HLS vs MPEG-DASH: Energy Consumption Comparison of Streaming Protocols

Dominik Myśliwiec
d.mysliwiec@student.utwente.nl
University of Twente
Enschede, The Netherlands

Muhammad Arifin Hidayat
muhammadarifinhidayat@student.utwente.nl
University of Twente
Enschede, The Netherlands

ABSTRACT

In this study, we will examine the energy usage of two popular streaming protocols: MPEG-Dynamic Adaptive Streaming over HTTP (MPEG-DASH) and HTTP Live Streaming (HLS). With the growing popularity of streaming services, more environmentally friendly software development must comprehend the energy implications of various streaming protocols. Through experimental analysis, this study seeks to measure, evaluate, and compare the energy consumption of MPEG-DASH and HLS, offering insights into their sustainability. The study's findings may aid in the optimization of streaming platforms' energy consumption and aid in the development of more environmentally friendly streaming technology.

ACM Reference Format:
Dominik Myśliwiec and Muhammad Arifin Hidayat. 2024. HLS vs MPEG-DASH: Energy Consumption Comparison of Streaming Protocols. In . ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Video streaming is one of the primary forms of entertainment in most parts of the world. In 2024, Sandvine reported that video streaming applications accounted for around 38% of the total worldwide downstream internet traffic, with YouTube, Facebook, and Netflix emerging as the primary providers [2].

Nevertheless, with the increasing demand for streaming, its energy usage and environmental impact also grow. The objective of this assignment is to analyze the energy usage of streaming protocols, specifically examining the HLS and MPEG-DASH protocols.

The streaming protocol known as HTTP Live Streaming (HLS) was developed by Apple to transmit video and audio information over the internet. The system operates by decomposing the content into compact, downloadable segments, which are then delivered to the viewer in a fixed order. This technique facilitates seamless playback, even with an unstable connection. This protocol can dynamically adjust the bitrate of a stream to various network conditions, therefore enhancing the user experience [14].

MPEG-DASH (Dynamic Adaptive Streaming over HTTP) is another popular streaming protocol that operates in a similar way to HLS. The technology partitions the content into parts and adapts the video quality according to the user's internet speed. The difference between MPEG-DASH and HLS is that MPEG-DASH is an open standard, enabling compatibility with more devices and systems [17].

We propose a study to compare the energy usage of these protocols under similar conditions and measurements. This document is organized into the following sections: Section 2 provides the background of the study, Section 3 examines relevant literature, Section 4 describes our approach to experiment planning, Section 5 provides information on the tools, environment and procedure of the experiment with a discussion on threats to validity in Section 6, and finally Section 7 and 8 show the result of the performed experiment and a discussion and interpretation of the gathered data.

2 CONTEXT

This study will focus on the energy usage comparison of the two most popular video streaming protocols — HLS and MPEG-DASH. The aim of the work is to support video providers and software developers in creating energy-efficient video streaming platforms. Experiments will be conducted using both protocols in client and server environments, where we will simulate real-world streaming conditions and measure energy consumption from both perspectives.

It was estimated by IEA that in 2020, streaming a Netflix video for an hour required 0.077 kWh of energy on average, of which 72% is used by the device's displaying hardware, 23% for data transmission, and 5% by data centers [1]. The energy usage of data transmission strongly indicates that a significant portion of the energy usage can be affected by the streaming protocol, which decides what bitrate is used by the client, but also how often packets are transferred and what size they are. Having established that the energy usage of video streaming can be improved by choosing the right protocols, it is important to evaluate the impact of standardizing a more efficient protocol from various aspects. This can be done by utilizing the framework of the five dimensions of sustainability [4].

2.1 ~~Individual sustainability~~

~~This dimension considers the impact of a solution on the well-being of an individual. Because this research does not impact the individuals in this dimension, individual sustainability can be disregarded for the purpose of this work.~~

2.2 Technical sustainability

This dimension is used to evaluate the adaptability and longevity of a chosen protocol. Though it is not the primary focus of this

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, July 2017, Washington, DC, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

research, it's important to mention that MPEG-DASH is a more adaptable open-source solution, but is not, as of yet, supported by many Apple devices, whereas HLS is less open, but has universal support. For this reason, it is important to have open standards to avoid monopolies. If only one company controls a specific technology, streaming for this case, it may favour its products that might not be affordable for everyone. An open standard, on the other hand, if adopted by the industry can ensure fairness in the operations and functioning of the industry, allowing for more competition between companies and providers, and creating more innovative solutions for the clients without major sacrifices with regards to support or functionality.

