

Propuesta de Investigación Doctoral

Análisis del impacto de los tamaños de segmento en el desempeño
de streaming de vídeo en Internet

Presenta:

M. en C. Erik Miguel Díaz Salazar

Responsable de investigación doctoral I, II y III:

Dr. Víctor Manuel Ramos Ramos
Departamento de Ingeniería Eléctrica, UAM - Iztapalapa

Jurado examinador:

Dra. Graciela Roman Alonso
Departamento de Ingeniería Eléctrica, UAM-Iztapalapa

Dr. Enrique Stevens Navarro
Departamento de Electrónica, Universidad Autónoma de San Luis Potosí

Dr. Francisco de Asís López Fuentes
Departamento de Tecnologías de la Información, UAM - Cuajimalpa

Noviembre de 2020
(Corregido, Enero 2021)

Propuesta de Investigación Doctoral

**Análisis del impacto de los tamaños de segmento en el desempeño
de streaming de vídeo en Internet**

Presenta:

M. en C. Erik Miguel Díaz Salazar

Responsable de investigación doctoral I, II y III:

Dr. Víctor Manuel Ramos Ramos
Departamento de Ingeniería Eléctrica, UAM - Iztapalapa

Jurado examinador:

Dra. Graciela Roman Alonso
Departamento de Ingeniería Eléctrica, UAM-Iztapalapa

Dr. Enrique Stevens Navarro
Departamento de Electrónica, Universidad Autónoma de San Luis Potosí

Dr. Francisco de Asís López Fuentes
Departamento de Tecnologías de la Información, UAM - Cuajimalpa

Noviembre de 2020
(Corregido, Enero 2021)

Contenido

1	Introducción	1
2	Antecedentes	3
2.1	Motivación	4
3	Estado del Arte	6
3.1	Algoritmos de Adaptación de Tasa de Bits (ABR)	6
3.1.1	Adaptación basada en el servidor	7
3.1.2	Adaptación basada en redes asistidas	7
3.1.3	Adaptación híbrida	7
3.1.4	Adaptación basada en el cliente	7
3.2	Segmentación en DASH	9
3.3	Compresión de Vídeo	12
3.3.1	H.265/HEVC	12
3.3.2	VP9	14
3.3.3	AV1	15
4	Objetivos	16
4.1	Objetivo General	16
4.2	Objetivos Particulares	16
5	Metodología de Investigación	17
6	Avances	18
7	Calendario	19
	Referencias	20

1 Introducción

El *streaming* de vídeo es una de las tecnologías más populares hoy en día, debido a ello, los proveedores de servicios se esmeran por transmitir de forma continua el contenido multimedia del servidor al cliente. Sin embargo, es necesario que constantemente se mejoren los métodos de *streaming*, debido a la capacidad de la red y a la heterogeneidad de los usuarios. Las tecnologías para la distribución de vídeo para ofrecer una mejor calidad a los usuarios de forma automática se han convertido en un reto muy importante para la industria. Mediante el uso de HTTP, que es el protocolo de nivel de aplicación por defecto para el servicio de Internet, se ha llevado a cabo una nueva investigación y desarrollo del método de *streaming*.

El *Streaming* Adaptativo sobre HTTP (HAS), a través de su estándar MPEG-DASH (Dynamic Adaptive *Streaming* Over HTTP) [1] se ha vuelto muy popular para la distribución de contenidos multimedia a través de Internet. Este modelo tiene varias ventajas, en primer lugar, la infraestructura de Internet ha evolucionado de tal manera que puede soportar eficientemente a HTTP. Por ejemplo, las Redes de Distribución de Contenidos (CDN, Content Delivery Network) proporcionan copias locales del contenido multimedia principalmente y que están almacenados en otros servidores alejados geográficamente. De esta forma, es posible acceder a dichos contenidos de una manera mucho más eficiente, logrando un mejor balance de la carga a la que están sometidos los servidores que alojan el contenido como los enlaces que interconectan las distintas secciones de la red, eliminando posibles cuellos de botella y sirviendo los datos en función de la cercanía geográfica del usuario final, minimizando los retardos. Además, casi todos los firewalls pueden abrir puertos que soportan conexiones HTTP. En segundo lugar, el cliente puede realizar la sesión de *streaming* sin tener que mantener una sesión en el servidor.

El contenido se encuentra dividido en segmentos de entre 2 a 10 segundos de duración. Las propiedades de estos y su localización en el servidor se encuentran descritos en un archivo XML denominado Media Presentation Description (MPD). Los segmentos de corta duración pueden aprovechar las fluctuaciones en el ancho de banda, evitando la inanición en el buffer del reproductor a cambio de realizar un mayor número de peticiones GET y una posible sobrecarga en la codificación del material de vídeo. En cambio, los segmentos de mayor duración buscan tener una mayor eficiencia en la codificación al contar con un mayor número de Group of Pictures (GoP).

Por otro lado, uno de los elementos clave en el *streaming* de vídeo son los algoritmos de tasa adaptativa de bits (Adaptative Bitrate, ABR), los cuales se ejecutan principalmente en cada cliente. Su objetivo, es garantizar la calidad de la experiencia (QoE, Quality of Experience) de los usuarios bajo la presencia de fluctuaciones del ancho de banda. Dichas fluctuaciones se deben a factores como la intensidad de la señal en conexiones inalámbricas, congestión de la red, eventos de convergencia en la red o de servicios digitales, entre otros. Los algoritmos ABR pueden tener en cuenta factores como las estimaciones del ancho de banda, la ocupación del buffer de reproducción, características del dispositivo, preferencias del espectador o características del contenido, aunque con diferentes pesos o calificaciones de calidad. La Figura 1, muestra el proceso realizado en DASH para reproducir el contenido según van cambiando las condiciones en la red.

Además de los métodos de streaming de vídeo y sus algoritmos ABR, se encuentran también las técnicas de compresión de vídeo, que de igual forma, están experimentando cierto crecimiento. Esto se debe a una estandarización de la compresión, la cual es necesaria para facilitar el intercambio de datos a nivel mundial a través de aplicaciones de *streaming* de vídeo. Una codificación

estándar es eficiente si en el diseño de los algoritmos de compresión se implementan un codificador y decodificador eficaces. Actualmente, la mayoría del contenido de vídeo está codificado mediante el estándar H.264/AVC (Advanced Video Coding) [2]. H.264 se desempeña muy bien en HD (720p) y Full HD (1080p). Sin embargo, en escenarios utilizando UHD (Ultra High Definition) se requiere una mayor eficiencia.

Dado que la QoE del espectador debe determinarse en tiempo real durante la reproducción, generalmente se utilizan métricas objetivas que incluyen el número de paradas de buffer, la duración del retardo de inicio, la frecuencia y la cantidad de oscilaciones, y la inestabilidad del contenido multimedia. Debido a su diseño, los estándares como DASH no exigen ningún algoritmo ABR en particular, dejando a los desarrolladores la tarea de innovar e implementar su propio método.

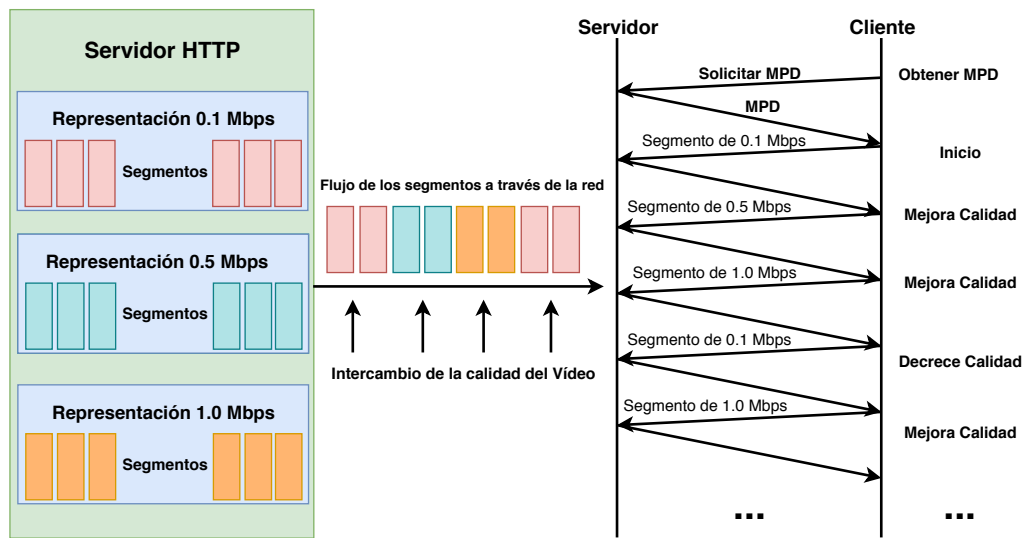


Figura 1: Cronograma del comportamiento adaptativo de DASH.

