



Análisis del servicio de *video streaming* basado en el algoritmo FDASH sobre LTE

Wilmar Yesid Campo-Muñoz^a ■ Andrés Felipe Escobar-Zapata^b
■ Juan Carlos Imbachi-Paz^c

Resumen: El *video streaming* es el servicio que mayor porcentaje de tráfico genera con respecto al total de datos en las redes móviles de evolución de largo plazo LTE (por sus siglas en inglés). Por otra parte, el protocolo de Streaming Adaptativo Dinámico sobre HTTP (DASH, por sus siglas en inglés) ha sido seleccionado para la transmisión del video en las redes LTE. Así, en este artículo, se presenta el análisis del comportamiento del servicio de video mediante una implementación de DASH bajo lógica difusa denominada FDASH (*fuzzy* DASH) en una red LTE, considerando diferente número de usuarios y traspasos (*handover*). Se analizan parámetros de calidad de servicio, como retardo, pérdida de paquetes, variación de pérdida de paquetes y rendimiento. Mediante este trabajo, se presenta FDASH como opción para el consumo del servicio de video, ya que permite adaptarse a las diferentes condiciones y exigencias de la tecnología inalámbrica móvil LTE.

Palabras clave: algoritmo FDASH, calidad de servicio, LTE, *video streaming*.

Fecha de recepción: 24 de octubre de 2017 **Fecha de aprobación:** 2 de octubre de 2018

Cómo citar: W. Y. Campo-Muñoz, A. F. Escobar-Zapata y J. C. Imbachi-Paz, "Análisis del servicio de *video streaming* basado en el algoritmo FDASH sobre LTE", *Ciencia e Ingeniería Neogranadina*, vol. 29(1), pp. 67-80. <https://doi.org/10.18359/rcin.3122>

- ^a Universidad del Quindío, Colombia. e-mail: wycampo@uniquindio.edu.co.
ORCID: <https://orcid.org/0000-0001-8585-706X>
- ^b Universidad del Quindío, Colombia. e-mail: afescobarz@uqvirtual.edu.co.
ORCID: <https://orcid.org/0000-0001-7351-3884>
- ^c Universidad del Quindío, Colombia. e-mail: jcimbachip@uqvirtual.edu.co.
ORCID: <https://orcid.org/0000-0002-0612-7315>

Assessment of Video Streaming Service based on the FDASH algorithm over LTE

Abstract: Video Streaming is the service that generates the highest percentage of data traffic in long-term evolution (LTE) mobile networks. Additionally, most video transmissions over LTE networks use the Dynamic Adaptive Streaming over HTTP (DASH) protocol. Hence, this study assesses the behavior of video service through DASH implementation based on fuzzy logic known as *Fuzzy* DASH (FDASH) over LTE networks, considering different numbers of users and handovers. This paper also addresses various quality of service parameters such as delay, packet loss, packet loss variation, and performance. Herein, FDASH is presented as a viable video service consumption option because it adapts streaming to the different conditions and requirements of LTE mobile wireless technology.

Keywords: FDASH Algorithm, Quality of Service, LTE, Video Streaming.

Análise do serviço de video streaming baseado no algoritmo FDASH sobre LTE

Resumo: Video Streaming é o serviço que gera a maior porcentagem de tráfego de dados nas redes móveis de evolução de longo prazo (LTE, pela sigla em inglês). Por outro lado, o protocolo de Streaming Adaptativo Dinâmico sobre HTTP (DASH, pela sigla em inglês) tem sido selecionado para a transmissão de vídeo nas redes LTE. Desta forma, neste artigo, está sendo apresentada uma análise do comportamento de serviço de vídeo mediante a implementação de DASH, baseada na lógica difusa, conhecida como FDASH (Fuzzy DASH) em uma rede LTE, considerando diferentes números de usuários e transferências (*handover*). Foram analisados parâmetros de qualidade de serviço como atraso, perda de pacotes, variação de perdas de pacote e rendimento. Com base neste trabalho, o FDASH se apresenta como opção para o consumo de serviço de vídeo, uma vez que permite adaptar-se às diferentes condições e exigências da tecnologia móvel sem fio LTE.

Palavras chave: algoritmo FDASH, qualidade de serviço, LTE, *video streaming*.

Introducción

La alta demanda y la globalización que se observa en las comunicaciones a través de dispositivos móviles han permitido que el mercado y el consumo de servicios tengan una nueva dinámica, impulsados también por el crecimiento de la población, las tecnologías de la información y de la comunicación (TIC) y los nuevos servicios. En las redes móviles, el porcentaje de tráfico de video con respecto al total de datos móviles fue del 60% en 2016 y se espera que supere el 78% para 2021 [1]. Por tanto, para apoyar estos pronósticos de tráfico, el sistema inalámbrico 3GPP LTE (3rd Generation Partnership Project Long Term Evolution) ha sido introducido. Este sistema permite transmisiones con alta velocidad de datos y baja latencia de extremo a extremo, que son requisitos clave en la tecnología *streaming* para el servicio de video bajo demanda, que consume grandes recursos de ancho de banda independiente del tamaño del archivo por transmitir.

El 3GPP Release 12 para la tecnología de LTE ha seleccionado el estándar Streaming Adaptativo Dinámico sobre HTTP (DASH, por sus siglas en inglés) [2] para la transmisión del video. Dicho estándar realiza una transmisión de bits adaptativa a diferentes velocidades, una técnica que permite manejar o adaptarse al ancho de banda, mediante la variación de parámetros de calidad de servicio QoS (Quality of Service, por sus siglas en inglés) en la red [3]. El protocolo DASH ha ganado la atención significativa por su capacidad de reproducir contenidos desde cualquier dispositivo gracias a estar soportado sobre protocolo de transferencia de hipertexto HTTP (por sus siglas en inglés). Además, mediante DASH un cliente de video logra saber su ancho de banda disponible y selecciona la velocidad de bits apropiada para el contenido de video, con el fin de maximizar su calidad.

Por otra parte, la fluctuación en el ancho de banda disponible en las redes inalámbricas móviles plantea un reto para el algoritmo de selección de velocidad de bits. En [4], se propone utilizar la regularidad en el movimiento de un usuario para mejorar el algoritmo de selección de velocidad de bits usando DASH. Sin embargo, para asegurar la QoS de *video streaming* en redes inalámbricas, es

necesario considerar fenómenos como la cobertura de la señal inalámbrica y la inestabilidad del canal inalámbrico, además de los parámetros de desempeño, como ancho de banda, retardo, rendimiento (*throughput*) y pérdida de paquetes [5].

En este artículo, se presenta un análisis sobre el comportamiento del servicio de video bajo la tecnología de *streaming* soportado por el protocolo DASH en redes LTE mediante la implementación del algoritmo de lógica difusa FDASH (fuzzy DASH) [6]. El algoritmo FDASH realiza la adaptación de las tramas de resolución de video para su transmisión de acuerdo con las condiciones del canal del usuario, con el objetivo de tomar la mejor decisión en cuanto a qué trama genera la resolución más adecuada [7].

