ISSN: 1692-7257 - Volume 1 – Number 41 - 2023

Revista Colombiana de
Tecnologías de Avanzada

Received: January 15th, 2023
Accepted: February 16th, 2023

AAASH ALGORITHM FOR VIDEO STREAMING SERVICE OVER MOBILE NETWORKS

ALGORITMO AAASH PARA EL SERVICIO DE VIDEOSTREAMING SOBRE REDES MÓVILES

PhD. Wilmar Yesid Campo-Muñoz*, **Ing. Juan Carlos Imbachi Paz***
Ing. Andrés Felipe Escobar Zapata *

* **Universidad del Quindío,** Facultad de Ingeniería, Programa de Ingeniería Electrónica.
Carrera 15 con calle 12 norte, Armenia, Quindío, Colombia.
Tel.: +57 (606) 7359300.
E-mail: wycampo@uniquindio.edu.co
E-mail:{afescobarz, jcimbachip}@uqvirtual.edu.co

How to cite: Campo Muñoz, W. Y., Imbachi Paz, J. C., & Escobar Zapata , A. F. (2023). ALGORITMO AAASH PARA EL SERVICIO DE VIDEOSTREAMING SOBRE REDES MÓVILES. REVISTA COLOMBIANA DE TECNOLOGIAS DE AVANZADA (RCTA), 1(41), 12–19. Retrieved from https://ojs.unipamplona.edu.co/index.php/rcta/article/view/2413

Copyright 2023 Revista Colombiana de Tecnologías de Avanzada (RCTA).
This work is licensed under an international license Creative Commons Atribución-NoComercial 4.0.

Abstract: The services supported by video streaming technology are the ones that generate the most traffic on mobile networks. One of the ways to deal with this situation is through the use of adaptive protocols, which allow the service traffic to be adapted to the network conditions in order to maintain quality conditions. So the Dynamic Adaptive Streaming over HTTP (DASH) arises. We in this paper applies Adaptation Algorithm for Adaptive Streaming over HTTP (AAASH) in clients of a cellular mobile network. The conditions of the experimental scenario are presented and the algorithm is executed in different scenarios increasing the number of users who consume video through DASH simultaneously. AAASH shows to be adequate, since quality of service parameters are obtained which satisfy the requirements.

Keywords: AAASH, mobile networks, QoS, video streaming.

Resumen: Los servicios soportados por la tecnología de videostreaming son los que más tráfico generan sobre las redes móviles. Una de las formas de hacer frente a esta situación, es mediante el uso de protocolos adaptativos, que permiten adecuar el tráfico del servicio a las condiciones de la red para mantener las condiciones de calidad. Por lo que surge el protocolo de transmisión adaptable y dinámica sobre HTTP (DASH - Dynamic Adaptive Streaming over HTTP). En este artículo se aplica el algoritmo adaptativo para streaming HTTP AAASH (Adaptation Algorithm for Adaptive Streaming over HTTP) en los clientes de una red móvil celular. Se presentan las condiciones del escenario de experimentación y el algoritmo se ejecuta en diferentes escenarios incrementando la cantidad de usuarios que consumen video mediante DASH de manera simultánea. AAASH muestra ser adecuado, puesto que se obtienen parámetros de calidad de servicio los cuales satisfacen los requerimientos.

Palabras claves: AAASH, QoS, red móvil, videostreaming.

1. INTRODUCTION

Currently, smartphones have capabilities comparable to those of a computer, allowing users to access various services, social networks, OTT (over-the-top) platforms, remote work, virtual education, and all types of content distribution offered by the Internet from anywhere. Among this content, video streaming stands out due to its high acceptance among end users and its impact on data networks, including mobile networks(Gadaleta, Chiariotti, Rossi, & Zanella, 2017), (Solano-Hurtado & Soto-Cordova, 2021). In this order of ideas, the 3GPP (3rd Generation Partnership Project) in its releases 12, considers the DASH (Dynamic Adaptive Streaming over HTTP), standard protocol for video transmission over cellular mobile networks (Campo-Muñoz, Andrés, Escobar-Zapata, Juan, & Imbachi-Paz, 2019).