2.3 Social sustainability

This dimension places the focus on factors that can build or erode trust in society. Both protocols considered in this work are commonly supported, though, for the purpose of retaining equality of access to videos, it's important that full support is ensured for the better-performing protocol. For HLS this is trivial, but MPEG-DASH support would have to be introduced for many Apple devices. Providing equal access to technology would increase fairness across different users.

2.4 Economic sustainability

Economic sustainability is centred around the financial viability of a solution. In the case of streaming protocols, using the more energy-efficient protocol places less stress on network infrastructure resulting in decreased operational expenses. It could be a clear advantage from an economic perspective. In addition, showcasing that an open standard protocol is as efficient, if not more efficient than one controlled by one company can improve the adaptation of said protocol and therefore decrease the influence of a single company over the market. This would allow for more competition to meet user's demands without giving up on functionality and prevent the company from imposing its political and economic interest on the target demographic.

2.5 Environmental sustainability

Finally, the environmental sustainability dimension is concerned with the long-term effects of a solution on the natural systems. The effects on the energy usage of video streaming are the primary purpose of this research for the purpose of improved environmental sustainability of global internet usage. An efficient video streaming protocol has the potential to significantly decrease the energy usage of video streaming and therefore reduce the impact internet usage has on the environment.

3 RELATED WORK

The optimization of energy consumption in adaptive video streaming received growing attention, as several studies have investigated their developments at both the client and server levels.

A study from 2018 evaluated energy consumption in video streaming using MPEG-DASH with a focus on video encoding, more specifically using HEVC. The authors found that GPU-based encoding significantly reduces energy use compared to CPU-based methods, particularly during content production. By comparing adaptation strategies like resolution, bitrate, and frame rate, the study concluded that frame rate adaptation was the most energy-efficient. Although the study provides relevant information on MPEG-DASH, it does not compare it against HLS [20].

~~In 2017, Santos-González et al. completed a comparative study of the performance of two video streaming platforms utilizing different protocols — RTSP and WebRTC [16]. Although the two protocols show different results, with the WebRTC platform providing a more pleasant viewing experience, RTSP and WebRTC are push-based protocols, unlike HLS and MPEG-DASH. This means that there are fundamental differences in how the protocols function, which makes the study not as relevant in comparing HLS and MPEG-DASH.~~

Marks and colleagues [12] highlighted the significant carbon footprint of streaming media, stating that it accounts for around 1 percent of worldwide greenhouse gas emissions, an amount projected to rise. As the number of users streaming higher-definition content for longer durations increases, the energy requirements of data centers, networks, and user devices also rise. Although streaming technologies have improved in efficiency, this has frequently resulted in increased overall usage. The authors proposed practical strategies, such as decreasing video resolution and promoting more environmentally friendly energy sources, to reduce the environmental impacts of streaming. The study provides fundamental groundwork for the environmental sustainability of the Internet but does not provide highly detailed information on video streaming protocols.

In a more recent review [9], the environmental impact of video streaming is examined, focusing on strategies to enhance energy efficiency on both the client and server sides. The review emphasized that client-side optimizations, such as adjusting display brightness and using efficient codecs like HEVC, can help reduce energy consumption. On the server side, employing energy-efficient hardware and optimizing CDNs are crucial. Emerging technologies, including artificial intelligence and edge computing, were also identified as promising avenues for further reducing energy usage.

Additionally, a survey by Afzal [3] discussed energy consumption in video streaming from production to consumption. The study proposed a taxonomy of factors affecting energy use, such as encoding schemes, hardware efficiency, and the lack of open energy measurement datasets. It also highlighted challenges in measuring the carbon footprint of streaming and the inefficiency of hardware in existing video players.

While previous studies have focused on energy consumption during specific aspects of video streaming, such as encoding efficiency and content delivery, our proposed experiment is unique in its comparative analysis of the energy efficiency between HLS and MPEG-DASH protocols using a locally deployed video hosting web application. By running server- and client-side experiments with the protocol as an isolated dependant variable, we aim to provide new insights into which protocol consumes less energy while maintaining quality. Unlike previous studies that have focused heavily on codecs or cloud environments, this experiment emphasizes the energy impact of the streaming protocol itself, offering a practical perspective for service providers aiming to reduce energy consumption at both ends.

4 EXPERIMENT PLANNING

To conduct an optimal method of comparing the two protocols, first, the goals and research questions have to be defined. A methodology that can be utilized for that purpose is the Goals, Questions, and Metrics (GQM) method formalized by Basili et al. [5]. This method consists of first defining the specific goal of the research, then extracting an appropriate research question and defining the metrics that can be used to answer it.

