclusion and future work.

2. Related Work

This section presents relevant previous works where the authors try to show how the video format affects parameters related to network performance such as bandwidth, time of service or Quality of Service (QoS) when video is distributed.

Analyzing the different works we found, it is possible to stablish two groups of studies. For example, there are authors like E. Ohwovoriole et al [11] that compared several codecs supported by HTML5. Their work was focused on working with rate-distortion performance under quantization or bitrate constraints. Authors did not take into account other factors such as delay, system complexity, etc. The test compared and extracted some interesting conclusions regarding to the use of the codecs High Efficiency Video Coding (HEVC), Advanced Video Coding (AVC/H.264) and VP8. The results showed that HEVC is able to reduce the bitrate up to 46% compared to AVC/H.264 while AVC H.264 reduced the bitrate by 21% compared to VP8.

Ohm et al. [12] presented a study that compared the compression of the video coding using the peak signal-to-noise ratio (PSNR) of the H.262/MPEG-2 Video, H.263, MPEG-4 Visual, H.264 codecs / MPEG-4 AVC and HEVC. According to the results, authors indicated that HEVC reached a subjective reproduction quality similar to H.264/MPEG-4 with a reduction of bit- rate around 50%. They recommended the use of HEVC for videos of low bit rates, high-resolution video content and low-delay communication applications.

Finally, Grois et al [13] presented a study that compared some other video coding standards. In this case, authors used H.264/MPEG-AVC, H.265/MPEG-HEVC and VP9 to see the coding performance of each one. According to the experimental results, H.265/MPEG-HEVC is able to save of 43.3% and 39.3% compared to the use of VP9 and H.264/MPEG-AVC respectively. Authors also indicate that VP9 produced an average bit-rate overhead of 8.4% compared to H.264/MPEG-AVC, with the same objective quality.

Regarding the coding times, the time spent by VP9 are more than 100 times higher than the one used by H.264/MPEG-AVC while the encoding time of VP9 with respect to H.265/MPEG-HEVC is reduced by a factor of 7.35 .

As we have seen, none of these works relates the use of codec and browsers with HTLM5-based web platforms. For example, López-Herreros et al [14] presented a study that analyzed the different video formats we also use in this work, i.e., MPEG-4, Ogg and WEBM with HTLM5-based transmissions for teaching purposes. Derived from this study, they were able to know the main characteristics of these three video formats. Lastly, authors designed an algorithm able to send the videos from a server using the most appropriate type of video format.

Finally, I. Mateos-Cañas et al. [15] proposed a decision algorithm which selected the most appropriate codec to compress a video as a function of its Chroma characteristics. The algorithm took into account the predominant color and the video quality to efficiently select the best codec for each case. The authors provided the protocol design for the proper operation of their system. From the results, the authors concluded that in terms of QoS and QoE, the H264 codec is the best option when the predominant color of videos are black or white while XVID offer the best results when video has as predominant color red, green or blue.

As we have seen, there are several studies on codecs comparisons but it is not easy to find practical tests that relate the use of codecs, web browsers and operating systems. Our work presents an improvement over the aforementioned works since our practical test benches include three web browsers (Google Chrome, Firefox and Opera) and three containers (MPEG-4, Ogg and WEBM) which are compatible with the three web browsers. The transmissions are sent through HTML5 connections. The study analyzes the network performance as a function of the video resolution and containers for Google Chrome, Firefox and Opera browsers in Windows 7 and Windows 10 for 1000BaseT networks.

3. Codecs for video and audio coding considered in this study

This section shows a summary of the main features of HTML5 standard. It also presents the different containers and codecs we have used to perform our experimental study. Moreover, we will show the compatibilities between the different browsers and these containers.

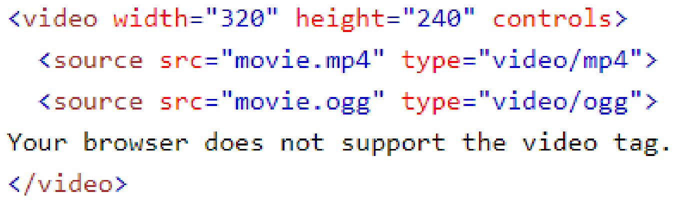
3.1 HTML5 Standard

HTML5 is a hypertext language based on XML tags with standardized names to be interpreted on any platform. In this way, HTML5 allows designers and developers to create a type of content that enables the transmission, decoding and reproduction of a video/audio file on a web browser. HTML5 incorporates the following capabilities:

1. Labels that exactly describe what they are designed to contain
2. Improved network communications.

3. Web Workers and Web Sockets to execute background processes and connections between client / server applications.
4. Support for CSS3 to manage GUI which means it can be content oriented.
5. SQL-based database API and JavaScript and Geolocation forms to incorporate services and improve communication between mobile and smart devices with the cloud.
6. Frames to add graphics and video without installing third-party plugins.