The DASH protocol improves the perceived quality for the user by allowing access to different video content resolutions through the Hypertext Transfer Protocol (HTTP), according to network conditions (23009-1:2012, 2012). However, this is a generic protocol not specialized, for example, for wireless networks, which has led to the emergence of its private counterparts such as the protocol HTTP Smooth Streaming HSS, HTTP Live Streaming – HLS, HTTP Dynamic Streaming – HDS (Van Der Hooft et al., 2016), (Tian & Liu, 2016). In this research, the adaptation scheme for DASH is applied using the adaptive algorithm for streaming HTTP AAASH (Adaptation Algorithm for Adaptive Streaming over HTTP) (Vergados, Michalas, Sgora, & Vergados, 2014) that is applied directly to DASH, and the behavior of quality of service parameters over a wireless network is analyzed.

AAASH adapts the video speed requested by the client using a complex adaptation mechanism with multiple conditions and configuration parameters. The algorithm focuses on the receiver and ensures that the multimedia file is divided into segments (Vergados et al., 2014). Each of the segments is provided at multiple bitrates, and the client is capable of selecting the appropriate rate for each of the segments (Miller, Quacchio, Gennari, & Wolisz, 2012). This approach aims to: correctly estimate the dynamics of the available network performance, control the client's buffer fill level to avoid underflows resulting in playback interruptions, maximize the quality of the streaming while avoiding unnecessary quality fluctuations, and minimize the delay between the user request and the start of playback (Miller et al., 2012).

The algorithm begins by selecting the lowest representation for the first segment to minimize the delay between the user's request to watch the video and the actual start of playback. This phase of more aggressive quick start is introduced to rapidly increase to the best feasible quality. The quick start phase proceeds as follows: for each subsequent download, the next highest representation is selected, while its bitrate is lower than a certain percentage of the throughput measured in the last few seconds. The percentage varies depending on the buffer level in use (Vergados et al., 2014) (Miller et al., 2012).

This article analyzes the behavior of the AAASH algorithm on a mobile network emulated using the NS-3 research tool and the LENA module (LTE/EPC Network Simulator) (Sabbah, Jarwan, Issa, & Ibnkahla, 2018). Quality of Service (QoS) parameters are obtained, such as delay, jitter, and packet retransmission are obtained for scenarios with different numbers of users.

The rest of the article is organized as follows: Section 2 presents related works, which highlight the importance of the topic and mainly emphasize the differences with the research presented here. Section 3 outlines the conditions of the experimental scenario. Section 4 contains the results and their discussion, and finally, Section 5 presents the conclusions and future works derived from this research.

2. RELATED WORKS

In this section, different works are presented that highlight the importance of videostreaming and its approach from the DASH protocol. The research presented in this article arises as a response to one of the future works presented in (Campo-Muñoz et al., 2019), where the authors study the fuzzy DASH (FDASH) algorithm, which is implemented in an LTE network. They analyze different scenarios with varying numbers of users and study QoS parameters to verify that the algorithm meets the minimum QoS values. There are also other proposals, such as the one presented in (Gadaleta et al., 2017) where they propose the D-DASH variant, a framework that uses deep learning and reinforcement learning techniques to optimize the quality of experience (QoE) of DASH.

Adaptive streaming technologies are used to deliver video content over data networks, but they still face problems in networks with limited bandwidth, especially for ultra-high-definition or higher videos, even when using high-efficiency video coding. In (Concolato et al., 2018) the authors propose a solution to these issues based on three elements: mosaic encoding with motion restriction; a specific derivation of the ISO base media file format (International Organization for Standardization); and its use in adaptive streaming of mosaic video content using DASH. Similarly, in (Solano-Hurtado & Soto-Cordova, 2021) a performance evaluation of the DASH protocol for video streaming is carried out, but no improvements are proposed. In (Orozco, Muñoz, & Hoyos, 2021) the authors propose the characterization of a video streaming service supported by the DASH protocol for an LTE network. They are based on emulated environments where different scenarios are constructed, traffic traces are obtained, and the probability density functions that describe them are found.

