Ad Hoc Networks 136 (2022) 102951

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

DASH Streaming traffic influence over energy efficient ethernet to improve energy savings

Tito Raúl Vargas *, Juan Carlos Guerri, Pau Arce

Institute of Telecommunications and Multimedia Applications (iTEAM), Universitat Politècnica de València, Spain

ARTICLE INFO

Keywords:
Network performance evaluation
Energy efficient ethernet
Multimedia traffic
IEEE 802.3az
Adaptive video streaming
DASH

ABSTRACT

Dynamic Streaming over HTTP (DASH) is the main standard used in online video streaming services, given that more than 1.2 billion pay subscribers around the world use this standard. This fact entails billions of streaming connections between video servers and client displays. These devices involved in the streaming connection use an Ethernet Interface Card that consumes energy. In order to reduce the energy consumption, IEEE proposed the 802.3az Energy Efficient Ethernet standard, with a mechanism to make the network card change to a low power consumption mode when it is not in transmission mode. This behavior will be beneficial for services where traffic is sent in bursts, for instance video packet bursts like in video streaming over Real Time Transport Protocol in IPTV or the widely used DASH standard. Therefore, in this study the Ethernet traffic pattern when transmitting online video content using DASH is characterized in order to analyze the efficiency of the IEEE 802.3az standard under this video streaming scenario, and to verify the convenience of activating this energy saving alternative at the network interface of billions of client devices. The experiments have been conducted using a test-bed consisting of a full DASH streaming architecture, comparing different video segment sizes and changing the available bandwidth during the experiments in different scenarios in order to analyze the effect of the DASH content segment size on the Ethernet traffic pattern to identify the trade-off between energy efficiency, the energy savings, and the impact on the performance of the dynamic adaptation on the video streaming and reproduction.

1. Introduction

The high demand that multimedia streaming services are experiencing nowadays makes that a great number of simultaneous connections has to be established between content servers and players at client side. This fact causes that at least a pair of Network Interface Cards (NIC) must be used for each video/audio stream, one at the server and one at the client. Particularly, the energy consumption at NICs have been found not to be negligible [1,2], therefore, finding alternatives to reduce energy consumption at NICs will benefit the efforts to reduce the total energy consumption of the devices involved in the streaming chain.

In this sense, to improve the energy savings specifically at network interface cards, the IEEE 802.3az Energy Efficient Ethernet (EEE) standard has been defined. The objective of EEE is to save energy by setting the NIC in low power mode when it has not any ethernet frame to transmit or receive [3]. Regarding the effectiveness of activating the low power mode in the NIC, the energy efficiency has been found to depend on the ethernet traffic pattern. Therefore, ethernet traffic patterns like those that shows transmissions of ethernet frames in bursts have been found to impact positively on the EEE energy efficiency because the time the NIC stays in high power mode (active) is large enough to compensate the energy consumption of putting into sleep and waking up the ethernet card [2,4].

Based on the aforementioned conclusion, the energy efficiency of the IEEE 802.3az has been analyzed when the card is used to transmit ethernet frames that convey multimedia traffic in managed networks like IPTV [5]. The studied multimedia traffic was H.264 encoded video contained in MP4 files and encapsulated using Real Time Transport Protocol (RTP) over User Datagram Protocol (UDP) over IP packets and transmitted over Ethernet frames. As a result, MP4/RTP/UDP/IP/Ethernet traffic was characterized to obtain these multimedia H.264 traffic pattern at application level and Ethernet level. Once those traffic patterns are obtained, this type of multimedia streams was found to generate ethernet bursts and it was demonstrated that it favors the EEE energy efficiency and the corresponding energy savings at the NIC.

Moreover, nowadays most of video/audio streaming applications are services over the top (OTT). OTT means that all the involved services run over the Internet, which is an unmanaged network. Additionally, the

* Corresponding author.
E-mail address: tivarher@upv.es (T.R. Vargas).

https://doi.org/10.1016/j.adhoc.2022.102951
Received 20 June 2022; Accepted 19 July 2022
Available online 30 July 2022

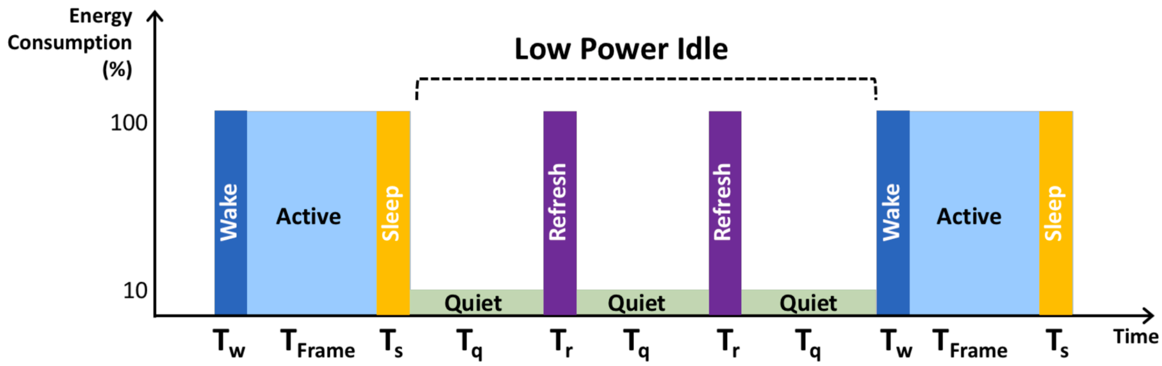


Fig. 1. Energy consumption and time periods in high and low power modes in 802.3az.

multimedia content is streamed over the HyperText Transport Protocol (HTTP)/Transmission Control Protocol (TCP) sessions, being the Dynamic Adaptive Streaming over HTTP (DASH), the standard widely used [6]. Streaming services such as Netflix, Amazon Prime, HBOMAX, among others, use the DASH standard for streaming adaptive content to compensate variable bandwidth during the whole movie transmission and the IP best-effort service that the Internet offers. For that reason, in this paper the DASH traffic pattern at application level and ethernet level is analyzed in order to identify the relationship that can be established with IEEE 802.az in the context of EEE energy efficiency and energy savings. In this work, the activities to study and analyze the influence of DASH traffic pattern on the EEE energy efficiency follows the next methodology.

Firstly, a DASH application test-bed is set up, involving all elements of a streaming service architecture: multimedia content, server and client entities, protocol stack and message encapsulation, network, and the ethernet network interfaces. The test-bed is complemented with a group of software scripts and tools. Using the test-bed makes possible to encode and adapt the videos for transmission via DASH according to the need for traffic analysis, such as segment size, quality and resolution. It also allows to run DASH streaming experiments in the same way as they would be performed in a real environment.

Secondly, multiple experiments using different scenarios are run to capture traces in order to analyze and determine the ethernet traffic pattern driven by multimedia content transmitted using DASH. Then, a thorough trace analysis is carried out to obtain important parameters that characterize the DASH traffic, and the related ethernet traffic is obtained running a set of specific scripts designed for that aim.

Finally, the efficiency of the IEEE 802.3az-EEE standard is calculated for each scenario to conclude over the convenience of the standard for energy savings using DASH.