Related to the last feature, HTML5 incorporates support for decoding the containers through the *<video>* tag. The service and video adjustment is controlled according to the MIME headers included in the HTML Media Video tag. By making use of these headers, browsers are able to determine how to process the file (in this case a video). Its structure for this task is as follows (see Fig. 1):



```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

Figure 1. Example of HTML Media Video tag

Where *type=”video/mp4* indicates that the file "movie.mp4" is a file of video type and it has been encapsulated using the MP4 format.

HTML5 is designed to include several sources to the video element. This allows the browser to search, among the type labels, the first one which is compatible to be reproduced in the browser. In this way, the compatibility between devices is significantly improved although it does not guarantee the optimization of the presented content.

Finally, the execution features of JavaScript codes and the use of PHP databases allow the interaction with the client regardless of the device used. This feature is useful for obtaining data and "feedback" from the client so that the data can be used to adjust the presentation of the optimized content.

3.2 Codecs and containers and its compatibilities

A container is a type of file that stores audio, video, metadata, synchronization, and error correction information. In our case, we will work with containers MP4, OGG and WebM for the transmission of video and audio over HTML5. On the other hand, a codec is an algorithm for encoding (transmission, storage and encryption) and decoding (playback and editing) multimedia content. The most commonly used codecs for encoding multimedia content are H.264, Theora and VP8. The video information as well as the audio information contained inside a container is usually compressed according to the specification of a specific codec. There are many containers and codecs that give rise to different formats of video files.

In this subsection, we present the most popular codecs which are supported by HTML5 [6] and will be used in our test benches.

3.2.1 MP4 – H.264

H.264 [16] or MPEG-4 AVC (Advanced Video Coding) is a video encoding format for recording and distributing FullHD video and audio signals. It was developed and maintained by the ITU-T Video Coding Experts Group (VCEG) with the ISO / IEC JTC1 Moving Picture Experts Group (MPEG).

The H.264 format is a video transmission method compatible with data networks which provides high quality images without consuming too much bandwidth. It is normally used for recording, compressing and distributing video content. Some of the most important features of this codec are:

- H-264 has been designed to consume a smaller bandwidth for high resolution video transmissions. H.264 uses a very efficient codec that provides lower latency than traditional video standards such as MPEG-2.
- According to the standard definitions, H.264 presents the 80% less bitrate than Motion JPEG video. It is estimated that the bitrate savings can reach up to 50% or more, compared to MPEG-2.
- H.264 is a video extension solution independent of the manufacturer. Users can mix H.264 equipment from different video manufacturers without having compatibility or ownership problems.

It is the standard used to provide DTT or webTV service and supports mobile devices. The MP4 container usually uses the H.264 / AVC for video transmission and AAC codec for audio transmission. However, the MP4 container can hold files with different encodings.

3.2.2 OGG – Theora

Ogg is a free and open source container format, developed and maintained by the Xiph.Org Foundation. It is designed to provide efficient flow diffusion and manipulation of high quality digital multimedia. Ogg is defined in RFC 3533 [17] and the recommended MIME type for Ogg files is application / ogg defined in RFC 3534 [18].Ogg is not just a video or audio codec, it is a container that includes audio, video and subtitles, and allows reproducing file on both computers and other devices with sufficient processing power. Files in this format have long been incorporated into the audiobooks of the Gutenberg project [19].

The Ogg container format can multiplex several independent streams for audio, video, text (such as subtitles) and metadata. Inside the Ogg multimedia framework, Theora [20] offers a video layer with losses. The audio layer is most commonly provided by the Vorbis format [21]. It is oriented to music files, but it is possible to use options such as the Opus compression codecs, FLAC for audio lossless and OggPCM.