4.1 Goal

The first step of following the GQM method is defining the goal. This includes defining the target of the research, its purpose, as well as the target demographic and context. The goal of this research can be formalized as the following:

Analyze HLS and MPEG-DASH
For the purpose of comparison
With respect to their energy usage
From the point of view of streaming providers or software developers
In the context of a university course

4.2 Questions

To achieve the defined goal, a set of research questions have to be answered. These questions will provide a transition mapping the goal to more specific metrics and methods.

Q1 What is the energy usage of streaming a video using the HLS and MPEG-DASH protocols?
~~**Q2** Is one of the protocols more energy efficient than the other?~~

We derived 3 more specific research questions based on the main research question.

Q1.1 How does the bitrate of a video stream impact the energy usage for HLS and MPEG-DASH protocols?
Q1.2 What is the difference in energy usage between client, server, and data transmitted for each protocol?
Q1.3 Is one of the protocols more energy efficient than the other?

4.3 Metrics

To answer the research questions, we will conduct experiments under varying conditions to evaluate the energy efficiency of each protocol. The following metrics are needed to answer the research questions:

Q1 Metrics Energy usage across experiments with different values of dependent variables
Q1.1 Metrics Energy used per bitrate
Q1.2 Metrics Energy used by the client, server, and transmitted data
Q1.3 Metrics The difference in energy consumption across experiments with different values of dependent variables

4.4 Hypotheses

The first step in planning an experiment which will help answer the posed research question is to state the hypotheses. The null hypothesis states that the energy usage of streaming a video using the HLS protocol E_H and the energy usage of streaming a video using the MPEG-DASH protocol E_D are the same.

$$H_0 : E_H = E_D$$

An alternative to the null hypothesis is that the energy usage of the two protocols is not equal. Since this is a hypothesis for a two-tailed test, this alternative hypothesis holds true no matter which protocol uses more energy.

$$H_a : E_H \neq E_D$$

4.5 Experimental units

To create an isolated experiment environment, we are planning to develop a locally hosted web application in two versions, one using the HLS protocol, and the other utilizing MPEG-DASH. The experimental unit is the whole system streaming the video; in other words, the server hosting the video, the web clients playing back the video, and the infrastructure used for networking between them.

4.6 Independent Variables

4.6.1 Video Streaming Protocol. In this study, we have this independent variable, which is the most vital aspect of the study – the streaming protocol. The protocol has two alternatives, which are the two most popular implementations of the compared protocols:

MPEG-DASH https://github.com/Dash-Industry-Forum/dash.js
HLS https://github.com/video-dev/hls.js

In a 'run' of the experiment, these two protocols can be compared by retaining all other independent variables through the configuration of these two implementations.

4.6.2 Video Bitrate. One of the main appeals of the two modern streaming protocols is the ability to support many different video bitrates. This is usually utilized by dynamically changing the bitrate to provide the highest possible quality of video without noticeable loading times depending on the capabilities of the user's device and network connection. That being said, it would still be valuable to compare the two protocols with regard to the same static bitrate, which increases how controlled the experiment is and can provide valuable insight into potential differences. For that reason, we will use the following alternatives for the video bitrate:

750kbps Extremely low quality used with slower networks
2500kbps Lower end of usual streaming bitrate
6000kbps Higher end of usual streaming bitrates
~~**Adaptive** The adaptive bitrate is the default setting for both protocols and allows the player to select the most optimal quality for video streaming according to the protocol implementation~~

4.7 Dependent Variables

The dependent variables are what will be observed and measured during the experiment to gain an understanding of the energy usage of the video streaming system:

Energy used by the server The energy used by the web application process to stream the video
Energy used by the client The energy used by the client to play the video
Number of transmitted packets Number of packets transmitted between the client and server

Transmitted data Sum of all the data from packets transmitted between the client and server

4.8 Workloads

In the case of our experiment, the workloads of the experiment are different configurations of the video streaming web application. There are many real-world usages of video streaming, too many to simulate for the purpose of this experiment. Because of this, we focus on the main configuration variables of the system which should have the highest correlation with the streaming protocol.

4.8.1 Video Streamed.
The web applications will be streaming the same short animated movie called "Big Buck Bunny" – an open-source video with a 10 minutes and 35 seconds duration created by the Blender Institute commonly used for video streaming testing [15]. The highest bitrate stream available as of planning the experiment, the 60fps 4k version of the short film was downloaded then FFmpeg [18] was used to create an MPEG-DASH .mpd file and an HLS .m3u8 playlist, both containing three streams with the bitrates utilized for the experiment.