Así, el aporte de este artículo se basa en evaluar el comportamiento del algoritmo FDASH para soportar la QoS del *video streaming* sobre una red LTE. Para esto, es necesario la puesta en funcionamiento de dicho algoritmo en los UE (User Equipment, por sus siglas en inglés) y la puesta en funcionamiento de un servidor DASH capaz de interoperar con la red troncal EPC (Evolved Packet Core, por sus siglas en inglés) de LTE. La metodología parte de la construcción de un modelo básico, que es simulado y validado, a partir del que se construyen los otros escenarios que se basan en el incremento en el número de usuarios y en el soporte de trasposos (*handover*).

De acuerdo con la literatura revisada se identificaron tres herramientas para realizar esta investigación Opnet Modeler, OMNeT++ y ns-3. La primera de estas es flexible y escalable a través de modelos jerárquicos, que representan estructuras de redes reales y puede ser implementada en sistemas operativos tipo Unix o Windows; sin embargo, el *software* es licenciado [8]. OMNeT++ es un simulador bien organizado y flexible, aunque posee reportes bastante pobres de los resultados de la simulación, por lo que los usuarios deben implementar mucho de los códigos para obtener las medidas [9]. El *software* ns-3 surge como un proyecto de código libre orientado hacia el desarrollo y la investigación de redes, en especial inalámbricas y está dirigido hacia la comunidad académica

dedicada a la investigación [6]. La comunidad científica ha desarrollado el módulo LTE para ns-3 que se denomina LENA (LTE/EPC Network Simulator). LENA está construido completamente en el lenguaje de programación C++ y ha sido fusionado desde la versión 10 de ns-3 [10].

El *software* seleccionado es ns-3 mediante LENA que corre sobre el sistema operativo GNU/Linux gracias a las siguientes características. Primero, por ser de carácter investigativo, ya que la herramienta ha sido utilizada en diferentes proyectos en los que se utiliza LENA, por ejemplo, en [11] se presenta un desarrollo para la simulación de LTE con ns-3 que se centra, principalmente, en el modelado de la parte E-UTRAN (Evolved Terrestrial Radio Access Network, por sus siglas en inglés) del sistema, con atención en los aspectos relacionados con canal, capas PHY (Physical Layer, por sus siglas en inglés) y MAC (Medium Access Control, por sus siglas en inglés); en [12] se presenta el análisis de los efectos de la elección del índice de calidad de canal en la transmisión de la información; en [13] los autores presentan una metodología y la puesta en funcionamiento de un sistema emulado capaz de recibir tráfico en tiempo real sobre una red simulada de LTE; en [14] se realizan estudios sobre el canal radio y su comportamiento con tecnología MIMO (Multiple-input Multiple-output, por sus siglas en inglés) que permite incrementar las velocidades en LTE. Segundo, por su tipo de licencia, ya que es de carácter de código libre, debido a esto se tiene una serie de libertades, como ejecutar el programa para cualquier propósito, estudiar cómo funciona, adecuarlo a la forma que se desee y distribuir el código de un usuario a otro. Tercero, por la existencia de LENA que permite el estudio de LTE sobre ns-3. Cuarto, por la capacidad de graficar los resultados, esta herramienta posee módulos para la generación de estadísticas gráficas que pueden ser manipuladas desde el mismo simulador o exportadas a algún procesador especializado [15].

Este artículo está organizado de la siguiente forma: en la sección uno, se describe el estándar para la transmisión de video en LTE; en la sección dos, se estudian los escenarios de experimentación; en la sección tres, se presenta el análisis de

resultados; en la sección cuatro, se ofrecen las conclusiones, seguido de los agradecimientos; y por último, las referencias.

Estándar para la transmisión de video en LTE DASH

LTE es un estándar para comunicaciones móviles de transmisión de datos de alta velocidad desarrollado por el 3GPP [16]. Esta tecnología es la primera de carácter inalámbrico para redes de área amplia (Wide Area Network, WAN, por sus siglas en inglés) en la que todos los servicios, incluida la voz, se soportan sobre el IP (Internet Protocol, por sus siglas en inglés) y sus velocidades pico de la interfaz radio se sitúan dentro del rango de 100 Mb/s y 1 Gb/s, ampliamente superiores a las conseguidas en los sistemas predecesores [17]. Asimismo, esta tecnología promete romper definitivamente las barreras que todavía impedían la consecución plena de una movilidad con capacidad multimedia [18].

Así, para el servicio multimedia de *video streaming*, el 3GPP, en su especificación técnica de LTE Advanced TS 26.234 v12.5.0 Release 12 [2], define DASH como el mecanismo adoptado para el soporte del servicio de *video streaming*. DASH se puede definir como un protocolo por el que se proporcionan formatos que habilitan la entrega eficiente y de alta calidad de servicios de *streaming*, que puede ser desplegado sobre la infraestructura actual de las redes de distribución HTTP, que permite seleccionar la calidad en función de la red y capacidad del dispositivo, en el que los cambios entre calidades son automáticos y transparentes al usuario, y que puede coexistir con tecnologías propietarias existentes [19]. Por lo anterior, una vez el canal radio se ve afectado por fenómenos como el desvanecimiento o el multirayecto, sus efectos son considerados por DASH para adaptar la calidad del video.

Por otra parte, el algoritmo FDASH promete ajustar de forma eficaz la velocidad de video entregada a cada usuario de acuerdo con las condiciones de red en cada momento, lo que se adapta a las mayores exigencias de las tecnologías inalámbricas móviles como LTE. Así, en este artículo,

los UE o dispositivos móviles son los encargados de soportar el algoritmo FDASH que implementa la norma MPEG-DASH (Moving Picture Expert Group, por sus siglas en inglés) para solicitar flujos de información desde un servidor de video DASH, que debe estar en la red troncal de LTE. Cada UE utiliza el algoritmo de adaptación de velocidad mediante el controlador difuso FDASH, para estimar la resolución del segmento de video siguiente. El algoritmo ejecuta el control del tiempo de *buffer* y la resolución del video para distribuir la posible mejor resolución de los segmentos de video y la entrega sin interrupciones en la reproducción del video sin subdesbordamientos de amortiguamiento en el cliente. Además, evita cambios innecesarios de la resolución de video, debido a las fluctuaciones frecuentes del rendimiento de la conexión disponible [6].

En FDASH, cada UE solicita el siguiente segmento de video con mayor o menor resolución en referencia al último segmento descargado por este. Un segmento corresponde a una sección de 2 s de video almacenada en el servidor DASH con una resolución que puede variar desde un valor bajo de tasa de bit (bit rate) de 45 000 bps hasta un valor máximo de tasa de bit de 4 220 000 bps. La decisión del segmento de video que deberá descargar el UE la realiza el algoritmo FDASH que usa dos entradas, una de las cuales corresponde al tiempo que demora el último segmento recibido por el UE hasta el instante en que empieza a reproducirse, esto es, el tiempo de almacenamiento en el *buffer*, y la segunda entrada corresponde a la diferencia entre el tiempo de *buffer* actual y la del segmento anterior. Con estos datos, el algoritmo obtiene una salida que es el tasa de bit del siguiente segmento de video que debe solicitarse al servidor DASH. Además, FDASH controla los cambios consecutivos de resolución sujetos a variaciones continuas del rendimiento de la red, que provocarían una mala experiencia de video para el usuario, por tanto, los cambios no pueden ser de forma rápida sino de forma suave [6].