The .Ogg container format is capable of storing both audio and video. In contrast, the .mp3 container can only store audio. However, in 2007, Xiph.org created the .ogv container and the Theora codec, both specific for video. Since then, .ogg containers are only recommended for storing audio. The Ogg Vorbis compression format is capable of obtaining audio qualities equal to or higher than an .mp3 files with a lower file weight.

Theora has the advantage of its lower complexity, which makes it especially useful for mobile devices. As a disadvantage, it presents low quality with respect to MP4 stands out.

3.2.3 WebM – VP8

The WEBM [22] file format is intended to be used in HTML5 for the reproduction of royalty-free videos based on the Matroska file container. Web browsers Mozilla Firefox, Opera and Google Chrome support the playback of WEBM video files, but Internet Explorer 9 and Safari require the assistance of external options (such as QuickTime) to play WEBM media. YouTube uses WEBM files in its HTML5 playback experiment. This file container was created by ON2, Xiph and Matroska, but it has been adopted and modified significantly by Google in 2010.

WebM is an open source multimedia container format that defines the video and audio components in its own structure. It is designed to support the current needs of the Internet. It is one of the best and most modern containers for integrating multimedia elements into a web page without affecting the user experience, since it allows the integration of high quality videos without affecting the load time too much or damaging its usability.

Its main advantages stem directly from the objective of the WebM project [23], which was to create a format that helps address the specific needs we have when streaming videos through a web page. These needs are:

1. Low computational load to allow playback on any device, including low-power netbooks, portable devices, tablets, etc.
2. Simple container format.
3. High quality real-time video delivery.
4. Minimum codec profiles and suboptions.

To achieve all these functionalities, the WebM files have a structure based on the Matroska container and it consist of compressed video sequences with the VP8 or VP9 codecs and compressed audio sequences with the Vorbis or Opus audio codecs. VP8 [24] and VP9 are very efficient video compression codecs in terms of resolution, datarate and framerate.

3.3 Compatibility between containers, codecs and browsers with HTML5

To select the operating systems and browsers that are going to study, it is convenient to analyze their use. On the one hand, as Fig. 2 shows, the number of users of Windows as a desktop operating system is higher (78%) than the rest of the operating systems followed by OS X (14%) [25]. For this reason, 2 versions of Windows (Windows 7 and Windows 10) have been selected to carry out this study. Windows 8 is ruled out since it has an architecture very similar to Windows 10.

Regarding to the use of web browsers [26], Fig. 3 shows that Google Chrome is the most used web browser on all continents, followed by Firefox. Opera is the web browser that registers the lowest users' number. However, it is capable of supporting the requirements of html5 videos.

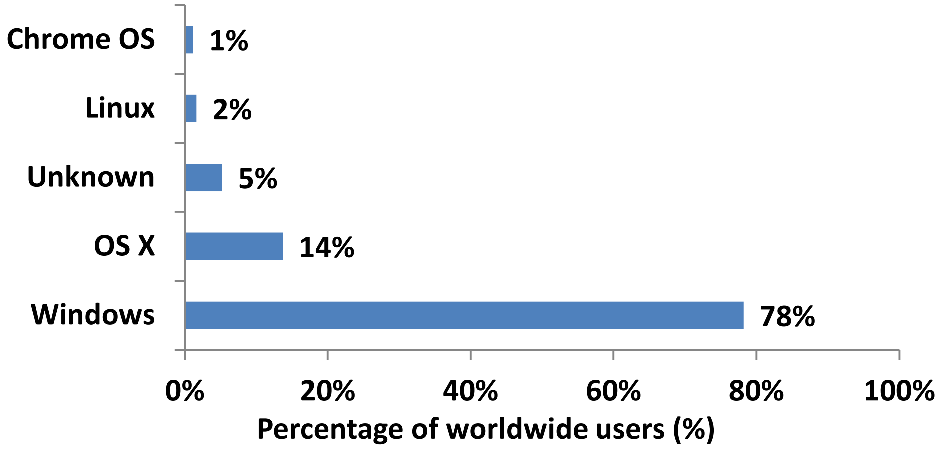


Figure 2. Worldwide users of operating systems – October 2018

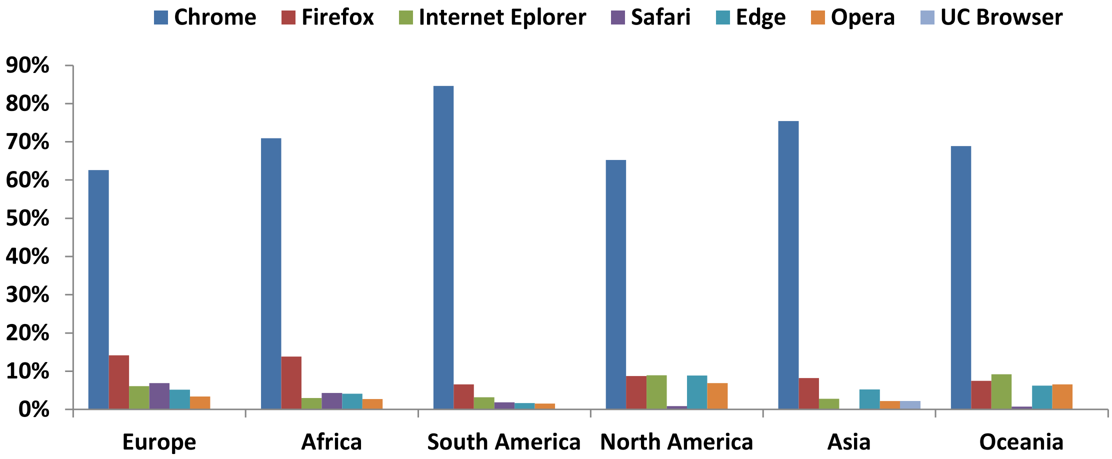


Figure 3. Use of web browsers per continent. October 2018

In addition, depending on the browser, we must take into account the compatibilities of containers and codecs with HTML5. The compatibilities between browsers and containers are shown in Table 1:

Table 1. Containers supported by HTML5 as a function of the web browsers.

Browser	Container		
	MP4	*WebM*	*Ogg*
Internet Explorer	Yes	No	No
Google Chrome	Yes	Yes	Yes
Firefox	Yes	Yes	Yes
Safari	Yes	No	No
Opera	Yes (from Opera V. 25)	Yes	Yes

Taking into account the aforementioned information, this study will consider Google Chrome, Firefox and Opera as web browsers. In addition, they are the ones that support the three video formats MP4, WebM and OGG. Regarding the operating systems, we will work on Windows10 and Windows7. Windows 8 has not been selected since the architecture of the system is very similar to the one used by Windows10.

4. Proposed test bench and tools

This section presents the scenario as well as the different video coding used to carry out our test bench.

4.1 Scenario