2 Antecedentes

La idea principal de DASH es distribuir el contenido multimedia en segmentos con duración de unos cuantos segundos, los cuales son codificados en diferentes tasas de bits y resoluciones llamadas *representaciones*. Estos segmentos son suministrados por un servidor web y descargados por peticiones HTTP-GET como cualquier otro elemento en un sitio web. En primer lugar, se obtiene el archivo manifiesto o MPD mediante HTTP, correo electrónico u otro medio de comunicación. Posteriormente, el MPD es analizado por el parser en el reproductor, con el que obtendrá información sobre la organización y distribución del contenido multimedia en el servidor, necesaria para la reproducción. Una vez obtenida la información requerida, el algoritmo ABR inicia las primeras estimaciones y el cliente comienza a realizar las peticiones de los segmentos por medio de HTTP mientras la reproducción esté en marcha. Este procedimiento puede verse en la Figura 2, la cual ilustra el proceso realizado en DASH para reproducir el contenido según van cambiando las condiciones en la red.

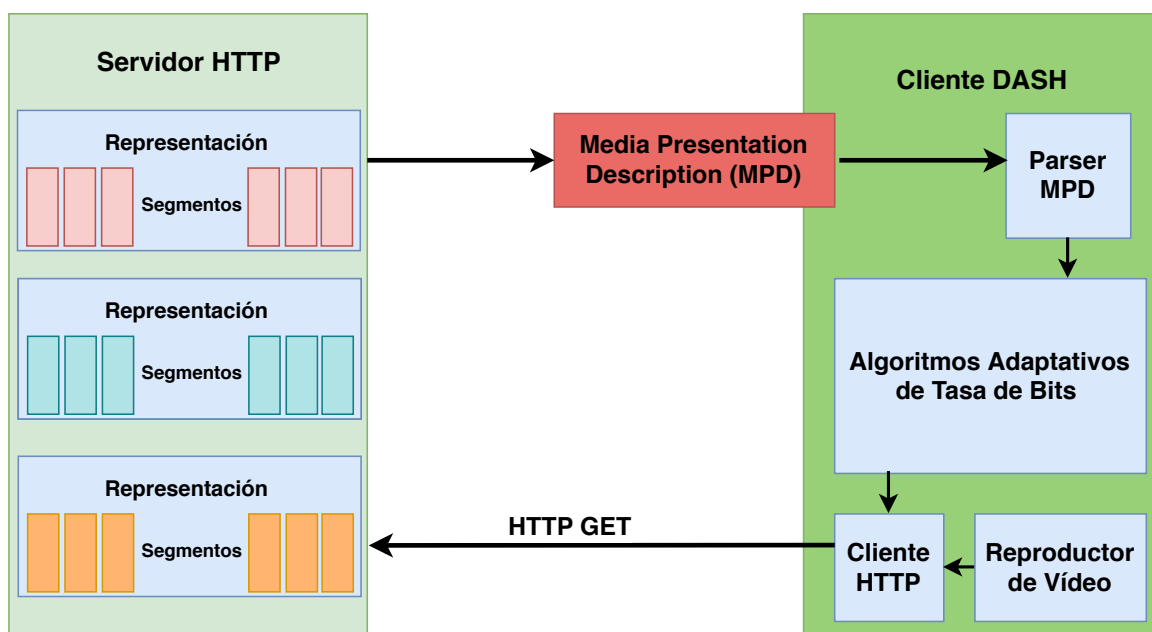


Figura 2: Escenario de un cliente DASH y un servidor HTTP.

Un archivo de vídeo es una imagen en movimiento creada como una secuencia de imágenes fijas o *frames*. De esta forma, al ser almacenados y luego reproducidos se crea una ilusión de movimiento. A partir de esto es necesario tener en consideración sus principales características, las cuales definen su formato y codificación: Tamaño del *Frame Size*, Relación de Aspecto, *Framerate*, y Tasa de Bits o *Bitrate* [3]. Por otro lado, es importante considerar varios puntos fundamentales en el *streaming* de contenido multimedia, los cuales pueden agruparse de la siguiente manera:

1. Protocolos de *Streaming*. Algunos sistemas de *streaming* requieren del uso de protocolos especializados, tal es el caso del Protocolo de Transporte en Tiempo Real o RTP (Real-time Transport Protocol) que trabaja en conjunto con el Protocolo de Control de RTP (RTCP)

y utilizan mecanismos que permiten dividir los flujos de los datos como el audio y vídeo en paquetes independientes para ser enviados y posteriormente sincronizados en su destino. Sin embargo, presentan varios inconvenientes como la incompatibilidad con Redes de Distribución de Contenidos o CDN (Content Distribution Network) o el caso de firewalls que no permiten el paso de paquetes que generan [4].

2. *Live streaming*. Como su nombre lo indica, es una transmisión de contenido que se está generando en tiempo real desde una fuente como una cámara o interfaz de audio, un codificador que digitalice el contenido, editor de medios y una CDN para su distribución y envío. Además, es necesario el uso de servidores de *streaming* para el control de transferencia de la información, asimismo utilizar protocolos de entrega como el Real Time Media Protocol o RTMP [5, 6].
3. *HTTP streaming*. Al establecerse una conexión TCP y solicitar el contenido multimedia por HTTP-Request, en el lado del servidor se responde a través de HTTP-Response, enviando el contenido como cualquier objeto de un sitio web. En el lado del cliente, una vez almacenada suficiente información mediante un búfer, la reproducción da inicio y de manera simultánea la descarga del resto del contenido prosigue. Sin embargo, este método no puede aprovechar características o parámetros de la red en la que está transmitiendo [7].
4. *Streaming Adaptativo sobre HTTP (HAS)*. Esta capacidad de adaptación permite variar la calidad del contenido multimedia según las condiciones de la red. La adaptación es posible cambiando de estrategia, de tal forma que haya una toma de decisiones en la elección de una mejor opción entre las tasas de bits disponibles en el lado del cliente. Por otro lado, en el servidor se encuentra el contenido segmentado, mencionado anteriormente. Según las condiciones de la red, en un tiempo determinado, un algoritmo de Adaptación de Tasa de Bits o ABR (Adaptative Bitrate) ejecutado en el cliente, realiza estimaciones utilizando parámetros de forma individual o en conjunto para determinar qué representación será la siguiente en ser descargada [8].

2.1 Motivación

El *streaming* en la modalidad de Video on Demand (VoD) se presta para ser el futuro en la distribución de contenido en línea. Hoy en día los usuarios prefieren el vídeo sobre otras aplicaciones de contenido, incluyendo correos electrónicos, redes sociales, entre otros. Debido a que el tiempo de visualización anual en las plataformas VoD va en aumento año con año como lo reporta Cisco Systems en su informe anual, donde predicen que el número de dispositivos conectados a Internet triplicará con creces la población mundial para 2023. Por otro lado, también se pronostica que el 70% de los usuarios a nivel mundial poseerá dispositivos móviles para entonces, además de que la demanda de contenido en 4K/UHD se verá duplicada [9].

Actualmente el *streaming* se está viendo involucrado en muchas de las actividades de la vida cotidiana, principalmente en el entretenimiento y el sector educativo. Hasta la fecha, existe una gran diversidad de propuestas por parte de la comunidad investigadora que busca mejorar el desempeño ante ambientes donde involucra principalmente problemas de congestión en la red ocasionando problemas como el retardo inicial en la reproducción del contenido, constantes interrupciones por la inanición del búfer del reproductor o una inadecuada adaptación del contenido, ocasionando cambios abruptos entre representaciones.