FDASH se basa en el concepto de *lógica difusa*, que permite tomar una decisión a partir de información indeterminada y aproximada. Se asume

que un flujo de video consiste en n segmentos de duración τ , que están disponibles en el servidor. Cada segmento está codificado en múltiples resoluciones de calidad. El rendimiento del segmento se calcula en el cliente según la ecuación (1) [6].

$$r_i = (b_i \times \tau) / (t_i^e - t_i^b) \quad (1)$$

Donde b_i denota la tasa de bits del segmento i y t_i^b y t_i^e respectivamente el momento en que el segmento i ha iniciado la descarga y el tiempo en que todo el segmento ha sido recibido por el cliente.

El aumento y la disminución de la resolución se determina por el algoritmo de adaptación de velocidad con controlador difuso. El algoritmo utiliza dos entradas suaves, es decir, el tiempo de autonomía en que el último segmento recibido espera en el cliente hasta que se inicia la reproducción y la diferencia del último tiempo de autonomía. FDASH trata de mantener el tiempo de amortiguación en el cliente por encima del tiempo de almacenamiento en el *buffer* de destino para evitar una caída y reducir cambios consecutivos de resolución de video sujetas a variaciones continuas del rendimiento de la red. Específicamente, las variables lingüísticas de la entrada del tiempo de almacenamiento del *buffer* se describen como *short*, *close* y *long*, para indicar la distancia del tiempo de almacenamiento temporal actual desde el objetivo T. Además, las variables lingüísticas para el diferencial de la entrada de tiempo de almacenamiento significan que la tasa entre los tiempos de almacenamiento está cayendo (*falling*), es estable (*steady*) o está subiendo (*rising*), mientras que las variables lingüísticas de la salida se describen como reducir (R), incremento pequeño (SI) y aumento (I), que declaran el factor de la disminución o el aumento de la resolución del siguiente segmento. Los valores lingüísticos de las variables de entrada y de salida están representados por números difusos triangulares; para más detalles, se puede consultar [7].

Las reglas que asocian la salida difusa para las entradas difusas podrían ampliarse a la sentencia if-then-else con las siguientes 9 reglas (r1...r9): Reglas (r1): if (short) and (falling) then R; (r2): if (close) and (falling) then SR; (r3): if (long) and (falling) then NC; (r4): if (short) and (steady) then SR; (r5):

if (close) and (steady) then NC; (r6): if (long) and (steady) then SI; (r7): if (short) and (rising) then NC; (r8): if (close) and (rising) then SI y (r9): if (long) and (rising) then I.

En esta investigación, se implementan los segmentos de video para diferentes resoluciones con las que el servidor DASH provee sus contenidos a los UE en que se implementa el controlador difuso, el video utilizado corresponde a BigBuckBunny presentado en [20].

Escenario de experimentación

Este estudio se basa en tres escenarios: los dos primeros se diferencian en el número de usuarios y el tercero considera la característica de traspaso (*handover*), en que se obtienen los parámetros de QoS para el algoritmo FDASH. Los parámetros de QoS para la transmisión de video debe ser menor de 0,240 s para el retardo y una pérdida de paquetes menor del 5% [21]. En esta investigación, los paquetes de *video streaming* están marcados con valores de punto de código de servicios diferenciados (DSCP, por sus siglas en inglés) [22] por parte del servidor DASH. Las resoluciones de video con las que se trabaja son 1920×1080 , 960×540 , 848×480 , 640×480 , 640×360 , 480×270 , 424×320 , 320×180 , 424×240 .

La construcción de los escenarios de experimentación inicia con el despliegue de un escenario básico consistente en una red LTE con un único usuario. En la figura 1, se observa el escenario básico construido en ns-3, que consta de un UE que consume el servicio de *video streaming*. El UE, a través de la interfaz aire Uu, con frecuencia de operación para el enlace de subida de 1930 MHz y para el enlace de bajada de 2120 MHz y una canalización de 5 MHz, accede a un Nodo B evolucionado (eNB) de la red LTE que corresponde a la red de accesos radio terrestre universal evolucionada EUTRAN; y a través de la red troncal EPC, a un servidor DASH [23]. El modelo usado dentro de LENA es el lena-simple-epc [23].

En el segundo escenario de simulación, se incrementa el número de UE a 10 y 23 usuarios. Los UE están conectados a un eNB de la red LTE; y a través del EPC, al servidor DASH. El número de

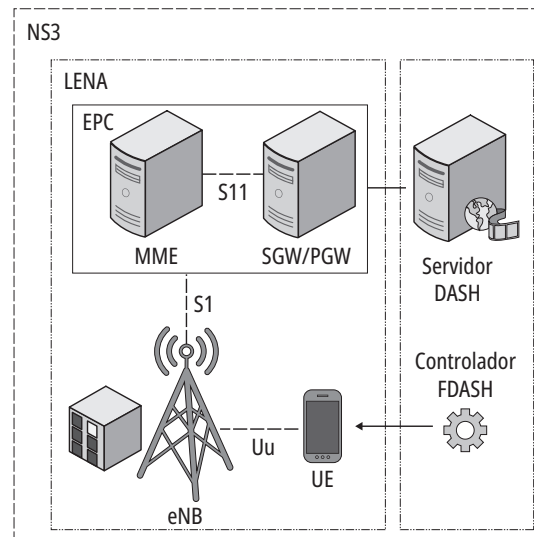


Figura 1. Escenario base LTE-LENA con DASH y FDASH. Fuente: Elaboración propia.

usuarios de 23 se escoge, ya que este es el máximo que permite el simulador, puesto que al superar dicho valor la simulación falla. La falla se presenta por el diseño del código LENA, que asocia este número máximo de 23 host por cada eNB. El tercer escenario considera la característica de *handover* con 1, 10 y 23 usuarios considerando un clúster de 3 celdas.

Mediante el módulo Flow Monitor de la herramienta de investigación ns-3 se obtienen los resultados de los escenarios de experimentación que arroja LENA. Este módulo proporciona un sistema flexible para medir el rendimiento de los protocolos de red. El módulo usa sondas, instaladas en nodos de red, para rastrear los paquetes intercambiados por los nodos y generar medidas de diferentes parámetros. Además, es posible acceder a las sondas directamente para solicitar estadísticas específicas sobre cada flujo. Las estadísticas de LENA se recogen para cada flujo y se exportan en formato XML (eXtensible Markup Language, por sus siglas en inglés). Por otra parte, puesto que el algoritmo FDASH y el servidor DASH no hacen parte de LENA, sus resultados son arrojados directamente por consola. Así, estas dos salidas (XML y consola) son procesadas para generar las diferentes curvas del comportamiento del servicio (figura 2).

El módulo datosNodo.sh de la figura 2 es un batch script en GNU/Linux, que fue escrito para

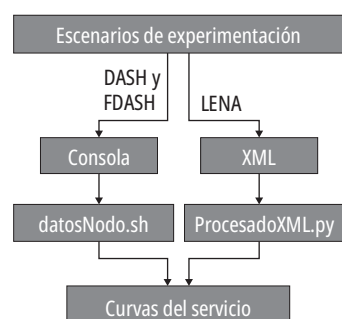


Figura 2. Proceso de obtención de resultados. Fuente: Elaboración propia.

permitir exportar a un fichero con extensión .txt los datos obtenidos en la salida estándar de la ejecución de los códigos de simulación de los escenarios en ns-3. Además, este módulo permite llamar a Gnuplot para graficar los resultados. El script genera las curvas para el comportamiento del tasa de bit y el *buffer*. El módulo *procesadoXML.py* fue escrito para permitir acceder a los archivos XML generados por LENA y extraer los datos. Este proceso de extracción de los datos para obtener los resultados se aplica a cada uno de los escenarios.