In order to carry out the network performance measurements, we have deployed the network shown in Fig. 4. It is composed by 2 desktop computers with very similar characteristics. Each one of them runs a different version of Windows. Transmissions are not made simultaneously to avoid bandwidth limitations. Both devices are connected to a router (192.168.0.1/24) which establishes the link between the end devices and the video server (192.168.0.10/24). To stream the videos, VLC Media Player has been used. All devices are connected using Cat6 Ethernet cable which allows up to 1Gbps transference rates.

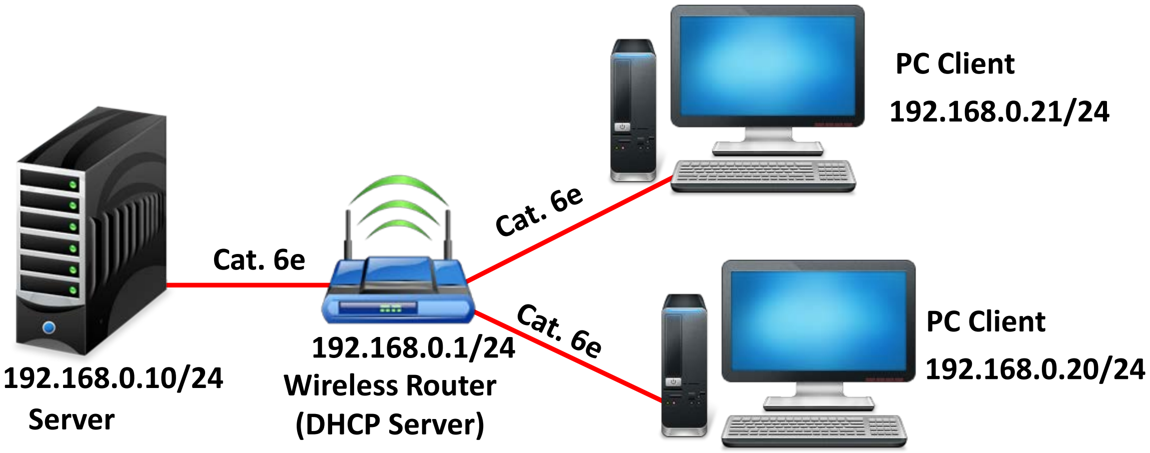


Figure 4. Scenario used to perform our tests

The hardware features of these devices are shown in Table 2.

4.2 Video used and coding performed

The video transmissions are performed using a free distribution video developed by the Blender Institute called Big Buck Bunny [27].The features of the original video are shown in Table 3.

In video transmissions through web pages, videos are traditionally used with different resolutions that are usually adapted to the different devices that request the videos. The network limitations can also be considered to select the best video. Therefore, the original video should be encoded with the different codecs discussed in Section 3. Because, not all devices present the same hardware and software features, it is also convenient to use different resolutions. In this case, the resolutions of 360p, 480p, 720p, 1080p and 4K (2160p) have been used. Table 4 shows the characteristics of the videos obtained after their coding.

Table 2. Features of equipment used in these tests.

Equipment	**Features**					
	Model	***Processor***	***Graphic***	***OS***	***Screen***	***Max. Resolution***
Server	MSI CX62 6QL	i7 6th GEN	Nvidia Geforce 940MX	Windows 10	17.3”	1366 x 768
Computer 1	Toshiba Satellite Pro c850-1j6	i5-3230M	Intel HD Graphics 4000	Windows 10	15.6”	1366 x 768
Computer 2	HP pavilion g6 2212es	i5-3210M	AMD Radon HD 7670M	Windows 7	15.6”	1366 x 768

Table 3. Characteristics of the original video in terms of size, duration, resolution and original container.

Video	**Original video features**						
	Size	***Original Container***	***Video format***	***Audio format***	***Duration***	***Overall bit rate***	***Width x Height (pixels)***
Original	85.5 MiB	MPEG-TS	AVC	MPEG Audio	56” 382ms	12.7 Mbps	4000x2250

Table 4. Formats, resolution, frame rate, bitrate and size of the videos to be transmitted.

Codec	**Features**						
	Video	***Resolution***	***FPS***	***Bitrate (KBPS)***	***Size (MIB)***	***Video Format***	***Audio Format***
None	Original	2250	60	127000	85,5	AVC	MPEG Audio
MP4		2160p	30	3950	77,4	AVC/AAC	MPEG Audio / AC-3
MP4		1080p	30	1872	26,6	AVC/AAC	MPEG Audio / AC-3
MP4		720p	30	1992	13,4	AVC/AAC	MPEG Audio / AC-3
MP4		480p	30	965	6,41	AVC/AAC	MPEG Audio / AC-3
MP4		360p	30	774	5,16	AVC/AAC	MPEG Audio / AC-3
Ogg		2160p	30	13200	88,6	Theora	Vorbis
Ogg		1080p	30	3239	21,8	Theora	Vorbis
Ogg		720p	30	1325	8,89	Theora	Vorbis
Ogg		480p	30	546	3,51	Theora	Vorbis
Ogg		360p	30	394	2,62	Theora	Vorbis
WebM		2160p	30	0,475	34	VP8	Vorbis
WebM		1080p	30	0,167	12	VP8	Vorbis
WebM		720p	30	0,055	5,69	VP8	Vorbis
WebM		480p	30	0,021	2,21	VP8	Vorbis
WebM		360p	30	0,036	2,61	VP8	Vorbis

Fig. 5 shows the video size evolution obtained for the different codecs depending on the resolution while Fig. 6 presents the bit rate as a function of the resolution after coding them.

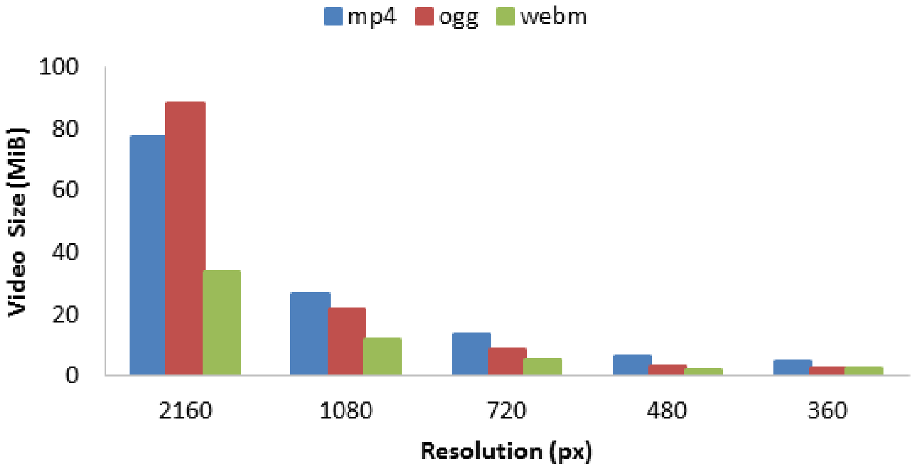


Figure 5. Video size depending on the resolution and container used

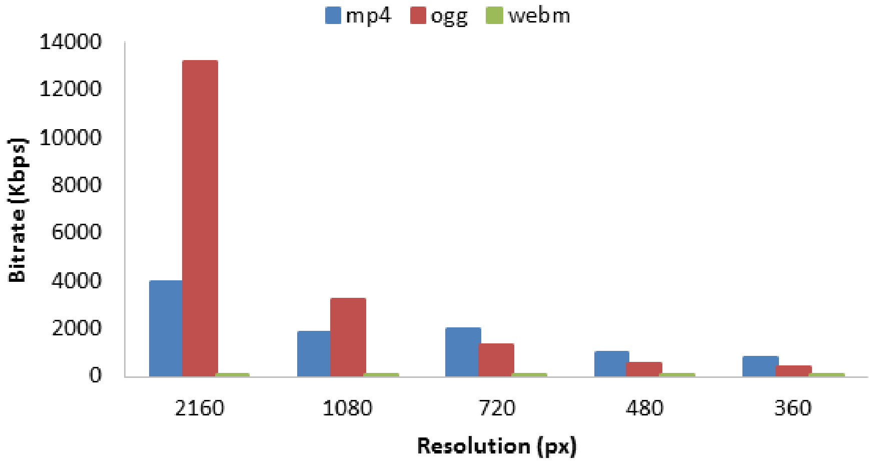


Figure 6. Bitrate depending on the resolution and container used

As we can see, the sizes of the obtained videos are similar for MP4 and OGG. Videos encoded with WebM present the smallest size reducing it almost half the size of mp4 videos. However, the bit rate in which the videos are encoded offers very different results. On the one hand, MP4 follows a linear trend while the bit rate employed by OGG is exponentially greater with the increase of the resolution. The bit rate for WEBM is so small with respect to the other containers.

5. Results

This section collects and explains the results obtained during the different terest we performed. The results have been divided for both cases (i.e. when Windows 7 is used and when Windows 10 is used) and the analyzed parameters have been Delay, bits error rate, data transfer rate and throughput.

5.1 Results for Windows 7

5.1.1 Delay in Windows 7

Fig. 7 shows the Delay in ms as a function of the video size and its container for Google Chrome, Firefox and Opera web browsers in Windows 7 for a 1000BaseT network.

