UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

TESIS DOCTORAL

**CONTRIBUCIÓN A LOS MECANISMOS DE ADAPTACIÓN
DINÁMICA PARA SERVICIOS DE DISTRIBUCIÓN
MULTIMEDIA SOBRE REDES MÓVILES**

MIGUEL ÁNGEL AGUAYO ORTUÑO

M.C. EN ELECTRÓNICA Y TELECOMUNICACIONES

2020

Departamento de Ingeniería de Sistemas Telemáticos
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE MADRID

TESIS DOCTORAL

**CONTRIBUCIÓN A LOS MECANISMOS DE
ADAPTACIÓN DINÁMICA PARA SERVICIOS DE
DISTRIBUCIÓN MULTIMEDIA SOBRE REDES
MÓVILES**

Autor

Miguel Ángel Aguayo Ortuño
M.C. en Electrónica y Telecomunicaciones

Director

Luis Bellido Triana
Doctor Ingeniero de Telecomunicación

Madrid, Diciembre 2020

POLITÉCNICA

Tribunal nombrado por el Magfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día X de diciembre de 2020.

Título: Contribución a los mecanismos de adaptación dinámica para servicios de distribución multimedia sobre redes móviles

Autor: D. Miguel Ángel Aguayo Ortuño

Director: D. Luis Bellido Triana

TRIBUNAL CALIFICADOR

Presidente:	D. David Larrabeiti López	Catedrático de Universidad Universidad Carlos III de Madrid
Vocales:	D. David Fernández Cambroneró	Profesor Titular de Universidad Universidad Politécnica de Madrid
	D. Alfonso Sánchez-Macián Pérez	Profesor Titular de Universidad Universidad Antonio de Nebrija
	D. Jorge E. López de Vergara Méndez	Profesor Titular de Universidad Universidad Autónoma de Madrid
Secretario:	Dña. Ana Belén García Hernando	Profesora Titular de Universidad Universidad Politécnica de Madrid
Suplentes:	Dña. Jasone Astorga Burgo	Profesora Titular de Universidad Universidad del País Vasco
	D. Ignacio Soto Campos	Profesor Titular de Universidad Universidad Carlos III de Madrid

Realizando el acto de defensa y lectura de la Tesis Doctoral en Madrid, el día x de diciembre del 2020.

CALIFICACIÓN:.....

El Presidente

Los Vocales

La Secretaria

*Para mi esposa Karla,
mi hijo Miguel Ángel y
mis padres Miguel y Susana*

Resumen

La amplia adopción de dispositivos móviles con capacidad para servicios multimedia, la disponibilidad de mejores redes con mayores anchos de banda y la disponibilidad de plataformas que ofrecen contenido digital han llevado a una creciente popularidad de los servicios de distribución de contenido multimedia. Sin embargo, los servicios de distribución de contenido multimedia pueden estar sujetos a diferentes factores que afectan a la calidad percibida por los usuarios, como las interrupciones del servicio o las oscilaciones de calidad debidas a las cambiantes condiciones de la red, especialmente en las redes móviles.

El estándar “Dynamic Adaptive Streaming over HTTP (DASH)” ha surgido como una solución para ofrecer el servicio de distribución del contenido multimedia. *DASH* aprovecha el uso de redes de distribución de contenido y las capacidades de los dispositivos multimedia para permitir a los reproductores multimedia adaptar dinámicamente la calidad de la transmisión multimedia al ancho de banda disponible y a las características del dispositivo. Como parte de la Tesis doctoral se realizó un estudio del estado del arte de la tecnología para ofrecer el servicio de distribución de contenido multimedia utilizando el estándar *DASH* con el fin de detectar limitaciones y/o problemas en el servicio. Si bien muchos elementos de *DASH* están estandarizados, los algoritmos que proporcionan la adaptación dinámica en el reproductor *DASH* no lo son.

La adaptación a menudo se basa en la estimación del rendimiento o un mecanismo de control del buffer, es por esto por lo que también se realizó un análisis de los distintos algoritmos de tasa adaptativa propuestos por la comunidad científica con el fin de detectar áreas de oportunidad para mejorar la calidad en los servicios de distribución multimedia para el estándar *DASH*.

El objetivo de la presente Tesis Doctoral es mejorar la calidad percibida por los usuarios mediante la propuesta de soluciones para los servicios de distribución multimedia en redes móviles. La contribución principal de la Tesis Doctoral se ha concretado en un nuevo algoritmo de tasa adaptativa para estimar el throughput del reproductor *DASH* basado en un método estadístico denominado “Adaptive Forgetting Factor (AFF)”. Usando este método la lógica de adaptación puede reaccionar apropiadamente a las diferentes condiciones de distintos tipos de redes de acceso, en especial en las redes móviles.

Para validar la propuesta del algoritmo de tasa adaptativa de estimación del throughput basado en el método *AFF*, se han desarrollado escenarios y varias plataformas virtualiza-

das para evaluar el algoritmo de tasa adaptativa. Este trabajo se ha realizado en el marco del proyecto INNPACTO LTextreme financiado por el Ministerio de Ciencia e Innovación de España.

Palabras Clave: Ancho de Banda, Comunicaciones Móviles, Comunicación Multimedia, Diseño y Análisis de Algoritmos, Distribución Adaptativa sobre HTTP, Estimación, Factor de Olvido Adaptativo, Multimedia, Rendimiento.

Abstract

The wide adoption of multimedia-capable mobile devices, the availability of better networks with higher bandwidths, and the availability of platforms offering digital content has led to an increasing popularity for multimedia streaming services. Nonetheless, multimedia streaming services can be subject to different factors that affect the quality perceived by the users, such as service disruptions or quality oscillations due to the changing network conditions, especially in mobile networks.

The Dynamic Adaptive Streaming over HTTP (DASH) standard has emerged as a solution for multimedia streaming services. *DASH* leverages the use of content-distribution networks and the capabilities of multimedia devices to allow multimedia players to dynamically adapt the quality of the media streaming to the available bandwidth and the device characteristics. As part of this thesis, a state of the art review of *DASH* multimedia streaming services has been conducted in order to detect limitations or problems with the service. While multiple elements of *DASH* are standardized, the algorithms providing the dynamic adaptation for the streaming are not.

The adaptation algorithm is often based on the estimation of the throughput or a buffer control mechanism. Therefore, a study of the different algorithms proposed by the scientific community has been performed to detect areas of opportunity to enhance the quality of *DASH* multimedia streaming services.

The overall objective of this thesis is to improve user perceived quality through the proposal of solutions for multimedia streaming services in mobile networks. The central thesis contribution is a new adaptive bitrate algorithm that estimates the throughput of the *DASH* player based on a statistical method named Adaptive Forgetting Factor (AFF). Using this method, the adaptation logic is able to react appropriately to the different conditions of different types of networks, especially mobile networks.

In order to validate the proposal of the adaptive bitrate algorithm based on the *AFF* estimation method, a set of experiments and several virtualized platforms have been developed to evaluate the adaptive rate algorithm. This work has been accomplished within the framework of the INNPACTO LTextreme project supported by the Ministry of Science and Innovation of Spain.

Key Words: Algorithm Design and Analysis, Adaptive Forgetting Factor, Adaptive Strea-

ming over HTTP, Bandwidth, Estimation, Multimedia, Mobile Communication, Multimedia Communication, Throughput.

Agradecimientos

En primer lugar, quisiera agradecer a mi director de tesis, Dr. Luis Bellido Triana, ya que su guía ha sido invaluable durante el transcurso de mi estancia en el Doctorado. Su dedicación y conocimientos han sido un pilar fundamental para poder finalizar este trabajo de investigación.

En segundo lugar, quisiera agradecer a la Dra. Encarna Pastor, por sus aportaciones y valiosas recomendaciones en el desarrollo de este trabajo de investigación. También quisiera agradecer al Dr. David Fernández por el apoyo para implementar los escenarios de prueba que sirvieron para el desarrollo de esta Tesis Doctoral. Por otra parte, quisiera agradecer a Carlos Lentisco y a Ricardo Flores su amistad la cual ha servido a que el tiempo dedicado al desarrollo de este trabajo de investigación haya sido mejor.

En tercer lugar, quisiera agradecer a la Secretaria de Educación Publica y a la Universidad de Colima por proporcionarme la financiación necesaria para alcanzar la meta del doctorado. Además, quisiera agradecer a la Universidad Politécnica de Madrid y en especial al Departamento de Ingeniería de Sistemas Telemáticos por permitirme realizar mis estudios de doctorado.

Finalmente quiero agradecer a mi esposa Karla la cual ha sido mi apoyo incondicional durante este largo tiempo en este país extranjero, te amo. Juntos empezamos este viaje sin saber las dificultades que encontraríamos, sin embargo, la vida nos ha dado el mejor regalo, que es el nacimiento de nuestro hijo Miguel Ángel, el cual ha logrado que todo sea mejor. Miguel Ángel, quiero decirte que junto a tu mamá eres lo más importante en mi vida y que haré todo lo que este en mis manos por ustedes dos, te amo. También quiero agradecer a mis padres Miguel y Susana sin los cuales no sería la persona que soy ahora y por todo el apoyo que siempre me han brindado.

ÍNDICE DE CONTENIDO

CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS	1
1.1 INTRODUCCIÓN Y MOTIVACIÓN	1
1.2 ÁMBITO DE LA TESIS DOCTORAL	4
1.3 OBJETIVOS DE LA TESIS DOCTORAL	5
1.4 METODOLOGÍA DE INVESTIGACIÓN	5
1.5 ESTRUCTURA DE LA MEMORIA	7
CAPÍTULO 2: SERVICIOS DE DISTRIBUCIÓN MULTIMEDIA	10
2.1 INTRODUCCIÓN	10
2.2 HISTORIA DE LOS SERVICIOS DE TRANSMISIÓN MULTIMEDIA	11
2.2.1 Real Time Streaming Protocol (RTSP)	12
2.2.2 Red de Distribución de Contenidos (CDN)	13
2.2.3 Progressive download streaming	15
2.2.4 HTTP Adaptive Bitrate Streaming	16
2.3 DYNAMIC ADAPTIVE STREAMING OVER HTTP (DASH)	17
2.3.1 MPD	19
2.3.2 Contenido Multimedia soportado en DASH	21
2.3.3 Arquitectura DASH	23
2.3.4 Métricas para medir de la calidad en DASH	26
2.3.5 Herramientas para DASH	28
2.4 ESQUEMAS DE DISTRIBUCIÓN DEL CONTENIDO DASH	31
2.4.1 Servicio de distribución multimedia bajo demanda	32
2.4.2 Servicio de distribución multimedia en vivo	32
2.4.3 Las CDN en el servicio de distribución multimedia	33
2.4.4 Servicio de difusión multimedia	34
2.4.5 Handover para servicios de difusión DASH	35

2.5	CONCLUSIONES	36
CAPÍTULO 3: ALGORITMOS DE ADAPTACIÓN EN DASH		38
3.1	INTRODUCCIÓN	38
3.2	ALGORITMOS DE TASA ADAPTATIVA (ABR)	39
3.3	ESTADO DEL ARTE SOBRE ALGORITMOS DE ADAPTACIÓN	44
3.3.1	Algoritmos basados en el throughput	45
3.3.2	Algoritmos basados en el buffer	49
3.3.3	Algoritmos basados en teoría de control o híbridos	54
3.3.4	Conclusiones sobre los algoritmos para DASH	65
3.4	MODELOS DE CALIDAD DE LA EXPERIENCIA (QoE) PARA DASH	66
3.5	CONCLUSIONES	71
CAPÍTULO 4: MÉTODO ADAPTIVE FORGETTING FACTOR (AFF)		73
4.1	INTRODUCCIÓN	73
4.2	ESTIMACIÓN DEL THROUGHPUT EN DASH	74
4.2.1	Qué es el método de estimación AFF	75
4.2.2	Actualización del factor adaptativo	78
4.2.3	Comportamiento óptimo del método AFF	83
4.3	PSEUDO-CÓDIGO PARA EL ALGORITMO AFF	84
4.4	COMPARACIÓN CON OTROS MÉTODOS DE ESTIMACIÓN	85
4.5	CONCLUSIONES	87
CAPÍTULO 5: ESCENARIOS Y MEDIDAS DE VALIDACIÓN		89
5.1	INTRODUCCIÓN	89
5.2	REQUISITOS PARA EVALUAR EL SERVICIO DE DISTRIBUCIÓN MULTIMEDIA . .	90
5.2.1	Contenido DASH	90
5.2.2	Reproductor DASH	91
5.3	DISEÑO Y DESARROLLO DE LA PLATAFORMA DE VALIDACIÓN	92
5.3.1	Antecedentes	93
5.3.2	Plataforma virtualizada	94
5.4	ESCENARIOS DE EVALUACIÓN	96
5.4.1	Perfiles de tráfico de una red móvil (LTE) real.	96
5.4.2	Perfiles de tráfico artificiales	100
5.5	CONCLUSIONES	102
CAPÍTULO 6: RESULTADOS		103
6.1	INTRODUCCIÓN	103
6.2	EVALUACIÓN DEL RENDIMIENTO	104
6.2.1	Escenario 1	105