Con el fin de mejorar el desempeño en los sistemas de *streaming* VoD, esta propuesta de investigación doctoral se centra en evaluar el desempeño que podría tener una plataforma de *streaming* desarrollando estrategias para DASH considerando varias líneas de investigación, que son importantes de atender:

- ¿Que beneficios traería el almacenar o generar segmentos con distinta duración? Live *streaming* [10] a diferencia de VoD, permite la generación dinámica de segmentos con diferentes duraciones, los cuales están referenciados en diferentes archivos MPD, mientras que en VoD la duración es fija y sólo permite el uso de un MPD.
- ¿Qué condiciones son necesarias para diseñar mecanismos en el *streaming* que consideren la adaptabilidad de las representaciones de vídeo basándose en la QoE? Gran parte del trabajo existente en este contexto considera la inclusión de varios parámetros objetivos, e.g., el ancho de banda, el tamaño del búfer, la variación del retardo, entre otros, para decidir la calidad del segmento de vídeo descargado. En contraste, una atención menor se ha dedicado a observar la calidad de la experiencia (QoE) percibida por el/los usuario(s). Este es un eje de investigación primordial, ya que la mayoría de los servicios comerciales se distribuyen a cantidades masivas de usuarios [11, 12, 13].
- ¿Cómo afectaría el rendimiento de una plataforma de *streaming* codificando las representaciones mediante H.265 u otra codificación de alta eficiencia [14]? La mayoría de las propuestas de la literatura utilizan conjuntos grandes de representaciones, las cuales están codificadas con H.264, con el propósito de suavizar el intercambio en el proceso de adaptabilidad. Sin embargo, no consideran los costos que esto representaría en almacenar todo este contenido en servidores. Por lo anterior, H.265 vendría a solucionar este problema. High Efficiency Video Coding (HEVC) o H.265 es un estándar que duplica la compresión manteniendo una calidad idéntica o similar comparada con H.264. Esta línea de investigación podría ser relevante.

3 Estado del Arte

3.1 Algoritmos de Adaptación de Tasa de Bits (ABR)

En DASH, los algoritmos de adaptación de tasa de bits o ABR (Adaptive BitRate) se ejecutan principalmente en el lado cliente de forma distribuida. El objetivo de estos algoritmos es garantizar la calidad de la experiencia (QoE) para los usuarios bajo la presencia de fluctuaciones del ancho de banda debido a factores como la intensidad de la señal (en conexiones inalámbricas), congestión de la red, eventos de convergencia en la red, entre otros. Estas fluctuaciones suelen ser comunes en Internet, también es posible que ocurran en redes domésticas o incluso en redes gestionadas donde a menudo se implementan controles de acceso y herramientas de calidad de servicio (QoS).

Los algoritmos ABR pueden tener en cuenta factores como las estimaciones del ancho de banda, un mapeado del buffer de reproducción, características del dispositivo, preferencias del espectador o características del contenido, determinadas por pesos o valoraciones en cuestiones de calidad. La QoE del espectador se determina en tiempo real durante la reproducción, generalmente se utilizan métricas objetivas que incluyen el número de paradas de buffer, la duración del retardo de inicio, la frecuencia y la cantidad de oscilaciones, y la inestabilidad del contenido multimedia. Debido a su diseño, estándares como DASH no exigen ningún algoritmo ABR en particular, dejando a los desarrolladores de las diferentes plataformas existentes la tarea de innovar e implementar su propio método. A través de la Figura 3, se muestra una clasificación de los diferentes esquemas de algoritmos ABR, según la entidad del sistema donde es implementada la lógica, los cuales son descritos a continuación [15].

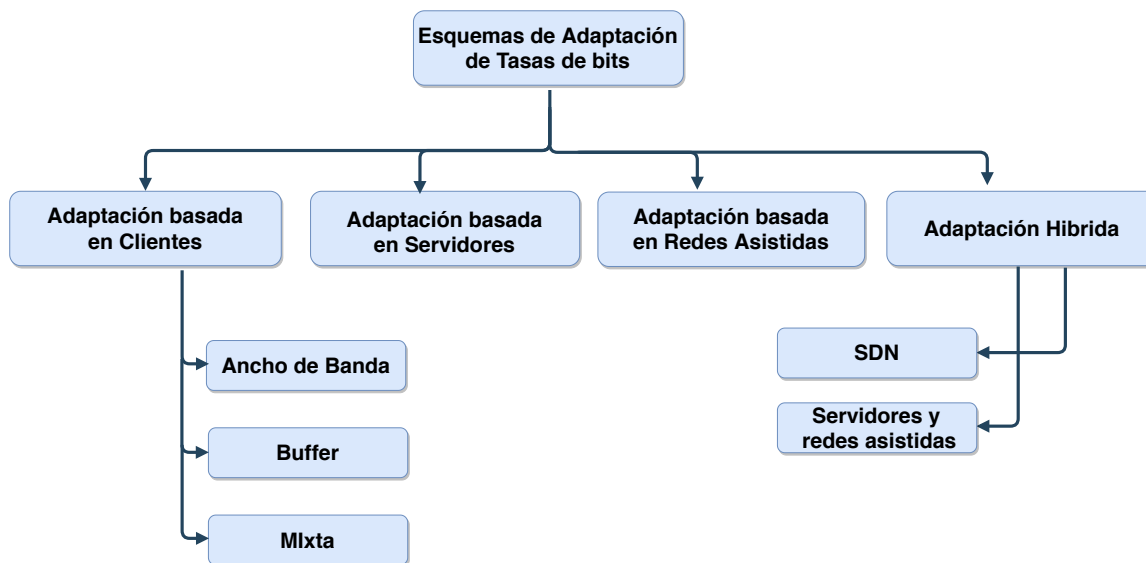


Figura 3: Clasificación de los esquemas de adaptación.

3.1.1 Adaptación basada en el servidor

Este tipo de esquemas utiliza un método de modelado de tasa de bits en el lado del servidor y no requiere ninguna cooperación por parte del cliente. El cliente aún toma algunas decisiones, pero es el servidor quien toma la decisión final. Sin embargo, dichos requieren de una gran complejidad, especialmente cuando el número de clientes aumenta. Estos esquemas también necesitan modificaciones en el archivo MPD o en un software personalizado para implementar la lógica de adaptación de la tasa de bits. Esto puede ser percibido como una violación de los principios de diseño estándar de DASH, debido a que el servidor debe ser un servidor HTTP estándar y los algoritmos de adaptación deben ejecutarse en el lado del cliente.

3.1.2 Adaptación basada en redes asistidas

Las redes asistidas permiten a clientes DASH tomar decisiones en la red durante el proceso de adaptación de la tasa de bits. Esto ocurre cuando se recopila información sobre las condiciones de la red y se informa a los clientes sobre las tasas de bits adecuadas que pueden seleccionarse. El proceso dentro de la red necesita componentes especiales (agente/proxy) desplegados en la red para monitorear el estado y las condiciones. También, ofrecen información a nivel de red que permite a los clientes DASH utilizar eficientemente los recursos.

3.1.3 Adaptación híbrida

Este tipo de adaptación, se compone de varias entidades de la red que colaboran entre sí y recopilan información útil sobre las condiciones de la red permitiendo a los clientes DASH seleccionar la mejor tasa de bits. Este esquema consiste en Redes Definidas por Software o SDN (Software Defined Networking), que permiten controlar los recursos de la red y monitoreo, simplificando programación y despliegue de recursos físicos. Por otro lado, se propone una arquitectura asistida por servidor y red para DASH o SAND (Server And Network Assisted DASH), la cual permite un intercambio bidireccional de mensajes entre clientes y elementos de la red que activa un mecanismo de control para priorizar flujos, reservas en el ancho de banda y la adaptación de tasas de bits en función de las condiciones de la red y de los clientes [16].

3.1.4 Adaptación basada en el cliente

La gran mayoría de los esquemas de adaptación residen en el lado del cliente [17], e intentan adaptarse a variaciones del ancho de banda cambiando a una tasa de bits en función de varios parámetros, como el ancho de banda o el *throughput* disponible, el tamaño del buffer de reproducción, entre otros. El cliente utiliza uno o más parámetros de entrada para su algoritmo ABR con el fin de elegir la tasa de tasa de bits adecuada para el siguiente segmento en ser descargado. Estos algoritmos tratan de evitar problemas comunes en el *streaming* como retardos al inicio de la reproducción, inestabilidad del vídeo, oscilaciones bruscas en la tasa de bits, inanición en el buffer y por otro lado, buscan mantener una QoE aceptable para el espectador. Este esquema se subdivide en tres subclases de adaptación: ancho de banda disponible, buffer de reproducción y adaptaciones mixtas, las cuales son descritas a continuación.