As we can see, the Google Chrome browser presents, in all cases, the highest delay (delay greater than 700 ms), being the MP4 container the one with the biggest delay (766 ms). For Firefox browser, the delay behaves in a predictable manner with an increasing exponential tendency depending on the size of the video. The delay experienced by the MP4 container is the biggest one followed by OGG and WEBM. The delay obtained when Opera browser is used maintains the aforementioned trend. MP4 (32 ms) is the container with the highest delay, followed by OGG and WEBM which recorded values of 28 ms and 16 ms, respectively.

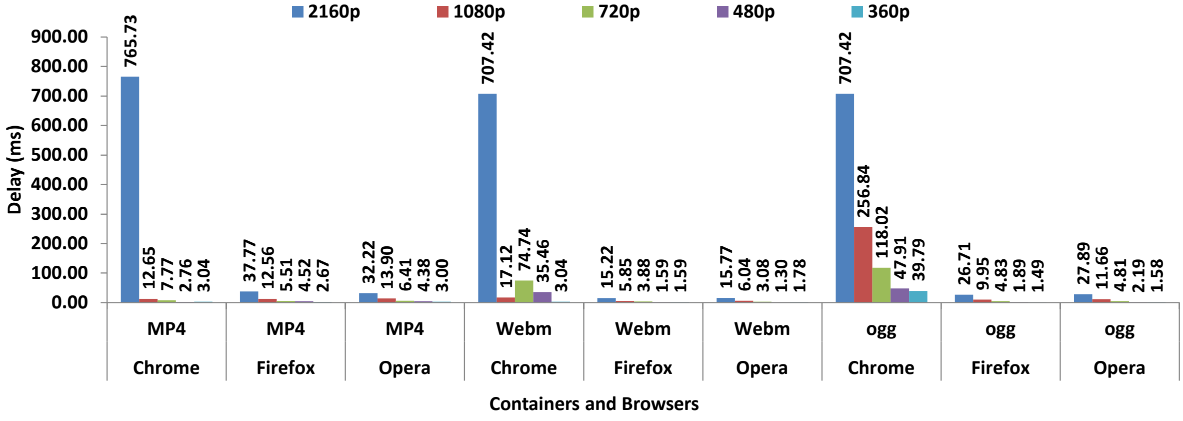


Figure 7. Delay as a function of the video resolution and containers for Google Chrome, Firefox and Opera browsers in Windows 7 for 1000BaseT networks.

In general, we can see that there is a tendency in terms of the delay experienced by the treated containers, with the delay experienced by MP4 always being higher, followed by OGG and by WEBM, which is the one with the least delay in all cases.

5.1.2 Bits Error in Windows 7

When analyzing the evolution of the transmission error (see Fig. 8), the results lack a clear trend of progression. Based on the data shown in Fig. 8, we observe that the transmission error remains around 5-7% in the most combinations of browsers and containers. We can only highlight the 2 high peaks that occur when Google Chrome browser is used with the MPA4 and WebM containers and when Opera browser is used with the MP4 container. These peaks may be due to the casuistry of the measured scenario, but we can clearly see that the average error rate is around 6%.

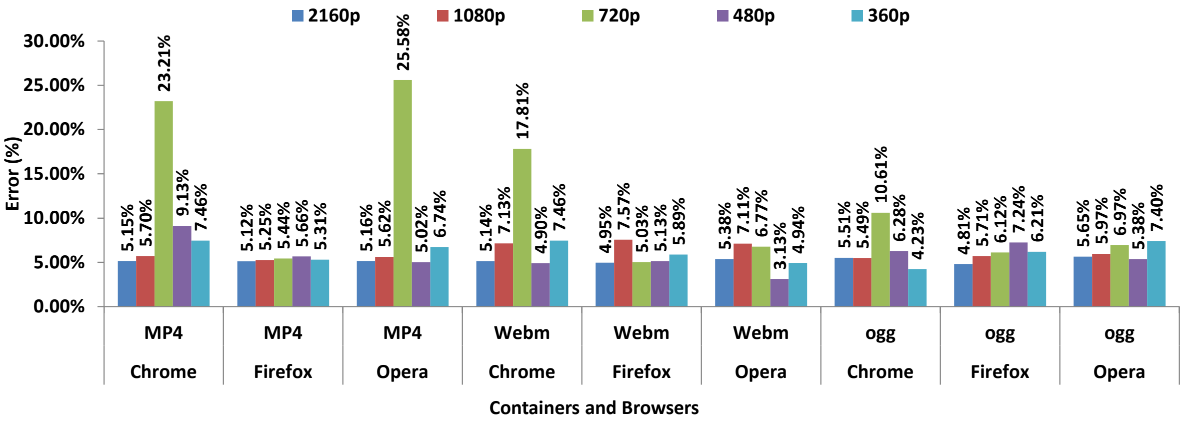


Figure 8. Error rate as a function of the video resolution and containers for Google Chrome, Firefox and Opera browsers in Windows 7 for 1000BaseT networks.

Therefore, from the results we can conclude that the error rate cannot be used to determine which browser presents better performance.

5.1.3 Data Transfer rate in Windows 7

Regarding to the data transfer rate for Google Chrome browser (see Fig. 9), we observe it oscillates between the 6.6 Mbps and 23.38 Mbps. We can also see that there is a big difference between the use of OGG and the rest of containers (MP4 and WebM) which present a data transfer rate almost 4 times higher than OGG for videos of high resolutions. Firefox reaches data transfer rates higher than 60Mbps when using the MP4 and OGG containers while WebM registers values around 6Mbps. Finally, the data transfer rate for Opera ranges between 6 Mbps (when the WebM container is used) and 23.6 Mbps when the OGG container is used.

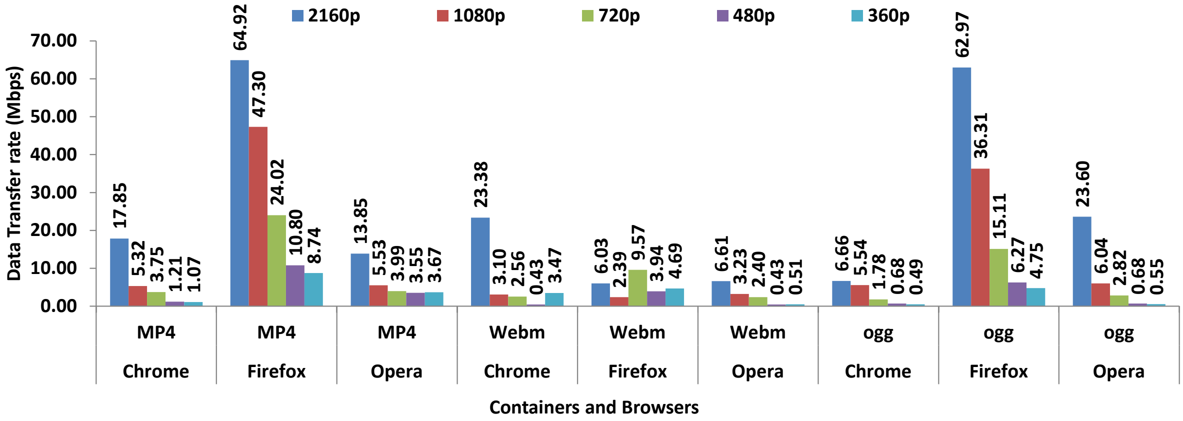


Figure 9. Data transfer rate as a function of the video size and containers for Google Chrome, Firefox and Opera browsers in Windows 7 for 1000BaseT networks.

From these results, we observe that in general the data transfer rate used by MP4 is the highest followed by WEBM and OGG whose data transfer rates are practically the same. This information is interesting since the videos encoded with OGG present a size very similar to those encoded with MP4. However, the videos encoded with OGG are transmitted using a lower bandwidth. The behavior of WebM is just the opposite, i.e., the videos encoded under WEBM present a size is smaller than the rest of videos and they require the highest data transfer rate.

5.1.4 Throughput in Windows 7

Fig. 10 shows the Throughput in Mbps as a function of the video size and its container for Google Chrome, Firefox and Opera web browsers in Windows 7 for a 1000BaseT network. Analyzing the throughput for the Google Chrome browser, we can see that the highest throughput value is recorded for the WEBM container for high resolution videos, reaching values of 22.17 Mbps. However, for low resolution videos, the most efficient container is OGG with values of 0.47Mbps for 320p videos. For Firefox, WEBM is the container with the lowest throughput in all cases. We must highlight that MP4 is the container with the highest throughput followed by OGG and WEBM. On the other hand, in the first two cases, the throughput generated is much higher for high resolution videos (7MBps) than in low resolutions (around 1MBps).WEBM presents a constant value around 1MBps for all video sizes. Finally, Opera, using the OGG container, presents higher throughput than MP4 container for high resolution videos.

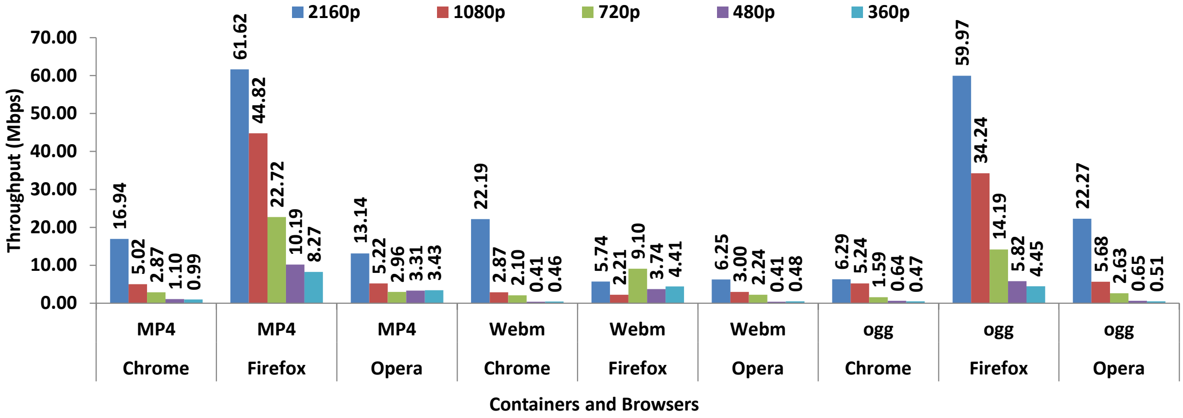


Figure 10. Throughput as a function of the video size and containers for Google Chrome, Firefox and Opera browsers in Windows 7 for 1000BaseT networks.

5.2 Results for Windows 10.

5.2.1 Delay in Windows 10

Fig. 11 shows the delay as a function of the video size and its container for Google Chrome, Firefox and Opera web browsers in Windows 7 for a 1000BaseT network. The value of delay recorded when using Google Chrome follows an exponential trend and it is the browser that registers the highest delay value for all resolutions. The delay of 4k resolution videos in Google Chrome can reach values higher than 800 ms.