6.2.2	Escenario 2	107
6.2.3	Escenario 3	109
6.2.4	Escenario 4	111
6.3	MÉTRICAS DE QOE DE LOS MÉTODOS DE ESTIMACIÓN	112
6.4	PRUEBA DE EQUIDAD (FAIRNESS)	115
6.5	CONCLUSIONES	116
CAPÍTULO 7: CONCLUSIONES Y TRABAJOS FUTUROS		118
7.1	INTRODUCCIÓN	118
7.2	OBJETIVOS Y RESULTADOS	118
7.3	CONCLUSIONES GENERALES	120
7.4	DIVULGACIÓN DE RESULTADOS	122
7.4.1	Artículo de Revista	122
7.4.2	Artículos de Congresos	122
7.4.3	Proyectos	123
7.5	TRABAJOS FUTUROS	124
BIBLIOGRAFÍA		126
LISTA DE ACRÓNIMOS		144

ÍNDICE DE FIGURAS

FIGURA 1.1. TRÁFICO DE ENTRETENIMIENTO EN TIEMPO REAL VNI [CISCO18]	1
FIGURA 1.2. ESQUEMA GENERAL QUE RESUME ESTA TESIS DOCTORAL	7
FIGURA 2.1. FUNCIONAMIENTO DEL PROTOCOLO RTSP	13
FIGURA 2.2. ELEMENTOS QUE COMPONEN UNA CDN [Buyya08]	14
FIGURA 2.3. DASH MODELO DE DATOS DE ALTO NIVEL [ISO14b]	20
FIGURA 2.4. ARQUITECTURA DASH DE ALTO NIVEL [ISO14b]	23
FIGURA 2.5. PARTES DEL CLIENTE DASH [3GPP18a]	24
FIGURA 2.6. FUNCIONAMIENTO DE LA ARQUITECTURA DASH	26
FIGURA 3.1. EL BUCLE DE CONTROL INTERNO Y EXTERNO	40
FIGURA 3.2. ESCENARIOS DE DESCARGA DE SEGMENTOS EN <i>DASH</i> [Varma15]	43
FIGURA 4.1. GRÁFICA DE CONTROL EWMA	75
FIGURA 4.2. ALGORITMO DE ESTIMACIÓN DEL THROUGHPUT AFF	85
FIGURA 4.3. AFF CONTRA OTROS MÉTODOS DE ESTIMACIÓN	87
FIGURA 5.1. DIAGRAMA DE BLOQUES DEL MECANISMO DE ADAPTACIÓN DE <i>dash.js</i>	92
FIGURA 5.2. PLATAFORMA VIRTUALIZADA PARA DIFUSIÓN	93
FIGURA 5.3. PLATAFORMA VIRTUALIZADA PARA DISTRIBUCIÓN	95
FIGURA 5.4. PLATAFORMA VIRTUALIZADA PARA DIFUSIÓN CLIENTE-SERVIDOR	97
FIGURA 5.5. PERFILES DE TRÁFICO DE UNA RED <i>LTE</i> REAL.	99
FIGURA 5.6. PERFILES DE TRÁFICO ARTIFICIALES.	100
FIGURA 5.7. ESCENARIO REAL PARA EVALUAR LA EQUIDAD.	101
FIGURA 6.1. THROUGHPUT ESTIMADO DE LOS MÉTODOS Y LAMBDA EN EL ESCENARIO 1.	106
FIGURA 6.2. (A) CDF DEL BITRATE Y (B) CDF DEL NIVEL DE BUFFER EN EL ESCENARIO 1.	106
FIGURA 6.3. THROUGHPUT ESTIMADO DE LOS MÉTODOS Y LAMBDA EN EL ESCENARIO 2.	107
FIGURA 6.4. (A) CDF DEL BITRATE Y (B) CDF DEL NIVEL DE BUFFER EN EL ESCENARIO 2.	108
FIGURA 6.5. THROUGHPUT ESTIMADO DE LOS MÉTODOS Y LAMBDA EN EL ESCENARIO 3.	109
FIGURA 6.6. (A) CDF DEL BITRATE Y (B) CDF DEL NIVEL DE BUFFER EN EL ESCENARIO 3.	110
FIGURA 6.7. THROUGHPUT ESTIMADO DE LOS MÉTODOS Y LAMBDA EN EL ESCENARIO 4.	111
FIGURA 6.8. (A) CDF DEL BITRATE Y (B) CDF DEL NIVEL DE BUFFER EN EL ESCENARIO 4.	112
FIGURA 6.9. DIAGRAMA DE CAJA PARA EL ANÁLISIS DE LA EQUIDAD DE LOS TRES MÉTODOS.	116

ÍNDICE DE TABLAS

TABLA 2.1. RESUMEN ESTÁNDARES <i>DASH</i> [SEUFERT15]	18
TABLA 2.2. CONTENIDO DE VÍDEO SOPORTADO EN <i>DASH</i> [DASH-IF18A]	22
TABLA 2.3. CONTENIDO DE AUDIO SOPORTADO EN <i>DASH</i> [DASH-IF18A]	22
TABLA 3.1. TAMAÑO DEL SEGMENTO Y ABR [SANI17]	44
TABLA 3.2. ALGORITMOS <i>DASH</i> Y SUS CARACTERÍSTICAS I.	63
TABLA 3.3. ALGORITMOS <i>DASH</i> Y SUS CARACTERÍSTICAS II	64
TABLA 3.4. EFECTO DE LOS PARÁMETROS EN LA PERCEPCIÓN [SEUFERT15].	69
TABLA 4.1. NOMENCLATURA DEL ALGORITMO <i>AFF</i>	76
TABLA 5.1. CARACTERÍSTICAS DEL VÍDEO CODIFICADO PARA <i>DASH</i>	91
TABLA 5.2. ATENUACIÓN DE LAS ANTENAS PARA PILOTO DE RED <i>LTE</i>	98
TABLA 5.3. ESTADÍSTICAS DE LOS PERFILES DE TRÁFICO DE REALES EN MBPS	98
TABLA 5.4. ESTADÍSTICAS DE LOS PERFILES DE TRÁFICO ARTIFICIALES EN MBPS	101
TABLA 6.1. RESULTADO DE LAS MÉTRICAS DE <i>QoE</i> DE LOS ALGORITMOS DE ESTIMACIÓN	113
TABLA 6.2. RESULTADOS DEL ANÁLISIS DE LA EQUIDAD.	115

CAPÍTULO 1

INTRODUCCIÓN Y OBJETIVOS

1.1. Introducción y Motivación

Actualmente, el tráfico de Internet se compone principalmente de datos de entretenimiento en tiempo real (audio y vídeo). Un informe reciente del uso del Internet [Sandvine18] muestra que los datos de entretenimiento en tiempo real consumen el 56 % del tráfico total de Internet. De acuerdo con [Cisco18], se espera que el vídeo en tiempo real alcance el 82 % (UHD 20.7 %, HD 57.1 % y SD 22.2 %) del tráfico total de Internet para el 2022. En la figura 1.1 se puede observar que para el 2022 el vídeo bajo demanda y en vivo será el 82 % del tráfico total de Internet.

Figura 1.1: Tráfico de entretenimiento en tiempo real VNI [Cisco18] de 2017 a 2022.

La explosión del contenido multimedia ha impulsado tanto a la comunidad científica

como a la industria privada a crear protocolos y arquitecturas para ofrecer servicios multimedia de calidad para todos los usuarios bajo la premisa de poder consumir el contenido multimedia con calidad en cualquier lugar, en cualquier momento y de cualquier manera [Negru04].

El estándar *DASH* se ha erigido como un nuevo protocolo con el que se puede ofrecer el servicio de distribución multimedia por Internet. Este estándar ha ganado notoriedad debido a las ventajas que ofrece con respecto a otras soluciones, ya que funciona utilizando el protocolo *Hypertext Transfer Protocol (HTTP)*. La principal ventaja de utilizar el protocolo *HTTP* es la de poder atravesar por la infraestructura de telecomunicaciones sin problemas, ya que desde el punto de vista de la seguridad informática el tráfico *HTTP* no genera bloqueos en equipos de detección, prevención de intrusos y cortafuegos.

La *International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)* ha definido el uso del *DASH* [ISO14b] como el estándar para la entrega continua de contenido multimedia en el Internet. El 3rd Generation Partnership Project (3GPP) ha definido el uso de *DASH* [3GPP18a] como el estándar para la entrega continua de contenido multimedia en redes móviles, específicamente en redes móviles 4G como la red *LTE*.

El estándar *DASH* define cómo se puede dividir el contenido multimedia en pequeños archivos o “segmentos”, y cómo almacenar la descripción del segmento en un archivo de metadatos denominado *Media Presentation Description (MPD)* para que un reproductor multimedia *DASH* pueda recuperar tanto los metadatos como la secuencia de segmentos utilizando el protocolo *HTTP* para el intercambio de datos.

Una de las ventajas del uso de *DASH* es que define diferentes representaciones, lo cual hace posible que el mismo contenido se pueda recuperar con diferentes calidades, dependiendo del terminal de usuario o del ancho de banda disponible. Para cada representación, el archivo *MPD* contiene información como: el tipo de medio, el códec, el ancho y la altura del vídeo, la velocidad de codificación del vídeo o la tasa de bits promedio (bitrate). Los archivos de contenido multimedia están identificados por un *Uniform Resource Locator (URL)*.

Otra de las ventajas de usar *HTTP* como medio de transporte para el servicio de distribución de contenido multimedia es que los servidores *HTTP* estándar y las técnicas de distribución de contenido se pueden reutilizar para almacenar y entregar el contenido multimedia.

Un reproductor *DASH* funciona implementando un algoritmo de adaptación, el cual

analiza las condiciones de la red (ancho de banda, retardo, etc.) y selecciona la representación adecuada para su descarga. El principal objetivo del algoritmo de adaptación es garantizar la reproducción continua del contenido multimedia. El objetivo mencionado anteriormente se logra mediante la selección de la representación con la calidad más adecuada, evitando al mismo tiempo los eventos de parada durante la reproducción del contenido multimedia.

Sin embargo, el estándar *DASH* [ISO14b] no se centra en los procedimientos del cliente o del servidor, sino en los formatos de datos utilizados para proporcionar una reproducción de medios *DASH*, es decir, los algoritmos de adaptación para *DASH* no están estandarizados. Solo se puede encontrar una guía con comentarios sobre el posible comportamiento del cliente en el documento *ISO/IEC* en [ISO15b].

El estado del arte muestra que existen diversas propuestas de algoritmos adaptativos para *DASH*. Se observa que muchos de los algoritmos se han diseñado para escenarios específicos, por ejemplo, redes inalámbricas móviles, en los que se establecen parámetros específicos a fin de lograr la mejor calidad permitida por las condiciones disponibles que ofrece el escenario de pruebas implementado (emulado, simulado o real). Sin embargo, el mismo algoritmo de adaptación en un escenario diferente puede sobreestimar o subestimar el ancho de banda.

De esta manera, los problemas que presenta el algoritmo de adaptación para el estándar *DASH* y que afectan de manera negativa la calidad percibida por el usuario se resumen en los siguientes tres casos que pueden suceder al reproducir el contenido:

1. Cuando un algoritmo de adaptación está estimando más ancho de banda de lo que ofrece la red, se puede producir un evento de parada (stalling event) en la reproducción del contenido multimedia debido a que se recuperan segmentos que requieren un mayor ancho de banda que el ofrecido por la red, pudiendo ocasionar el agotamiento de los datos en la memoria del reproductor (buffer). Un efecto no deseado de un evento de parada es la aparición de un evento de rellenado del buffer (rebuffering event), el cual provoca que la reproducción tarde en empezar de nuevo, ya que se espera a acumular segmentos hasta un nivel deseable para garantizar la reproducción continua.
2. Por otro lado, si un algoritmo de adaptación subestima el ancho de banda de la red, el reproductor de vídeo recupera calidades de los segmentos multimedia que son inferiores a las permitidas por el ancho de banda ofrecido por la red y, por lo tanto, la calidad percibida por el usuario se verá afectada de manera negativa.

3. Otro problema que se puede presentar es que el reproductor esté cambiando constantemente entre las distintas calidades del contenido multimedia (bitrate switching) debido a fluctuaciones persistentes en el ancho de banda, efecto que sucede frecuentemente en redes móviles, lo cual impacta negativamente en la calidad percibida por el usuario.