4.8.2 Video Coding Format.
The video coding format can highly influence its playback. This is why this variable should be taken into account when comparing two protocols. There are two formats supported by both HLS and MPEG-DASH that can be used as alternatives:

H.264 This is the older, but still popular, video coding format. It is less efficient, meaning it requires a higher bitrate to retain the same quality of video.

H.265 This is the newer format that works very similarly but is better at utilizing a lower bitrate to play the video at an equally high quality.

In this experiment, we will select only H.264, due to more limited support for H.264 and the lack of time needed to ensure it works properly with the used libraries within the scope of this project.

4.8.3 Video Bitrate.
For each workload, we will use one of the following video bitrate:

750kbps Extremely low quality used with slower networks
2500kbps Lower end of usual streaming bitrate
6000kbps Higher end of usual streaming bitrates

4.9 Analysis Procedure

The result of the experiment will be analyzed using statistical approaches to gain insight into energy consumption and comparison of HLS and MPEG-DASH. We plan to use several methods:

Descriptive statistics We will utilize basic statistical data from the experiment to summarize the result and visualize it for easier understanding

Paired sample t-test We will perform the Paired sample t-test to analyze whether there is a significant difference in the energy consumption of the system. Our observations are paired measurements of energy consumption from both HLS and MPEG-DASH protocols under identical bitrate. Here, the energy consumption of the system is the sum of the energy consumption of the server, client, and the estimated energy consumption of the network infrastructure.

Shapiro-Wilk This test will be used for assessing the normality of the differences between paired observations.

Mann-Whitney U If normality is not met, we will use the Mann-Whitney U test that does not assume normality.

5 MEASUREMENT TOOLS, ENVIRONMENT, AND PROCEDURE

This section outlines the measurement tools, environment, and procedure used in the research to ensure accurate and reliable results.

5.1 Measurement Tools

In the upcoming experiments, we will monitor energy consumption on both the server and client sides to comprehensively evaluate energy consumption. Running Average Power Limit (RAPL) will be employed to gather the necessary details on the process, such as energy usage when possible, or otherwise power usage over time for an accurate approximation [8]. RAPL is chosen because of its low-level interface that provides a measure of the energy consumption of a CPU and others without the need for additional hardware. Moreover, RAPL is designed by Intel so that is compatible with our experimental setup. Additionally, a network analysis tool such as Wireshark will be used to estimate the impact of the protocol concerning data transmission[10] and use 10W/GB [1], [11] to calculate the energy consumption. Together these measurements will form the foundation for our energy consumption analysis.

5.2 Environment

To perform the experiment, 2 different devices are utilized. The first one is an MSI-GS65 Stealth Thin 8RE laptop with 8th Gen. Intel® Core™ i7-8750H Processor at 2.2 GHz, 16 GB of DDR4-2666 RAM, GeForce® GTX 1060 with 6GB GDDR5, 256 GB SSD running Ubuntu 24.04.1 (Noble Numbat). The latest Google Chrome browser for Linux (version 130.0.6723.58) will be selected over the other browsers because of its popularity and utilization[13]. The laptop will be plugged in (not on battery mode), and the brightness will be set at the highest level, 100% streaming player volume, and 30% laptop volume.

The second device is a 16-inch MacBook Pro from 2019, with a 2.6GHz 6-core Intel Core i7 CPU, 512GB SSD, Intel UHD Graphics 630 GPU running macOS version Sonoma 14.6.1 and Google Chrome browser for MacOS (version 130.0.6723.59). The laptop will also be plugged in, and the brightness will be set at the highest level, 100% streaming player volume, and 30% laptop volume.

Although two devices are required, for consistency and accuracy all measurements will be obtained from the system running Linux. This means that in a single run of the experiment, the experiment will be executed multiple times with the Linux machine fulfilling each role once so that that aspect of the experiment is measured without interference.

5.3 Procedure

The experiment will be started by running the server. After that, we will synchronize the measurement by running RAPL, then loading the video on the client browser and playing it's full duration. Additionally, to maintain measurement accuracy and reliability, some procedures will be employed:

- The workloads will be executed in a controlled environment that is as constant as possible. This includes performing a network speed test before each experiment to ensure all runs use a network connection speed within 10% of the mean of the connection speeds of all runs. All tests are also run at night, due to a lower risk of network interference.
- Each run of RAPL is done with a timeout of 650 seconds, where the video is loaded and played by the client within 1 second of starting the measurement. The fixed time allows for consistent padding of passive runtime regardless of the differences in time elapsed between starting the measurement and playing the video.
- Each test run consists of executing the experiment 3 times, once to measure energy usage as a server, another for energy usage as the client, and one for network analysis. This is done to increase result accuracy by measuring all elements on the same system.
- Each test run will be iterated 5 times for each combination of bitrate and protocol to ensure no outlying results.
- Energy will be measured using perf [6]. Every measurement of energy will be done using the `power/energy-pkg/` counter, which contains the value of energy consumed by the whole package and is available for most Intel CPUs. The GPU energy is of less importance in this experiment, as both the HLS and DASH systems will be displaying the same video in the same bitrate.