In (Colonnese, Conti, Scarano, Rubin, & Cuomo, 2022), a bandwidth-aware allocation strategy is proposed for the video service supported by the DASH protocol, transparent to the client. The approach ensures smooth playback for all users and provides priority-based services for Premium users. This proposal can be adopted by each internet service provider, where it would need to be customized or adapted to different profiles, which could become much more cumbersome.

In (Atehortua, Yesid Campo, & Elías Chanchí, 2022) the authors propose a tool that estimates QoE for the video streaming service based on the DASH protocol, using QoS parameters according to the ITU-T P.1203 standard. This proposal is interesting as a starting point for service providers to take corrective actions in real-time to improve QoE.

According to the literature, there is extensive research involving the DASH protocol and its impact on QoS and QoE. Unlike these studies, this article applies the AAASH algorithm by implementing it on the client, making it extrapolable to other technologies such as 5G or WiFi networks. Additionally, QoS parameters are analyzed.

3. EXPERIMENTAL SCENARIO

For the experimentation scenario, the NS-3 research tool was used due to its open-source nature. The LTE network technology was chosen for being a mature technology and the one that currently supports the most traffic, and according to (Ericsson, 2022) it will support more than 50% of the total traffic by 2025. To construct the LTE scenario, the LENA library is used. For the implementation of the AAASH algorithm, it is necessary to specify the type of communication to be established on each client, in this caseTCP Sockets, using the “ns3::TcpSocketFactory” Class, and then define each of the application attributes, such as “targetDt” which corresponds to the Target Buffering Time (Tiempo de Búfer Objetivo) with an initial value of 35s, and the average period for throughput estimation (Window where 10s is used in this case (Campo-Muñoz et al., 2019). The H.264/MPEG-4 standard is employed for encoding. The Signal to Interference plus Noise Ratio (SINR) takes values between –5 dB and 20 dB. The operating frequency for the uplink is 1930 MHz and for the downlink is 2120 MHz, with a 5 MHz channel bandwidth. The model used within LENA is the lena-simple-epc (Malinverno et al., 2020), (NS-3 Project, 2023).

The QoS parameters analyzed are delay, jitter, and packet loss. The UE connects to an evolved Node B (eNB) of the LTE network, which corresponds to the Evolved-UMTS Terrestrial Radio Access Network (EUTRAN); and through the Evolved Packet Core (EPC) backbone network, to a DASH server (NS-3 Project, 2023).

Three scenarios are created, and the QoS parameters are analyzed for each. To perform the analysis of these three scenarios, a basic scenario with a single user was used as the starting point, from which additional scenarios are built by increasing the number of users.

4. RESULTS AND DISCUSSIONS

In this section, the results are presented by scenarios: the first is the basic one, consisting of a single user; Scenario II consists of 10 users, and Scenario III consists of 23 users.

able 1 presents the values for delay and delay fluctuation parameters in milliseconds (ms) and the percentage of packet loss (Handbook, 2020), (Cisco, 2017) against which the results of each scenario are compared. It is important to highlight that for the packet loss parameter, comparisons are made against retransmitted packets, as the DASH protocol at layer 4 of the OSI model operates with the TCP protocol. Therefore, packets that do not receive an acknowledgment (ack) are retransmitted.

Table 1: QoS parameters

Parameter	
Delay (retardo)	150 ms
Jitter (Fluctuación del retardo)	30 ms
Packet Loss (Paquetes pérdidos)	< 1 %

4.1 Scenario I

In the first scenario, a user acquires the video streaming service with a buffer size of 35 seconds, but with the characteristics previously specified in the experimentation scenario.