1.2. Ámbito de la Tesis Doctoral

El ámbito de la Tesis Doctoral es la mejora de la calidad percibida por los usuarios mediante la propuesta de soluciones para los servicios de distribución multimedia en redes móviles. De forma más concreta se propondrá un nuevo algoritmo de adaptación dinámica para *DASH* con el fin de mejorar la calidad en la experiencia del usuario en los reproductores *DASH* y que pueda ser implementado en distintos escenarios de distribución sin la necesidad de realizar cambios para cada tipo de red (fija, inalámbrica o móvil).

La evaluación de los servicios de distribución multimedia sobre redes móviles comprende muchas áreas de trabajo, ya que es necesario considerar todos los factores que afectan al servicio. Existen factores externos e internos [Sani17] que influyen en la continua reproducción del contenido multimedia utilizando el estándar *DASH*.

Los factores internos que afectan a un servicio de distribución multimedia utilizando el estándar *DASH* son: el algoritmo de estimación de los recursos, el algoritmo de planificación de nuevos segmentos y el algoritmo de adaptación dinámica. El algoritmo de estimación de recursos normalmente toma en cuenta cómo se realiza la estimación del throughput y cómo se controla el buffer. El algoritmo de planificación de nuevos segmentos establece si se realizan peticiones de nuevos segmentos en paralelo o de forma secuencial. El algoritmo de adaptación dinámica realiza la decisión de la adaptación con teoría de control, de manera heurística, con un algoritmo de optimización o utilizando inteligencia artificial.

Los factores externos, tienen que ver con factores que afectan al servicio pero que no afectan directamente a los algoritmos mencionados anteriormente. En este caso, los factores externos serían: la naturaleza del contenido multimedia, el contexto del servicio y los mecanismos de operación de la “*Content Delivery Network (CDN)*”. La naturaleza del contenido multimedia se refiere a cómo la preparación del contenido multimedia tiene un impacto en la calidad del servicio, es decir, el codificador usado, la duración en segundos de los segmentos, las tasas de las distintas representaciones, etc. El contexto del servicio implica lo referente a la información del nivel del sistema donde se desean desplegar los

servicios de contenido multimedia. Los mecanismos de operación de la *CDN* describen cómo el uso de estas tecnologías incrementa la disponibilidad y la escalabilidad, y además, reduce la latencia en el acceso al contenido multimedia.

El enfoque de la tesis se centrará en los aspectos que están más relacionados con los factores internos, es decir, los factores que afectan al algoritmo de tasa adaptativa o “*Adaptive Bitrate (ABR)*”, específicamente en los algoritmos de estimación de los recursos. También se considerarán los factores externos, sobre todo en lo que se refiera a los distintos métodos de entrega en la distribución de los servicios multimedia a los usuarios, como pueden ser, la transmisión del servicio multimedia (unicast), la difusión del servicio multimedia (broadcast) y mecanismos de traspaso de un servicio entre diferentes tipos de redes de acceso (handover) inalámbricas.

1.3. Objetivos de la Tesis Doctoral

El objetivo general de la Tesis Doctoral es mejorar la calidad percibida por los usuarios mediante la propuesta de soluciones para los servicios de distribución multimedia en redes móviles. En la búsqueda de estas soluciones se considera como condición fundamental la obtención de información verificable que sea útil y relevante para la investigación, con el fin de alcanzar el objetivo general propuesto.

Para alcanzar el objetivo general de la Tesis, se han planteado los siguientes objetivos específicos:

1. Proponer un algoritmo de tasa adaptativa que mejore la estimación del throughput para incrementar la calidad de la experiencia de los usuarios del servicio de distribución multimedia y que pueda ser implementado en distintos escenarios de distribución.
2. Validar el algoritmo propuesto para analizar su factibilidad mediante una plataforma en la que se emule el servicio de distribución multimedia utilizando diferentes perfiles de tráfico de una red móvil.
3. Difundir los resultados obtenidos mediante su publicación en congresos y revistas de impacto.

1.4. Metodología de Investigación

Para dar cumplimiento a los objetivos establecidos en la sección anterior, es necesario establecer una metodología de investigación, en la cual se plasmen los pasos necesarios a

realizar para validar científicamente los resultados que se obtendrán de esta investigación.

La Tesis Doctoral seguirá el método de investigación científico analítico. Para ello, se desglosa en secciones independientes para su estudio cada una de las partes que conforman los servicios de distribución multimedia en redes móviles. Esto permite analizar con profundidad cada elemento por separado y de esta manera conocer las interacciones de los servicios de distribución multimedia en redes móviles.

Para lograr el objetivo general plasmado en la sección 1.3 de este capítulo, se han planteado los siguientes pasos con los cuales se desglosa el servicio de distribución multimedia:

- a) Como primer paso es necesario realizar el estudio del estado del arte que permita analizar la situación actual de los servicios de distribución multimedia, para obtener los conocimientos necesarios y poder identificar las limitaciones y/o problemas del servicio.
- b) Después es primordial analizar los algoritmos de tasa adaptativa para identificar las principales propuestas de estos algoritmos y, con esto, determinar áreas de oportunidad para proponer soluciones que mejoren la calidad en los servicios de distribución multimedia.
- c) Posteriormente, una vez cumplimentado el primer objetivo específico definido en la sección anterior, se requiere seleccionar y/o crear una plataforma en la que se pueda replicar y/o emular el servicio de distribución multimedia sobre redes móviles usando diferentes escenarios de la red móvil (perfiles de tráfico). Esto, permitirá comparar el algoritmo propuesto con el resultado del análisis de los algoritmos propuestos en el punto anterior.
- d) Como último paso es necesario concluir si el algoritmo propuesto puede coadyuvar a lograr el segundo objetivo específico propuesto en la sección anterior mediante el análisis de los resultados obtenidos utilizando la plataforma descrita en el punto anterior y los distintos perfiles de tráfico de redes móviles.

En la figura 1.2 se muestra el esquema general que resume el trabajo realizado por esta tesis doctoral, el cual se basa por una parte en el objetivo general y los objetivos específicos del documento como con el desglose de los elementos planteados en esta sección.

Figura 1.2: Esquema general que resume esta tesis doctoral

1.5. Estructura de la Memoria

La investigación realizada en la presente tesis doctoral es del tipo documental y experimental. En lo que respecta a la parte documental de la investigación, la metodología utilizada consistió en realizar una revisión exhaustiva del estado del arte desde el punto de vista teórico y de propuestas, con el objetivo de organizar sistemáticamente el conocimiento adquirido e identificar los problemas existentes en los servicios de distribución multimedia, así como en las propuestas que tratan de resolverlos. Este proceso permitió definir un conjunto de objetivos específicos para abordar los problemas identificados y que representan las contribuciones de la Tesis Doctoral. En el proceso de definición de objetivos se ha considerado la propuesta de un algoritmo de tasa adaptativa para mejorar la calidad que experimentan los usuarios del servicio.

En lo que respecta a la parte experimental de la investigación, se ha implementado una plataforma virtualizada que sirve para evaluar los servicios de distribución multimedia para validar las propuestas del algoritmo de tasa adaptativa presentado en la Tesis Doctoral. Para desarrollar la prueba de concepto se han implementado diferentes tipos de entrega del contenido multimedia. En el escenario de validación se han realizado diferentes experimentos con el fin de observar el comportamiento del algoritmo de tasa adaptativa propuesto y con esto obtener resultados que demuestren la factibilidad de la propuesta.

La memoria de esta Tesis Doctoral se ha estructurado de la siguiente manera:

- **Capítulo 2**

En este capítulo se establecen los conceptos básicos sobre los servicios de distribución multimedia en los que se basa la tesis, haciendo especial énfasis en los conceptos que han servido de partida para el desarrollo de este trabajo de investigación.

- **Capítulo 3**

En este capítulo se presentan y discuten las propuestas y trabajos sobre los algoritmos de adaptación para el estándar *DASH*. Este capítulo incluye los conceptos básicos del “Adaptive Bitrate Streaming (ABR)” y las propuestas de algoritmos de adaptación comúnmente utilizados en los reproductores *DASH*. También se incluye un estudio de diferentes propuestas de modelos de métricas de *QoE*.

- **Capítulo 4**

En este capítulo se propone un nuevo método para la estimación del throughput en los reproductores de los servicios de distribución multimedia. El método se llama “Adaptive Forgetting Factor (AFF)” y funciona estimando de forma adaptativa el rendimiento de la transmisión del vídeo, y al encontrar fluctuaciones a corto plazo, se colocan mayores pesos en las observaciones más recientes (suavizado exponencial), lo cual resulta en un método que “olvida” las mediciones antiguas rápidamente. También se compara el método *AFF* con otros métodos comúnmente utilizados en los algoritmos de tasa adaptativa. Por último, se muestra la implementación del método *AFF* como pseudo-código para su posterior adaptación en un reproductor *DASH*.

- **Capítulo 5**

En este capítulo se explica la implementación de una plataforma virtualizada para evaluar el algoritmo descrito en el capítulo 4. En este capítulo se describen todos los elementos necesarios para evaluar los servicios de distribución multimedia. La plataforma considera distintos tipos en la entrega de los servicios de distribución multimedia.

- **Capítulo 6**

En este capítulo se presentan los resultados de la evaluación del algoritmo descrito en el capítulo 4, en donde se compara el algoritmo *AFF* con otros algoritmos de estimación del throughput propuestos por la comunidad científica, con lo que se demuestra la viabilidad de la propuesta del algoritmo de adaptación basado en *AFF*.

- **Capítulo 7**

Por último, en este capítulo se presentan las conclusiones generales y particulares sobre el cumplimiento de los objetivos propuestos en el desarrollo de la tesis. Además, se presenta la divulgación de resultados realizada por medio de publicaciones en una revista con factor de impacto, en congresos internacionales y los resultados de la participación en proyectos de investigación españoles. En la parte final del capítulo se presentan las propuestas para trabajos futuros con los cuales se pueda seguir desarrollando el tema de investigación de esta tesis doctoral.

CAPÍTULO 2

SERVICIOS DE DISTRIBUCIÓN MULTIMEDIA

2.1. Introducción

En este capítulo se establecen los conceptos básicos sobre el servicio de distribución de contenido multimedia en el cual se basa la tesis, haciendo especial énfasis en los conceptos que han servido de partida para el desarrollo de este trabajo de investigación. El objetivo principal de este capítulo es el de detallar los diferentes componentes del servicio de distribución multimedia, especialmente los elementos que componen el estándar *DASH*, con el fin de identificar limitaciones y/o problemas del servicio y con esto mejorar la calidad de la experiencia de los usuarios del servicio de distribución multimedia.

En el apartado 2.2, se describe una breve historia del inicio del servicio de distribución de contenido multimedia y las tecnologías que permitieron el incremento en el uso del servicio mencionado anteriormente. También, el apartado 2.2 trata sobre las tecnologías de streaming adaptativo que precedieron a la creación del estándar *DASH*. En el apartado 2.3 se analizan las distintas partes que componen el estándar *DASH* el cual se define en el documento de la *ISO/IEC* [ISO14b]: el archivo que define el contenido, el tipo de contenido multimedia soportado por el estándar, el tipo de arquitectura que implementa, los elementos funcionales del cliente, las métricas que se definen para medir la calidad del servicio y las herramientas actuales, tanto libres como comerciales, para desplegar servicios *DASH*. En el apartado 2.4 se definen las distintas maneras en que se pueden desplegar los servicios *DASH*. Por último, en el apartado 2.5 se presentan las conclusiones del capítulo.

2.2. Historia de los servicios de transmisión multimedia

El 5 de septiembre de 1995, la compañía ESPN realizó la primera transmisión multimedia (audio) en vivo al retransmitir por Internet las narraciones del partido de béisbol entre los Yankees de Nueva York y los Marineros de Seattle utilizando el software RealAudio, el cual contenía tecnología punta desarrollada por la empresa Real Networks (antes Progressive Networks).

La empresa RealNetworks junto a Netscape y la Universidad de Columbia propusieron como borrador en 1996 un nuevo estándar para la transmisión multimedia en tiempo real llamado *Real Time Streaming Protocol (RTSP)* [Schulzrinne98]. Este nuevo protocolo fue publicado por la *Internet Engineering Task Force (IETF)* en el documento RFC 2326 en 1998, dentro del grupo Multiparty Multimedia Session Control Working Group (MMUSIC WG).

Sin embargo, el problema al que se enfrentaban las empresas que ofrecían servicios de distribución multimedia fue el que se pudiera transmitir exitosamente vídeo sobre líneas de modem con capacidad de 56kbps. Se estima que a principios del año 2000 existían 150 millones de conexiones por modem en los países pertenecientes a la Organización para la Cooperación y el Desarrollo Económicos (OCDE).