6 THREATS TO VALIDITY

In this study, ensuring validity is crucial to ensure the results' reliability. We gathered some potential threats to the construct, internal, external, and conclusion validity [7]:

6.1 Construct Validity

An unclear definition of energy usage for each protocol may lead to incomplete results. For example, the measurement of a different dependent variable for one streaming protocol but not the other or a different measurement method may make the comparison unequal. We plan to apply all workloads for each protocol and measure the same dependent variables while executing multiple tests involving energy usage and network usage to gain more accurate results that reflect the actual energy used by each protocol.

6.2 Internal Validity

Network conditions might influence the experiment results validity because of unexpected fluctuations during the experiment. The other threat that may arise comes from environmental setup. If settings are assigned differently for each protocol, the result may not be able to express the comparison. To mitigate the risk, we will try to execute the experiment under the same configurations and the same network. To ensure this is the case, we will measure the network connection speed before every test and avoid running tests under different conditions.

6.3 External Validity

We plan to execute the experiment only in a limited environment as we stated before. The result of the experiment may not be able to represent all possible cases such as different devices or platforms. It could be mitigated by involving more experiments with different setups to increase generalization, but unfortunately, it is out of the scope of this research and is left to future researchers.

6.4 Conclusion Validity

Some conclusion validity threats that we found are insufficient variable size and biased measurements. If the variables are not defined correctly, for example, short testing time, it may result in biased measurements because it may not detect significant differences. Additionally, measurement tools themselves could lead to inaccurate measurements if they are not chosen correctly. For those reasons, we select measurement tools carefully that are tested for similar research subjects and try to formulate sufficient variables to minimize the impact.

7 RESULTS

Results for RQ1.1 are presented in Section 8.1, describing the impact of bitrate on energy consumption for HLS and MPEG-DASH. In Section 8.2, we answer RQ1.2 by investigating the distribution of energy usage between client, server, and data transmission. Finally, in Section 8.3, RQ1.3 is answered by comparing energy consumption between protocols. All measurement results can be seen from Figure 1. All graphs will be presented using mean values. We performed the Shapiro-Wilk Test [19] for each measurement of bitrate for a given protocol to check the correctness of the assumption of normality and received a p-value above 0.05 for all paired bitrate results.

7.1 Impact of Bitrate on Energy Usage for HLS and MPEG-DASH

Figure 2 shows the energy consumption across three different bitrates. At the lowest bitrate, MPEG-DASH consumes more energy around 4566.32 Joules on average compared to HLS with 4371.13 Joules. At 2500 kbps, HLS energy usage slightly surpasses that of MPEG-DASH, with average values of approximately 9327.82 Joules for HLS and 8768.53 Joules for MPEG-DASH. MPEG-DASH again displays higher energy consumption for the highest bitrate (6000kbps) at an average of 22412.11 Joules compared to HLS's 20647.93 Joules. The result suggests that energy consumption increases with higher bitrates for both HLS and MPEG-DASH. We can conclude that for overall energy consumption, HLS consumes less energy than MPEG-DASH, but for a specific bitrate of 2500kbps, MPEG-DASH is more efficient.