- Adaptación basada en el ancho de banda disponible, el cliente basa sus estimaciones en el ancho de banda de red, que normalmente calcula como el tamaño del segmento dividido por el tiempo de descarga. En [18], se propone un algoritmo de adaptación que intenta detectar fluctuaciones del ancho de banda y la congestión utilizando un flujo de red suavizado basado

en un tiempo de obtención de segmento o SFT (Segment Fetch Time) que mide el tiempo que inicia con el envío de un mensaje HTTP GET hasta recibir el último byte del segmento. Probe AND Adapt (PANDA) [19] estima con precisión el ancho de banda disponible y trata de eliminar el problema del estado estacionario ON-OFF, así como de reducir las oscilaciones de la tasa de bits cuando varios clientes están en un cuello de botella en el mismo enlace que comparten.

- Adaptación basada en el buffer de reproducción, el cliente se basa en los niveles de ocupación del buffer como criterio para seleccionar la tasa de bits correspondiente a la siguiente representación en ser descargada. Se determinan varias regiones de ocupación, y una curva $f(B)$ dentro de la región viable puede generar un rango de tasas de bits R_{\min} y R_{\max} dada la ocupación del buffer. La tasa de bits más baja (R_{\min}) se selecciona cuando el nivel del buffer se encuentra en un área de reserva B , una vez sobrepasada esta zona, la tasa de bits se basa en función del algoritmo que define a $f(B)$. El área intermedia entre la reserva y el punto en el que $f(B)$ es igual a R_{\max} se define como un amortiguador y pasado este punto se encuentra un área denominada reserva superior. El amortiguador siempre debe de mantenerse por encima de B para que sea capaz de absorber las fluctuaciones causadas por variaciones de capacidad. Por lo tanto, $f(B)$ siempre descargará un segmento antes de que el buffer contraiga al área de la reserva dentro del área segura, de lo contrario se encontrará en el área de riesgo [20, 21, 22].
- Adaptación mixta. El cliente hace su selección basándose en una combinación de parámetros que incluyen el ancho de banda disponible, la ocupación del buffer, el tamaño y/o duración del segmento, entre otros. ELASTIC (fEedback Linearization Adaptive STreamIng Controller), es un estimador basado en la teoría del control de retroalimentación, que genera un flujo TCP de larga duración, evitando un comportamiento de estado estacionario ON-OFF que conduce a sobreestimaciones de ancho de banda [23]. ELASTIC puede garantizar una equidad del ancho de banda entre clientes, pero sin tener en cuenta la QoE del espectador. SARA (Segment-Aware Rate Adaptation) se basa en variaciones del tamaño de cada segmento, estimación del ancho de banda disponible y la ocupación del buffer [24]. Como HTTP utiliza TCP, el rendimiento de un segmento depende del tamaño del archivo y, por lo tanto, los autores proponen personalizar el MPD (Media Presentation Description) para incluir este parámetro. Para cada nueva descarga de segmento, el estimador decide la representación del nuevo segmento basándose en el ancho de banda estimado y el estado actual del buffer. GTA (Game Theoretic Approach) es un esquema que utiliza la toma de decisiones de un conjunto de estimadores tomados de la literatura y aplica la teoría de juegos para lograr equidad entre conjuntos de clientes que compiten por el ancho de banda disponible [25].

3.2 Segmentación en DASH

En DASH, la segmentación del contenido multimedia es imprescindible, ya que esto permite el intercambio entre las diferentes representaciones o calidades del audio y/o vídeo durante una sesión de *streaming* según las condiciones de la red. Sin embargo, se plantea la pregunta ¿Cuál puede ser el tamaño óptimo para los segmentos sin afectar la calidad del contenido? Existen diferentes factores que pueden influir como el entorno que puede ser de acceso fijo o móvil, el tipo de contenido que será reproducido (animación, live action, deportes, etc.) o el servicio con el que se cuenta (básico o premium). Enfocándose en las condiciones de la red, por ejemplo, los segmentos de corta duración son buenos para adaptarse rápidamente a las fluctuaciones en el ancho de banda y evitar interrupciones durante la reproducción, en cambio los segmentos con mayor duración pueden tener una mejor eficiencia y calidad de codificación [8].

Para establecer una buena alternancia entre las diferentes representaciones o tasas de bits en distintos esquemas de *streaming* adaptativo como HLS o HTTP Live-*streaming*, es necesario comprender que los segmentos están constituidos por varios grupos de imágenes o GOP (Group of Pictures). El primer fotograma de un GOP es un intra-frame mejor conocido como I-frame, el cual contiene la mayor parte de información para una secuencia que va hacia adelante con respecto del tiempo, seguido de los fotogramas predictivos (P-frames), y los fotogramas bidireccionales (B-frames), que son predictivos tanto adelante como atrás. Los B-frames estiman cambios de fotogramas anteriores y posteriores. Los I-frames tienen un costo mayor en términos de bytes, ya que deben ser almacenados y transmitidos, mientras que los B-frames y P-frames aseguran una eficiencia sobre la codificación del vídeo. La posición de los I-frames depende de dos propiedades. En primer lugar, los I-frames son necesarios en el momento de realizar cambios de escena, en los que la imagen visualizada suele diferir significativamente. En segundo lugar, los I-frames se insertan para reducir errores una vez que se alcanza cierto umbral para una duración máxima en los GoP [8], esto puede verse de forma más clara en la Figura 4.

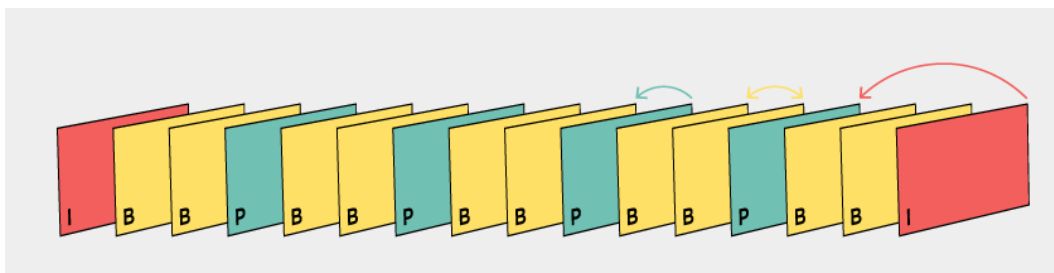


Figura 4: Secuencia de fotogramas que forman un GoP en un stream MPEG-DASH.

Por otro lado, es indispensable que los I-frames mantengan una posición fija, por ejemplo, después de 48 fotogramas, un I-frame tiene que posicionarse en un segmento de vídeo de 24 fps y su duración debe ser de dos segundos. Esto es necesario para garantizar que los I-frames se localicen al comienzo de cada segmento, lo que facilitaría poder alternar las representaciones entre diferentes segmentos. Y el decodificador no necesitará ninguna referencia ya sea de fotogramas o segmentos anteriores y, por lo tanto, el nuevo segmento puede tener diferentes tasas de bits. Asimismo, esto permitirá a los usuarios poder manipular la barra de reproducción, a fin de encontrar la escena deseada, la cual comenzará con un I-frame. Para poder establecer las posiciones fijas de los I-frames, es necesario que haya una restricción en el tamaño de los GOP con respecto al tamaño

del segmento. Los segmentos cortos tienden a estar en desventaja con respecto a los de mayor duración en la codificación final, y debido a ello, también se necesitan más I-frames para garantizar el cambio de representación en los límites de los segmentos. Esto conduce a una deficiencia en la codificación porque los I-frames no pueden aprovechar la predicción temporal, toda vez que necesitan de más bits para la codificación que los P-frames de predicción y, por lo tanto, la calidad general del contenido empeorará en comparación con una codificación convencional en la misma tasa de bits, como la utilizada para una descarga de tipo progresiva HTTP o los segmentos con tamaños de segmento más largos. Es un problema muy conocido y que debe ser considerado en la generación de contenido para DASH.

Los segmentos de menor tamaño pueden afectar la calidad del contenido en hasta 1.5 dB según el Peak Signal-to-Noise Ratio (PSNR) [26] obtenido. Sin embargo, la influencia de este efecto se va reduciendo de forma significativa conforme va aumentando la duración del segmento. Una duración inferior a dos segundos presenta un rendimiento muy pobre. En combinación con otros factores, como las características de la red, en general se deben de evitar segmentos pequeños principalmente de un segundo [27]. La Tabla 1, muestra los resultados obtenidos al hacer una comparación a segmentos de 1, 2, 4, 6, 10 y 15 segundos de tres de las 20 representaciones con las que se cuentan. La Figura 5 muestra cómo al incrementar la duración de los segmentos, independientemente de qué representación o tasa de bits se trate, va aumentando la calidad en términos del PSNR.