Análisis de resultados

Los parámetros de QoS sobre el protocolo FDASH en la red LTE estudiados son el rendimiento (*throughput*), el retardo, la fluctuación del retardo (*jitter*) y la pérdida de paquetes. Para la codificación, se emplea el estándar de video avanzado H.264/MPEG-4. La relación señal a ruido más interferencia (SINR) toma valores entre -5 dB y 20 dB. El tiempo de simulación es de 500 s; se asigna este tiempo, puesto que el comportamiento de las curvas se repite después de este valor.

Escenario uno con un usuario

Los resultados del algoritmo FDASH para el escenario 1 se aprecian en la figura 3, esto es, las curvas que arroja el script *datosNodos.sh*. La curva de color azul representa el rendimiento. Las curvas de color verde (bitrate nuevo, *NewBitrate*) y rojo (bitrate antiguo, *OldBitrate*) representan la tasa de bits del usuario en dos instantes, según como trabaja el algoritmo FDASH (ecuación (1)). El rendimiento no corresponde a una resolución en

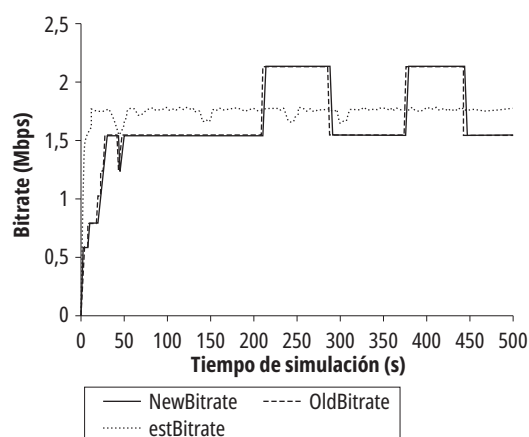


Figura 3. Rendimiento de un usuario con algoritmo FDASH. Fuente: Elaboración propia.

particular, puesto que lo que hace el algoritmo es cambiar las resoluciones con tal de mantener precisamente el rendimiento.

El tamaño por defecto del *buffer* del algoritmo FDASH es de 35 s. Este tiempo de *buffer* varía según la tasa de bits recibida, además que pretende estabilizarse para obtener una definición de la imagen correcta. Al analizar el rendimiento en la figura 3, se observa que el UE comienza a aumentar el bitrate hasta un tiempo de 35 s en que la resolución del video aumentará gracias al protocolo FDASH. Sin embargo, a los 47 s, tiene que reducir el envío de información, debido a la tasa de bits, para luego estabilizarla sin que se produzcan pérdida de paquetes ni se detenga la reproducción del video. La tasa de bits se estabiliza en 1,75 Mbps.

En la figura 4, se observa el comportamiento de nivel del *buffer*, en que en los primeros 40 s de simulación aproximadamente se incrementa hasta el valor de 35 s, para estabilizar, pero por la estimación de la velocidad de bits del algoritmo sigue aumentando, por lo que se obtiene un valor máximo de 57 s en tamaño de *buffer* a los 200 s de simulación. A partir de este tiempo, tiene un comportamiento oscilante, que es resultado de la lógica del algoritmo FDASH.

Los procesos de cálculo según la información arrojada por LENA en los archivos XML es la presentada en las ecuaciones (2) y (3), para los valores promedios de retardo y variación del retardo (*jitter*), respectivamente.

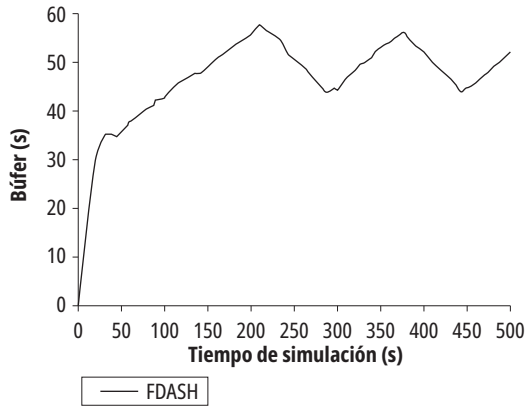


Figura 4. Nivel del *buffer* de un usuario con algoritmo FDASH. Fuente: Elaboración propia.

$$\overline{\text{Retardo}} = \frac{\text{delaySum}}{\text{rxPackets}} \quad (2)$$

$$\overline{\text{Jitter}} = \frac{\text{jitterSum}}{\text{rxPackets}-1} \quad (3)$$

Así, el retardo promedio para este escenario básico es de 19 ms y el *jitter* promedio es de 0,94 ms. Estos valores son inferiores a los exigidos para QoS para el servicio de *video streaming*, cuyo valor máximo es de 240 ms para el retardo [21], y el valor del *jitter*, al ser menor de los 50 ms recomendado por Cisco en [24] puede ser compensado con el *buffer*. Por otra parte, otro parámetro para medir la QoS es la pérdida de paquetes que, en este escenario, fue del 0% que no debe ser superior del 5% para el servicio de *video streaming* [21]. De acuerdo con lo anterior, el escenario uno cumple con los parámetros establecidos para QoS, según los parámetros dados en [21] y [24] para el *video streaming*.

En los siguientes escenarios, se presentan para cada uno de los parámetros estudiados los valores promedio de todos los usuarios, el promedio por usuario y el promedio móvil mediante ecuaciones, en que la variable es el usuario. También se presenta por parámetro el usuario con peor comportamiento tal que permite verificar si se cumple o no con los requisitos de QoS.

Escenario dos con aumento de usuarios

En las figuras 5 y 6 con 10 y 23 usuarios respectivamente, se observa que el servicio de *video streaming* cumple con los parámetros de QoS para el algoritmo FDASH en cuanto a porcentaje de paquetes perdidos por usuario y en promedio para todos los usuarios, que es menor al 5% estimado en [21]. Analizando el escenario con 10 usuarios, el promedio del porcentaje de paquetes perdidos es del 0,39% que corresponde a 846,6 paquetes. Además, en la figura 5a, se observa que para el usuario 3 se presenta el máximo porcentaje promedio de paquetes perdidos que es del 0,43%, correspondiente a 919 paquetes, como se observa en la figura 5b. Los promedios móviles (figuras 5a y 5b), para el porcentaje y número de paquetes perdidos, están dados por las ecuaciones (4) y (5), respectivamente.

$$Y = -2 \times 10^{-6}x + 0,0039 \quad (4)$$

$$Y = -0,1818x + 847,5 \quad (5)$$

Analizando el escenario con 23 usuarios, el promedio del porcentaje de paquetes perdidos es del 1,47% que corresponde a 1415 paquetes. Además, en la figura 6a, se observa que para el usuario 19 se presenta el máximo porcentaje promedio de paquetes perdidos que es del 1,56%, correspondiente a 1470 paquetes, como se observa en la figura 6b. Los promedios móviles (figuras 6a y 6b), para el porcentaje y número de paquetes perdidos, están dados por las ecuaciones (6) y (7), respectivamente.

$$Y = 1 \times 10^{-5}x + 0,0146 \quad (6)$$

$$Y = -0,413x + 1409,7 \quad (7)$$

En el escenario con 10 usuarios, el retardo promedio es de 29,41 ms, su valor máximo es de 29,45 ms que se presenta sobre el usuario 1 (figura 7a), y el promedio móvil está dado por la ecuación (8). En el escenario con 23 usuarios, el retardo promedio es de 31,86 ms, su valor máximo es de 31,89 ms que se presenta sobre el usuario 1 (figura 7b), y el promedio móvil del retardo está dado por la ecuación (9). Estos valores son inferiores a los establecidos en [21].