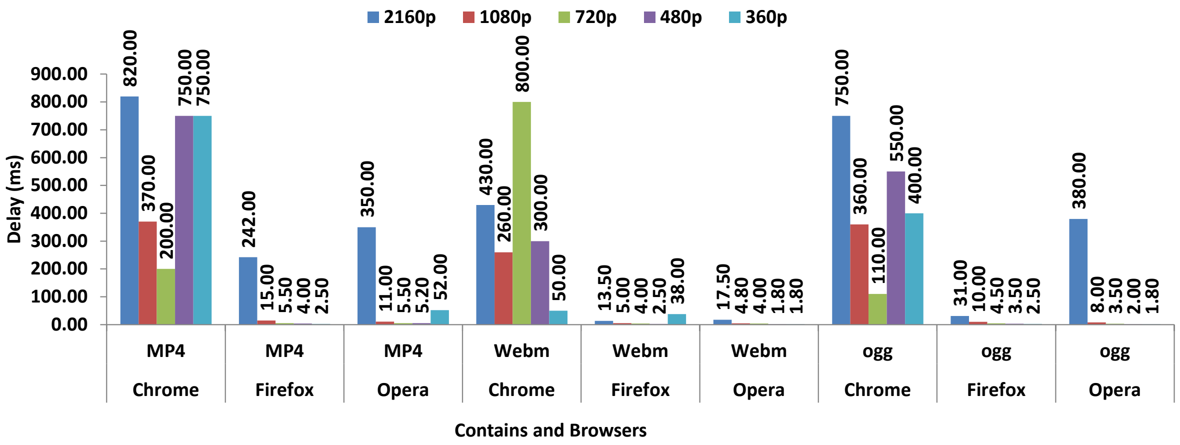


Figure 11. Delay as a function of the video size and containers for Google Chrome, Firefox and Opera browsers in Windows 10 for 1000BaseT networks.

For Firefox, the delay follows an exponential trend, reaching values close to 242 ms for 4k resolution videos (using the MP4 container) and values lower than 15 ms for the rest of resolutions and containers. On the other hand, it is interesting to note that the delay for Firefox and Opera is practically identical for all containers, and it multiplies the value for Google Chrome up to 20 times higher compared to 4k.

5.2.2 Error rate in Windows 10

Fig. 12 shows the percentage error rate when using the 3 browsers under study. The container that has generated the highest error rate is Ogg over the Opera browser and 4K videos. The use of Chrome with the MP4 container also has a high error rate, compared with the other options, with values of error close to 1.5% for 4K videos. The error presented in Firefox with the MP4 container is practically nil, as well as for OGG and for WEBM in low qualities. When using Opera, MP4 presents an error rate below 0.25% in all qualities and OGG and WEBM ranges around 0.5% and 2% for high qualities, although low video resolutions present very small error rates.

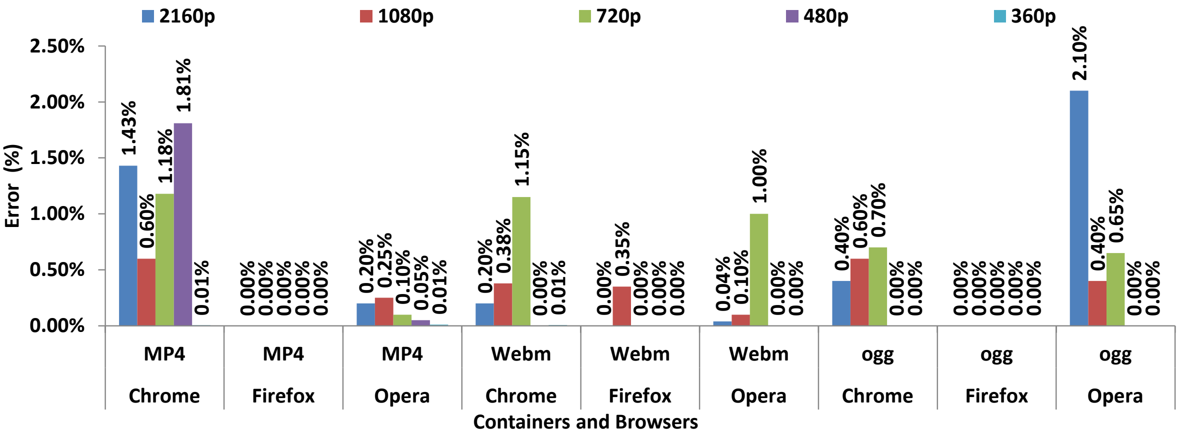


Figure 12. Error rate as a function of the video size and containers for Google Chrome, Firefox and Opera browsers in Windows 10 for 1000BaseT networks.

5.2.3 Data Transfer rate in Windows 10

Fig. 13 shows the data transfer rate as a function of the video size and containers for Google Chrome, Firefox and Opera browsers in Windows 10 for 1000BaseT networks. Regarding the codec and the transfer rate (Fig. 13), we see that there is a big difference between OGG, MP4 and WEBM for high resolution videos in Firefox. As Figure 13 shows, MP4 is the container with the highest data transfer rate with values of 192 Mbps for video of 4k, followed by WEBM and OGG. However, this difference becomes much less significant in the rest of browsers.

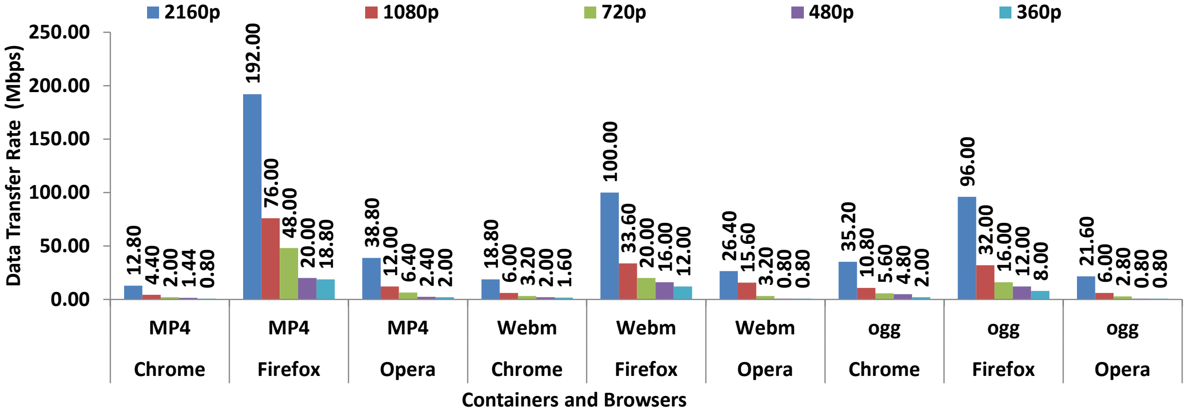


Figure 13. Data transfer rate as a function of the video size and containers for Google Chrome, Firefox and Opera browsers in Windows 10 for 1000BaseT networks.

For Firefox, MP4 presents the highest data transfer rate followed by OGG and WEBM, which has a very similar rate. Firefox presents a data transfer rate, for 4k videos, up to 192 Mbps for MP4, 100Mbps for WebM and 96 Mbps for Ogg. This is because Firefox uses a larger transmission window size, which makes the size of the packets larger and the total transmission time decreases considerably. The data transfer rate for Opera ranges between 21.6 and 38.8 Mbps for 2160p videos while for videos with lower resolutions (320p, 480p and 720p) the data transfer rates do not exceed 6.4Mbps. Google Chrome presents a behavior very similar to Opera reaching the lowest data transfer rates when using MP4 container for all resolutions.

5.2.4 Throughput in Windows 10

Fig. 14 shows the throughput in Mbps as a function of the video size and containers for Google Chrome, Firefox and Opera browsers in Windows 10 for 1000BaseT networks. As we can see, the throughput for Google Chrome is higher in the case of OGG for high resolution videos and similar to MP4 up to 1080p. These results match with the analysis of the data transfer rate since both values must be similar due to low error rate. In the case of Firefox, the throughput is higher (96 Mbps) in the case of MP4 for 2160p videos followed by Ogg (93.6 Mbps) and slightly lower for low resolutions videos. The use of WEBM together with Firefox presents the lowest throughput with maximum values of 12Mbps for 2160p videos. Finally, the throughput for Opera is higher in the case of OGG for high resolution videos (20.8Mbps) and very similar for the MP4 container up to 1080p with maximum values of 4.8Mbps.

WEBM the container that presents lower throughput, with maximum values of 5.6 Mbps for 4k videos.

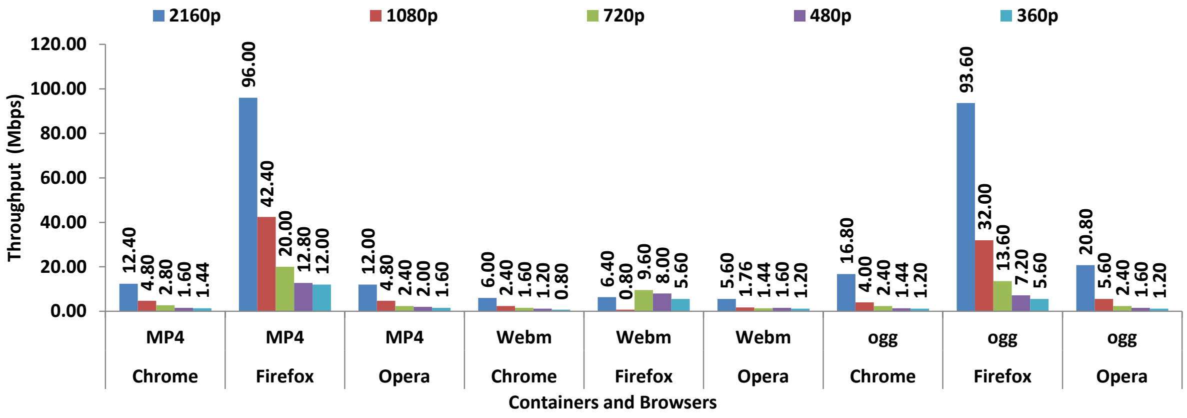


Figure 14. Throughput as a function of the video size and containers for Google Chrome, Firefox and Opera browsers in Windows 10 for 1000BaseT networks.

6. Conclusion and Future Work