The simulation time is 500 seconds, which is sufficient to evaluate the behavior of the service over the AAASH protocol. In Fig. 1, the curve labeled NewBitrate and OldBitrate represent the user's bit rate at two points in time, according to how the AAASH algorithm operates. The performance does not correspond to a specific resolution, as the algorithm changes resolutions in order to maintain the performance, which is represented by the curve labeled estBitRate. The performance of the segment is calculated on the client side and corresponds to a ratio between a product and a difference. The product is obtained by multiplying the segment's bit rate by its duration, while the difference is obtained by subtracting the time when the entire segment has been received by the client from the time the segment begins downloading. Regarding performance, it is observed that the protocol takes 12 seconds to stabilize the bit rate at 1.21 Mbps.

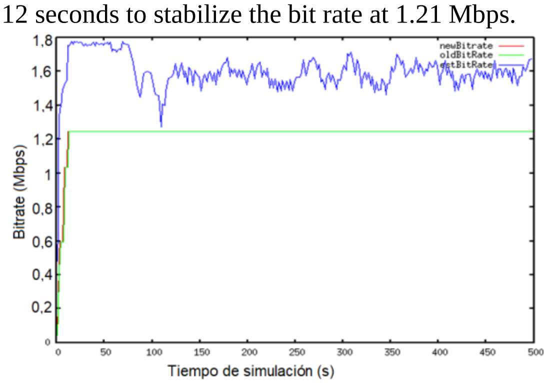

Fig. 1. Throughput Level of a User with the AAASH Algorithm

In Fig. 2, the buffer behavior for the AAASH algorithm is shown. In the first few seconds of the simulation, the buffer quickly increases to approximately 25 seconds during the first 10 seconds of playback. Then, it reaches a buffer value of 50 seconds between 1 second and 70 seconds of playback. Afterward, it decreases to 35 seconds of buffer at the 110-second mark of playback, where it remains stable, with slight fluctuations in its behavior.

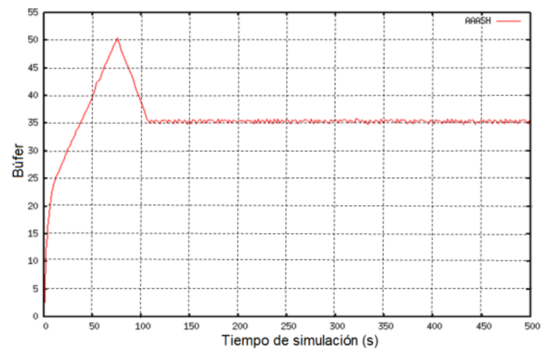

Fig. 2. Buffer level of a user with the AAASH algorithm

3.1.1 Delay for a user

For the calculation of the average delay, according to the NS-3 documentation and its FlowMonitor, it is calculated by dividing the delaySum by the received packets (rxPackets), whose values are shown in Fig. 3. Thus, once the calculations are performed, the average delay value is found to be 15.05 ms. In Fig. 4, the delay behavior associated with the number of received packets is observed, with the sum matching the received packets (rxPackets) value in Fig. 3.

<Flow timesForwarded="0" lostPackets="382" rxPackets="150700" txPackets="151089" rxBytes="86412410" txBytes="86636068" lastDelay="+15064648.0ns" jitterSum="+77657631353.0ns" delaySum="+2268284396931.0ns" timeLastRxPacket="+499986999999.0ns" timeLastTxPacket="+499971935351.0ns" timeFirstRxPacket="+284999999.0ns" timeFirstTxPacket="+271930314.0ns" flowId="2">

Fig. 3. Flow Monitoring

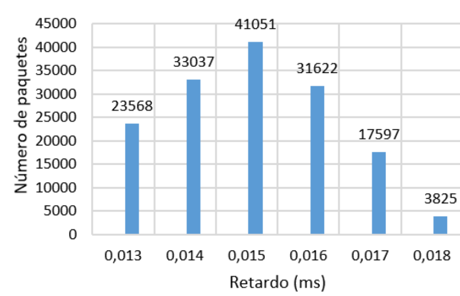

Fig. 4. Histogram of delay for a user

Comparing the delay values obtained for Scenario I with those specified in Table 1, the value of this parameter is met for this scenario. That is, it complies with the QoS parameter, as the packets with the highest delay are 0.018 ms lower than the 50 ms threshold.