La empresa Microsoft ganó la batalla con la solución Windows Media Framework, la cual especificaba el software y protocolos propietarios para ofrecer un servicio de transmisión multimedia. El protocolo propietario de Microsoft para la transmisión de contenido multimedia se llamó *Microsoft Media Server (MMS)*. En 2003, con el lanzamiento de la versión 9 de Windows Media Services (WMS) la compañía decidió dejar de utilizar *MMS* e incorporar protocolos estandarizados como *RTSP*.

A mediados de la década del 2000 Macromedia (posteriormente adquirida por Adobe Systems en 2005) conquistó la cuota de mercado utilizando su reproductor Flash Player. Esto se debió a que el servicio que ofrecía Flash combinaba la interactividad, la Web 2.0 y los contenidos multimedia. Sin embargo, a pesar de esta nueva tecnología, seguían apareciendo problemas con el ancho de banda, la escalabilidad y la extensión de los servicios de distribución multimedia. La tecnología del Flash Player se basaba en el uso de *Small Web Format (SWF)*, que es un formato de archivo para multimedia que utiliza gráficos vectoriales y un lenguaje llamado ActionScript el cual se generaba utilizando el software de Adobe Shockwave. Flash Player en su versión 6 del 2002 ya soportaba la inclusión de vídeo dentro del *SWF*.

Para finales de la década del 2000, en los países de la OCDE, más del 90 % de las suscripciones de acceso a Internet usaban banda ancha, este tipo de acceso había aumentado a más de 300 millones de suscripciones, lo cual propició el consumo de servicios de distribución de contenidos multimedia.

Sin embargo, la verdadera revolución para los servicios de distribución de contenido multimedia surgió cuando se empezaron a utilizar redes de distribución de contenidos *CDN*, que permiten replicar los datos (contenido multimedia) en servidores que se encuentran ubicados al borde de la red, y así acercar esos datos a los usuarios, incrementando la accesibilidad al contenido.

El incremento de las suscripciones de banda ancha en los hogares y la habilitación de tecnologías como las *CDN* propició la creación de nuevas técnicas para la distribución de contenido multimedia utilizando el protocolo *HTTP*. La mayoría de estas técnicas se basan en la patente de Estados Unidos de América número 7818444 [Brueck10] creada por la empresa Move Networks (antes Echostar).

El estándar *DASH* fue creado entre el grupo de trabajo “Moving Picture Experts Group” de la *ISO/IEC* y el *3rd Generation Partnership Project (3GPP)*. *DASH* empezó a desarrollarse en enero de 2009 y fue finalizado en marzo de 2010, para después convertirse en una norma de la *ISO/IEC* bajo el estándar 23009-1 [ISO14b] en 2012. También tiene su definición dentro del *3GPP* como la especificación técnica 26 247 [3GPP18a].

En los siguientes apartados se profundiza sobre las tecnologías referentes al protocolo *RTSP* y las *CDN*.

2.2.1. Real Time Streaming Protocol (RTSP)

El protocolo *RTSP* fue creado con el propósito de establecer y controlar sesiones de transmisión multimedia hacia los clientes. El protocolo utiliza comandos como los que se usan en las grabadoras de vídeo (play, record, pause, etc.) para facilitar el control en tiempo real de la transmisión de medios desde el servidor al cliente.

El protocolo *RTSP* se transporta encapsulado en el protocolo *User Datagram Protocol (UDP)* en la capa de transporte, el protocolo *UDP* permite el envío de datos a través de la red sin que se haya establecido previamente una conexión, como se muestra en la figura 2.1. El RFC 2326 fue actualizado a la versión 2.0 en el 2016 por el RFC 7826 [Schulzrinne16].

El protocolo *RTSP* combina su funcionamiento principalmente con estos tres protocolos:

- *Real-Time Transport Protocol (RTP)* y *Real-Time Control Protocol (RTCP)*, especi-

Figura 2.1: Funcionamiento del protocolo RTSP

ficado por el RFC 3550 [Schulzrinne03].

- *Session Description Protocol (SDP)*, especificado por el RFC 4566 [Handley06].

El protocolo *RTP* es el encargado del transporte de contenido multimedia de extremo a extremo en tiempo real. El protocolo proporciona facilidades para compensar el jitter y la detección de datos fuera de secuencia. El protocolo *RTCP* es el encargado de proporcionar estadísticas de Calidad del Servicio (QoS), información de control y sincronización para una sesión *RTP*. El protocolo *SDP* describe sesiones de comunicación multimedia mediante el anuncio de la sesión, invitación a la sesión y negociación de los parámetros iniciales.

El problema de utilizar servicios de streaming de datos multimedia utilizando el protocolo *RTSP* es que cuando el tráfico sale del dominio de la red del cliente puede atravesar servicios de traducción de direcciones de red y cortafuegos que podrían impedir el correcto funcionamiento del servicio.

2.2.2. Red de Distribución de Contenidos (CDN)

Una red de distribución de contenidos o *CDN*, es una red virtual de nodos enlazados lógicamente que se encuentran estratégicamente distribuidos, con el fin de maximizar la disponibilidad y el rendimiento del servicio (transmisión de contenido multimedia, descargas de software, aceleración de contenido web, etc.) al estar cerca de los usuarios finales. Esto permite disminuir los tiempos de respuesta y las pérdidas en la entrega de la información al usuario final, así como obtener estadísticas del comportamiento de los usuarios basado en el registro de las páginas visitadas y la ubicación geográfica.

La primera generación de *CDN* apareció hacia finales de los 90, precedida por innovaciones tecnológicas como el almacenamiento en proxy cachés, almacenamiento en cachés

jerárquicos y las granjas de servidores. Esta primera generación de *CDN* fue diseñada para acelerar el acceso a sitios Web con contenido estático y dinámico.

La segunda generación de *CDN* apareció a mediados del año 2000 y fue creada para abordar la mayor demanda de almacenamiento de contenido multimedia y la transmisión de este contenido a los usuarios.

La tercera generación de *CDN*, se enfocó en ofrecer servicios de comunidad como cómputo distribuido en la nube, servicios autónomos y auto-manejados para ofrecer *QoE* al usuario final.

Los objetivos de un servicio *CDN* se enfocan en tres aspectos esenciales que son:

- Escalabilidad, que se refiere a la capacidad del sistema de expandirse para manejar cantidades nuevas y grandes de datos, usuarios y transacciones sin una disminución significativa en el rendimiento.
- Seguridad, que se refiere la protección del contenido contra el acceso y la modificación no autorizados del contenido confidencial y de alto valor del usuario.
- Fiabilidad, que se refiere a la disponibilidad del servicio y cuáles son los límites en las interrupciones del servicio que se pueden esperar.

Figura 2.2: Elementos que componen una CDN [Buyya08]

Como se muestra en la figura 2.2 una *CDN* está compuesta por cuatro elementos, los cuales son:

- El sistema de entrega de contenido, que consiste en un servidor origen y un grupo de servidores réplica, los cuales entregan la información al usuario final.
- El sistema de enrutado, el cual se encarga de dirigir la petición del cliente al servidor réplica apropiado, así como llevar un registro detallado de los contenidos que se encuentran en los servidores réplica.
- El sistema de distribución, el cual se encarga de mover el contenido apropiado desde el servidor origen a los servidores réplica.
- El sistema de administración, que mantiene un registro del acceso de los clientes y el uso de los servidores de la *CDN* para poder facturar en base al uso del servicio de *CDN*, ya sea a una organización o a un tercero.

2.2.3. Progressive download streaming

A mediados de la década del 2000 la mayoría del tráfico de Internet era *HTTP* y las *CDN* se utilizaban para entregar contenido multimedia a grandes audiencias. Por otro lado, los servicios de difusión multimedia que utilizaban el protocolo *RTSP* basado en conexiones *UDP* se encontraron con problemas al no poder satisfacer la demanda de los usuarios.

Para estas fechas, los investigadores aceptaron que el protocolo “*Transmission Control Protocol (TCP)*” [Wang08] ofrecía beneficios para transmisiones de vídeo tolerantes al retardo. Se introdujo un buffer de reproducción en la capa de aplicación para compensar las fluctuaciones de velocidad de *TCP* y las congestiones. A esta versión para la difusión de vídeo multimedia utilizando el protocolo *HTTP* sobre *TCP* se le llamó Progressive Download [Apple02].

En progressive download el cliente descarga el fichero de vídeo entero tan rápido como *TCP* lo permite. El reproductor de vídeo comienza la reproducción del vídeo antes de que la descarga se complete. La diferencia entre los servicios de distribución de contenido multimedia basados en *RTSP* y la descarga progresiva radica en la forma en que los datos con el contenido multimedia son recibidos y almacenados por el dispositivo del usuario final que está accediendo a los medios digitales. En la mayoría de los servicios basados en *RTSP* se utiliza el protocolo *UDP* para la entrega de los datos de audio y vídeo y el protocolo *TCP* para los datos de control del reproductor, mientras que los servicios de descarga progresiva utilizan el protocolo *TCP*.

El principal inconveniente de la técnica de descarga progresiva es que no es posible avanzar rápidamente a través de ninguna parte del contenido multimedia hasta que este se ha almacenado por completo todo el contenido anterior en el dispositivo del usuario.

La reproducción multimedia que utiliza descarga progresiva depende de un archivo de metadatos que se encuentra en el servidor y de un mecanismo de almacenamiento temporal (buffer) que está implementado en el cliente. Cuando una cantidad específica de datos se encuentra almacenada en el buffer el cliente puede empezar la reproducción del contenido. La cantidad específica de datos para empezar la reproducción se encuentra definida en el archivo de metadatos.

2.2.4. HTTP Adaptive Bitrate Streaming

En el año 2007 la empresa Move Networks introdujo una tecnología que cambiaría la industria multimedia que se llamó transmisión de vídeo adaptativo basado en *HTTP* (HTTP-based adaptive bitrate streaming).

El 19 de octubre del 2010 se aprobó la patente US7818444 con el nombre "Apparatus, system, and method for multi-bitrate content streaming" [Brueck10] a la empresa Move Networks. Esta tecnología basó su uso en el protocolo dominante *HTTP* para entregar pequeños trozos de archivos multimedia mientras la aplicación del cliente supervisaba las velocidades de descarga con el fin de solicitar trozos de tamaño variable en respuesta a las condiciones cambiantes de la red para adaptar la calidad.

Esta tecnología tuvo un gran impacto ya que permitió que los servicios multimedia se distribuyeran utilizando la tecnología de las *CDN* y aprovechar el almacenamiento en caché mejorara la eficiencia del servicio de distribución de contenido multimedia.

Pronto otras soluciones para ofrecer servicios de transmisión de vídeo adaptativo basado en *HTTP* surgieron de compañías punteras, como fueron:

- **Microsoft Smooth Streaming (MSS) [Zambelli09] (2008):** Smooth Streaming es una extensión de los servicios Internet Information Services (IIS) con el cual Microsoft ofrece un servicio de transmisión de vídeo adaptativo basado en *HTTP*. Smooth Streaming utiliza un formato basado en "ISO base media file format" el cual fue estandarizado como Formato de Archivo Protegido e Interoperable.
- **Apple HTTP Live Streaming (HLS) [Pantos17] (2009):** *HTTP* Live Streaming (HLS) es un protocolo de comunicaciones para la transmisión de vídeo adaptativo basado en *HTTP* implementado por la compañía Apple como parte del software

QuickTime X y el sistema operativo de los teléfonos de apple iOS. HLS soporta vídeo en vivo, así como vídeo bajo demanda.

- **Adobe HTTP Dynamic Streaming (HDS) [Borg13] (2010):** *HTTP* Dynamic streaming es la solución para proveer servicios de transmisión de vídeo adaptativo basado en *HTTP* de la compañía Adobe. La solución se basa en usar *Open Source Media Framework (OSMF)* [Hamer12] para la entrega de este contenido, así como en la utilización del software Flash Player 10.

Pronto los servicios de transmisión de vídeo adaptativo basado en *HTTP* se convirtieron en la elección favorita para eventos de transmisión en vivo de alto perfil (Olimpiadas de Vancouver y Londres, los torneos de tenis Wimbledon y Roland Garros, etc.) y también la tecnología favorita para servicios de pago para vídeo bajo demanda (Netflix, Hulu, Amazon Instant Video, etc.).

Sin embargo, al no existir un estándar oficial para servicios de transmisión de vídeo adaptativo basado en *HTTP*, la industria se unió y a mediados de la década del 2000 se creó el estándar *DASH*. En la siguiente sección se explica este estándar.

2.3. Dynamic Adaptive Streaming over HTTP (DASH)

El estándar *DASH* fue creado entre el grupo de trabajo “Moving Picture Experts Group” de la *ISO/IEC* y el *3GPP*. *DASH* empezó a desarrollarse en enero de 2009 y se completó en marzo de 2010.