The improvement in the network capacity and the placing on the market of new devices has generated the appearance of new codecs and containers for the efficient transmission of videos with many purposes, among which the teaching aims stand out. The wide variety of existing devices on the current market makes necessary to adapt these videos for achieving the best levels of QoS and QoE. Therefore, in this article we have presented a practical study that analyzes network performance when videos processed with different containers for reproducing in a web browser based on HTML5. The videos have been reproduced using a personal computers running Windows 7 and Windows 10 operating systems. After carrying out our tests, we can draw the following conclusions:

On the one hand, when we work on a desktop computer with Windows 7 and videos are reproduced through Google Chrome browser, MP4 presents some problems when decoding the video at the client side. Regarding to the rest of containers, Ogg is presented as the container with the lowest data transfer rate and throughput compared to WEBM. However, WEBM is the preferred container for transmitting low resolution videos. For the Firefox browser, WEBM and OGG contain a low data transfer rate compared to MP4. In addition, WEBM presents less delay, less utilization and lower throughput, especially for cases of high quality videos where its behavior is excellent. Finally, for the Opera browser, the data transfer rate of the containers is similar, being MP4 the container with the highest data rate. In terms of delay, errors and throughput, WEBM is the container with the best performance for all resolutions, followed by MP4 for high quality videos and OGG for low quality videos.

On the other hand, when we work on a desktop computer with Windows 10 and videos are reproduced through Google Chrome browser, MP4 is the container with greater delay, higher error rate and lower data transfer rate, although its value of throughput is lower compared to other containers. WEBM presents the best results in terms of delay, error and use of the channel. For the Firefox browser, all the containers present a low error rate, although WEBM is slightly worse for high quality videos. WEBM presents better statistics regarding the rate of data transfer rate and delay, and a stable throughput for all video resolutions. Ogg presents very similar statistics to WEBM in terms of data transfer rate for low quality videos. Lastly, for the Opera browser, MP4 has a stable and very low error rate especially for high resolution videos. WEBM again gives the best results in terms of delay, utilization and throughput, but presents a higher error rate when transmitting high quality videos.

As future works, we would like to extend these tests to other kind of devices, including smartphones and tablets, and operating systems. We would design an intelligent algorithm for real-time transcoding [28] for distributing video in different media, or applications based on the use of HTML5 for e-learning or video conference purposes [29].

Acknowledgement

This work has been partially supported by the European Union through the ERANETMED (Euromediterranean Cooperation through ERANET joint activities and beyond) project ERANETMED3-227 SMARTWATIR and by the “Ministerio de Educación, Cultura y Deporte”, through the “Convocatoria 2016 - Proyectos I+D+I - Programa Estatal De Investigación, Desarrollo e Innovación Orientada a los retos de la sociedad” (Project TEC2016-76795-C6-4-R) and through the “Convocatoria 2017 - Proyectos I+D+I - Programa Estatal de Investigación, Desarrollo e Innovación, convocatoria excelencia” (Project TIN2017-84802-C2-1-P).

References

[1] Aronowitz, S. “Technoscience and cyberculture”. Francis and Taylor Group, 2014. 1st Edition.

[2] Oshima, K. and Muramatsu, Y., “Current situation and issues related to ICT utilization in primary and secondary education”. Fujitsu Scientific & Technical Journal, 2015, Vol. 51, Issue 1, Pp.3-8.

[3] Sendra, S.; Jimenez, JM.; Parra Boronat, L.; Lloret, J. “Blended Learning in a Postgraduate ICT course”. In 1st International Conference on Higher Education Advances (HEAD' 15). June 24-26, 2015. Valencia, Spain. Pp, 516-525.

[4] Bartolomé-Pina, A-R, Steffens, K. “¿Son los MOOC una alternativa de aprendizaje?= Are MOOCs Promising Learning Environments?”, Comunicar, Vol. 22, Issue 44, Pp. 91-99. 2015. http://dx.doi.org/10.3916/C44-2015-10

[5] Estepa, A. J., Estepa, R., Vozmediano, J., Carrillo, P., “Dynamic VoIP codec selection on smartphones”, Network Protocols and Algorithms, Vol. 6, Issue 2, Pp. 22-37. 2014. https://doi.org/10.5296/npa.v6i2.5370

[6] Abdullah, M. T. A., Lloret, J., Canovas Solbes, A., García-García, L. Survey of Transportation of Adaptive Multimedia Streaming service in Internet. Network Protocols and

Algorithms, Vol. 9, No. 1-2, pp. 85-125, 2017. https://doi.org/10.5296/npa.v9i1-2.12412