2. Background

In order to understand and analyze the DASH streaming traffic influence over Energy Efficient Ethernet to improve energy savings, it is important to know the 802.3az standard, its characteristics, operation and weak points, and thus be able to relate it to the DASH transmission system. Having that into consideration, this section is presented as a review of the definition of the Energy Efficient Ethernet (EEE) standard, the IEEE 802.3az, and the Dynamic Adaptive Streaming over HTTP (DASH) standard. Regarding EEE, conclusions from previous studies are also presented taking into account the energy efficiency in general and energy efficiency regarding multimedia streams, being such studies a guideline of the present study.

2.1. Energy efficient ethernet (EEE) and multimedia streams

As aforementioned, to reduce energy consumption at Ethernet NICs the IEEE 802.3az Energy Efficient Ethernet standard has been proposed.

Table 1
Energy Efficient Ethernet time parameter definitions.

Symbol	Name	Definition
T_w	Wake-up time	Time needed by the card to go to active mode from the low power idle mode, so the card is ready for transmission.
T_s	Sleep time	Time needed by the card to go from the active mode to the low power idle mode, so the card can start saving energy.
T_r	Refresh signal time	Time needed by the card to exchange signaling for synchronization between receivers, signal needed during idle times (low-power mode).
T_q	Maximum quiet period	Time where the card is in the low-power idle mode, no transmissions. A T_r is required when the low-power idle mode exceeds the T_q.
T_{Frame}	Frame transmission time	Time needed to transmit an Ethernet Frame.

Table 2
η_{Frame} and minimum T_w, T_s and T_{Frame} for different link speeds in IEEE 802.3az.

Protocol	Min T_w (µs)	Min T_s (µs)	T_{Frame} (1500B) (µs)	η_{Frame}	T_{Frame} (150B) (µs)	η_{Frame}
100BASE-TX	30	200	120	34.3%	12	4.9%
1000BASE-T	16	182	12	5.7%	1.2	0.6%
10GBASE-T	4.48	2.88	1.2	14.0%	0.12	1.6%

The main function of the proposed standard is to make the network interface card commute to low-power consumption mode (low-power idle) when there are not ethernet frames to transmit or receive. In this mode, the NIC reduces energy consumption to 10% of the high-power transmission mode. Fig. 1 depicts the operation and energy consumption of the Energy Efficient Ethernet standard, with the abbreviations defined in Table 1.

The standard defines the energy efficiency in terms of the ethernet interface card transmission rate and the ethernet frame size [3]. The ethernet frame efficiency η_{Frame} is defined by the relationship between the ethernet frame transmission time and the time needed to change from low power mode to high power mode and back, every time the card needs to transmit a frame. Eq. (1) describes the ethernet frame efficiency, which relates the T_W wake-up time, T_{Frame} frame transmission time (active) and T_s sleeping time.

$$\eta_{Frame} = T_{Frame}/T_W + T_{Frame} + T_s \quad (1)$$

As a reference, in a 1000BASE-T ethernet link, the frame efficiency η_{Frame} is 0.6% for a frame transmission time of 1.2 µs with a frame size of 1500 Bytes, whereas a frame transmission time of 12 µs with a frame size of 150 Bytes results in a frame efficiency of 5.7% . Table 2 describes the

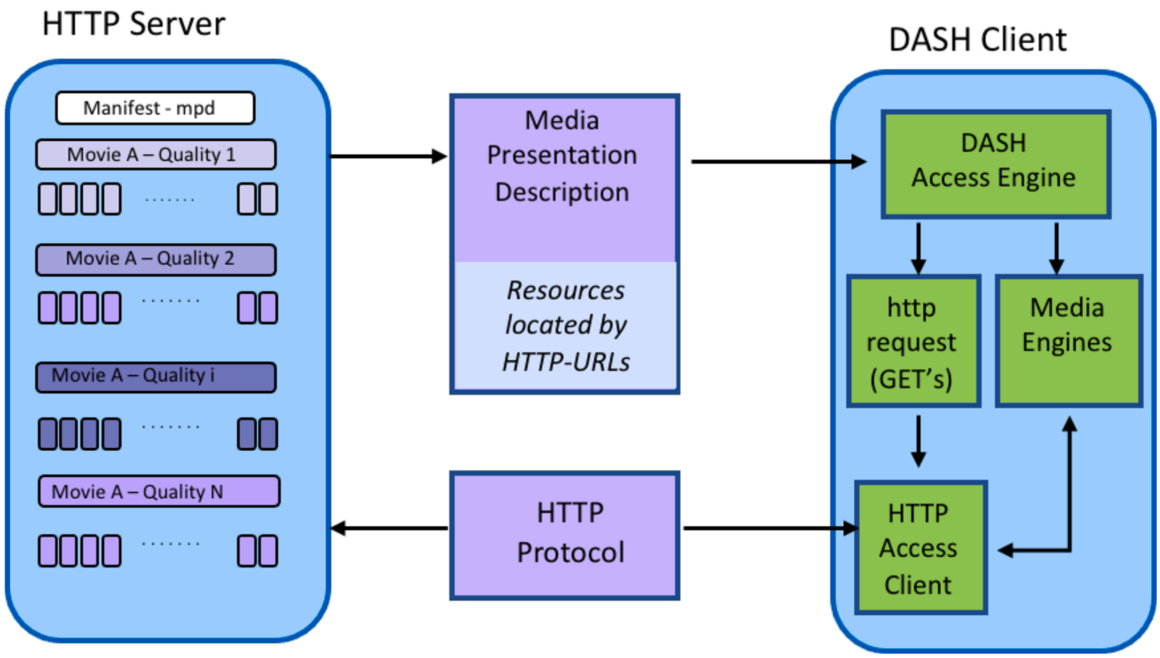


Fig. 2. Dynamic Adaptive Streaming over HTTP application and service architecture.

frame efficiency of frames with size 1500 Bytes and 150 Bytes for different link transmission speeds (100BASE-TX, 1000BASE-T and 10GBASE-T).

Eq. (1) entails that the fact of changing to low power mode is not efficient because the time needed to transmit a frame is short compared with sleeping and waking times, making the EEE standard impractical, as could be seen in Table 2. Therefore, other studies have concluded that transmitting a burst of ethernet frames in the high power mode rather than one frame results in better efficiency [2,4]. Thus, the burst efficiency η_{Burst} is defined as in Eq. (2), where T_{Burst} is the ethernet burst transmission time that depends on the aggregated number of frames.

$$\eta_{Burst} = T_{Burst}/T_W + T_{Burst} + T_s \quad (2)$$

Regarding multimedia streams and its relationship with the Energy Efficient Ethernet standard, a previous study has calculated the energy efficiency when ethernet traffic conveys multimedia streams [5]. That study was carried out by means of setting up a test-bed to characterize the traffic generated by an IPTV video server that streams H.264 videos encapsulated in MP4/RTP/UDP/IP over Ethernet. The set-up considers a managed IP network scenario since it is the common situation in IPTV. Additionally, the video traffic profile could be reviewed in detail for different scenarios of video quality [5]. The IPTV traffic analysis shows that in this case the Ethernet traffic presents a bursty nature, thus because there is a relationship between the video Group of Pictures (GoP) (or group of video frames) encoding and transmission mechanism, and the Ethernet Bursts actually transmitted. In terms of efficiency, but taking now into account the burst efficiency, a GoP efficiency was defined η_{GoP} as could be seen in Eq. (3).