En abril del 2012, la *ISO/IEC* ratificó la versión de la nueva generación del estándar para la transmisión de vídeo digital adaptativo (adaptive video streaming). Este nuevo estándar se llama “MPEG-DASH” [ISO14b], *Moving Picture Experts Group (MPEG)-DASH* y está registrado como un estándar internacional bajo el nombre ISO/IEC 23009 Information technology – Dynamic adaptive streaming over HTTP (DASH).

DASH fue creado con el fin de contar con un único estándar que reemplazara las múltiples soluciones propietarias (Microsoft, Apple y Adobe), y que ofreciera la interoperabilidad y convergencia necesaria para el crecimiento a gran escala de soluciones de vídeo streaming. Entre las compañías que participaron en la creación de este estándar están: Microsoft, Apple, Netflix, Qualcomm, Ericsson y Samsung.

La asociación de grupos de telecomunicaciones conocida como 3rd Generation Partnership Project (3GPP) ha definido el uso de *DASH* en la especificación 26.247 [3GPP18a]

como el estándar para la difusión de vídeo digital en las redes móviles: 3G *Global System for Mobile Communications (GSM)* y 4G *LTE*.

Una de las ventajas de usar *DASH* es que el streaming de vídeo va encapsulado en el protocolo *HTTP* versión 1.1, también llamado *HTTP/1.1*. El uso de este protocolo permite la reutilización de la infraestructura actual de Internet (clientes y servidores web) y también, la reutilización de las técnicas de distribución del contenido multimedia utilizando *CDN*. Otra ventaja que presenta el uso de *DASH* es que este tipo de contenido encapsulado en *HTTP* no presenta problemas al pasar por cortafuegos (firewall) o con el protocolo *Network Address Translation (NAT)*, piedra angular de los despliegues de servicio de Internet residenciales (*Asymmetric digital subscriber line (ADSL)*, fibra óptica, etc.).

La norma ISO/IEC 23009-1:2014 [ISO14b] especifica los formatos para la Descripción y los Segmentos de Presentación de Medios para la transmisión dinámica adaptable de medios MPEG a través de *HTTP*. Es aplicable a los servicios de transmisión a través de Internet. En la norma ISO/IEC 23009-1:2014 se definen dos elementos indispensables para ofrecer un servicio de distribución de contenido multimedia. Estos elementos son:

- La especificación del formato que debe tener el archivo de presentación de los contenidos multimedia o *MPD* utilizando el lenguaje *eXtensible Markup Language (XML)*, el cual define las características del contenido multimedia y la localización de este dentro del servidor *HTTP* utilizando identificadores basados en *URL*.
- El tipo de contenido multimedia, es decir las características de codificación necesarias para que este contenido multimedia funcione en reproductores estándar, así como la manera en que este contenido multimedia se divide en pequeños archivos (segmentos) para facilitar el transporte por redes de datos.

Tabla 2.1: Resumen estándares *DASH* [Seufert15]

Solución Estándar	ISO-DASH	3GP-DASH
Tipo	Estándar	Estándar
Descriptor de los datos	MPD (XML)	MPD (XML)
Códec de vídeo	Cualquiera	H.264
Códec de audio	Cualquiera	AAC
Formato	ISOBMFF	Formato de archivo 3GPP
Tamaño del segmento	No especificado	No especificado

En la tabla 2.1 se muestran las diferencias entre el estándar de la *ISO/IEC* [ISO14b] y el estándar del 3GPP [3GPP18a] para la distribución de contenido multimedia utilizando *DASH*.

La norma ISO/IEC 23009-2:2017 [ISO17b] especifica el software de conformidad y referencia que implementa las cláusulas normativas de ISO / IEC 23009-1, es decir, vectores de prueba que comprenden descripciones de presentaciones de medios, segmentos y combinaciones de estos, los cuales se ajustan o no a las cláusulas normativas de ISO / IEC 23009-1 y los correspondientes módulos de software.

La norma ISO/IEC TR 23009-3:2015 [ISO15b] proporciona directrices técnicas para implementar y desplegar sistemas basados en ISO/IEC 23009-1.

ISO/IEC 23009-4:2013 [ISO13b] especifica los mecanismos de encriptación y señalización independientes del formato para su uso con cualquier formato de segmento de medios utilizado en *DASH*. También especifica los mecanismos que garantizan la integridad y autenticidad de los segmentos utilizados en *DASH*.

La norma ISO/IEC 23009-5:2017 [ISO17c] define la arquitectura funcional Server and Network Assisted *DASH* (SAND), SAND proporciona información sobre las características operativas en tiempo real de redes, servidores, proxies, cachés y CDN, como también, el rendimiento y estado del cliente *DASH*. La arquitectura identifica los elementos de red SAND, el protocolo de comunicación SAND, la semántica de los mensajes SAND, los esquemas de codificación de los mensajes SAND y la naturaleza de los intercambios de mensajes SAND.

La norma ISO/IEC 23009-6:2017 [ISO17d] especifica el transporte de presentaciones de medios MPEG-*DASH* sobre protocolos full duplex compatibles con *HTTP*, particularmente *HTTP/2* y *WebSocket*. Esta norma aprovecha las características que estos protocolos soportan a través de *HTTP/1.1* para mejorar el rendimiento en la entrega, particularmente para distribuir vídeo en vivo con baja latencia, al tiempo que mantiene la compatibilidad con versiones anteriores.

2.3.1. MPD

El archivo *MPD* proporciona la suficiente información al cliente para proporcionar un servicio de transmisión de vídeo por medio del acceso a segmentos especificados en el estándar *HTTP/1.1*, como se muestra en la figura 2.3.

El archivo *MPD* se compone de Períodos. Dentro del documento se describe la secuencia de Períodos en el tiempo que componen la transmisión multimedia completa. Un Período normalmente representa una cantidad finita de contenido multimedia durante el cual está disponible un conjunto consistente de versiones codificadas del contenido multimedia disponible, es decir, el conjunto de tasas de bits, idiomas, subtítulos, etc.

Figura 2.3: DASH Modelo de datos de alto nivel [ISO14b]

Dentro de un Período, el contenido multimedia está organizado por Conjuntos de Adaptación. Un conjunto de adaptación representa un conjunto de versiones codificadas intercambiables de uno o varios componentes de contenido multimedia. Por ejemplo, puede haber un conjunto de adaptación para el componente de vídeo principal y otro para el componente de audio principal. Si hay otro material disponible, subtítulos o descripciones de audio, estos pueden tener un conjunto de adaptación por separado.

Un conjunto de adaptación contiene un conjunto de representaciones. Una Representación describe una versión codificada de componentes de contenido multimedia. Cualquier Representación única dentro de un Conjunto de Adaptación es suficiente para representar los componentes de contenido multimedia.

Esto significa normalmente que los clientes pueden cambiar dinámicamente de Representación a Representación dentro de un Conjunto de Adaptación para adaptarse a las condiciones de la red u otros factores. La conmutación se refiere a la presentación de datos descodificados de una representación hasta cierto tiempo t , y la presentación de datos descodificados de otra representación desde el tiempo t en adelante.

Dentro de una Representación, el contenido se puede dividir en Segmentos de tiempo para un correcto acceso y entrega. Para acceder a un Segmento, se proporciona una URL

para cada Segmento. En consecuencia, un Segmento, es la unidad de datos que se puede recuperar con una sola solicitud *HTTP*.

2.3.2. Contenido Multimedia soportado en DASH

La otra parte importante que se define en el estándar *DASH* [ISO14b] es el tipo de contenido multimedia que es compatible. La encapsulación del contenido multimedia se realiza de acuerdo con el estándar de la *ISO/IEC* llamado “ISO Base Media File Format” (ISOBMFF) [ISO15a]. En el estándar *DASH*, se utiliza el documento RFC 6381 llamado “The ‘Codecs’ and ‘Profiles’ Parameters for “Bucket” Media Types” [Gellens11] para definir los códecs aceptados de contenido multimedia.

En el documento que define el estándar *DASH* en el 3GPP [3GPP18a] se utiliza el documento TS 26.234 del 3GPP llamado “Protocols and codecs” [3GPP18b], en el cual se definen los diferentes tipos de códecs de audio y vídeo soportados. Otro documento que define los códecs de contenido multimedia que se pueden utilizar en el estándar *DASH* es el documento creado por el foro DASH-IF en la guía llamada “Guidelines for Implementation: DASH-IF Interoperability Points” [DASH-IF18a]. En esta guía se define una descripción general de las tecnologías aceptadas para el contenido multimedia y cómo encajan en los aspectos relacionados con *DASH*.

2.3.2.1. Vídeo

Dentro del estándar *DASH*, se considera el códec H.264, utilizando “Advanced Video Coding” (AVC) [ISO14a] en el perfil “High” 3.1 y una resolución de 1280×720 a 30 FPS como la codificación básica que debería ser soportada por todos los reproductores *DASH*. Sin embargo, si se quisiera transmitir el contenido de vídeo en High Definition (HD) se tendría que utilizar el perfil “high” 4.0 para alcanzar una resolución de 2048×1024 a 30 FPS. Con el fin de soportar los nuevos contenidos a una resolución Ultra HD (UHD) o 4K es necesario utilizar el códec H.265 “High Efficiency Video Coding” (HEVC) [ISO17a] en el perfil “Main-10” 5.1 a una resolución de 4096×2160 a 60 FPS.

El códec H.264 utiliza 8 bits por muestra, lo cual se traduce a 256 tonos por color primario o 16 millones de colores. Dentro de cada perfil el máximo bitrate soportado es de: 10 Mbps para el perfil 3.0, 14 Mbps para el perfil 3.1 y 20 Mbps para el perfil 4.0.

El códec H.265 en el perfil “Main-10” soporta 10 bits por muestra lo que se traduce a 1024 tonos de color primario o 1 billón de colores. Dentro de cada perfil, el máximo bitrate

soportado es de: 10 Mbps para el perfil 3.1, 50 Mbps para el perfil 4.1 y 160 Mbps para el perfil 5.1.

En la tabla 2.2 se muestra las configuraciones del códec de vídeo que soportan los reproductores *DASH* en la actualidad. Con estas codificaciones se puede alcanzar las velocidades en el bitrate de vídeo más altas. También se menciona en el estándar que todos los perfiles anteriores tanto para el códec H.264 y H.265 están soportados.

Tabla 2.2: Contenido de Vídeo soportado en *DASH* [DASH-IF18a]

Codec	Profile	Level	Parameter	Resolution
H.264	AVC Progressive Main Profile	3.0	avc1.4DY01E	720×480@30 FPS (SD)
	AVC Progressive High Profile	3.1	avc1.64Y01F	1,280×720@30 FPS (HD)
		4.0	avc1.64Y028	2,048×1024@30 FPS (2K)
H.265	HEVC Main	3.1	hev1.1.2.L93.B0 hvc1.1.2.L93.B0	1280×720@33.7 FPS (HD)
		4.1	hev1.1.2.L123.B0 hvc1.12.L123.B0	2048×1080@60 FPS (2K)
	HEVC Main-10	4.1	hev1.1.2.L93.B0 hvc1.1.2.L93.B0	2048×1080@60 FPS (2K)
	HEVC Main-10	5.1	hev1.2.4.L153.B0 hvc1.2.4.L153.B0	3840×2160@60 FPS (4K)

2.3.2.2. Audio

Dentro del estándar *DASH*, el contenido de audio debe ser codificado utilizando el códec “Advanced Audio Coding” (AAC) [ISO13a] en estéreo a una velocidad de 128 Kbps. Se puede utilizar cualquier versión del códec AAC, es decir High Efficiency (HE) AAC y HE-AAC v2.

Las ventajas que presenta utilizar el códec AAC es que se muestrean más frecuencias que con el estándar MP3 (de 8 a 96 kHz en AAC por 16 a 48 kHz en MP3), soporta hasta 48 canales de audio y un mejor manejo de las frecuencias por encima de 16 kHz.

En la tabla 2.3 se muestran las configuraciones del códec de audio que soportan los reproductores *DASH* en la actualidad.

Tabla 2.3: Contenido de Audio soportado en *DASH* [DASH-IF18a]

Codec	Profile	Parameter	ISO BMFF Encapsulation
MPEG-4	AAC Profile	mp4a.40.2	ISO/IEC 14496-14
	HE-AAC Profile	mp4a.40.5	
	HE-AAC v2 Profile	mp4a.40.29	

2.3.3. Arquitectura DASH

El estándar *DASH* utiliza la arquitectura cliente-servidor [Migues89], que es la arquitectura de referencia del protocolo *HTTP*. El estándar *DASH* utiliza esta arquitectura ya que presenta las siguientes ventajas:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor.
- Escalabilidad: se puede mejorar la capacidad de clientes y servidores por separado.
- Fácil mantenimiento: la centralización facilita la tarea de poner al día datos u otros recursos, mientras que los clientes no se verán afectados.
- Seguridad: Un cliente defectuoso o no autorizado no puede dañar el sistema.