4.1.2 Fluctuation of delay for a user

The average delay fluctuation, according to the NS-3 documentation and its FlowMonitor, is calculated by dividing the jitterSum by the number of received packets minus one (rxPackets), as shown in Fig. 3. After performing the calculations, the average fluctuation value is found to be 0.51 ms, meaning that most of the packets exhibit a delay fluctuation close to zero, as seen in Fig. 5. Fig. 5 shows the behavior of the delay fluctuation associated with the number of received packets, where the sum matches the received packets (rxPackets) from Fig. 3. In Fig. 5, no bar is visible for the delay fluctuation of 0.005 s due to the scale of the number of packets, as the number of packets associated with this value is only 16.

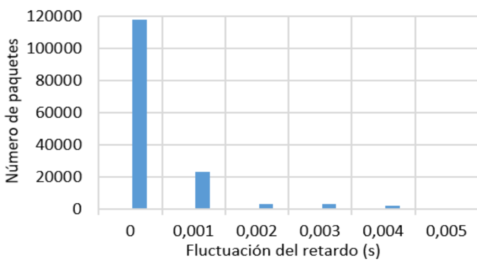

Fig. 5. Delay fluctuation for a user

According to Table 1, where values between 30ms and 50ms are set for video, the delay fluctuation for this scenario (0.51 ms) does not pose a problem, meaning it meets the QoS parameters.

4.1.3 Packet Retransmission

In this scenario, packet retransmission corresponds to what LENA refers to as lostPackets. It is important to remember that the transmission occurs at layer 4, using the TCP protocol, which is identified in the IP header as protocol 6 (protocol= "6" ver Fig. 6, this means that packets that are lost are subsequently retransmitted. The lost packet count is 382, as shown in Fig. 3. This value represents 0.25% of the total transmitted packets (txPackets). Therefore, it meets the QoS parameter as specified in Table 1.

```
<Ipv4FlowClassifier>
    <Flow flowId="2" destinationPort="49153" sourcePort="80" protocol="6"
        destinationAddress="7.0.0.2" sourceAddress="1.0.0.2"/>
```

Fig. 6. Transmission under the TCP protocol

4.2 Scenario II with 10 users

This section shows the behavior of the QoS parameters for 10 users simultaneously consuming a video streaming service with the AAASH algorithm.

4.2.1 Delay for 10 users

In Fig. 7, the delay for each of the 10 users is shown; this parameter represents the average value of the flows for each user. User 3 exhibits the highest value with 27.7 ms, while user 5 shows the lowest value with 27.3 ms. Therefore, the delay experienced by each user in this scenario is very similar. Furthermore, according to Table 1, the delay for Scenario II meets the QoS parameters.

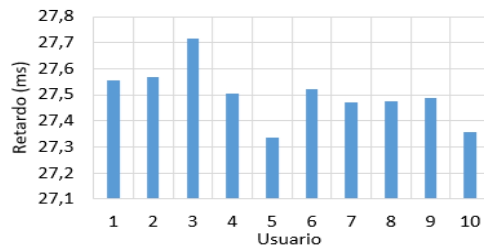

Fig. 7. Delay for 10 users

4.2.2 Fluctuation of delay for 10 users

In Fig. 8, the average delay fluctuation for each of the 10 users is shown. User 4 exhibits the highest delay fluctuation value, reaching 0.53 ms. According to Table 1, the delay fluctuation for this scenario meets the QoS parameters.