$$\eta_{GoP} = \frac{GoPSize \ * \ E(B) \ * T_{Frame}}{T_w + GoPSize * E(B) * T_{Frame} + T_S} \quad (3)$$

Usually, the GoP size is equal to 12 video frames (one I frame and 12 P and B video frames), especially when using a frame rate of 24 fps, so each video GoP generates 12*$E(B) \approx 135kBytes$ ethernet burst that is transmitted every 12*$41.66ms = 500ms$. The main conclusion of that study of IPTV streams is that the traffic pattern of video transmissions favors the aggregation of frames to conform ethernet bursts, mainly because the video is structured in series of image frames or Group of Pictures (GoPs), leading to a GoP burtification [5]. The aforementioned aspect improves the efficiency to 98% approximately when transmitting multimedia traffic because the active time mode of the NIC is around 500 ms, which is the average time used to transmit a GoP or ethernet burst, and this time is now larger compared to the waking and sleeping times of the network card.

2.2. *Dynamic adaptive streaming over HTTP (DASH)*

The Dynamic Adaptive Streaming over HTTP standard (or simply DASH) is a widely deployed content distribution standard for unmanaged networks like the Internet. This technique implies dynamic changes of quality of the video or audio being transmitted depending on the available bandwidth between the client and server. Thus, the client player requests a specific quality (bitrate, resolution, etc.) of the same content depending on measurements of the available bandwidth and the playback buffer. A specific number of available content qualities is known beforehand by the player prior to starting the streaming process. The dynamic adaptation to the network conditions prevents playing interruptions and video frame losses.

The MPEG–DASH – ISO/IEC 23009 standard [6] provides the features and framework to offer OTT services through the Internet, like the widely known Netflix, HBO, and so on and, at the same time, allows them to coexist together avoiding congestion using the available network resources to the fullest at each specific time. Nowadays, DASH has been implemented by different players and video and audio encoding software such as Shaka Player [7] and ffmpeg encoding tool.

The DASH application architecture mainly consists of a web server, which is a software entity in charge of serving the content. The video and audio content to be streamed is hosted in this web server and using different files that represent many qualities of the same content using, for instance, MP4 containers. Each quality container file is divided in segment files or, alternatively, one single file with a hint track of segments with a specific time duration. Available qualities, segments and codec parameters are indexed in an XML file called Media Presentation Descriptor (MPD). Additionally, on the other side of this DASH architecture there is the player client, that could be implemented as an independent player or a web application built to run on a web browser. Fig. 2 presents this DASH architecture.

At this point, it is worth mentioning the fact that different player

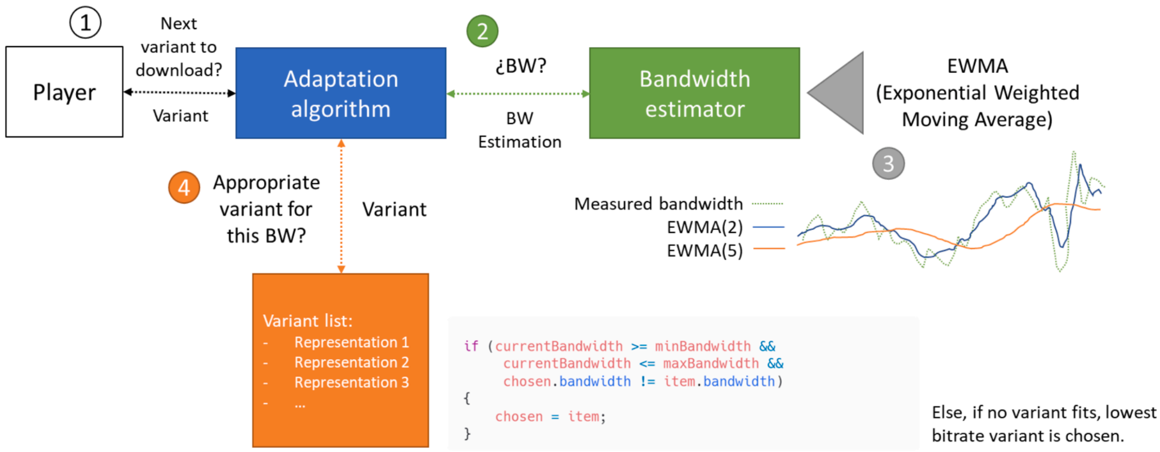


Fig. 3. Adaptation algorithm of Shaka player.

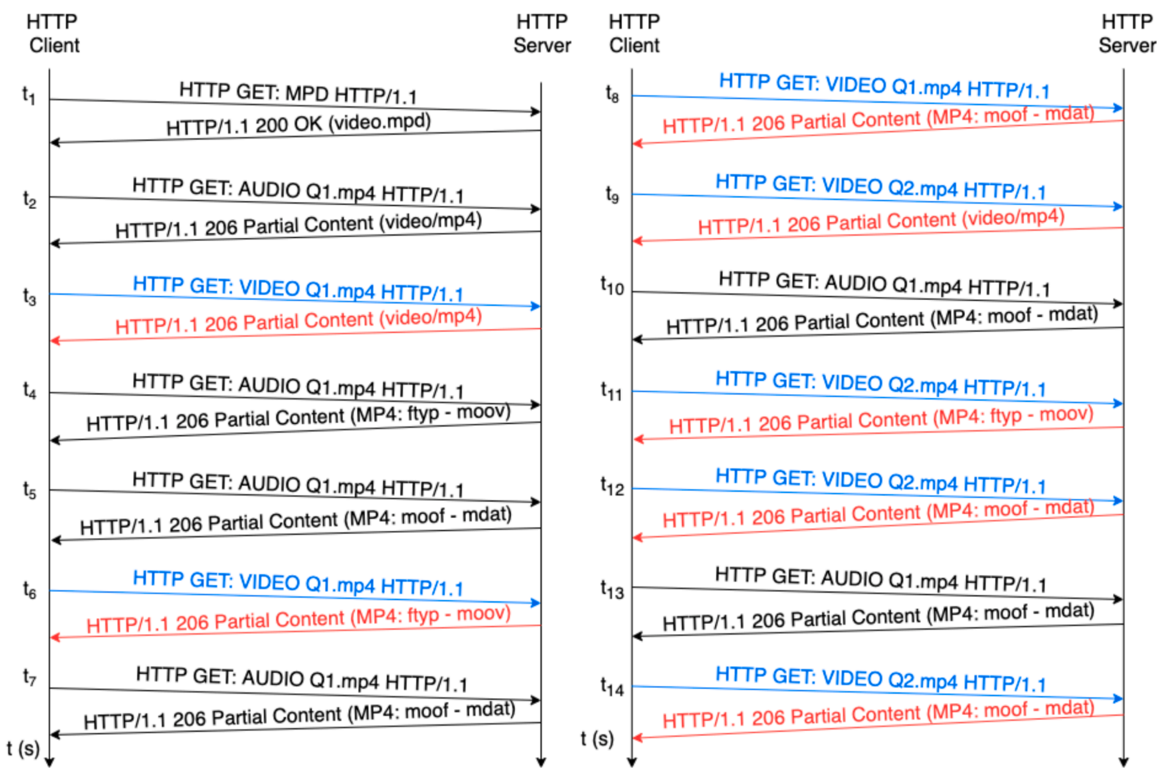


Fig. 4. DASH session protocol message exchange.