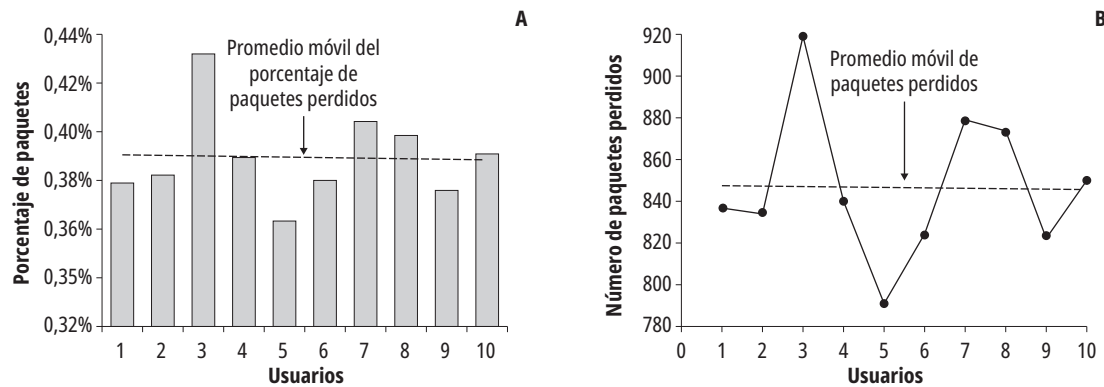


Figura 5. Paquetes perdidos y promedio móvil para 10 usuarios: a) Porcentaje de paquetes, b) Número de paquetes. Fuente: Elaboración propia.

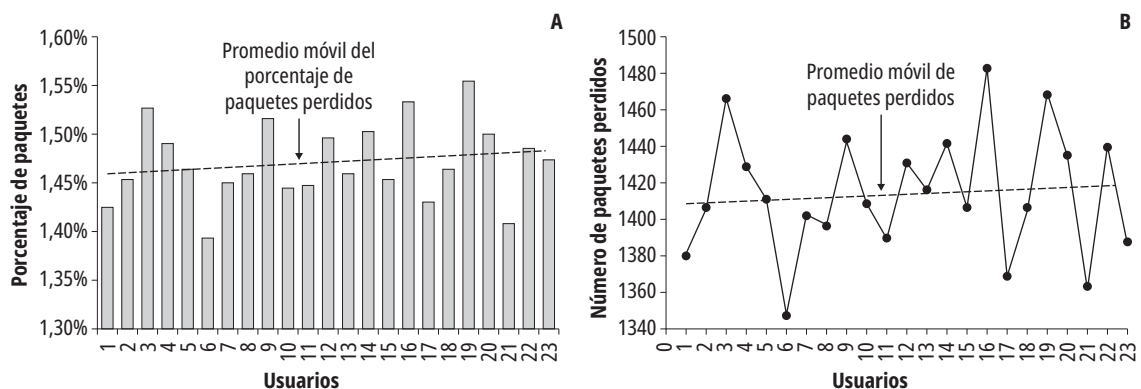


Figura 6. Paquetes perdidos y promedio móvil para 23 usuarios: a) Porcentaje de paquetes, b) Número de paquetes. Fuente: Elaboración propia.

$$Y = -0,0046x + 29,438 \quad (8)$$

$$Y = -0,0002x + 0,5177 \quad (10)$$

$$Y = -0,0007x + 31,857 \quad (9)$$

$$Y = 5 \times 10^{-5}x + 0,5613 \quad (11)$$

En el escenario con 10 usuarios, el valor promedio del *jitter* es de 0,51 ms, su valor máximo es de 0,52 ms que se presenta sobre el usuario 3 (figura 8a), y el promedio móvil del *jitter* está dado por la ecuación (10). En el escenario con 23 usuarios, el valor promedio del *jitter* es de 0,561 ms, su valor máximo es de 0,565 ms que se presenta sobre el usuario 23 (figura 8b), y el promedio móvil del *jitter* está dado por la ecuación (11). Estos valores están por debajo de los 50 ms máximos aceptados en [24].

De acuerdo con los parámetros de pérdida de paquetes, retardo y *jitter*, se observa que el algoritmo FDASH cumple con los parámetros de QoS en el escenario dos.

Escenario tres con *handover*

En este escenario, se observa el comportamiento de un usuario en movimiento que consume el servicio de *video streaming* usando el algoritmo FDASH. En la figura 9, se muestra el rendimiento, en que existe una disminución de la tasa de bit (*bit rate*) hasta los 0,4, 0,75, 0,4 y 0,3 Mbps en cada uno de los momentos que se da el *handover*, alrededor de los 80 s, 160 s, 420 s y 480 s, respectivamente. Se observa, además, que el rendimiento no logra estabilizarse, debido a que cada vez que el usuario experimenta el *handover* se produce una disminución del *bitrate*.

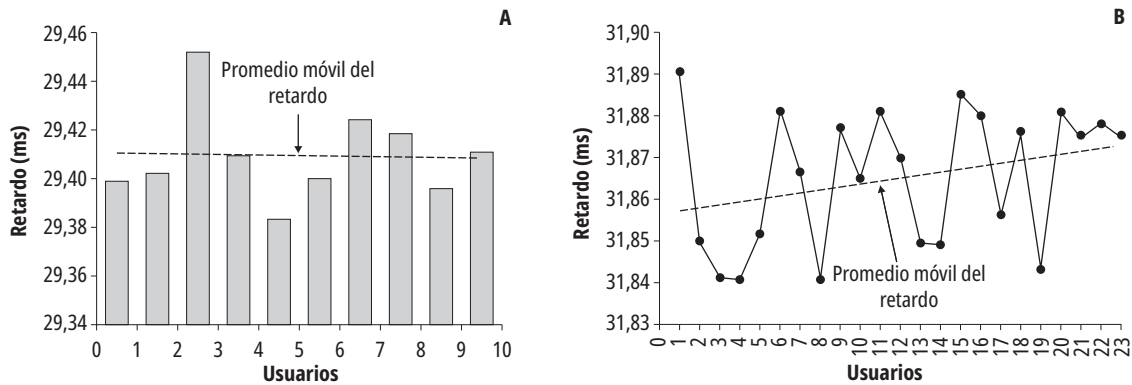


Figura 7. Retardo y promedio móvil del retardo: a) Para 10 usuarios, b) Para 23 usuarios. Fuente: Elaboración propia.