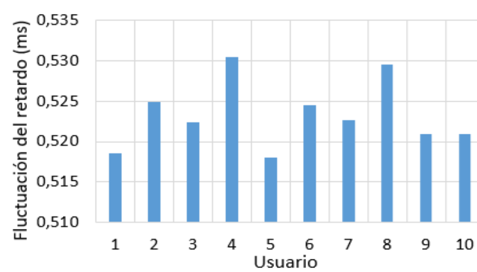

Fig. 8. Fluctuation of delay for 10 users

4.2.3 Packet retransmission

Fig. 9 shows the percentage of lost packets, retransmitted packets, and dropped packets. User 10 requires the highest number of retransmitted packets, totaling 1,056, which represents only 0.46% of the total packets. For instance, User 4 has only 816 retransmitted packets, accounting for 0.45% of the total packets. In any case, as shown in

Fig. 9, the percentage of retransmitted packets for each user is below 1%, thus meeting the QoS parameter as per Table 1.

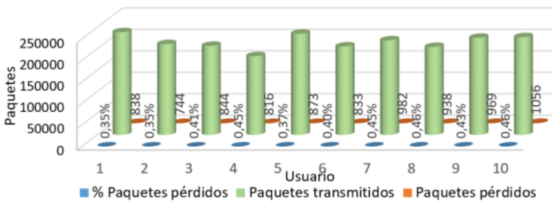

Fig. 9. Lost Packets for 10 Users

4.3 Scenario III with 23 users

In this section, the behavior of the QoS parameters for 23 users with the AAASH algorithm simultaneously consuming a video streaming service is shown. The number of 23 users in Scenario III is chosen because this is the maximum number allowed by the NS-3 tool, since exceeding this value causes the tool to malfunction.

4.3.1 Delay for 23 users

In Fig. 10, it can be observed how the delay for 23 users has increased for each of them. In the worst case, it has increased by 3.4 ms compared to the scenario with 10 users. Thus, it is noted that the lowest value is 30.52 ms for user 4, while the highest value is 30.77 ms for user 14. Therefore, the delay experienced by each user in this scenario is very similar. Additionally, according to Table 1 the delay for this scenario meets the QoS parameters, see Table 1.

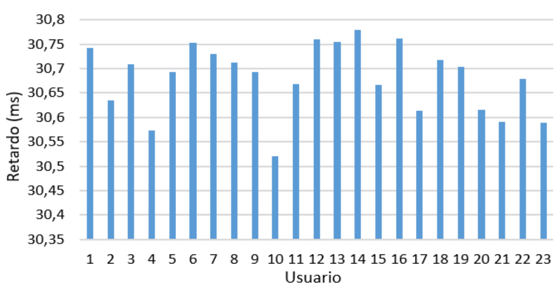

Fig. 10. Delay for 23 users

4.3.2 Jitter for 23 users

In Fig. 11, the jitter for each of the 23 users is shown, which has increased across all users. In the worst case, it increased by 0.05 ms compared to the values recorded for 10 users. The maximum value is presented by user 16, with a jitter of 0.582 ms. According to Table 1, the jitter for this scenario complies with the QoS parameters, see Table I.

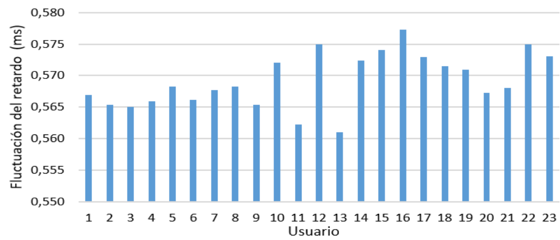

Fig. 11. Jitter for 23 users

4.3.3 Packet Retransmission

Figure 12 presents the percentage of lost packets, retransmitted packets, and lost packets. Thus, user 21 is the one who requires the highest number of retransmissions, totaling 1,391 packets, which represents 1.47% of the total. Comparing Figures 9 from Scenario II with 10 users versus Figure 12 from Scenario III with 23 users, an increase in the number of retransmissions and the percentage they represent is observed. This indicates that as the number of users increases, the QoS parameters deteriorate.