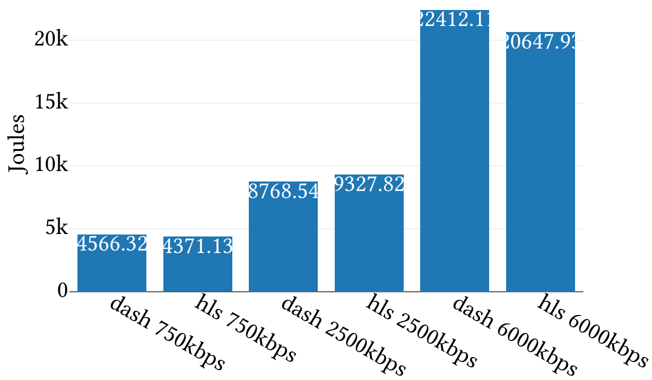

Figure 2: Energy Consumption Results

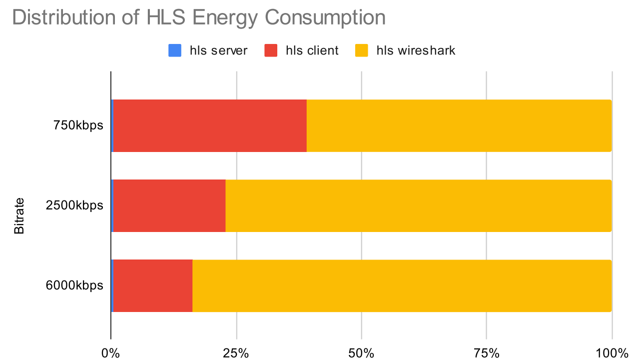

Figure 4: Distribution of HLS Energy Usage

7.2 Distribution of Energy Usage Between Client, Server, and Data Transmission

Further analysis segmented energy usage across client, server, and data transmission components for each protocol. Figure 3 and Figure 4 indicate that the data transmission consistently accounts for the majority of energy consumption across all bitrates of both protocols. The server consumes significantly less energy than the others with less than 1% of total energy usage. The range of client energy usage is around 20-40% while data transmission can take up to 80% of energy consumption.

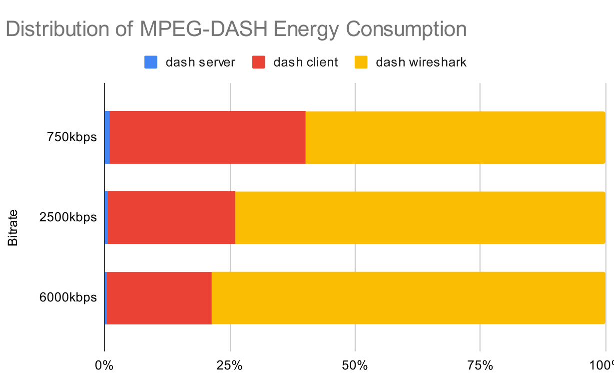

Figure 3: Distribution of MPEG-DASH Energy Usage

7.3 Energy Efficiency Comparison Between Protocols

To examine energy efficiency, we measured the total energy and weighted average of all protocols. We use this formula to calculate the weighted average.

$$\frac{\sum(\text{Energy Usage for bitrate} \times \text{Bitrate})}{\sum \text{Bitrate}}$$

Figure 5 shows that MPEG-DASH consumes more energy than HLS in both total and weighted average. The total energy of MPEG-DASH and HLS are around 35746.97 and 34346.89 Joules respectively. HLS is more energy efficient with 16268.7 Joules compared to MPEG-DASH with 17277.7 Joules. Because each run contains data that was most similar in the environment due to a similar time of execution, We used the paired-sample T-Test with a significance level of 0.05 to measure whether there is a significant difference in the energy consumption of the system because we know that the data is normally distributed based on the result of a Shapiro-Wilk Test. For the bitrate of 750kbps, we see a p-value of 0.004 which means H_0 is rejected in favor of $H_1 : E_H < E_D$. H_0 is also rejected for the bitrate of 2500kbps in favor of $H_1 : E_D < E_H$ and bitrate 6000kbps in favor of $H_1 : E_H < E_D$ because their p-value are 0.005 and <0.001 respectively.

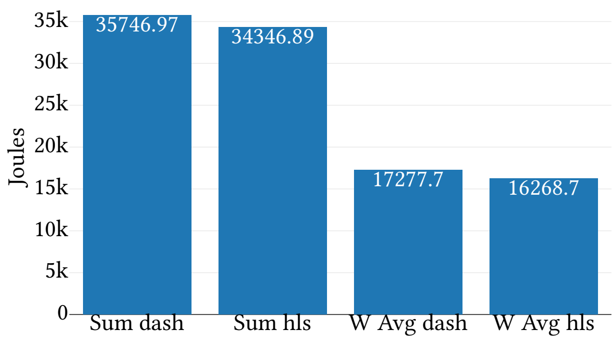

Figure 5: Total and Weighted Average of Energy Consumption

8 DISCUSSION

This experiment reveals several insights regarding the energy consumption of HLS and MPEG-DASH. Firstly, bitrate significantly impacts the energy usage of both protocols. As bitrate increases, both protocols require more energy, with the MPEG-DASH protocol consuming slightly more energy than HLS, especially in low and high bitrate scenarios. In a medium bitrate scenario, which often corresponds to streaming a video in a resolution of 1280 x 720 pixels, we see a different result that shows more efficiency of MPEG-DASH. It is clear based on the collected data that this difference stems from less overhead in the size of network transmission, but it is difficult to tell what this could be attributed to. It is possible that MPEG-DASH is more efficient with streams of this bitrate and this size, perhaps losing fewer packets or simply the dash.js library having better optimization for the resolution of 720x1280 pixels. Overall, it is clear lower bitrates can serve as an effective strategy for reducing energy demands, albeit with trade-offs in video quality, which must be considered based on user experience requirements and network capabilities.