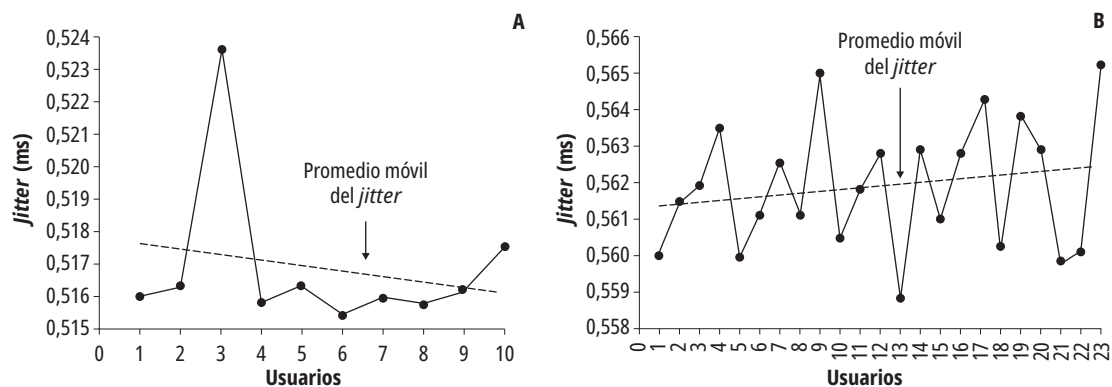


Figura 8. Jitter promedio y promedio móvil: a) Para 10 usuarios, b) Para 23 usuarios. Fuente: Elaboración propia.

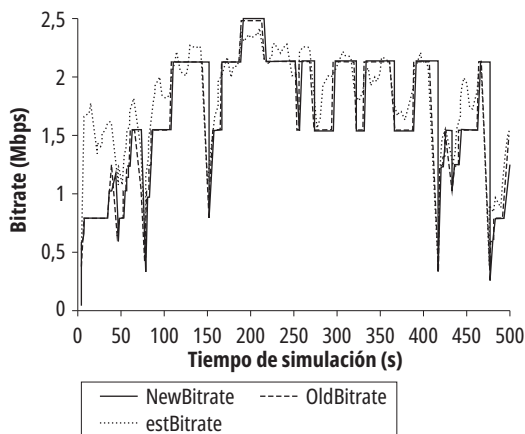


Figura 9. Rendimiento para un usuario en movimiento con algoritmo FDASH. Fuente: Elaboración propia.

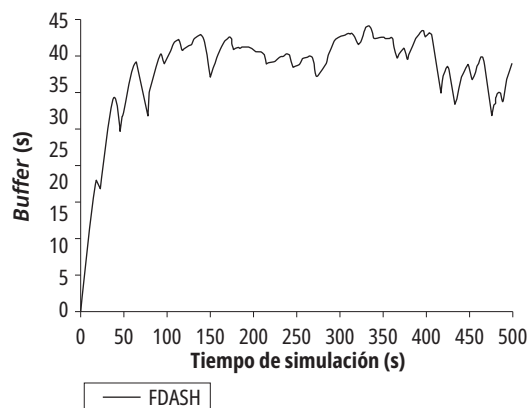


Figura 10. Comportamiento del *buffer*. Fuente: Elaboración propia.

Al comparar la figura 9 (con un usuario y con *handover*), con su equivalente la figura 3 (con un usuario y sin *handover*), se puede observar que debido a la característica del *handover* el rendimiento

no logra estabilizarse y su valor promedio es menor.

En la figura 10, se observa el comportamiento del *buffer* que es variable y alcanza un valor

máximo de 45 s, lo que es coherente con el comportamiento del rendimiento.

Para el escenario con un usuario en condiciones de *handover*, el valor medio del *jitter* es de 1,52 ms y del retardo es de 17,27 ms, valores que son calculados mediante las ecuaciones (1) y (2).

Escenarios con 10 y 23 usuarios y *handover*

En el escenario con 10 usuarios y *handover*, el retardo promedio es de 21,57 ms, su valor máximo es de 23 ms que se presenta sobre el usuario 6 (figura 11a), y el promedio móvil del retardo está dado por la ecuación (12). En el escenario con 23 usuarios y *handover*, el retardo promedio es de 22,99 ms, su valor máximo es de 30 ms que se presenta sobre el usuario 3 (figura 11b). El promedio móvil del

retardo está dado por la ecuación (13). Estos valores son inferiores a los establecidos en [21].

Para el escenario con 10 usuarios y *handover*, el promedio del porcentaje de paquetes perdidos es del 0,76% que corresponde a 651,3 paquetes. Además, en la figura 12a, se observa que el usuario 9 presenta el máximo porcentaje promedio de paquetes perdidos que es del 1%, valor que es inferior al porcentaje estimado del 5% [21] y corresponde a 868 paquetes, como se observa en la figura 12b. Los promedios móviles (figuras 13a y 13b), para el porcentaje y número de paquetes perdidos, están dados por las ecuaciones (14) y (15), respectivamente.

$$Y = -0,0341x + 21,387 \quad (12)$$

$$Y = -0,0553x + 22,328 \quad (13)$$

$$Y = 7 \times 10^{-5}x + 0,0072 \quad (14)$$

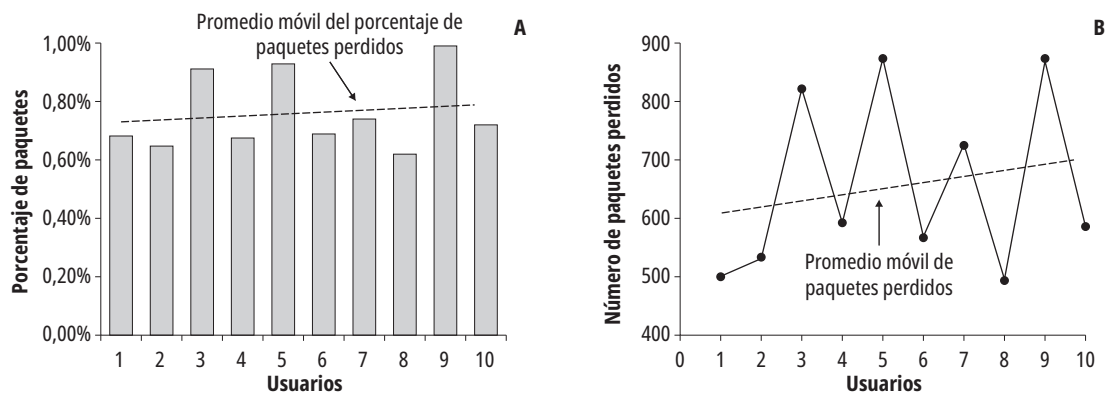


Figura 11. Retardo promedio y promedio móvil con *handover*: a) Para 10 usuarios, b) Para 23 usuarios. Fuente: Elaboración propia.