implementations can have different adaptation algorithms, so the way the player changes from a video quality to another may differ from player to player. There can be more aggressive players which change fastly to a better quality at the moment they detect an instant increase in available bandwidth. The available bandwidth is often calculated as the quotient of a segment size and the time it took to download it. Other players can show a more conservative performance, waiting for a confirmation of the increase of bandwidth long enough to be sure that they do not to fall back to a worse quality in a few seconds, since the constant change of qualities in a short space of time can also be annoying for the viewer. Particularly, the adaptation algorithm of the Shaka Player takes all this into account and shows a fast adaptation when lowering to a worse quality to avoid interruptions, but adapts slowly the streaming to changes that entail increasing the quality of the video representation or variant.

In this sense, Fig. 3 depicts the Shaka Player default adaptation algorithm. In order to perform the bandwidth estimation based on this decrease-fast, increase-slow concept, the adaptation algorithm makes use of two exponential weighted moving averages (EWMA) obtained from the measured bandwidth, with windows of 2 s and 5 s of sample history, respectively. The objective of using this two EWMA is to take the minimum of these two estimates. This should have the effect of adapting down quickly, but up more slowly. Once the bandwidth is estimated, the adaptation algorithm checks if any of the available variants fits in that bandwidth. Then, the appropriate variant is chosen, or the lowest quality representation is selected otherwise, and finally the player is informed so it can perform the quality switch.

When a DASH streaming session starts, the player first makes an HTTP GET request to download the MPD. After analyzing the contents of the MPD, the player can now perform the following HTTP GET requests to fetch the video and audio container files, specifically, the segments of the selected quality according to the available bandwidth at the exact

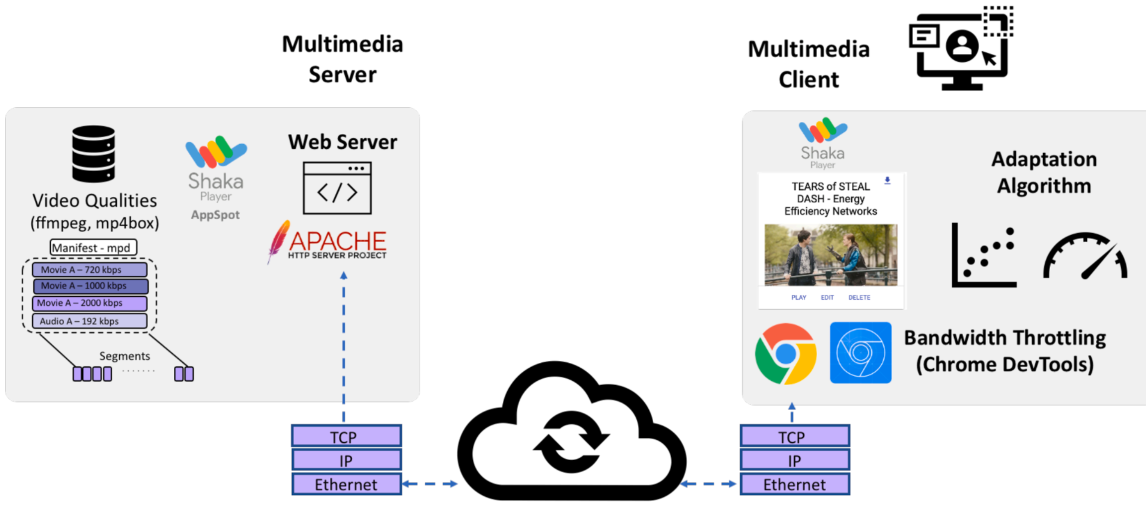
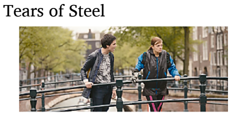


Fig. 5. Dynamic Adaptive Streaming over HTTP test-bed.

Table 3
Video available representations.

Movie	Content Type	Target Coding Bitrate (kbps)	Avg. Bitrate (kbps)	Resolution (px)	Segment Size (s)
Tears of Steel 	High Quality Video	2000	1672	2592 × 1080	4, 6, 10
	Medium Quality Video	1000	830	1728 × 720	4, 6, 10
	Low Quality Video	720	429	864 × 369	4, 6, 10
	Single Quality Audio	192	131	–	4, 6, 10

reproduction time. The player will continue getting segments at the selected quality until the content playback reaches the end. In Fig. 4, the protocol message exchange in a DASH session is presented. In particular, the HTTP GET request message to download the MPD XML file can be seen, as well as the audio segments in one quality (Q1) and the video segments in qualities Q1 and Q2 and, finally, the corresponding HTTP Response messages.

3. DASH traffic pattern characterization

One of the main goals of this study is to characterize the DASH traffic pattern and analyze its relationship with the ethernet traffic in order to obtain conclusions about the efficiency of the IEEE 802.3az – EEE standard taking into account the results in the previous studies about multimedia traffic and the energy efficiency. In this section, the experiments carried out and the resulting traffic patterns are presented.

3.1. Dynamic adaptive streaming over HTTP test-bed

In order to characterize the ethernet traffic pattern in a DASH-based streaming platform, a DASH test-bed was deployed as shown in Fig. 5. The test-bed consists of a web server acting as a multimedia server and a web video player that will be the multimedia client. The multimedia content is hosted at the server. This content is a movie encoded in three different representations or qualities using the ffmpeg tools, so the multimedia content involves several MP4 containers and the MPD file. Additionally, in order to be able to perform an extensive experiment, the videos are available in different segment sizes, which are 4, 6 and 10 s. The DASH player is a Javascript web application, so although the Shaka Player [7] is also hosted at the server, the player application will run at the client side. Apache has been used as the web server entity. At the multimedia end-point, the Chrome browser is used as the HTTP client entity, where the Shaka Player will be loaded. In order to control the bandwidth available for the target streams, the Chrome bandwidth throttling from developer tools has been used. The test-bed uses the TCP/IP protocol stack over Ethernet 802.3 1000BASE-T.

Regarding the DASH content, an open movie called Tears of Steel from the Blender Mongo Project [8] has been used, encoded and prepared for streaming using DASH. Specifically, a portion of 180 s of the movie has been used, encoding three different video qualities in H.264, using the MP4 container. The same video and qualities are also available in 4, 6 and 10 s for the DASH video segment duration. Target encoding bitrate, average bitrate and resolution of the video representations are shown in Table 3. Additionally, one audio track is encoded and used in the streaming service. The difference between the target encoding bitrate and the average bitrate is due to the variability of the video and audio content and the nature of the codec, which is able to compress contiguous similar frames and samples more than different ones.