Tabla 1: PSNR obtenido de muestras de segmentos de vídeo de diferentes tamaños.

Tamaño del Segmento	1 seg. (24 frames)	2 seg. (48 frames)	4 seg. (96 frames)	6 seg. (144 frames)	10 seg. (240 frames)	15 seg. (360 frames)
300 kbit/s	41.48	41.65	43.80	49.04	49.15	49.18
1032 kbit/s	43.86	43.88	45.85	50.85	50.96	50.98
4200 kbit/s	45.03	45.91	50.98	51.23	51.87	51.92

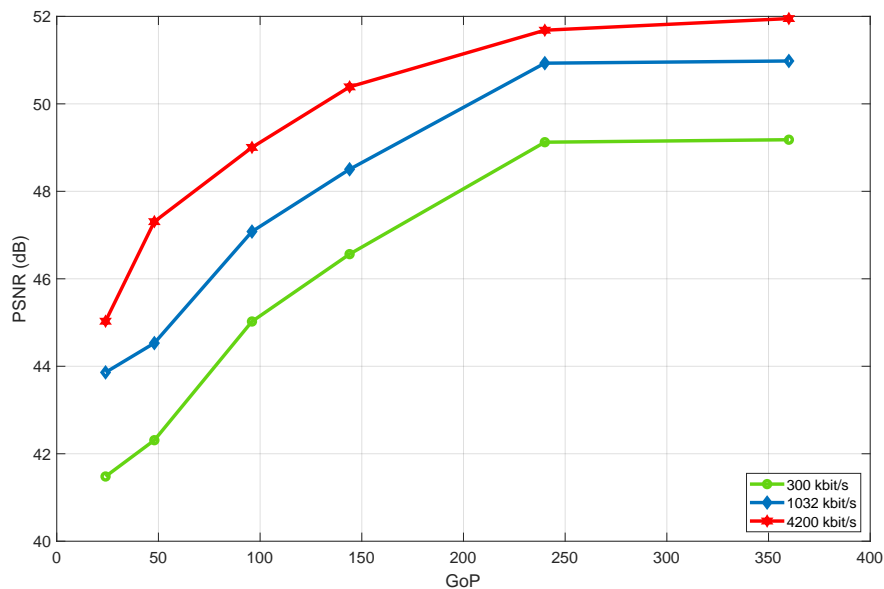


Figura 5: PSNR según el tamaño de los segmentos entre diferentes tasas de bits.

Como se ha mencionado anteriormente, los segmentos de mayor tamaño pueden causar interrupciones según las condiciones de la red debido a fluctuaciones en el ancho de banda. Por otro lado, la duración de segmentos muy cortos puede ocasionar un desempeño ineficiente en una sesión de *streaming* debido a sobrecargas ocasionadas por múltiples peticiones. En [28], se propone evaluar diferentes estándares de conexiones de Internet, a fin de mostrar diferentes niveles de impacto según el tamaño de los segmentos en un *streaming* por el throughput. Principalmente, consiste en probar la configuración del servidor HTTP, permitiendo conexiones persistentes o Keep Alive. Se utilizaron retardos en la red de entre 100 y 150 ms. Se analizaron los tamaños de los segmentos adecuados para una configuración de la red en los que las conexiones, sean persistentes o no, a través de los segmentos de 1, 2, 4, 6, 10 y 15 segundos de Big Buck Bunny del conjunto de datos. El tamaño de los segmentos que muestra un rendimiento óptimo para una conexión persistente está en el rango de entre 2 a 4, mientras que entre 5 a 10 segundos para el caso contrario. Cuando se utilizan segmentos más largos, un cliente no podría adaptarse con la misma rapidez que con segmentos más cortos y, por lo tanto, la tasa de bits en general tiende a deteriorarse para el caso de segmentos de tamaño mayor. Por otra parte, la influencia del tiempo de ida y vuelta en la red tiende a aumentar cuando se utilizan segmentos cortos, lo que afecta el rendimiento de conexiones no persistentes, porque en este caso se requiere establecer una conexión TCP con el servidor por cada segmento. Pero también, los resultados de la conexión persistente tienden a sufrir la influencia del retardo cuando se utilizan segmentos demasiado cortos. En conclusión, es recomendable usar segmentos para DASH o HLS de alrededor de 4 a 6 segundos, lo cual ayuda a establecer una mejor eficiencia entre la codificación y la flexibilidad para la adaptación a los cambios del ancho de banda. También, es importante conocer las características que permitan establecer conexiones HTTP persistentes, ya que es una forma fácil de aumentar el rendimiento en una sesión de *streaming*.

La mayoría de la literatura sobre DASH, se enfoca en optimizar algoritmos de tasa de bits, que utilizan tamaños de segmentos con duración fija y actúan variando la calidad entre estos a través de las representaciones que serán descargadas. Por otro lado, existen pocos trabajos cuyo objetivo de estudio se enfoca en utilizar el tamaño de los segmentos como un parámetro que puede ser ajustado de acuerdo con los cambios que se den, según las condiciones de la red. Sin embargo, no consideran que almacenar más segmentos requiere de un mayor espacio en los servidores lo que conlleva a un aumento en sus costos. Por ello, es importante definir qué estrategia de codificación (compresión) se utilizará en la creación de los segmentos en cada una de las representaciones de tasa de bits requeridas. Por lo anterior, surge la necesidad de utilizar contenido de vídeo más eficiente a través de algoritmos de compresión como es el caso de H.265/HEVC, capaz de preservar más de un 50% de la información [29]. La propuesta presentada en [30], da lugar a una ventana de oportunidad para establecer un sistema de *streaming* basado en DASH que permita la utilización distintos de algoritmos para optimizar tres frentes: Estimar las condiciones de la red por medio de algoritmos ABR, implementar un mecanismo que permita alternar al menos dos tamaños de segmentos y aprovechar las ventajas que cada uno representan dado el análisis previamente presentado.

3.3 Compresión de Vídeo

3.3.1 H.265/HEVC

H.265/HEVC (High Efficiency Video Coding) [31] está planteado para ser el sucesor de H.264, fue diseñado siguiendo los mismos principios, pero añadiendo mejoras en herramientas de codificación que permiten ahorrar hasta el 50% de la tasa de bits, conservando una calidad de percepción bastante comparable. Estas mejoras permiten aumentar el tamaño de la unidad básica de procesamiento hasta en 64 x 64 píxeles, añadiendo más modalidades de predicción intra-frame (33 en comparación con los 8 de AVC) o mejorando la Codificación Aritmética Binaria Adaptativa de Contenido (Content-Adaptive Binary Arithmetic Coding, CABAC), entre otros. Además, se introduce un filtro Offset Adaptativo de Muestras (SAO Sample Adaptive Offset) para mejorar la reconstrucción de amplitudes de la señal [32].

HEVC fue diseñado específicamente para proporcionar una gama de colores más amplia para la tecnología HDR y UHD, en lugar HD y SDR que ofrece H.264. La elección de usar un códec en particular se basa en parte en el hardware disponible. Con las CPU de la última década, más o menos, volviéndose cada vez más potentes, más baratas y abundantes, ha habido una tendencia en toda la industria a cambiar el espacio de almacenamiento en los discos duros por la computación a través de las CPU [33].

HEVC sigue los pasos de una tendencia que busca intercambiar la capacidad de almacenamiento por poder computacional. Esto significa que las redes que ya empleamos para entregar el vídeo en HD, serán capaces de entregar más datos en dicho vídeo, con un mayor rango dinámico, una gama de colores más amplia y resoluciones más grandes. La infraestructura actual no requerirá inmediatamente más ancho de banda. Por otro lado, las CPU en televisores, tablets y teléfonos inteligentes pueden decodificar más datos de las redes sin requerir más ancho de banda.

HEVC permite una menor velocidad de datos con una calidad visualmente equivalente en comparación a otros esquemas de compresión. Esto se debe a que, mientras H.264 descompone los fotogramas en cuadrados de píxeles llamados “*macrobloques*”, en HEVC los fotogramas son descompuestos en Unidades de Árbol de Codificación (CTU Coding Tree Units), los cuales pueden ser de hasta 64 x 64 píxeles. Cada uno de los macrobloques dentro de H.264 puede hacer sólo una predicción inter-frame o intra-frame, pero no ambas.