Secondly, differences in energy demands across client, server, and data transmission further clarify the role of each component in the overall energy usage. Data transmission is the biggest energy consumption among the others. This suggests that energy optimization efforts should prioritize data transmission improvements. The amount of energy used by the client also shows a potential for improving the battery life of used devices by optimizing their video streaming. It is our assumption that the proportionately small energy draw of the server providing the video stems from the simplicity of the application. A server hosted on a consumer-grade device can have significant differences in complexity from a data center or industrial server, which are industry standards for providing video. Additionally, due to the ease of access of the video on the device SSD shared by the server's memory, the energy required to access the memory containing the video is significantly smaller than it could be in a different scenario. Regardless, similarly to results obtained by IEA in *The carbon footprint of streaming video: fact-checking the headlines*, server energy usage is a minor component of the system.

The statistical advantage in energy usage of HLS over MPEG-DASH seems to largely stem from the client. HLS consistently outperforms MPEG-DASH, especially in a high bitrate scenario, where we observed it using only 70% of the energy consumed by the same bitrate when streamed using MPEG-DASH. There is a very high likelihood these differences stem from the differences in support for HLS and MPEG-DASH on different devices and browsers as of writing this paper. Currently, HLS is much more widely adopted, with many devices supporting it natively using device-based decoders, while MPEG-DASH is forced to rely on third parties regardless of the environment. It is possible that this difference extends even into the quality of third-party support, with HLS decoders having higher demand and better resources than decoders of MPEG-DASH due to its much lesser popularity.

Finally, HLS appears to offer a more energy-efficient option than MPEG-DASH in terms of overall results. This result does not necessarily mean that HLS should be adopted more widely but perhaps highlights the need for improvement of other protocols like MPEG-DASH, which has other advantages over HLS as mentioned in 2. Regardless, if the energy usage of a video streaming application is the highest priority, there is a higher chance of achieving low energy consumption numbers by utilizing the HLS protocol over MPEG-DASH.

9 CONCLUDING REMARKS

This study presents the groundwork for a proposed experiment aimed at comparing the energy usage of the HLS and MPEG-DASH streaming protocols. The study aims to support green software development by offering valuable insights into the implementation of energy-efficient streaming systems.

More comprehensive results can be gained for further research by comparing different implementations and parameters, such as adaptive streaming, different encodings, and programming languages. Moreover, it would be very beneficial to perform tests on systems of different scales, with more clients per server, and more videos available per server.

REFERENCES

[1] The carbon footprint of streaming video: fact-checking the headlines. Technical report, IEA, 2020. URL https://www.iea.org/commentaries/the-carbon-footprint-of-streaming-video-fact-checking-the-headlines.
[2] 2024 global internet phenomena report. Technical report, Sandvine, 2024. URL https://www.sandvine.com/phenomena.
[3] Samira Afzal, Narges Mehran, Zoha Azimi Ourimi, Farzad Tashtarian, Hadi Amirpour, Radu Prodan, and Christian Timmerer. A survey on energy consumption and environmental impact of video streaming. arXiv preprint arXiv:2401.09854, 2024.
[4] Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. Sustainability design and software: The karlskrona manifesto. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, volume 2, pages 467–476, 2015. doi: 10.1109/ICSE.2015.179.
[5] Victor R Basili-Gianluigi Caldiera and H Dieter Rombach. Goal question metric paradigm. Encyclopedia of software engineering, 1(528-532):6, 1994.
[6] Linux Foundation. Perf. URL https://perfwiki.github.io/main/.
[7] Andreas Jedlitschka, Marcus Ciolkowski, and Dietmar Pfahl. Reporting experiments in software engineering, pages 201–228. Springer, Germany, 2008. ISBN 978-1-84800-043-8. doi: 10.1007/978-1-84800-044-5_8.
[8] Kashif Khan, Mikael Hirki, Tapio Niemi, Jukka Nurminen, and Zhonghong Ou. Rapl in action: Experiences in using rapl for power measurements. ACM

Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS), 3, 01 2018. doi: 10.1145/3177754.