3.2. DASH traffic pattern

To obtain the traffic pattern for the DASH application and describe it at application and Ethernet levels, experiments were performed using the DASH test-bed described in Fig. 5. Four transmission scenarios were used that differ from each other in bandwidth limitation at the DASH client. Bandwidth throttling is carried out using the Google Chrome developer tools throttling feature. Bandwidth limitation for each scenario is adjusted to 1000 kbps, 2000 kbps and 3500 kbps, that are values close to the encoding objective of each one of the available video qualities, resulting in three scenarios with fixed bandwidth, and a fourth scenario with variable bandwidth every 30 s is also added to analyze the performance on variable scenarios. That four transmission scenarios are analyzed also for three different values of DASH video segment sizes, 4, 6 and 10 s. Conducting the experiments allows to obtain traces in the client and the server and to carry out the required processing and analysis to obtain the DASH traffic profile and the corresponding generated Ethernet traffic pattern.

The DASH application-level message exchange regarding the conducted experiments could be seen in Fig. 6, which shows the messages from the loading, buffering and the steady states of the streaming

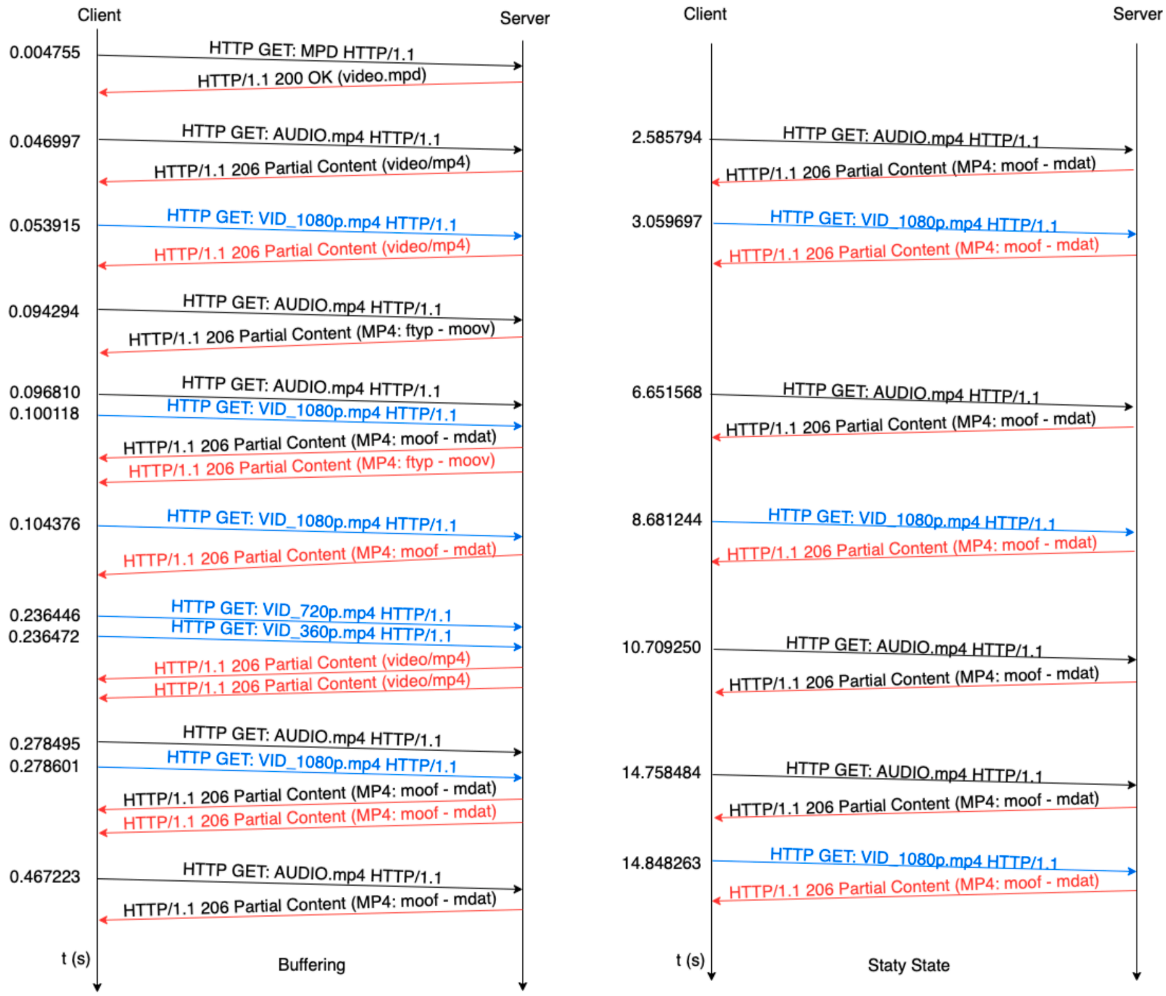


Fig. 6. Buffering and steady state in a DASH session.

Table 4
Client-side DASH traffic pattern.

Available Bandwidth (kbps)	Segment Size (s)	Avg. Segment Download Time (s)	Avg. Segment Size (Bytes)	Avg. Segment Throughput (bps)
3500	4	2.99	1.2 M	3.3 M
2000		3.09	643 k	1.7 M
1000		3.22	332 k	820 k
3500–1000		3.05	791 k	2.1 M
3500	6	4.56	1.9 M	3.5 M
2000		4.70	977 k	1.7 M
1000		4.89	505 k	827 k
3500–1000		4.56	1.2 M	2.2 M
3500	10	7.23	3.1 M	3.4 M
2000		7.48	1.5 M	1.6 M
1000		7.79	805 k	826 k
3500–1000		7.48	2.3 M	2.1 M

session. This message exchange shown in the figure corresponds to an experiment of a highest throttling and 4 s segment case scenario. As it could be seen, at buffering state the DASH player requests the MPD file to the server, downloads it and reads it to start the retrieving process of the available segments of video content in a selected quality in subsequent HTTP GET messages. Also, it could be observed that the client starts different HTTP/TCP stream sessions to download the video segments for each one of the available qualities, thus requesting the first segment for each available representation. As Fig. 6 depicts, the first group of HTTP request messages asking the first video segments of each different quality are those that have the corresponding HTTP 206 replay messages carrying a Partial Content (video/mp4). This HTTP response messages convey the first part of the MP4 containers named VID_1080p, VID_720p, VID_360p. Then, the player based on the available throttling (bandwidth) continues requesting the next segment of the selected quality.

On the other hand, at the steady state the DASH client requests segments of a selected quality each time it needs a new segment, and the requested quality depends on the available bandwidth. As it could be seen in Fig. 6, the segment retrieving process is done with seconds of

Table 5
Server-side DASH traffic pattern.