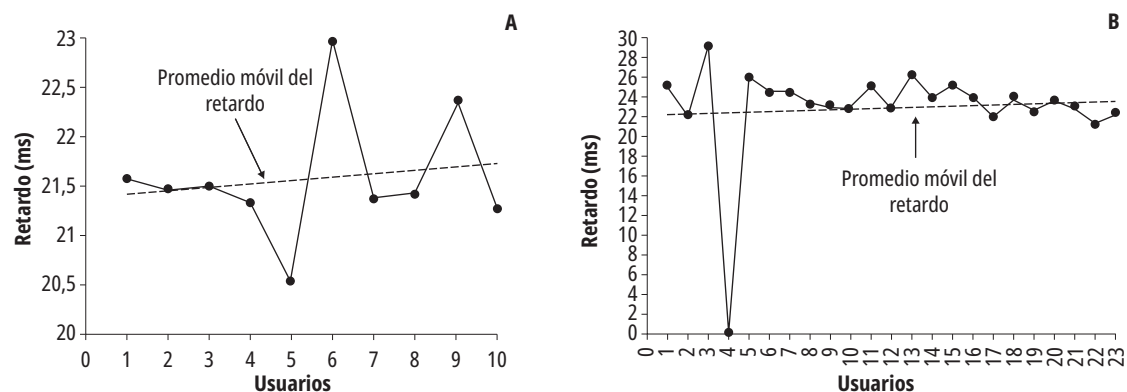


Figura 12. Porcentaje de paquetes perdidos y promedio móvil con *handover*: a) Para 10 usuarios, b) Para 23 usuarios. Fuente: Elaboración propia.

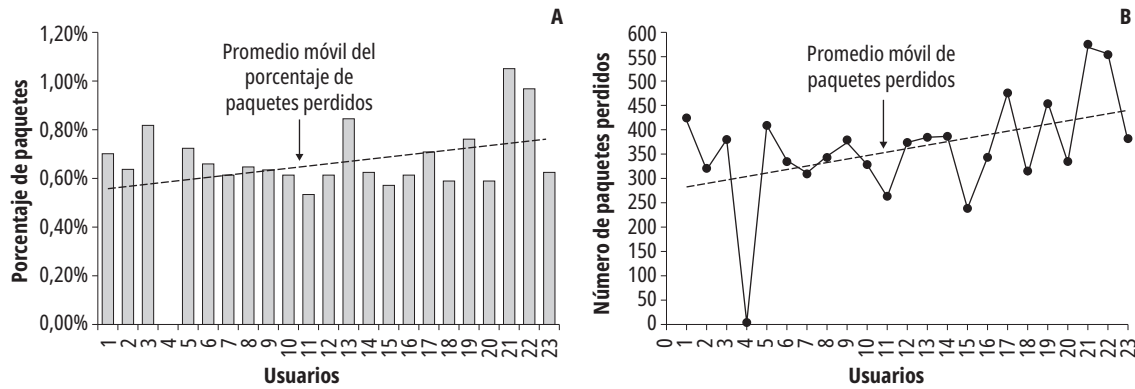


Figura 13. Paquetes perdidos y promedio móvil para 23 usuarios con *handover*: a) Porcentaje, b) Número de paquetes. Fuente: Elaboración propia.

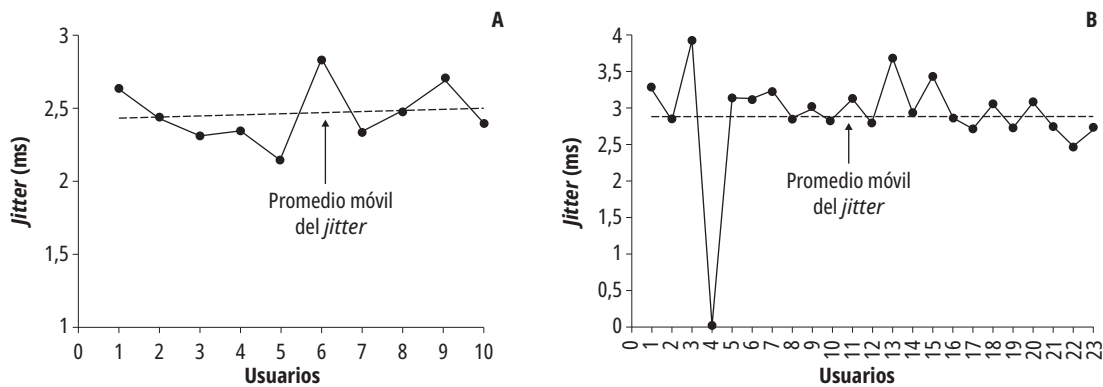


Figura 14. Jitter promedio y promedio móvil con *handover*: a) Para 10 usuarios, b) Para 23 usuarios. Fuente: Elaboración propia.

$$Y = -9,8727x + 597 \quad (15)$$

Para el escenario con 23 usuarios y *handover*, el promedio del porcentaje de paquetes perdidos es del 66% que corresponde a 356,6 paquetes. Además, en la figura 13a, se observa que el usuario 21 presenta el máximo porcentaje promedio de paquetes perdidos que es del 1,05%, valor que es inferior al porcentaje estimado del 5% [21] y corresponde a 572 paquetes, como se observa en la figura 13b. Los promedios móviles (figuras 14a y 14b), para el porcentaje y número de paquetes perdidos, están dados por las ecuaciones (16) y (17), respectivamente.

$$Y = 9 \times 10^{-5}x + 0,0055 \quad (16)$$

$$Y = 7,1789x + 270,46 \quad (17)$$

En cuanto al *jitter* en el escenario con 10 usuarios y *handover* (figura 14a), su valor promedio es de 2,45 ms, su valor máximo es de 2,8 ms que se presenta sobre el usuario 6, y el promedio móvil del *jitter* está dado por la ecuación (18). En el escenario con 23 usuarios y *handover* (figura 14b), su valor promedio es de 2,89 ms, su valor máximo es de 3,9 ms que se presenta sobre el usuario 3 y el promedio móvil del *jitter* está dado por la ecuación (19). Estos valores están por debajo de los 50 ms máximos aceptados en [24].

$$Y = 0,0079x + 2,4135 \quad (18)$$

$$Y = 0,0006x + 2,89186 \quad (19)$$

De acuerdo con los resultados obtenidos para este escenario, se observa que los parámetros de QoS, como el retardo y el *jitter*, se ven afectados

por el *handover*. Sin embargo, sus valores máximos no superan los establecidos en [21] y [24]. Mientras que el porcentaje de paquetes perdidos no se ve afectado respecto de los otros escenarios por el *handover*, este resultado muestra cómo el algoritmo FDASH ante condiciones desfavorables escoge disminuir la tasa de bit, como se observa en la figura 9, esto es, conmutará entre diferentes calidades del video.

Conclusiones

El tamaño del *buffer* en el uso del algoritmo FDASH es una característica relevante, de modo que es necesario que los dispositivos móviles UE tengan una buena capacidad de memoria del orden de las decenas de segundo. Con el avance de la tecnología en cuanto a los teléfonos móviles inteligentes, esta característica está garantizada.

Se observa que el tamaño del *buffer* es mayor en el escenario con un usuario y sin *handover* respecto de su equivalente un usuario y con *handover*. Este comportamiento, aunque pudiera parecer contradictorio, lo que significa es que en un caso ideal en que en una red solo exista un usuario donde no tenga que cambiar de celdas o portadoras (sin *handover*) la red estará en capacidad de ofrecer un mayor rendimiento (*throughput*), lo que exige mayores capacidades del *buffer*. Así, al existir un mayor rendimiento o una mayor tasa de bits en el canal radio, asimismo existe una mayor probabilidad de que varíe el retardo de los flujos entre el transmisor y el receptor, esto es, el valor del *jitter* aumentará. Esta es la razón por la que en el escenario básico presenta el mayor de los valores del *jitter* que alcanza un 0,94 ms, lejos de los 50 ms máximos recomendados por Cisco.