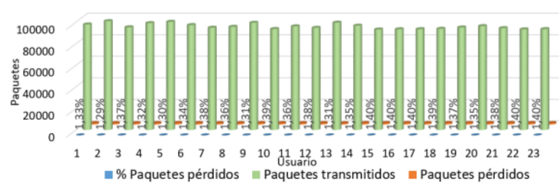

Fig. 12. Lost packets for 23 users

In principle, if we compare the packet loss percentage for each of the 23 users, we observe that the value exceeds the 1% reference threshold in Table 1. Therefore, none of the users would be meeting this QoS parameter. However, it is important to clarify that we are working with the DASH protocol, which operates at the transport layer (Layer 4) of the OSI model using the TCP protocol. As a result, what actually occurs is packet retransmission, not loss. This explains the importance of the AAASH buffer of 35 seconds, meaning there is no actual packet loss, and the effects of the retransmission should be absorbed by the buffer. Additionally, it should be noted that the experiments are conducted over a 5 MHz channel, which can support peak speeds ranging from 1.6 Mbps to 21 Mbps (Agusti et al., 2010) therefore, each of the 23 users could transmit the video at a very low rate, between 69.5 Kbps and 913 Kbps, lower than the ideal value of 1.21 Mbps calculated for Scenario I.

5. CONCLUSIONES

This article presents the AAASH algorithm on the client side of a cellular mobile network in order to analyze its performance as the number of users increases while consuming a video streaming service under the DASH protocol. The results show that the algorithm allows for compliance with QoS parameters despite the increase in users consuming the service simultaneously.

It is important to highlight the behavior of the delay fluctuation parameter, as according to Table 1, the maximum allowed value is 30 ms. However, the maximum reached was only 0.58 ms, which represents only 1.9% of the maximum value. Similarly, in the case of delay, the maximum allowed value is 150 ms, but the maximum value reached was 30.75 ms, representing only 20% of the maximum value.

Packet loss is the most critical parameter, as in the scenario with 23 users, the specifications in Table 1 are not met. However, this behavior should be interpreted from two perspectives: i) it is important to remember that the DASH protocol prevents packet loss when transported over TCP. Therefore, the user's device needs to have enough memory capacity to support a 35-second buffer, which, given the current capabilities of devices, seems like an easily achievable requirement; ii) Scenario III exceeds the capacity since it does not support the minimum required bit rate of 1.21 Mbps, which was calculated for Scenario I.

Based on the results, it can be concluded that the AAASH algorithm, when executed on the client side, is a valid option to support video streaming in a cellular mobile network.

As future work, it is suggested to study the QoE for these same scenarios to understand the effect of packet retransmissions and determine whether stalling events occur.

ACKNOWLEDGMENTS

The authors of this article would like to express their sincere gratitude to the University of Quindío for the support provided in carrying out this research.

REFERENCES

23009-1:2012, I. (2012). ISO - ISO/IEC 23009-1:2012 - Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats. Retrieved August 3, 2022, from https://www.iso.org/standard/57623.html

Agusti, R., Bernardo, F., Casadevall, F., Ferrús, R., Pérez-Romero, J., & Sallent, O. (2010). LTE: NUEVAS TENDENCIAS EN COMUNICACIONES MÓVILES AUTORES.

Atehortua, J., Yesid Campo, W., & Elías Chanchí, G. (2022). Proposal of a tool for the automatic estimation of quality of experience in MPEG-DASH scenarios. *Przeglad Elektrotechniczny*. https://doi.org/10.15199/48.2022.08.26

Campo-Muñoz, W. Y., Andrés, Escobar-Zapata, F., Juan, & Imbachi-Paz, C. (2019). Análisis del servicio de video streaming basado en el algoritmo FDASH sobre LTE. *Ciencia e Ingeniería Neogranadina*, *29*(1), 67–80. https://doi.org/10.18359/RCIN.3122

Cisco. (2017). Video Quality of Service (QOS) Tutorial - Cisco. Retrieved April 1, 2023, from https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-video/212134-Video-Quality-of-Service-QOS-Tutorial.html