La compresión inter-frame es cuando los píxeles se toman de cuadros adyacentes en un vídeo, mientras que la compresión intra-frame es cuando los píxeles se toman prestados del mismo cuadro de vídeo. En HEVC, las CTU utilizan una combinación de compresión inter e intra-frame, lo que significa que hay más métodos para preservar el detalle en imágenes complejas [34]. En las Figuras 6 y 7 se pueden distinguir cómo son seccionados los fotogramas de acuerdo en cada enfoque.

Los bloques estimados pueden ser codificados en tamaños diferentes según un error residual. Cada CTU se codifica primero como un cuadrante de predicción, donde a profundidad el codificador decide si codifica con mezcla/salto, inter o intracodificación. El resultado de dichas predicciones se codifica entonces con un segundo cuadrante de predicción que, de forma opcional, se puede tener una mayor profundidad que el cuadrante de predicción. Esto permite que el error residual de una CTU intercodificada de 32x32 pueda representarse por una mezcla de bloques de 16 x 16, 8 x 8 y 4 x 4, como se muestra en la Figura 8.



Figura 6: Macrobloques de 16 x 16.



Figura 7: CTU de 64 x 64

Entre las mejoras a comparación de H.264, se pueden destacar:

- HEVC puede codificar vectores de movimiento con mucha más precisión, logrando una mejor predicción de bloques con menor error residual. Por ejemplo, existen 35 direcciones intra-frames, en comparación de sólo 9 para H.264.
- Se incluye un método de predicción adaptativa para vectores de movimiento para mejorar la inter-predicción.
- Un filtro de desbloqueo mejorado.
- Un filtro Offset Adaptativo de Muestra, que reduce los artefactos en los bordes de los bloques.

HEVC comprime los datos de manera mucho más eficiente, ya que al utilizarlo como una herramienta de compresión de vídeo, sus requisitos de ancho de banda y almacenamiento se reducen aproximadamente en un 50%. La Tabla 1, compara el ancho de banda para la codificación H.264 vs. H.265.

Tabla 2: Comparación entre las capacidades de compresión de H.264 y H.265.

Resolución	H.264/AVC	H.265/HEVC
480p	1.5 mbps	0.75 mbps
720p	3 mbps	1.5 mbps
1080p	6 mbps	3 mbps
4K	32 mbps	15 mbps

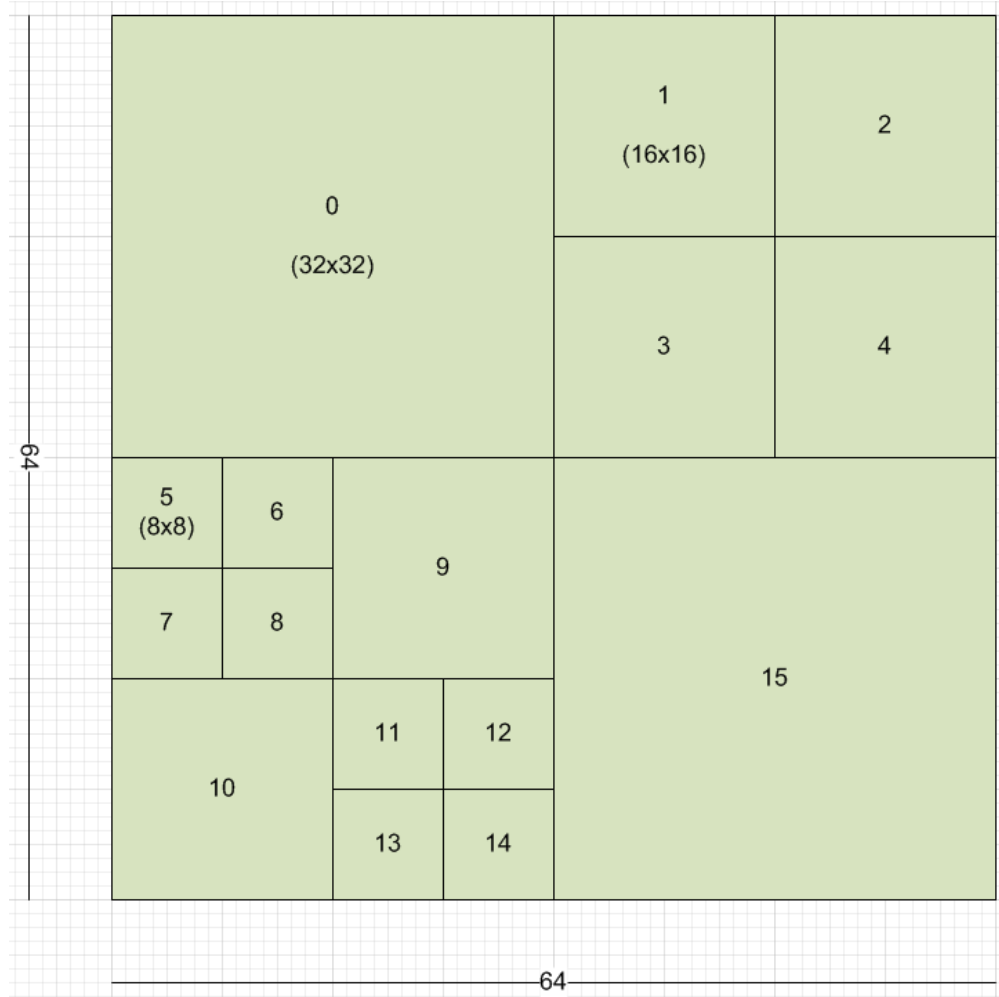


Figura 8: Partición en bloques de diferentes dimensiones para HEVC.

Moving Picture Experts Group (MPEG) fue el encargado de supervisar el desarrollo de HEVC, de igual forma que MPEG-2, MP3 y H.264. Debido a su complejidad, HEVC no ha sido aceptado de igual manera como AVC, por lo que han surgido otras opciones que son descritas más adelante.

3.3.2 VP9

VP9 es un formato de compresión de vídeo de código abierto que en la actualidad está en desarrollo por WebM y Google. El trabajo en VP9 comenzó a finales de 2011 y sustituye a VP8. Su objetivo principal es proporcionar soluciones de vídeo basándose en una web libre. VP9 transmite el vídeo de forma más eficiente que VP8, una de las mayores ventajas de este códec es que está libre de derechos de autor. HEVC por su lado, cobra el uso por regalías de patente y debido a esto Google espera que VP9 alcance un uso mayor al emplearlo en Youtube [35].

3.3.3 AV1

Por otro lado, un conjunto de empresas entre las que destacan Amazon, Cisco, Intel, Microsoft, Mozilla y Netflix crearon una alianza con la que se anuncia la creación de un nuevo códec en desarrollo, conocido como AV1. Este códec puede utilizarse tanto en vídeo como en imágenes. Para el primer caso, puede llegar a superar la compresión de VP9 y HEVC en hasta un 40%, aunque por lo regular se encuentra en un 25%. En el segundo, supera por un 15% en eficiencia a HEIF (High Efficiency Image File Format), que a su vez reduce a la mitad el tamaño de JPEG [36].

Aunque muchos dispositivos de reproducción, sistemas operativos, entre otros soportan HEVC, los proveedores de servicios de *streaming* se preparan para adoptar AV1. HEVC puede ser útil para señales de televisión ATSC 3.0 [37]. Sin embargo Google, Amazon, Apple, Hulu y Netflix pretenden adoptar la especificación AV1 1.0. A partir de este año, Netflix ha comenzado el *streaming* en dispositivos Android en este formato para algunos de sus títulos como fase de pruebas [38]. Como lo muestran las Figuras, 4, 5 y 6 [39, 40, 41] el soporte para HEVC, WebM y AV1 respectivamente es aún disparado debido a lo antes mencionado, siendo que el contenido de vídeo en HEVC puede reproducirse mediante aplicaciones de terceros con el propósito de realización de pruebas.

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser
	12-18			3.1-10.1		3.2-10.3							4	
6-10	79	2-73	4-79	1-12.1	10-65	11-13.2		2.1-4.4.4	12-12.1				5-10.1	
11	80	74	80	13	66	13.3	all	80	46	80	68	12.12	11.1	1.2
		75-76	81-83	TP										

Figura 9: Soporte para HEVC en browsers.

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser
	12-13	2-3.6	4-5		10.1									
	14-18	4-27	6-24	3.1-12	11.5-15	3.2-12.1		2.1-2.2					4	
6-10	79	28-73	25-79	12.1	16-65	12.2-13.2		2.3-4.4.4	12-12.1				5-10.1	
11	80	74	80	13	66	13.3	all	80	46	80	68	12.12	11.1	1.2
		75-76	81-83	TP										

Figura 10: Soporte para WebM en browsers.