Al analizar el *buffer* en el uso del algoritmo, FDASH oscila junto con el rendimiento (*throughput*), por lo que se ve un efecto rebote, en que se aumenta la calidad y el *buffer*, asimismo si disminuye el *buffer* también lo hace la calidad de forma constante.

Así, de acuerdo con los resultados, se concluye que el algoritmo FDASH al iniciar la transferencia de datos asigna a todos los usuarios una misma tasa de bits en un tiempo corto, con la pretensión

de subir la tasa de bits; sin embargo, esto es regulado según la cantidad de usuarios. En todo caso, el algoritmo FDASH no desborda los parámetros de QoS en cuanto a pérdida de paquetes, retardo y *jitter*, en ninguno de los escenarios analizados. Por tanto, es un algoritmo totalmente viable para soportar la tecnología de *video streaming* sobre redes LTE, bajo escenarios como los descritos en este artículo.

Como trabajo futuro, se propone la comparación del algoritmo FDASH versus el algoritmo AAASH (por sus siglas en inglés) para redes LTE.

Agradecimientos

A la Universidad del Quindío por su financiación al proyecto 839.

Referencias

- [1] Cisco. (febr. 2019). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022* [En línea]. Disponible en: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf>
- [2] C. Arzelier. (en. 2017). *The Mobile Broadband Standard, 3GPP a Global Initiative, Release 12. 3GPP. 3GPP Multimedia Broadcast and Multicast Service (MBMS)* [En línea]. Disponible en: http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/ (enero de 2017).
- [3] T. Stockhammer, "Dynamic adaptive streaming over HTTP --: standards and design principles", en *MMSys '11 Proceedings of the second annual ACM conference on Multimedia systems*, Nueva York, 2011, pp. 133-144. doi>10.1145/1943552.1943572
- [4] T. Xu y L. Ma, "Predictive prefetching for MPEG DASH over LTE networks", en *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3432-3436, 2015. DOI: 10.1109/ICIP.2015.7351441
- [5] O. B. Maia, H. C. Yehia y L. de Errico, "A concise review of the quality of experience assessment for video streaming", *Computer Communications*, vol. 57, pp. 1-12, febr., 2015. <https://doi.org/10.1016/j.comcom.2014.11.005>
- [6] D. J. Vergados, A. Michalas, A. Sgora, D. D. Vergados y P. Chatzimisios, "FDASH: A fuzzy-based MPEG/DASH adaptation algorithm", *IEEE Systems Journal*, vol. 10, no. 2, pp. 859-868, jun., 2016. DOI: 10.1109/JSYST.2015.2478879

- [7] D. J. Vergados, A. Michalas, A. Sgora y D. D. Vergados, "A fuzzy controller for rate adaptation in MPEG-DASH clients", en *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, 2014, pp. 2008-2012. DOI: 10.1109/PIMRC.2014.7136501
- [8] W. Y. Campo-Muñoz, G. E. Chanchí-Golondrino y M. C. Camacho-Ojeda, "Uso de técnicas de emulación en la construcción de un modelo de tráfico para un servicio multimedia", *Ingeniería, Investigación y Tecnología*, vol. 18, no. 2, pp. 209-221, abr.-jun. 2017.
- [9] M. Morales Rodríguez, M.A. Calle Pérez, J. D. Tovar Vanegas y J. C. Cuéllar Quiñonez, *Simulando con OM-NET: selección de la herramienta y su utilización*. Cali: Universidad Icesi, 2013.
- [10] G. Piro, N. Baldo y M. Miozzo, "An LTE module for the ns-3 network simulator", en *SIMUTools '11 Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, Barcelona, España, 2011.
- [11] B. Herman, D. Petrov, J. Puttonen y J. Kurjenniemi, "A3-Based Measurements and Handover Model for ns-3 LTE", en *MOBILITY: The Third International Conference on Mobile Services, Resources, and Users*, pp. 20-23, 2013.
- [12] J. G. Torres y R. Bustamante, "Analysis of the effects of CQI feedback for LTE networks on ns-3", *IEEE Latin America Transactions*, vol. 13, no. 11, pp. 3538-3543, nov., 2015. DOI: 10.1109/TLA.2015.7387927
- [13] T. Molloy, Z. Yuan y G. M. Muntean, "Real time emulation of an LTE network using ns-3", en *25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*, pp. 251-257, 2014. DOI: 10.1049/cp.2014.0694
- [14] F. Guidolin, L. Badia y M. Zorzi, "Implementation of 2 x 2 MIMO in an LTE module for the ns3 simulator", en *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2012 IEEE 17th International Workshop on, Barcelona, España, pp. 281-285, 2012. DOI: 10.1109/CAMAD.2012.6335351
- [15] W. Y. Campo-Muñoz, E. Astaiza-Hoyos y L. F. Muñoz-Sanabria, "Traffic modelling of the video-on-demand service through ns-3", *Dyna*, vol. 84, no. 202, pp. 55-64, jul.-sept., 2017. <http://dx.doi.org/10.15446/dyna.v84n202.61650>
- [16] J. M. Hernando Rábanos, L. Mendo Tomás y J. M. Riera Salís, *Comunicaciones móviles*. Madrid: Universitaria Ramón Areces, 2015.
- [17] R. Agusti, F. Bernardo, F. Casadevall, R. Ferrús, J. Pérez-Romero y O. Sallent, *LTE nuevas tendencias en comunicaciones móviles*. Madrid: Fundación Vodafone España, 2010.
- [18] E. Dahlman, S. Parkvall y J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*, 2.^a ed. Ámsterdam: Elsevier, 2013.
- [19] D. Ortiz Trujillo, G. E. Chanchí Golondrino, D. F. Durán Dorado, W. Y. Campo Muñoz y J. L. Arciniegas Herrera, "Coding multimedia content using DASH standard", en *2016 IEEE 11th Colombian Computing Conference (CCC)*, Popayán, Colombia, pp. 1-7, 2016. DOI: 10.1109/ColumbianCC.2016.7750782
- [20] ITEC. Dynamic Adaptive Streaming over HTTP [En línea]. Disponible en: http://www-itec.uni-klu.ac.at/dash/?page_id=746
- [21] Autoridad Reguladora de los Servicios Públicos, *Reglamento de prestación y calidad de los servicios*. San José de Costa Rica: Autoridad Reguladora de los Servicios Públicos, 2009.
- [22] Cisco. (15 febr. 2008). *Implementación de políticas de calidad del servicio (QoS) con DSCP* [En línea]. Disponible en: https://www.cisco.com/c/es_mx/support/docs/quality-of-service-qos/qos-packet-marking/10103-dscpvalues.pdf
- [23] ns-3. (2017). *lena-simple-epc.cc* [El línea]. Disponible en: https://www.nsnam.org/doxygen/lena-simple-epc_8cc_source.html
- [24] T. Szigeti, K. McMenamy, R. Saville y A. Glowacki, *Cisco TelePresence Fundamentals*. Indianápolis: Cisco Press, 2009.