Figura 2.4: Arquitectura DASH de alto nivel [ISO14b]

En la figura 2.4 se muestran los diferentes elementos que componen la arquitectura *DASH*. Los elementos obligatorios de *DASH* aparecen en líneas continuas, los elementos que aparecen en líneas discontinuas son optativos. En la figura 2.4, también se muestra que el cliente es el único elemento necesario por el usuario para la reproducción de contenido multimedia. A continuación, se describen cada uno de los elementos que aparecen en la figura 2.4.

2.3.3.1. Servidor DASH

El servidor *DASH* se compone varios elementos, siendo el principal el elemento *DASH Segment Delivery Function*, el cual es un servidor estándar *HTTP* en el cual se encuentran accesibles el archivo de metadatos *MPD* y los segmentos de contenido multimedia (servicio

bajo demanda). En el caso de un servicio en vivo, también es necesario un elemento que actualice constantemente el archivo de metadatos MPD. En el servidor también se puede encontrar el software utilizado para la creación (codificador) del contenido multimedia, y se considera la implementación de caches de *HTTP* y el uso de *CDN*.

2.3.3.2. Cliente DASH

El cliente es el único elemento necesario por el usuario para la reproducción de contenido multimedia. El cliente *DASH* basado en el estándar 3GPP 26.247 [3GPP18a] se compone de dos elementos lógicos como se muestra en la figura 2.5, estos elementos lógicos son:

- Mecanismo de acceso *DASH*
- Mecanismo del contenido multimedia

El mecanismo de acceso *DASH* recibe el archivo de metadatos MPD, con el cual construye y emite solicitudes de los segmentos de contenido multimedia utilizando peticiones estándar del protocolo *HTTP*. El resultado del mecanismo de acceso *DASH* son los segmentos descodificados en el formato de contenedor de acuerdo con el estándar *ISO/IEC 14496-12* ISO Base Media File Format (ISOBMFF) [ISO15a] y la sincronización entre el tiempo interno del segmento de contenido multimedia y la línea de tiempo del reproductor. También es el encargado de la correcta interpretación de los eventos que pueden suceder en la reproducción del contenido multimedia, por ejemplo, la inserción de anuncios comerciales.

El mecanismo del contenido multimedia es el encargado de codificar o descodificar el flujo de datos digitales y de obtener el contenido multimedia para su posterior reproducción. Este mecanismo lee los segmentos de contenido multimedia en el formato ISOBMFF [ISO15a] junto con los eventos y la información del tiempo para posteriormente mostrar el contenido multimedia en la pantalla del reproductor.

Figura 2.5: Partes del cliente DASH [3GPP18a]

Como se menciona en el capítulo 1, el estándar *ISO/IEC 27009* [ISO14b] no especifica los procedimientos para la implementación del cliente. En el estándar *ISO/IEC 27009-3:2015* [ISO15b] se especifican las directrices técnicas para implementar y desplegar servicios *DASH* y en la especificación 26.247 del *3GPP* [3GPP18a] (Anexo A) se incluye la guía “Example DASH Client Behaviour” en donde se proporcionan ejemplos sobre lo que se espera en cuanto al comportamiento en un cliente estándar. Otro documento en el cual se incluyen requisitos y una guía para el estándar *DASH* es el documento de “interoperabilidad” del foro *DASH-IF* [DASH-IF18a].

2.3.3.3. Funcionamiento de la arquitectura DASH

Como se menciona en la sección 2.4.3, la arquitectura *DASH* se basa en una implementación “cliente-servidor”, en donde existe una conversación entre el cliente *DASH* y el servidor *DASH* que facilita el intercambio de información para la reproducción del contenido multimedia. A continuación, se describen los pasos necesarios para que exista un servicio de distribución multimedia utilizando el estándar *DASH*:

1. El cliente solicita al servidor *DASH* el archivo *MPD*, en el cual se especifican todas las características de la sesión de contenido multimedia (tipo de contenido multimedia (vídeo o audio), duración, las distintas calidades (bitrate), etc.)
2. Como se menciona en la definición del cliente, el algoritmo de adaptación dinámica empieza a solicitar segmentos multimedia. Además, monitoriza el tiempo que dura la descarga de estos segmentos multimedia y también almacena los segmentos descargados en la memoria temporal del reproductor (búfer).
3. Con los datos mencionados en el punto anterior, el algoritmo de adaptación dinámica calcula cual es la calidad adecuada según el ancho de banda disponible en el acceso al servidor *DASH*.

Como se mencionó anteriormente, el algoritmo de adaptación dinámica que se encuentra en el cliente no se define en el estándar *ISO/IEC 23009*. Sin embargo, el estándar sí especifica una serie de parámetros de *Calidad de Servicio (QoS)* y *Calidad de Experiencia (QoE)* en los que se puede basar para implementar el comportamiento del algoritmo adaptativo.

La *Calidad de Servicio (QoS)* es definida por la ITU-T en la recomendación E.800 [ITU-T17a] como: ‘La totalidad de características de un servicio de telecomunicaciones

Figura 2.6: Funcionamiento de la arquitectura DASH

que influyen en su capacidad para satisfacer las necesidades declaradas e implícitas del usuario del servicio.” (Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service).

La *Calidad de Experiencia (QoE)* es definida por la ITU-T en el documento P.10/G.100 [ITU-T17b] como: “El grado de deleite o molestia del usuario de una aplicación o servicio.” (The degree of delight or annoyance of the user of an application or service).

Estos parámetros tienen como objetivo evaluar de manera general la calidad en el servicio de distribución multimedia. A continuación, se describen las diferentes métricas de *Quality of Service (QoS)* y *QoE* definidas en el estándar ISO/IEC 23009.

2.3.4. Métricas para medir de la calidad en DASH

El estándar *DASH* de la *ISO/IEC* [ISO14b] define un conjunto de métricas y un mecanismo que el proveedor de servicios puede utilizar para iniciar la recopilación de métricas y la generación de informes en los clientes, si se diera el caso de que existe un mecanismo generador de informes.

El estándar *DASH* del 3GPP [3GPP18a] define no solo métricas de *QoS*, sino también métricas para calcular la *QoE* del usuario. El estándar del 3GPP además, define un protocolo para reportar las métricas de calidad llamado “Quality Reporting Protocol” en la sección 10 del estándar, utilizando un informe de *QoE* con formato XML que se llamará “*application/3gpdash-qoe-report+xml*”. Asimismo, se incluye en el anexo K una normativa para realizar una estimación del *Mean Opinion Score (MOS)* para audio y vídeo, el cual se basa en la norma P.1203 de la ITU-T [ITU-T16].

La norma P.1203 define un modelo de evaluación objetiva para el estándar *DASH*, el

cual predice el *MOS* para audio y vídeo por separado, y también, de manera conjunta en una escala de clasificación de categoría absoluta de 5 puntos para analizar la calidad en la reproducción del contenido multimedia.

A continuación, se muestran las métricas para medir la *QoS* y la *QoE* definidas tanto por la *ISO/IEC* como por el 3GPP:

- **TCP connections:** Esta métrica define una tabla la cual contiene todos los datos relacionados con las conexiones *TCP* con las cuales se hicieron las transacciones de los segmentos de contenido multimedia. En el estándar de la *ISO/IEC* [ISO14b] se define en la tabla D.1.
- **HTTP request/response transactions:** Esta métrica define una tabla, la cual contiene todos los datos relacionados con las conexiones *HTTP* con las cuales se hicieron las transacciones de petición y respuesta de los segmentos de contenido multimedia. Esta es una métrica para medir la *QoS*. En el estándar de la *ISO/IEC* [ISO14b] se define en la tabla D.2.
- **Average Throughput:** Esta métrica define una lista de la cual se puede obtener el throughput promedio que se observa en el cliente durante el intervalo de medición. Esta es una métrica para medir la *QoE* del servicio. En el estándar del 3GPP [3GPP18a] se define en la tabla 27.
- **Initial Playout Delay:** Esta métrica señala el retraso en la reproducción del contenido multimedia. Esta es una métrica para medir la *QoE* del servicio. En el estándar del 3GPP [3GPP18a] se define en la tabla 28 y en la tabla 31a.
- **Representation Switch Events:** Esta métrica define una lista de la cual se obtienen los eventos de cambio de representación del contenido multimedia. Esta es una métrica para medir la *QoS* del servicio. En el estándar de la *ISO/IEC* [ISO14b] se define en la tabla D.3.
- **Buffer Level:** Esta métrica muestra la lista de mediciones de nivel de ocupación de buffer (milisegundos) durante la reproducción a velocidad normal. Esta es una métrica para medir la *QoS*. En el estándar de la *ISO/IEC* [ISO14b] se define en la tabla D.4.
- **Play List:** Esta métrica contiene la lista de periodos de reproducción. Un período de reproducción es la acción del usuario que ocurre entre un intervalo de tiempo y

la siguiente acción del usuario, el final de la reproducción o una falla que detiene la reproducción. Esta es una métrica para medir la *QoS*. En el estándar de la *ISO/IEC* [ISO14b] se define en la tabla D.5.

- **MPD Information:** Esta métrica contiene una lista que se puede utilizar para reportar información acerca de la representación del MPD, de modo que los servidores de informes sin acceso directo al MPD puedan comprender las características de los medios utilizados. Esta es una métrica para medir la *QoE* del servicio. En el estándar del 3GPP [3GPP18a] se define en la tabla 31.
- **Device information:** Esta métrica contiene una lista con la información sobre las características del dispositivo físico del usuario, es decir la resolución de la pantalla y si el contenido multimedia se está reproduciendo en la pantalla completa o no. Esta es una métrica para medir la *QoE* del servicio. En el estándar del 3GPP [3GPP18a] se define en la tabla 31b.

2.3.5. Herramientas para DASH

Actualmente existe disponible una amplia variedad de software tanto libre como comercial para implementar servicios basados en *DASH*. En esta sección se describen brevemente las distintas herramientas consideradas en el ámbito de la tesis.

2.3.5.1. Implementaciones Software Cliente DASH de Código Libre

En esta sección se describen los distintos reproductores de código libre para la reproducción de contenido multimedia utilizando el estándar *DASH* que existen actualmente.

- *dash.js*: Es una iniciativa del Foro *DASH-IF* para establecer una producción dentro de un marco de calidad para crear reproductores de contenido multimedia compatibles con el estándar *DASH* utilizando bibliotecas JavaScript y el “Media Source Extensions API” definido por el W3C.
- *dash.as*: CastLab creó el reproductor *dash.as* para reproducir contenido *DASH* en cualquier dispositivo compatible con Adobe Flash. Está escrito en el lenguaje “ActionScript” de Adobe y fue diseñado para ser liviano con un gran rendimiento. El proyecto se encuentra abandonado desde 2016

- Osmo4 de GPAC: Es un cliente que puede reproducir contenido multimedia del estándar *DASH*, soportando tanto los formatos de MPEG-2 como de ISO-BMFF. El cliente fue creado por el proyecto GPAC que pertenece al grupo de multimedia de Telecom ParisTech. El proyecto se encuentra en la versión 0.8.0 para el reproductor Osmo4.
- VLC de VideoLAN: VLC media player es un reproductor y framework multimedia, libre y de código abierto desarrollado por el proyecto VideoLAN. VLC puede reproducir contenido *DASH* desde su versión 3.0 (Vetinari) de manera nativa.
- Javascript and MediaSourceAppend by YouTube team at Google: Un reproductor para contenido *DASH* creado por la empresa YouTube (Google) para realizar pruebas y validaciones del estándar *DASH*. El proyecto se encuentra abandonado desde 2016.

Después de realizar un análisis profundo de las diferentes implementaciones de clientes *DASH* de código libre, se ha seleccionado el reproductor *dash.js* debido a la facilidad de modificar el código JavaScript, a que es un proyecto vigente que la comunidad científica sigue mejorando y a la portabilidad que presenta, ya que el reproductor *dash.js* funciona con casi todos los navegadores *HTTP* para computadoras de escritorio (Edge, Firefox, Chrome, Safari) y para algunas versiones de navegadores en el sistema operativo Android.

2.3.5.2. Implementaciones Comerciales para DASH

En esta sección se presentan de manera breve las distintas soluciones comerciales de herramientas para ofrecer el servicio de distribución multimedia utilizando el estándar *DASH*. El problema que presentan estas soluciones es que es necesario contar con licencias (costo) para poder implementar y modificar la solución que proponen para la reproducción de contenido multimedia basado en el estándar *DASH*. A continuación, se presentan las distintas soluciones:

- Aricent: Aricent ha desarrollado el Software Developer Kit (SDK) el cual es un marco de desarrollo de aplicaciones para reproducir contenido *DASH*.
- AvidBeam: AvidBeam Technologies es una empresa que se dedica a realizar implementaciones para ofrecer servicios multimedia basado en *DASH*.
- Bitdash: es un conjunto de clientes *DASH* altamente optimizados para diversas plataformas y dispositivos.