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser
		2-54												
		55-60												
		61-62												
		63-64												
	12-17	65	4-66											
	18	66	67-69		10-56									
6-10	79	67-73	70-79	3.1-12.1	57-65	3.2-13.2		2.1-4.4.4	12-12.1				4-10.1	
11	80	74	80	13	66	13.3	all	80	46	80	68	12.12	11.1	1.2
		75-76	81-83	TP										

Figura 11: Soporte para AV1 en browsers.

4 Objetivos

En esta sección, se plantean los objetivos que se desean alcanzar, basándose en los antecedentes y el estado del arte previamente presentados, así como en el trabajo de investigación realizado hasta ahora. Se distinguen varias líneas de investigación que son importantes de atender para el desarrollo de este proyecto, así como la metodología que se seguirá para cumplirlos.

4.1 Objetivo General

Establecer un mecanismo alternativo que permita mejorar la adaptación de tasa de bits basado en las implementaciones de *streaming* de vídeo actuales, principalmente en aquellos cuya funcionalidad depende de las condiciones en la red del lado del cliente. Dicho mecanismo será apoyado con el uso de dos o más series de segmentos de diferente duración, y permitirá identificar algunas ventajas y limitaciones de enfoques existentes. Por otro lado, y acorde con dicha implementación se pretende analizar el desempeño de al menos un mecanismo de codificación para el *streaming* de vídeo que ayude a mejorar su funcionamiento de forma adecuada.

4.2 Objetivos Particulares

Los objetivos particulares de esta propuesta de investigación se centran principalmente en el mejoramiento de los algoritmos de estimación de tasa de bits apoyándose en una serie de parámetros que harán más eficiente la toma de decisiones durante una sesión de *streaming*

- Realizar un estudio sobre los mecanismos que componen al *streaming* de vídeo, identificando ventajas y limitantes que permitan establecer nuevas métricas o parámetros de dichos mecanismos considerando también la calidad de la experiencia (QoE) percibida y hacer más eficientes las estimaciones de los algoritmos de adaptación de tasa de bits, de acuerdo al entorno de red en que se ejecuten.
- Diseñar e implementar un mecanismo de *streaming* que contemple utilizar de manera dinámica al menos dos conjuntos de segmentos de vídeo de distinta duración. Estos estarán apoyados por las estimaciones dadas las condiciones de la red y evaluar su desempeño.
- Estudiar y analizar el posible desempeño de integrar a la plataforma de *streaming* un módulo que utilice codificación de alta eficiencia, e.g. H.265, VP9, entre otros. Esto con base en la fragmentación de los diferentes esquemas y su dependencia con los navegadores más utilizados y reproductores compatibles.

5 Metodología de Investigación

A continuación, se describen las principales actividades propuestas para el desarrollo de este proyecto de investigación y así cumplir con los objetivos anteriormente mencionados.

1. *Revisión del estado del arte.* Realizar un profundo estudio de la bibliografía especializada que permita un alto grado de comprensión y aplicación de los conceptos relacionados. Lo anterior, llevará a detectar ventajas y limitaciones en los esquemas de adaptación y la segmentación dinámica.
2. *Herramientas.* Elaboración del conjunto de herramientas para dar soporte a los mecanismos propuestos, tales como el contenido multimedia que será reproducido en una plataforma de *streaming* de código abierto que permita la implementación de diversos módulos personalizables según las necesidades con las que se deseen experimentar.
3. *Modelado.* Modelar e implementar un mecanismo basado en la selección de las mejores propuestas para implementar nuevas características en la plataforma de *streaming*. El enfoque es principalmente en los algoritmos de adaptación de tasa de bits, lo que permitirá la identificación de posibles escenarios compatibles y alternos. Lo anterior, en combinación con el uso dinámico de segmentos de vídeo de diferente duración en distintos tipos de contenido. Con ello, se podrán establecer nuevos parámetros y métricas relacionadas a la calidad de la experiencia (QoE) para establecer una toma de decisiones más exacta.
4. *Desempeño y resultados.* Realizar pruebas con base en escenarios de los mecanismos y métodos de *streaming*, para obtener posibles comparaciones, y así establecer nuevas pruebas. Esto ayudará a seleccionar mecanismos y parámetros más robustos de acuerdo a los resultados obtenidos de dichas pruebas. Con esto, también se podrán realizar las adecuaciones necesarias para optimizar sus características.

6 Avances

El trabajo realizado hasta el momento, comprende una nueva implementación de una plataforma de *streaming* basada en DASH, la cual está conformada por tres componentes: clientes, emulador de red y un servidor web, como lo muestra la Figura 12.

El reproductor DASH en el cliente se encarga de solicitar y descargar el archivo manifiesto o MPD, que será analizado por el *Parser* que permitirá establecer cómo se encuentra conformada la organización de los segmentos almacenados en el servidor web. Mientras que el emulador Dummynet permitirá crear y manipular diferentes escenarios de tráfico en la red, emulando el comportamiento de uno o más enlaces.

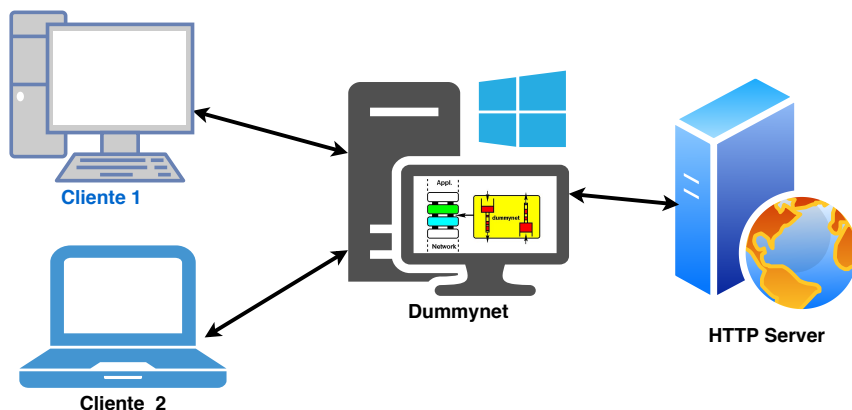


Figura 12: Organización del banco de trabajo para pruebas.

Por otro lado, el trabajo también se centra en la generación de nuevas fuentes de vídeo con segmentos de diferente duración (4, 8 y 12 segundos). En la Tabla 3 se muestran las características que tendrán. El *parser* del reproductor DASH puede ser adaptado a la lectura de diferentes parámetros como por ejemplo, el tamaño de los segmentos o el peso de percepción de calidad como el SSIM o SSIMPLUS. Esto dependerá del tipo de contenido con el propósito de generar distintos escenarios que permitan establecer un mecanismo de intercambio dinámico entre segmentos ayudando a mejorar los procesos de adaptación basándose en la propuesta de [42]. Este busca obtener un promedio de un conjunto de estimaciones para definir una probabilidad que le ayude a determinar si es necesario realizar un cambio de los segmentos utilizados en ese momento. En cambio, este proyecto de investigación busca apoyarse de los algoritmos ABR como un parámetro más preciso para determinar este cambio.

Por otro lado, para poder utilizar HEVC en Firefox y Chrome, se requerirá la implementación o adecuación de una biblioteca que pueda adaptarse a DASH. Tal es el caso de libde265 [43], una biblioteca en Javascript que permite visualizar contenido en HEVC a través de un sitio web. Sin embargo, se requerirá trabajo de optimización, toda vez que no funciona de forma óptima en resoluciones arriba de 720p.

Tabla 3: Tasas de bits a codificar en HECV, WebM, AV1 y AVC.

Tasa de bits (Kbps)	Resolución (Píxeles)	Cuadros por Segundo (fps)
45	320 x 240	30
88	320 x 240	30
128	320 x 240	30
177	480 x 360	30
217	480 x 360	30
255	480 x 360	30
323	480 x 360	30
378	480 x 360	30
509	854 x 480	30
577	854 x 480	30
782	1280 x 720	30
1008	1280 x 720	30
1207	1280 x 720	30
1473	1280 x 720	30
2087	1920 x 1080	30
2409	1920 x 1080	30
3340	1920 x 1080	30
4000	1920 x 1080	30
5000	1920 x 1080	30
6000	1920 x 1080	30

7 Calendario

Tomando en consideración los objetivos particulares y la metodología antes mencionados, la Tabla 4 muestra el cronograma de actividades en el que se reflejan los tiempos establecidos para la consecución del proyecto de investigación, que va del 1 de enero de 2020 al 31 de diciembre de 2023, restando tres años a la presentación original del presente documento.