Available Bandwidth (kbps)	Segment Size (s)	Avg. Segment Upload Time (s)	Avg. Segment Size (Bytes)	Avg. Segment Throughput (Mbps)
3500	4	1.73	1.38 M	16
2000		1.24	661 k	22
1000		0.99	344 k	36
3500–1000		0.98	891 k	67
3500	6	3.25	1.99 M	7.6
2000		3.15	997 k	6.4
1000		2.34	516 k	10
3500–1000		1.76	1.3 M	22
3500	10	6.02	3.33 M	4.1
2000		5.45	1.66 M	2.1
1000		4.43	859 k	1.4
3500–1000		2.93	2.22 M	2.9

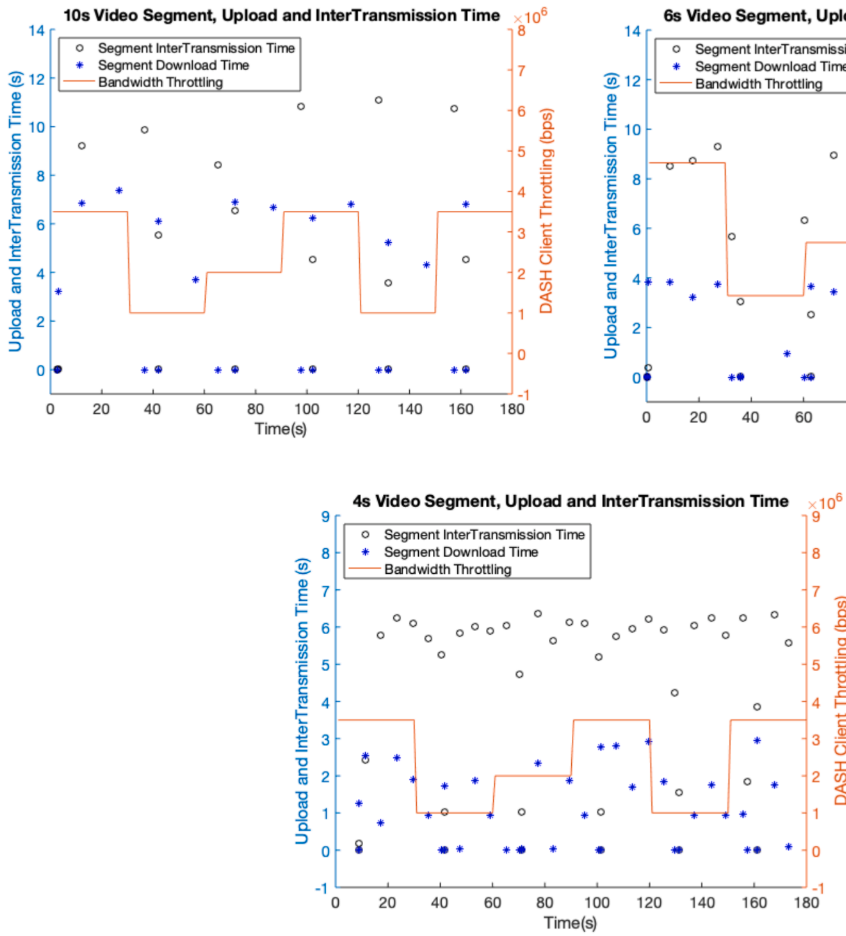


Fig. 7. Segments inter-transmission and transmission time at server with variable bandwidth throttling.

difference depending on the segment duration. If the available bandwidth changes during the streaming of the video, the DASH player will use the other HTTP/TCP stream sessions initialized beforehand to download the next low or high quality video segment.

Running the experiments for the scenarios described above, the DASH traffic pattern is obtained. Description of the DASH traffic pattern at application level is presented in Table 4, where there are parameters from the DASH segment perspective that includes: average segment download time (seconds), average segment size (Bytes) and the segment transmission average throughput (bps) attending to the different bandwidth throttling scenarios. These results are measured at the client where the bandwidth throttling is applied. These parameters are shown to characterize and understand the behavior of the DASH traffic and are measured at application level.

After illustrating this traffic characterization, the DASH traffic profile at the Ethernet connection level is presented below in Table 5. These measurements correspond to the DASH traffic pattern at server side for the same streaming scenarios. The results presented are the average segment upload time (s), average segment size (Bytes) and segment throughput (Mbps).

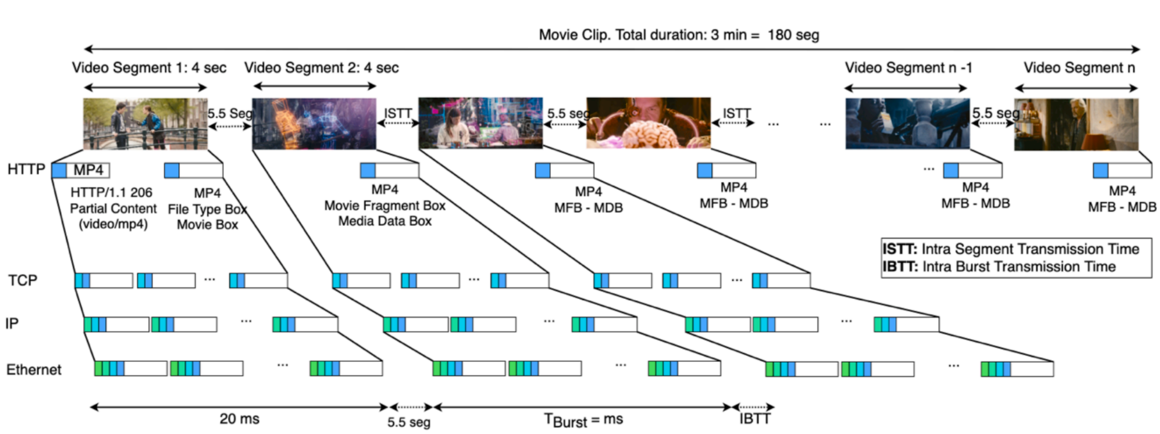


Fig. 8. DASH protocol stack encapsulation.

Table 6
Ethernet DASH traffic at the server.

Bandwidth (kbps)	Segment Size (s)	Avg. Time Between Ethernet Burst (s)	Avg. Ethernet Burst Size E(B) (Frames)
3500	4	3.97	1104.87
2000		4.48	568.70
1000		4.73	326.97
3500 – 1000		4.21	753.63
3500	6	5.16	1706.10
2000		5.31	939.90
1000		5.95	502.85
3500 – 1000		5.85	1163.47
3500	10	8.00	2926.17
2000		8.05	1561.59
1000		9.46	820.67
3500 – 1000		5.97	1976.8

Visually, the same DASH traffic characteristics are presented in Fig. 7, showing for each video segment in the transmission the upload time and the segment request interarrival time. Understanding the start times of new segment transmission, the time between segment transmissions, the segment download time, the segment size in Bytes is relevant to determine the traffic nature of the corresponding Ethernet traffic generated. In the next section, the traffic pattern is analyzed in terms of ethernet frames.

3.3. Ethernet DASH traffic pattern

Considering the outcomes obtained and presented in the previous section about the trace analysis in DASH at segment level, this section focuses on the DASH traffic pattern at Ethernet level.

First, the encapsulation of DASH content over Ethernet can be graphically observed depicted in Fig. 8. At application layer, the MP4 video messages are shown, which represent the DASH movie segments, i.e. 4 s of movie in this case. Also Fig. 8 presets the Intra Segment Transmission Time (ISTT), i.e. 5.5 s in average. Those video segments are transmitted in HTTP messages on TCP streams via IP datagrams. The group of IP packets that convey all MP4 messages (video segments) generates a burst of ethernet frames (E(B)) to be transmitted by the Network Interface Card. Each E(B) has a duration time, T_{Burst}, depending on the number of ethernet frames. For instance, a $T_{Burst} = 20$ ms is presented for an ethernet burst of 400 frames. Also, the Intra Burst Transmission Time (IBTT) is represented, which is approximately equal to the ISTT.