Colonnese, S., Conti, F., Scarano, G., Rubin, I., & Cuomo, F. (2022). Premium quality or guaranteed fluidity? Client-transparent DASH-aware bandwidth allocation at the radio access network. *Journal of Communications and Networks*, *24*(1), 59–67. https://doi.org/10.23919/JCN.2021.000046

Concolato, C., Le Feuvre, J., Denoual, F., Mazé, F., Nassor, E., Ouedraogo, N., & Taquet, J. (2018). Adaptive Streaming of HEVC Tiled Videos Using MPEG-DASH. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(8), 1981–1992. https://doi.org/10.1109/TCSVT.2017.2688491

Ericsson. (2022). Past and present reports - Mobility Report - Ericsson. Retrieved April 1, 2023, from https://www.ericsson.com/en/reports-and-papers/mobility-report/reports

Gadaleta, M., Chiariotti, F., Rossi, M., & Zanella, A. (2017). D-DASH: A Deep Q-Learning Framework for DASH Video Streaming. *IEEE Transactions on Cognitive Communications and Networking*, *3*(4), 703–718. https://doi.org/10.1109/TCCN.2017.2755007

Handbook, C. Q. (2020). Cisco QoS Handbook 2nd Edition.

Malinverno, M., Mangues-Bafalluy, J., Casetti, C. E., Chiasserini, C. F., Requena-Esteso, M., & Baranda, J. (2020). An edge-based framework for enhanced road safety of connected cars. *IEEE Access*, *8*, 58018–58031. https://doi.org/10.1109/ACCESS.2020.2980902

Miller, K., Quacchio, E., Gennari, G., & Wolisz, A. (2012). Adaptation algorithm for adaptive streaming over HTTP. *2012 19th International Packet Video Workshop, PV 2012*, 173–178. https://doi.org/10.1109/PV.2012.6229732

NS-3 Project. (2023). ns-3: src/lte/examples/lena-simple-epc.cc Source File. Retrieved April 1, 2023, from https://www.nsnam.org/docs/release/3.19/doxygen/lena-simple-epc_8cc_source.html

Orozco, H. F. B., Muñoz, W. Y. C., & Hoyos, E. A. (2021). Revista Ingeniería. *Ingeniería*, *26*(3), 381–400. https://doi.org/10.14483/23448393.17960

Sabbah, A., Jarwan, A., Issa, O., & Ibnkahla, M. (2018). Enabling LTE emulation by integrating CORE emulator and LTE-EPC Network (LENA) Simulator. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, *2017-October*, 1–6. https://doi.org/10.1109/PIMRC.2017.8292642

Solano-Hurtado, L. R., & Soto-Cordova, M. M. (2021). A Study on Video Streaming Quality of DASH Scheme in Multimedia Services. *2021 7th Congreso Internacional de Innovacion y Tendencias En Ingenieria, CONIITI 2021 - Conference Proceedings*. https://doi.org/10.1109/CONIITI53815.2021.9619720

Tian, G., & Liu, Y. (2016). Towards Agile and Smooth Video Adaptation in HTTP Adaptive Streaming. *IEEE/ACM Transactions on Networking*, *24*(4), 2386–2399. https://doi.org/10.1109/TNET.2015.2464700

Van Der Hooft, J., Petrangeli, S., Wauters, T., Huysegems, R., Alface, P. R., Bostoen, T., & De Turck, F. (2016). HTTP/2-Based Adaptive Streaming of HEVC Video over 4G/LTE Networks. *IEEE Communications Letters*, *20*(11), 2177–2180. https://doi.org/10.1109/LCOMM.2016.2601087

Vergados, D. J., Michalas, A., Sgora, A., & Vergados, D. D. (2014). A fuzzy controller for rate adaptation in MPEG-DASH clients. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, *2014-June*, 2008–2012. https://doi.org/10.1109/PIMRC.2014.7136501