Tabla 4: Distribución temporal de actividades para el desarrollo.

Actividad	19-O	20-I	20-P	20-O	21-I	21-P	21-O	22-I	22-P	22-O	23-I	23-P	23-O
Revisión del estado del arte													
Definición de alcances del proyecto													
Realización de pruebas y experimentos													
Preparación y presentación del examen pre-doctoral													
Implementación de las propuestas													
Realización de pruebas y experimentos por etapas													
Redacción y presentación de artículo de congreso													
Redacción de tesis doctoral													
Redacción de artículo de revista													
Preparación y defensa de la tesis doctoral													

Referencias

- [1] “ISO/IEC 23009-1:2019.” [Online]. Available: <https://www.iso.org/standard/75485.html/>, Aug 2019.
- [2] Tsbmail, “H.264 : Advanced Video Coding for Generic Aaudiovisual Services.” [Online]. Available: <https://www.itu.int/rec/T-REC-H.264-201906-I/en>, Jan 2020.
- [3] “Video Basics.” [Online]. Available: <http://wsa.wikidot.com/tbm:video-basics>.
- [4] “Video Delivery in HTTP.” [Online]. Available: <http://www.roman10.net/video-delivery-in-http>.
- [5] Apple, “HTTP Live Streaming.” [Online]. Available: <https://developer.apple.com/streaming/>.
- [6] “Real-Time Messaging Protocol (RTMP) Specification.” [Online]. Available: <https://www.adobe.com/devnet/rtmp.htm>.
- [7] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, “A Survey on Quality of Experience of HTTP adaptive streaming,” *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.
- [8] I. Sodagar, “The MPEG-DASH Standard for Multimedia Streaming Over the Internet,” *IEEE Multimedia*, vol. 18, pp. 62–67, Apr. 2011.
- [9] “Cisco Visual Networking Index: Forecast and Trends, 2018–2023 White Paper.” [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html/>, Mar 2020.
- [10] Y. Jing and Q. Gao, “Design and Implementation of Live Streaming System for Wearable Devices,” in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, IEEE, Nov. 2018.
- [11] V. P. K. M and S. Mahapatra, “Quality of Experience Driven Rate Adaptation for Adaptive HTTP Streaming,” *IEEE Transactions on Broadcasting*, vol. 64, pp. 602–620, June 2018.
- [12] C. Yao, J. Xiao, Y. Zhao, and A. Ming, “Video Streaming Adaptation Strategy for Multiview Navigation over DASH,” *IEEE Transactions on Broadcasting*, vol. 65, pp. 521–533, Sept. 2019.
- [13] Z. Akhtar, K. Siddique, A. Rattani, S. L. Lutfi, and T. H. Falk, “Why is Multimedia Quality of Experience Assessment a Challenging Problem?,” *IEEE Access*, vol. 7, pp. 117897–117915, 2019.
- [14] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1649–1668, Dec. 2012.
- [15] A. Bentalb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, “A Survey on Bitrate Adaptation Schemes for Streaming Media over HTTP,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 562–585, 2019.
- [16] “MPEG DASH Requirements for a Webpush Protocol.” [Online]. Available: <https://tools.ietf.org/html/draft-begen-webpush-dash-reqs-00>, Oct 2014.
- [17] T. Stockhammer, “Dynamic Adaptive Streaming over HTTP –,” in *Proceedings of the second annual ACM conference on Multimedia systems*, ACM Press, 2011.
- [18] C. Liu, I. Bouazizi, and M. Gabbouj, “Rate Adaptation for Adaptive HTTP Streaming,” in *Proceedings of the second annual ACM conference on Multimedia systems*, ACM Press, 2011.
- [19] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, “Probe and Adapt: Rate Adaptation for HTTP Video Streaming at Scale,” *IEEE Journal on Selected Areas in Communications*, vol. 32, pp. 719–733, Apr. 2014.

- [20] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, “A Buffer-Based Approach to Rate Adaptation,” in *Proceedings of the 2014 (ACM) conference on (SIGCOMM) - (SIGCOMM)*, ACM Press, 2014.
- [21] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, “BOLA: Near-Optimal Bitrate Adaptation for Online Videos,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, IEEE, Apr. 2016.
- [22] “DASH Industry Forum: Catalyzing the Adoption of MPEG.” [Online]. Available: <https://dashif.org/>.
- [23] L. D. Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, “ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH),” in *2013 20th International Packet Video Workshop*, IEEE, Dec. 2013.
- [24] P. Juluri, V. Tamarapalli, and D. Medhi, “SARA: Segment Aware Rate Adaptation Algorithm for Dynamic Adaptive Streaming over HTTP,” in *2015 IEEE International Conference on Communication Workshop (ICCW)*, IEEE, June 2015.
- [25] A. Bentaléb, A. C. Begen, S. Harous, and R. Zimmermann, “Want to Play DASH?,” in *Proceedings of the 9th ACM Multimedia Systems Conference on - (MMSys)*, ACM Press, 2018.
- [26] A. Horé and D. Ziou, “Image Quality Metrics: PSNR vs. SSIM,” in *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369, 2010.
- [27] J. Ozer, “Choosing the Optimal Segment Duration - Streaming Learning Center.” [Online]. Available: <https://streaminglearningcenter.com/blogs/choosing-the-optimal-segment-duration.html>, 2016.
- [28] S. Lederer, “MPEG-DASH and HLS Segment Length for Adaptive Streaming.” [Online]. Available: <https://bitmovin.com/mpeg-dash-hls-segment-length/>, 2020.
- [29] O. Zach and M. Slanina, “Content Aware Segment Length Optimization for Adaptive Streaming over HTTP,” *Radioengineering*, vol. 27, no. 3, p. 819–826, 2018.
- [30] L. Bedogni, M. Di Felice, and L. Bononi, “Dynamic Segment Size Selection in HTTP based Adaptive Video Streaming,” in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 665–670, 2017.
- [31] “ITU-T H.265 High Efficiency Video Coding.” [Online]. Available: <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11885>, Nov 2019.
- [32] O. Zach and M. Slanina, “A Comparison of H.265/HEVC Implementations,” in *Proceedings ELMAR-2014*, pp. 1–4, Sep. 2014.
- [33] “Everything You Need to Know About 4K HDR TVs.” [Online]. Available: <https://www.consumerreports.org/tvs/everything-you-need-to-know-about-4k-hdr-tvs/>, Feb 2020.
- [34] M. Malhotra, A. V. Singh, and R. Matam, “Comparative Performance Issues with H.264 vs H.265,” in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 283–288, Feb 2019.
- [35] “VP9: Faster, Better, Buffer-Free YouTube Videos.” [Online]. Available: <https://youtube-eng.googleblog.com/2015/04/vp9-faster-better-buffer-free-youtube.html>, Apr 2015.
- [36] Z. Deng and I. Moccagatta, “Hardware-Friendly Inter Pprediction Techniques for AV1 Video Coding,” in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 948–952, Sep. 2017.
- [37] “¿Qué es ATSC 3.0? Y Por Qué es Importante Conocerlo.” [Online]. Available: <https://www.videoservicios.com.mx/que-es-atsc-3-0-y-por-que-debemos-conocerlo/>, Jul 2018.
- [38] J. Porter, “Netflix Begins Streaming Data-Saving AV1 Videos on Android.” [Online]. Available: <https://www.theverge.com/2020/2/6/21126039/>

- `netflix-android-av1-video-codec-efficient-support-bandwidth-quality-vp9`,
Feb 2020.
- [39] “HEVC/H.265 Video Format.” [Online]. Available: <https://caniuse.com/#feat=hevc>, Mar 2020.
 - [40] “WebM Video Format.” [Online]. Available: <https://caniuse.com/#feat=webm>, Mar 2020.
 - [41] “AV1 Video Format.” [Online]. Available: <https://caniuse.com/#feat=av1>, Mar 2020.
 - [42] H. Azwar and Hendrawan, “H.265 Video Ddelivery Uusing Ddynamic Aadaptive Streaming over HTTP (DASH) on LAN Network,” in *2014 8th International Conference on Telecommunication Systems Services and Applications (TSSA)*, pp. 1–6, Oct 2014.
 - [43] Copyright (c) 2014 Struktur AG, “libde265.” [Online]. Available: <https://github.com/strukturag/libde265.js>, Jul 2015.