For the 4 bandwidth scenarios and the 3 different segment size use cases of the experiments carried out, the average time between ethernet burst in seconds and the average burst size E(B) in number of ethernet frames is obtained, as shown in Table 6. These values characterize the ethernet traffic pattern using DASH. This characterization of the ethernet traffic handled by DASH segments presents valuable values that allow the Ethernet Burst Size in number of frames and the transmission time to be calculated and, thus, the calculation and computation of the energy efficiency and energy savings, which will be presented in the following section.

4. EEE energy efficiency on DASH

The DASH ethernet traffic pattern described in the previous section is profiled using parameters like video segment inter-arrival time, video segment download and upload time, ethernet burst size E(B) in frames and time between ethernet bursts. These parameters make possible to calculate the time the NIC is in active and low power idle modes. Therefore, the DASH segment efficiency can be proposed and calculated using the efficiency relationship of the total data transmission time over the total active and transitions time of the NIC, as in Eq. (4). This DASH segment efficiency can be achieved because the observed DASH traffic pattern implies transmissions of MP4 content in segments, so the DASH content streams conform ethernet bursts with E(B) size.

Table 7
Segment efficiency.

Bandwidth (kbps)	Segment Size (s)	Segment Efficiency $\eta_{Segment}$
3500	4	0.9829
2000		0.9674
1000		0.9446
3500–1000		0.9752
3500	6	0.9889
2000		0.9800
1000		0.9632
3500–1000		0.9838
3500	10	0.9934
2000		0.9879
1000		0.9771
3500–1000		0.9904

$$\eta_{segment} = \frac{Total\ segment\ trasmission\ time}{Total\ active\ and\ transitions\ time} \tag{4}$$

The DASH segment efficiency will be around 100% when wake-up time, Tw, and sleep time, Ts, are negligible from the ethernet burst transmission time perspective. The ethernet burst transmission time is equal to the ethernet burst size in frames, E(B), times the ethernet frame transmission time, T_{Frame}. Based on the GoP efficiency calculation method used in the previous IPTV and EEE efficiency study [5], the video segment efficiency for DASH streaming is presented in Eq. (5).

$$\eta_{segment} = \frac{E(B) * T_{Frame}}{T_w + E(B) * T_{Frame} + T_S} \tag{5}$$

As an example, Eq. (6) presents the DASH video segment efficiency calculated for the case of the 3500 kbps bandwidth throttling and 4 s segment size. In Table 7, the segment efficiency is presented for the other DASH streaming experiments carried out in this study, for different available throttling, and different video segment sizes. E(B) values used are those presented in Table 6 and the $T_{Frame,}$ Tw and Ts are constant values defined by the EEE 802.3az standard for the 1000Base-T transmission speed.

$$\eta_{segment} = \frac{1104.87 * 12\mu s}{30\mu s + 1104.87 * 12\mu s + 200\mu s} = 0.9829 \tag{6}$$

As shown in Table 7, it could be concluded that the segment efficiency takes advantage of the transmission of burst of ethernet frames, being higher when high bandwidth is available to send bursts of the corresponding DASH segments, thus implying that the NIC card could stay longer in low-power idle mode than high-power active mode. The calculated DASH segment efficiency is related to the energy consumption ratio of the wake-up and activation processes of the network card, going from low-power idle mode to active transmission mode. Then, DASH traffic pattern driven by video segments allows better efficiency ratio for IEEE 802.3az.

Table 8
Segment efficiency and energy saving.

Bandwidth (kbps)	Segment Size (s)	Segment Efficiency $\eta_{Segment}$	Avg. Energy Saving (%)
3500	4	0.9829	61.81
2000		0.9674	69.71
1000		0.9446	73.64
3500–1000		0.9752	65.51
3500	6	0.9889	51.68
2000		0.9800	53.11
1000		0.9632	62.88
3500–1000		0.9838	61.84
3500	10	0.9934	48.90
2000		0.9879	49.20
1000		0.9771	57.80
3500–1000		0.9904	53.08

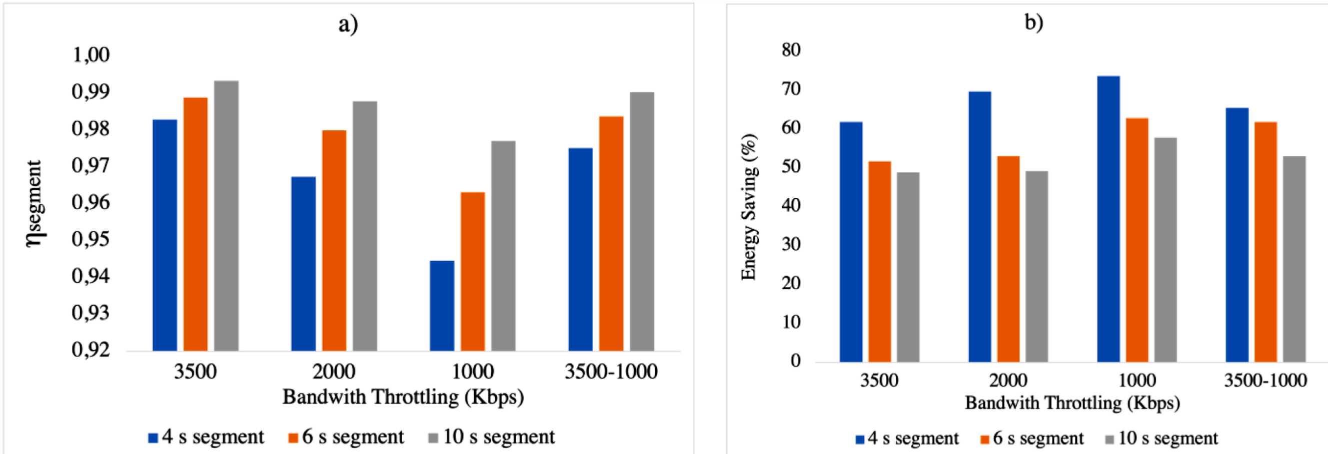


Fig. 9. a) segment efficiency, b) energy saving (%).

Another important aspect of the DASH traffic pattern is the time between video segment transmissions. As presented in the previous section, the average time between video segments or Ethernet burst is approximately 3.97 s in the 3500 kbps case. Bearing this in mind, it is possible to calculate the total time of the transmission of the movie when the card is in low energy consumption mode and in high energy consumption mode, which enables the possibility to verify the energy savings achieved by activating the standard 802.3az on the network interface card.

As an example, the calculation is made for the scenario with bandwidth variations where different video segment file sizes are transmitted from different qualities of the video because of the variable bandwidth. The average time between ethernet bursts is approximately 4.21 s and 42 Ethernet Bursts are transmitted, which takes the ethernet card to be in low power energy consumption mode approximately 117.9 s. For a duration of the movie of 180 s, it is equivalent to an energy saving of 65.51% when implementing the EEE standard. Table 8 shows the average energy saving calculated for each streaming scenario.