[9] Koffka Khan. Advancements in sustainable video streaming: A comprehensive review on enhancing energy efficiency in adaptive video streaming systems. International Journal of Multidisciplinary Research and Publications, 6:202–207, 2024.

[10] HyunHo Kim, HoonJae Lee, and HyoTaek Lim. Performance of packet analysis between observer and wireshark. In 2020 22nd International Conference on Advanced Communication Technology (ICACT), pages 268–271, 2020. doi: 10.23919/ICACT48636.2020.9061452.

[11] Kungl. Ingenjörsvetenskapsakademien IVA. Science & Society Forum: Växande IKT-sektor och fler datacenter – hur påverkas elförsörjningen?, 2024. URL https://www.youtube.com/watch?v=Xo0PB5i_b4Y&t=2520s&ab_channel=Kungl.Ingenj%C3%B6rsvetenskapsakademienIVA. Accessed: 2024-10-25.

[12] Laura U. Marks, Joseph Clark, Jason Livingston, Denise Oleksijczuk, and Lucas Hilderbrand. Streaming media's environmental impact. Media+Environment, 2 (1), 10 2020. doi: 10.1525/001c.17242.

[13] Azza Mohamed and Ibrahim Ismail. A performance comparative on most popular internet web browsers. Procedia Computer Science, 215:589–597, 2022. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2022.12.061. URL https://www.sciencedirect.com/science/article/pii/S1877050922021329. 4th International Conference on Innovative Data Communication Technology and Application.

[14] Roger Pantos and William May. Http live streaming. Technical report, 2017.

[15] Ton Roosendaal. Big buck bunny. In ACM SIGGRAPH ASIA 2008 computer animation festival, pages 62–62. 2008.

[16] Iván Santos-González, Alexandra Rivero-García, Jezabel Molina-Gil, and Pino Caballero-Gil. Implementation and analysis of real-time streaming protocols. Sensors, 17(4):846, 2017.

[17] Iraj Sodagar. The mpeg-dash standard for multimedia streaming over the internet. IEEE multimedia, 18(4):62–67, 2011.

[18] FFmpeg Team. Ffmpeg. URL https://ffmpeg.org/.

[19] HC Thode. Testing for normality, 2002.

[20] Mikko Uitto and Martti Forsell. Towards energy-efficient adaptive mpeg-dash streaming using hevc. In 2018 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pages 1–6, 2018. doi: 10.1109/ICMEW.2018.8551534.

	dash server	dash client	dash wireshark	hls server	hls client	hls wireshark	dash energy	hls energy
750kbps run 1	44.03	1746	2736	27.12	1632	2664	4526.03	4,323.12
750kbps run 2	42.69	1845	2736	26.61	1657	2664	4623.69	4347.61
750kbps run 3	43.33	1798	2736	25.55	1733	2700	4577.33	4458.55
750kbps run 4	40.99	1729	2736	26.25	1698	2664	4505.99	4388.25
750kbps run 5	41.57	1821	2736	26.14	1684	2628	4598.57	4338.14
Averages	42.522	1787.8	2736	26.334	1680.8	2664	4566.322	4371.134
	dash server	**dash client**	**dash wireshark**	**hls server**	**hls client**	**hls wireshark**	**dash energy**	**hls energy**
2500kbps run 1	63.52	2308	6516	41.82	1974	7236	8887.52	9,251.82
2500kbps run 2	62.75	2350	6444	44.21	2173	7236	8856.75	9453.21
2500kbps run 3	60.24	2079	6444	42.57	2285	7164	8583.24	9491.57
2500kbps run 4	61.19	2256	6480	43.46	1928	7200	8797.19	9171.46
2500kbps run 5	64.98	2137	6516	46.03	2061	7164	8717.98	9271.03
Averages	62.536	2226	6480	43.618	2084.2	7200	8768.536	9327.818
	dash server	**dash client**	**dash wireshark**	**hls server**	**hls client**	**hls wireshark**	**dash energy**	**hls energy**
6000kbps run 1	102.24	4512	17640	92.88	3348	17496	22254.24	20,936.88
6000kbps run 2	94.72	4583	17568	101.76	3076	17208	22245.72	20385.76
6000kbps run 3	97.39	4767	17712	97.62	3462	17316	22576.39	20875.62
6000kbps run 4	96.49	4675	17712	95.93	3169	17388	22483.49	20652.93
6000kbps run 5	101.7	4831	17568	93.48	3303	16992	22500.7	20388.48
Averages	98.508	4673.6	17640	96.334	3271.6	17280	22412.108	20647.934

Figure 1: Measurement Results