- BuyDRM: La plataforma KeyOS de BuyDRM ofrece soluciones tecnologías centradas en DASH que incluyen tanto la parte del cliente como la del servidor.
- Helix: Ofrece la distribución de contenido multimedia *DASH* con calidad mediante el desarrollo de Helix SDK para crear clientes Android.
- InterDigital: ofrece la implementación de servicios *DASH* para el diseño de clientes para dispositivos móviles, como teléfonos inteligentes y tabletas con Android.
- NexPlayer: NexStreaming es una compañía global de software multimedia enfocada en el desarrollo de software multimedia premium.
- PRESTOplay: PRESTOplay desarrolla software para crear reproductores a medida de alta calidad para sus clientes.
- Qualcomm: Qualcomm ha desarrollado un cliente para la reproducción *DASH* especial para la plataforma Android que se ejecuta en los procesadores Snapdragon.
- StreamOne: StreamOne Player es un reproductor de vídeo HTML5 flexible de alto rendimiento que soporta *DASH*.
- TheoPlayer: THEOplayer es el reproductor de contenido multimedia *DASH* basado en HTML5 que ofrece una experiencia de reproducción de primera clase sin plugins.
- VisualON: Cuenta con un kit de desarrollo de reproductores multimedia comerciales que permite la recuperación y reproducción de contenido *DASH* multiplataforma.

2.3.5.3. Software para validar contenido DASH

En esta sección se presentan de manera breve las distintas soluciones para validar que el contenido multimedia se ajusta a lo establecido por el estándar *DASH*. A continuación, se presentan estas soluciones:

- *DASH-IF* Validator: Software validador y conformador para contenido DASH (MPD y segmentos basados en ISO-BMFF)
- Manifest Validator: Software que ofrece la posibilidad de validar archivos MPD para DASH. El software tiene la capacidad para hacerlo en la nube. Además, no revisa solo el cumplimiento del esquema de DASH sino también el cumplimiento de la semántica en el archivo de descripción.

- CPIX Document Validator: Esta herramienta valida la conformación del documento XML (MPD) con la especificación del intercambio de información de protección de contenido (CPIX).

2.3.5.4. Marco de evaluación para algoritmos DASH

En esta sección se presentan de manera breve las distintas soluciones para evaluar el servicio de distribución multimedia utilizando el estándar *DASH*.

- Sabre [Spiteri19]: Es un entorno de simulación de código abierto para algoritmos *ABR*. Es una herramienta de Python que facilita el desarrollo inicial y la evaluación rápida de algoritmos en un entorno similar a los reproductores reales, sin requerir que los investigadores del algoritmo aprendan los detalles de implementación a menudo compleja del reproductor *DASH*. El proyecto se encuentra activo desde 2017.
- Advise [Zabrovskiy17]: Es una herramienta que permite realizar experimentos de una manera automática y realizar un análisis profundo de los resultados utilizando diferentes parámetros para el servicio de distribución multimedia. La herramienta permite calcular métricas tanto de *QoS* como de *QoE*. El proyecto se encuentra activo desde 2017.
- Dashbed [Raca19]: Es una herramienta flexible, escalable y personalizable que permite realizar el análisis de algoritmos *ABR* en un ambiente inalámbrico. El proyecto se encuentra activo desde 2017.

2.4. Esquemas de distribución del contenido DASH

La distribución de contenido multimedia [Nussbaumer95] se refiere a la situación en que un usuario observa vídeo digital o escucha audio digital desde un cliente (teléfono, computadora, etc.) a través de Internet. Sin embargo, existen distintas formas en la manera en que se distribuye a los clientes el contenido multimedia que se encuentra en Internet.

En el estándar *DASH* existen dos secciones específicas que marcan las diferencias entre servicios de distribución bajo demanda (on-demand) y en vivo (live). En los servicios de distribución bajo demanda, el contenido se encuentra de manera completa en el servidor *HTTP* y el usuario accede a ellos en el momento y lugar que desea. En los servicios de distribución en vivo, se complica el esquema, ya que es necesario tener la fuente del

contenido multimedia, un codificador para digitalizar la información con las características necesarias y una *CDN* para distribuir el contenido.

A continuación, se explican los distintos tipos de distribución de contenido multimedia utilizando el estándar *DASH*.

2.4.1. Servicio de distribución multimedia bajo demanda

El contenido multimedia bajo demanda es un sistema de programación que permite a los usuarios seleccionar y ver/escuchar contenido de vídeo o audio como películas y programas de TV cada vez que elijan, en lugar de a una hora de transmisión programada.

Una de las desventajas de usar la distribución del contenido multimedia bajo demanda es que es necesario proporcionar un flujo de vídeo para cada solicitud del usuario lo cual resulta en un uso poco eficiente del ancho de banda del cliente y del operador [Choi12].

En el estándar *DASH*, el MPD es típicamente un documento estático que describe los diversos aspectos del contenido multimedia. Todos los segmentos de la presentación multimedia están disponibles en el servidor una vez que se ha generado y almacenado el contenido multimedia.

En el estándar *DASH*, dentro del archivo de metadatos MPD se define *type="static"* para referirse a que el contenido multimedia puede ser accedido bajo demanda. También se define dentro del MPD el esquema *xsi:schemaLocation* de *DASH* con el valor *urn:mpeg:dash:profile:isoff-on-demand:2011*.

Para servicios de distribución de contenido multimedia bajo demanda una buena estrategia sería utilizar una calidad baja para utilizar menos ancho de banda en el inicio de la reproducción y un buffer en el cliente de gran tamaño [Thang14].

2.4.2. Servicio de distribución multimedia en vivo

Tradicionalmente la transmisión de servicios de distribución de contenido multimedia se ha realizado mediante el uso de la televisión abierta. No obstante, el estándar *DASH* ha habilitado el uso de Internet para la distribución de contenido multimedia en vivo [Seufert15] a pesar de unas restricciones de latencia estrictas en comparación con los servicios de distribución bajo demanda.

El desafío de la transmisión en vivo es minimizar el retardo de extremo a extremo entre la generación de contenido y la reproducción, lo que se traduce en la “vivacidad” de la reproducción del contenido multimedia. Sin embargo, los procesos de codificación, seg-

mentación y transporte del contenido multimedia en respuesta a las solicitudes del cliente en el extremo del servidor agregan retardos adicionales.

En el estándar *DASH*, dentro del archivo MPD se define para los servicios en vivo el valor de *type*="dynamic" para referirse a que el contenido multimedia puede ser accedido en vivo. También se define dentro del MPD el esquema *xsi:schemaLocation* de *DASH* con el valor *urn:mpeg:dash:profile:isof-live:2011*. Uno de los desafíos que presenta el servicio de distribución de contenido multimedia en vivo es cómo el archivo MPD se va actualizando durante la reproducción del contenido multimedia.

2.4.3. Las CDN en el servicio de distribución multimedia

Algunas de las principales compañías comerciales en la distribución de servicios multimedia basados en *DASH* como Hulu y Netflix [Adhikari15] ofrecen servicios *Over-The-Top (OTT)* [Erman11]. *OTT* se refiere a la transmisión del contenido multimedia sin el control del operador sobre el servicio, es decir, que el *Internet Service Provider (ISP)* solo actúa como una red de transporte de datos. El principal reto para estas compañías de distribución de contenido multimedia basado en el estándar *DASH* es cómo transmitir este contenido a miles de usuarios simultáneos mientras se ofrece una reproducción continua y con una alta interactividad.

Para lograr este objetivo, las compañías de distribución de contenido multimedia utilizan las redes de distribución de contenido o *CDN*. Una *CDN* consiste en múltiples grupos de servidores (cluster) distribuidos geográficamente por el mundo. Estos grupos de servidores tienen como objetivo ofrecer el contenido multimedia desde la localización más cercana al usuario. A continuación, se muestran los tipos de *CDN* que existen para el servicio de distribución de contenido multimedia:

- **CDN privada:** Este tipo de *CDN* es propiedad del proveedor de contenido multimedia. Es el caso de YouTube.
- **CDN de un tercero:** Este tipo de *CDN* distribuye el contenido multimedia de múltiples proveedores, por ejemplo, Akamai [Dilley02] fue durante un tiempo la empresa encargada de distribuir el contenido multimedia de las plataformas Netflix y Hulu.

Uno de los principales problemas que presenta el servicio de distribución de contenido multimedia se debe a la alta heterogeneidad y a la dinámica cambiante de Internet, es decir, un grupo de servidores geográficamente cercanos al usuario no siempre significa un

acceso cercano en términos de red (retardo y variabilidad del ancho de banda en el tiempo) [Pathan14].

Por lo tanto, se prefieren estrategias de despliegado de la *CDN* dependientes de los factores de red [Almashor15], es decir, seleccionando como parámetros para la elección del servidor basados en: el retardo, las pérdidas de paquetes, el desempeño del ancho de banda, la carga del grupo de servidores, el costo de la entrega de los datos por el *ISP*, etc.

2.4.4. Servicio de difusión multimedia

El servicio distribución de contenido multimedia es intrínsecamente un servicio personalizado con la característica de interacción uno-a-uno, por lo que requiere un flujo dedicado para atender cada solicitud del usuario [Choi12].

Sin embargo, proporcionar un flujo de contenido multimedia por separado (de gran ancho de banda) para cada solicitud del usuario no es rentable para el operador del servicio. Es por esto por lo que se han creado servicios de difusión del contenido multimedia con los cuales se pretende ahorrar en el ancho de banda ya que solo se genera un flujo del contenido multimedia para todos los usuarios del servicio.

Dentro del estándar *DASH* no se especifica la manera en la que el contenido se pueda utilizar para ofrecer servicios de difusión (broadcast) multimedia. Sin embargo, en las redes móviles existe el servicio *Multimedia Broadcast and Multicast Service (MBMS)*, el cual es el encargado de transmitir los datos de una sola entidad de origen a múltiples destinatarios. El 3GPP define el servicio *MBMS* en la especificación técnica 23 246 llamada “Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description” [3GPP17a]. El servicio *MBMS* utiliza el estándar *DASH* para la codificación del contenido multimedia y también dentro del servicio para la recuperación de segmentos perdidos o que llegan dañados al cliente.

El termino *MBMS* aplica a redes móviles 2G y 3G. Sin embargo, para redes 4G, específicamente *LTE*, el servicio se denomina *Evolved Multimedia Broadcast and Multicast Service (eMBMS)*. Broadcast es el término utilizado para describir la comunicación en la que se envía una información desde un solo emisor a todos los receptores conectados.

Dentro del servicio *eMBMS* se utiliza la especificación técnica 26.346 del 3GPP llamado “Protocols and codecs” [3GPP17b], en la cual se definen los diferentes tipos de códecs de contenido multimedia soportados por el servicio. El servicio *MBMS* y *eMBMS* utiliza el estándar *DASH* para codificar y segmentar el contenido multimedia para su envío por la red móvil.

Debido a que *HTTP* no se puede utilizar con enlaces unidireccionales, se introdujo un protocolo unidireccional, este es el protocolo *File Delivery over Unidirectional Transport (FLUTE)* definido en el RFC 3926 [Paila04]. El protocolo *FLUTE* se transmite a través del protocolo *UDP* hacia los dispositivos del cliente. El uso del protocolo *UDP* hace que la entrega de datos no sea fiable, por lo que el estándar del *3GPP* define el uso de mecanismos de recuperación de errores mediante códigos *Forward Error Correction (FEC)*, y si los códigos fallan, la recuperación de segmentos perdidos se realiza mediante peticiones *HTTP* usando el acceso “unicast” del cliente *LTE*.

El servicio *eMBMS* cubre el servicio para muchos usuarios ubicados en la misma área, los cuales consumen el mismo contenido al mismo tiempo [Lecompte12]. Por ejemplo, muchos usuarios ubicados en un estadio para ver un juego o un evento en vivo en sus dispositivos.

El servicio *eMBMS* es eficiente para enviar rápidamente el mismo contenido a muchos dispositivos al mismo tiempo sin interacción con el usuario; por ejemplo, almacenamiento de contenido multimedia, eventos en vivo, noticias, clips de música o inyección de actualizaciones de software.

El servicio *eMBMS* utiliza canales reservados en la trama de la red móvil. En la red *LTE* las tramas son llamadas *Physical Resource Block (PRB)* y su tamaño depende de la frecuencia utilizada para su despliegue.