Fig. 9a evidences the fact that the average segment efficiency improves when the available bandwidth and the segment size are higher. For a given bandwidth, higher segment sizes will improve the segment efficiency. In the same way, for a given segment size, segment efficiency will be higher when the available bandwidth is also higher. Therefore, since the total energy saving, Fig. 9b, for the entire video stream depends on the segment size in time and bytes, using 4 s segments, for instance, in low quality video of about 1000 kbps for the target encoding bitrate causes the NICs to stay longer in low-power idle mode than when using 6 or 10 s segments.

Moreover, observing the video segment duration and its relationship with energy savings (Fig. 9b), it is found that short video segments, e.g. 4 s, favor energy savings for all the streaming scenarios, especially in the case of low bandwidth and low quality video segments, because the video segment size and its transmission time present influence on longer low-power transmission times, and consequently, this entails transmitting faster short video segments in short high-power transmission times.

In contrast, a higher segment efficiency is obtained with longer video segments with a little reduction in energy savings with respect to shorter video segments. However, the energy saving is still around 50% so implementing EEE is favorable on servers and clients involved with DASH traffic, regardless the video segment duration selected, although the impact of these variations at the application level do not affect energy efficiency much.

5. Conclusion and future work

The EEE standard allows switching to low power mode when the network card is not transmitting, which saves energy efficiently when sending bursty ethernet traffic. Consequently, the analysis and experiments carried out in this paper have shown that the ethernet traffic pattern generated by DASH streams implies burst of ethernet frames to be transmitted back-to-back, during short periods of time, in a periodic manner based on the DASH video segments size. That characteristic of the ethernet traffic pattern impacts positively on the efficiency and energy savings when implementing the IEEE 802.3az standard.

In order to carry out these analyses, the traffic pattern of a DASH streaming service has been characterized at application level and at ethernet level, using different segment sizes. This DASH service has been used to demonstrate under several bandwidth use case scenarios the improvements in segment efficiency, energy efficiency and energy savings that can be achieved using the EEE standard.

Given the large number of existing DASH video clients worldwide, this study aims to encourage the possibility of implementing the EEE standard on as many network interface cards as possible. This fact could lead to a high decrease in energy consumption globally. Energy efficiency research today is not only relevant due to the energy high cost and global warming, but it is also important to meet the Quality of Service (QoS) requirements of DASH streaming, the User Quality of Experience (QoE) as well as the total energy needed to perform all functions of the video distribution network, as is analyzed in other studies [9,10] and still there are challenges to meet and asses requirements of QoE to address in balance with energy efficiency [11]. As a future work the Live Low-Latency DASH could be characterized to study the influence in Energy Efficient Ethernet to improve energy savings under streaming live conditions using other encoding techniques.

CCS Concepts

- Networks → Network performance evaluation → **Network experimentation.**
- Information systems → Information systems applications → Multimedia information systems → **Multimedia streaming**.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, J. Maestro, IEEE 802.3 az: the road to energy efficient ethernet, IEEE Commun. Mag. 48 (11) (2010) 50–56.
[2] P. Reviriego, K. Christensen, J. Rabanillo, J. Maestro, An initial evaluation of energy efficient ethernet, IEEE Commun. Lett. 15 (5) (2011) 578–580.
[3] IEEE Std 802.3az Energy Efficient Ethernet-2010, IEEE Standard for Information technology– local and metropolitan area networks– specific requirements– Part 3: CSMA/CD Access Method and Physical Layer Specifications Amendment 5: media Access Control Parameters, Phys. Layers Manag. Parameters Energy-Efficient Ethernet (2010).
[4] P. Reviriego, J.A. Maestro, J.A. Hernández, D. Larrabeiti, Burst transmission for Energy Efficient Ethernet, IEEE Internet Comput. 14 (4) (2010) 30–37.
[5] A. de la Oliva, R.T. Vargas Hernández, J.C. Guerri, J.A. Hernández, P. Reviriego, Performance analysis of Energy Efficient Ethernet on video streaming servers, Comput. Netw. (2012), https://doi.org/10.1016/j.comnet.2012.09.019. Available online 11 October 2012, ISSN 1389-1286.
[6] ISO/IEC 23009-1, Dynamic adaptive streaming over HTTP (DASH) -Part 1: media presentation description and segment formats, 2012. (2017).
[7] Shaka Player Demo. [Online]. Available: https://shaka-player-demo.appspot.com. [Accessed: 01-Jul-2021].
[8] Tears of Steel Movie, Mongo Open Movie project. Blender Foundation. Available online: https://mango.blender.org/ (Accessed 01-Jul-2021).
[9] R. Bolla, R. Bruschi, F. Davoli, E.V. Depasquale, Energy-efficient management and control in video distribution networks: 'legacy' hardware-based solutions and perspectives of virtualized networking environments, in: A. Popescu (Ed.), Greening Video Distribution Networks. Computer Communications and Networks, Springer, Cham, 2018, https://doi.org/10.1007/978-3-319-71718-0_2.
[10] A. Popescu, Y. Yao, M. Fiedler, X. Ducloux, Video distribution networks: models and performance, in: A. Popescu (Ed.), Greening Video Distribution Networks. Computer Communications and Networks, Springer, Cham, 2018, https://doi.org/10.1007/978-3-319-71718-0_3.
[11] K. Bouraqia, E. Sabir, M. Sadik, L. Ladid, Quality of experience for streaming services: measurements, challenges and insights, IEEE Access 8 (2020) 13341–13361.

Tito Raúl Vargas Hernández received his Telecommunications Engineering degree from Universidad Santo Tomás, Bucaramanga, Colombia, in 2003. The Spanish Education and Science Ministry recognized him as Telecommunications Engineer in 2005. He received the M.Sc degree in Telematics Engineering from Universitat Politècnica de València (UPV), Spain in 2008. From 2005 to 2010 he worked as ICT research technician and research engineer at AI2 and iTEAM Research Institutes at UPV. Since 2010 he has been working as lecturer, master degree director, ICT Consultant, Research Engineer, Digital Transformation Center director and Virtual Campus director at Universidad Santo Tomás, Bucaramanga, Colombia. Since 2022 he works as a researcher at iTEAM research institute. His research interests include multimedia streaming over IP networks, IoT applications architectures, energy efficiency in communications, fiber sensor applications and traffic analysis and evaluation.

Juan Carlos Guerri received the M.S. and Ph.D. degree, both in telecommunication engineering, from the Universitat Politècnica de València (UPV), Valencia, Spain, in 1993 and 1997, respectively. He is a Professor in E.T.S. Telecommunications Engineering at the UPV, where he leads the Multimedia Communications research group (COMM) of the iTEAM Institute. He is currently involved in research and development projects for the application of multimedia to industry, medicine, education, and communications.

Pau Arce, received his Telecommunications Engineering degree, M.S. in Telematics and the Ph.D. degree in Telecommunications from the Universitat Politècnica de València (UPV), Spain, in 2005, 2007 and 2014 respectively. Currently he works as a researcher at the Institute of Telecommunications and Multimedia Applications (iTEAM). His research interests include multimedia QoS, routing on wireless ad hoc networks and performance evaluation of computer systems.