[7] Fabrega, L., Jove, T., "A review of the architecture of admission control schemes in the Internet", Network Protocols and Algorithms, Vol. 5, Issue 3, Pp.1-32, 2013. https://doi.org/10.5296/npa.v5i3.3726

[8] The 13th annual Complete VNI Forecast. Explore global and regional IP traffic growth and trends (2017 – 2022). In Cisco website. Available at: https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html. [Last access: November 17, 2018]

[9] Secades, V. A., Arranz, O., "Big data & eLearning: A binomial to the future of the knowledge society". International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 3, Issue 6, Pp. 29-33, 2016. http://10.9781/ijimai.2016.364

[10] Anthes, G., "HTML5 leads a web revolution". Communications of the ACM, Vol. 55, Issue 7, Pp. 16-17, 2012. http:// 10.1145/2209249.2209256

[11] Ohwovoriole E., Andreopoulos Y., "Comparison of Google / WebM VP8, AVC / H. 264 and TMuC HEVC, C TMuC ", In Proc. London Communications Symposium (LCS2010), Sep 10, 2010, London (UK). Pp. 1-4.

[12] Ohm J. R., Sullivan G. J., Schwarz H., Tan T. K., Wiegand T., "Comparison of the coding efficiency of video coding standards—including High Efficiency Video Coding (HEVC)", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, Issue 12, pp. 1669 - 1684, 2012. https://doi.org/10.1109/TCSVT.2012.2221192

[13] Grois D., Marpe D., Mulayoff A., Itzhaky B., Hadar O., "Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders", 2013 Picture Coding Symposium (PCS 2013), December 8-11, 2013, San Jose, CA, USA. https://doi.org/10.1109/PCS.2013.6737766

[14] López-Herreros A., Canovas A., Jimenez J. M., Lloret J., "A New IP Video Delivery System for Heterogeneous Networks using HTML5", 2015 IEEE International Conference on Communications (ICC), 8-12 June 2015, London (UK), https://doi.org/10.1109/ICC.2015.7249451

[15] Mateos-Cañas, I., Sendra,S., Lloret, J., Jimenez, J.M., "Autonomous video compression system for environmental monitoring", Network Protocols and Algorithms, Vol 9, No 1-2, Pp. 48-70. 2017. Doi: https://doi.org/10.5296/npa.v9i1-2.12386

[16] Wang, Y.-K., Even, R., Kristensen, T., Jesup, R., "RTP Payload Format for H.264 Video (RFC 6184)". (2001) In Internet Engineering Task Force (IETF) web site. Available at: https://tools.ietf.org/html/rfc6184 [Last access: Nov. 17, 2018]

[17] Pfeiffer, S., T"he Ogg Encapsulation Format Version 0 (RFC 3533)", (2003). In Internet Engineering Task Force (IETF) web site. Available at: https://tools.ietf.org/html/rfc3533 [Last access: Nov. 17, 2018]

[18] Walleij, L., "The application/ogg Media Type (RFC 3534)", 2003, In Internet Engineering Task Force (IETF) web site. Available at: https://tools.ietf.org/html/rfc3534 [Last access: Nov. 17, 2018]

[19] Project Gutenberg. Available at: https://www.gutenberg.org/[Last access: Nov. 17, 2018]

[20] Theora features. Available at: https://www.theora.org/doc/Theora.pdf [Last access: Nov. 17, 2018]

[21] Barbato, L., "RTP Payload Format for Vorbis Encoded Audio (RFC 5215)", 2008, In Internet Engineering Task Force (IETF) web site. Available at: https://tools.ietf.org/html/rfc5215 [Last access: Nov. 17, 2018]

[22] WebM codec features Project. In WebMproject web site. Available at: https://www.webmproject.org/docs/container/[Last access: Nov. 17, 2018]

[23] WebM Project. In WebMproject web site. Available at: https://www.webmproject.org/[Last access: Nov. 17, 2018]

[24] Bankoski, J., Koleszar, J., Quillio, L., Salonen, J., Wilkins, P., "VP8 Data Format and Decoding Guide (RFC 6386)", 2011, In Internet Engineering Task Force (IETF) web site. Available at: https://tools.ietf.org/html/rfc6386 [Last access: Nov. 17, 2018]

[25] Statistics of worldwide operating systems users. In Statcounter webs site. Available at: http://gs.statcounter.com/os-market-share/desktop/worldwide [Last access: Nov. 17, 2018]

[26] Statistics of web browser users. In Statcounter webs site. Available at: http://gs.statcounter.com/browser-market-share/desktop/worldwide [Last access: Nov. 17, 2018]

[27] Big Buck Bunny video. In Blender Foundation web site. Available at: https://peach.blender.org/download/ [Last access: Nov. 17, 2018]

[28] Mowafi, M. Y., Awad, F. H., Al-Batati, M. A. "Opportunistic Network Coding for Real-Time Transmission over Wireless Networks", Network Protocols and Algorithms, Vol 5, No 1 (2013). Pp. 1-19. https://doi.org/10.5296/npa.v5i1.2303

[29] Atenas, M., Sendra, S., Garcia, M., Lloret, J., "IPTV Performance in IEEE 802.11 n WLANs", IEEE Global Communications Conference (IEEE Globecom 2010), Miami (USA), December 6 – 10 , 2010. Pp. 929-933. 10.1109/GLOCOMW.2010.5700461

Copyright Disclaimer