2.4.5. Handover para servicios de difusión DASH

Los usuarios y las aplicaciones móviles presionan a los operadores para que mejoren la continuidad en el traspaso (“*vertical handover*”) de dispositivos y servicios, una presión que continuará en el futuro [Fernandes12]. El *vertical handover* se refiere a un dispositivo o servicio que cambia el tipo de conectividad que utiliza para acceder a una infraestructura de red, por lo general para soportar la movilidad de dicho dispositivo o servicio.

La disponibilidad de un gran número de redes inalámbricas es el resultado del despliegue de bajo costo de los puntos de acceso locales y las estrategias a corto plazo de los operadores para cubrir áreas geográficas más pequeñas a bajo costo [Ahmed14]. La llegada de los puntos de acceso de femtocelda, por ejemplo, sin duda mejorará la cobertura en interiores y proporcionará conectividad confiable sin la necesidad del despliegue ineficiente de costosas estaciones base adicionales.

Una femtocelda es una estación base de telefonía móvil muy pequeña que está conectada a la red telefónica a través de Internet, que se utiliza generalmente en áreas donde la señal

móvil es débil. En áreas urbanas densas, los dispositivos móviles o servicios comúnmente detectan señales de otros puntos de acceso y operadores que en ocasiones proporcionan mejor conectividad.

Con los avances recientes en la tecnología de software de radio, la mayoría de los dispositivos móviles de hoy en día, son capaces de comunicarse a través de diferentes tecnologías (Bluetooth, IEEE 802.11ac, IEEE 802.16(WiMAX), LTE). Por lo tanto, los dispositivos y servicios prometen mejorar la *QoE* del usuario en un escenario “siempre con la mejor conexión posible”.

En el mundo de los dispositivos móviles existen varios mecanismos propuestos que habilitan el *vertical handover* entre tecnologías de conectividad. Sin embargo, para los servicios no existe tanta literatura sobre mecanismos para realizar *vertical handover*, especialmente para servicios *DASH*, en donde es necesario cambiar entre un esquema de difusión mediante el servicio *eMBMS* que utiliza el protocolo *UDP* y los mecanismos de funcionamiento normales en *DASH*, es decir, peticiones usando el protocolo *HTTP* que van encapsuladas en el protocolo *TCP*.

2.5. Conclusiones

En este capítulo se han establecido los conceptos básicos sobre el servicio de distribución de contenido multimedia, haciendo especial énfasis en los conceptos que han servido de partida para el desarrollo de este trabajo de investigación. Este énfasis se muestra en la sección que define al estándar *DASH* sobre el cual se basa la implementación de un nuevo algoritmo para mejorar la estimación del throughput, con el fin de incrementar la calidad de la experiencia de los usuarios del servicio de distribución multimedia y que pueda ser implementado en distintos escenarios de distribución.

En el apartado 2.2 se mostró la historia de los servicios de distribución multimedia y las tecnologías que habilitaron el incremento en el uso de servicios de este tipo.

En el apartado 2.3 se analizaron las tecnologías que precedieron al estándar *DASH*, las cuales se llaman Progressive download streaming y *HTTP* adaptive streaming (HAS). Debido a que estas tecnologías resultaron en soluciones comerciales, se decidió crear el estándar *DASH* de la *ISO/IEC* [ISO14b].

En el apartado 2.4 se analizaron las diferentes partes que definen al estándar *DASH* [ISO14b]: la definición del archivo de metadatos MPD, el tipo de contenido multimedia que soporta el estándar, la arquitectura que utiliza, los diferentes componentes del cliente

y del servidor y las métricas de calidad que se definen en el estándar. Además, se indican las distintas herramientas existentes consideradas en la tesis.

En el apartado 2.5 se explicaron las diferentes maneras que existen para distribuir el contenido entre los clientes. Los servicios pueden ser bajo demanda o en vivo. Además, se muestran las ventajas de utilizar las *CDN* para la distribución del contenido multimedia. También existen diferentes tipos de entrega en ambientes con muchos clientes para ahorrar ancho de banda mediante técnicas de difusión del contenido multimedia.

Como conclusión final de este capítulo, al desarrollar el análisis sobre la tecnología para el servicio de distribución de contenido multimedia basado en el estándar *DASH*, se ha encontrado que los algoritmos de adaptación dinámica en el reproductor multimedia no se encuentran estandarizados. En el Capítulo 3 se analizan en detalle los principales algoritmos de adaptación dinámica propuestos por la comunidad científica y la industria privada haciendo especial énfasis en sus ventajas y desventajas.

CAPÍTULO 3

ALGORITMOS DE ADAPTACIÓN EN DASH

3.1. Introducción

En el capítulo anterior se analizaron las distintas opciones existentes para ofrecer los servicios de distribución multimedia, usando el estándar *DASH*. Una vez analizadas las partes que componen el estándar *DASH*, se comprobó que los algoritmos que rigen la adaptación dinámica para los servicios basados en *DASH* no forman parte del alcance del estándar, por lo que el esfuerzo investigador se debe dirigir por una parte al estudio de los principios que rigen a estos algoritmos y por otra parte al análisis de los distintos algoritmos de adaptación dinámica que han sido propuestos por la comunidad científica. Todo ello, con el fin de encontrar soluciones que mejoren la calidad percibida por los usuarios para los servicios de distribución multimedia en redes móviles.

El capítulo se estructura de la siguiente manera: En el apartado 3.2 se muestran los principios que rigen los algoritmos de adaptación en el estándar *DASH*. A continuación, el apartado 3.3 presenta y discute el estado del arte sobre algoritmos de adaptación para *DASH*, en el cual se incluyen los principales métodos con los que se intenta mejorar la adaptación de los reproductores *DASH*. En el apartado 3.4, se muestran los modelos existentes para medir la *QoE* del usuario para la distribución de contenido multimedia basado en el estándar *DASH*. Finalmente, en el apartado 3.5, se presentan las conclusiones de este capítulo.

3.2. Algoritmos de Tasa Adaptativa (ABR)

Un servicio de streaming de vídeo tiene unos requisitos en cuanto al ancho de banda, al retardo y a la pérdida de paquetes [Wu00]. Estos requisitos [Rao11] surgen porque el dispositivo de reproducción de vídeo del cliente espera que los datos estén constantemente disponibles para poder seguir actualizando la pantalla a un ritmo constante, generalmente a un número alto de fotogramas por segundo. No obstante, si se produce una interrupción (stalling event) en este proceso y no hay datos disponibles en la memoria de almacenamiento temporal (buffer under-run), se muestra una pantalla congelada temporalmente mientras el dispositivo del cliente termina de recuperar los datos (rebuffering event) para continuar con la reproducción continua del contenido multimedia. Los algoritmos de tasa adaptativa “ABR” buscan evitar estos efectos que afectan a la *QoE* de los usuarios.

Los algoritmos *ABR* son una técnica utilizada en el servicio de transmisión multimedia, y funciona detectando el ancho de banda del usuario en tiempo real y ajustando la calidad de la transmisión de contenido multimedia en consecuencia (bitrate change). Esto se realiza de acuerdo con una serie de factores como son:

- Las capacidades del terminal del cliente (CPU, tamaño de pantalla, etc.).
- El tipo de red de acceso al contenido multimedia (fija, móvil, etc.).
- El ancho de banda disponible.

La técnica *ABR* funciona en dos bucles, el bucle de control interno de *TCP* y el bucle de control externo *ABR* [Varma15]. El bucle de control interno de *TCP* se adapta a la congestión de la red e intenta igualar la tasa de envío del transmisor *TCP* con la tasa que en realidad soporta la red. Este primer bucle opera en un orden de un periodo de tiempo que es aproximadamente igual al *Round Trip Time (RTT)*, es decir, decenas de milisegundos. El bucle de control externo *ABR* funciona intentando igualar la tasa que se obtiene de *TCP* a una estimación suavizada de la medida *TCP* para después cuantizar la estimación a una calidad discreta (bitrate) del contenido multimedia. Este bucle opera en un periodo de tiempo grande, es decir del orden de segundos a decenas de segundos. En la figura 3.1 se muestra la diferencia entre el bucle interno y el bucle externo con respecto a la descarga de segmentos multimedia solicitados por el reproductor *DASH*.

Esto produce un efecto de promediado natural porque *ABR* no intenta igualar el rendimiento fluctuante a corto plazo de *TCP*, sino estimar un rendimiento suavizado que se promedia en varios segundos. Pero, aun así, es un problema no trivial estimar el promedio

Figura 3.1: El bucle de control interno y externo

TCP “verdadero” a partir de las mediciones realizadas en el receptor, lo que puede ocasionar una sobreestimación o subestimación del rendimiento de TCP que da como resultado oscilaciones en la tasa de vídeo seleccionadas.

Los objetivos generales de la técnica *ABR* [Varma15] son:

1. Evitar interrupciones en la reproducción del vídeo debido al agotamiento del buffer de almacenamiento temporal.
2. Maximizar la calidad del vídeo, es decir, encontrar el equilibrio entre evitar paradas y seleccionar la mejor calidad del vídeo para incrementar calidad de la experiencia para el usuario.
3. Fomentar la estabilidad, es decir, realizar cambios en la calidad del vídeo solo por necesidad.
4. Promover la equidad (Fairness) entre los reproductores *DASH*.

El desafío central de los algoritmos *ABR* es evitar que el buffer de reproducción del cliente se quede sin contenido multimedia (buffer starvation), al tiempo que se maximiza la *QoE* percibida por el usuario al adaptarse a las condiciones de la red que cambian dinámicamente [Kua17].

Durante una transmisión típica de una sesión de contenido multimedia utilizando el estándar *DASH* se pueden observar dos estados marcados, los cuales son: una fase de acumulación de segmentos de vídeo (buffering phase) y una fase de estado estable (steady state phase) [Rao11].

En la primera fase, el cliente empieza a reproducir el vídeo cuando una cantidad suficiente de datos están guardados en el buffer. En *DASH*, la variable *MinBufferTime*

del MPD fija el tamaño de buffer necesario para empezar la reproducción. Esta primera fase garantiza que el cliente cuenta con una cantidad suficiente de datos para compensar la varianza del ancho de banda disponible para el reproductor. En el estándar *DASH*, esta primera fase intenta descargar segmentos de contenido multimedia con la calidad baja, con el fin de agilizar la reproducción de vídeo y para realizar las primeras estimaciones del ancho de banda que el cliente percibe.

En la segunda fase, el cliente descarga nuevos segmentos de contenido multimedia de acuerdo con el ancho de banda detectado al descargar los segmentos anteriores y con el nivel de ocupación del buffer. Este comportamiento produce un ciclo ON-OFF [Kupka12]. En el periodo ON el cliente descarga un bloque de datos a la velocidad que permite el protocolo *TCP*, en el periodo OFF la conexión *TCP* está libre.

Existe otra fase, la cual sucede solo en el caso en que el buffer de datos se quede sin contenido multimedia (rebuffering phase), el cual es parecido a la primera fase, en donde el reproductor decide acumular segmentos suficientes para garantizar la reproducción continua del contenido multimedia. La aparición de un evento de parada y su posterior espera para obtener los datos necesarios conlleva el deterioro de la *QoE* percibida por el usuario.

El algoritmo de tasa adaptativa controla dos características principales en el cliente: la velocidad a la que el receptor envía solicitudes *HTTP* a la fuente de vídeo y la calidad de vídeo apropiada para su descarga, cuya tasa se aproxima a la velocidad que la red ofrece al cliente [Varma15].

Típicamente un algoritmo de tasa adaptativa cuenta con las siguientes definiciones para su funcionamiento [Varma15]:

$V = \{V(1), \dots, V(L)\}$ Conjunto de tasa de bits para el flujo de vídeo que están disponibles en el transmisor, con $0 < V(n) < V(m)$ para $n < m$.

U_n : La velocidad de vídeo de entre los del conjunto V que se usa para el n -ésimo segmento, que es un resultado del algoritmo de tasa adaptativa.

τ : Tamaño de un segmento (Chunk) en segundos, en términos del tiempo real dedicado a la reproducción del vídeo contenido en él.

R_n : Throughput obtenido al descargar el n -ésimo trozo de vídeo medido en el receptor.

$\hat{T}_n$ Intervalo entre solicitudes *HTTP* entre la n -ésima y $(n+1)$ solicitudes. Esto también es un resultado del algoritmo de tasa adaptativa.

$\tilde{T}_n$ Duración de descarga para el n -ésimo trozo.

B_n : Duración del vídeo almacenado en el buffer al final del n -ésimo período.

Usualmente los algoritmos de tasa adaptativa siguen un proceso de tres pasos, los cuales

son: estimación del throughput, determinación de la tasa del vídeo del siguiente segmento y programación del siguiente segmento.

- El primer paso, se estima el throughput (throughput instantáneo) del n -ésimo segmento de vídeo utilizando la siguiente ecuación:

$$R_n = \frac{U_n \cdot \tau}{\tilde{T}_n}, n \geq 1 \quad (3.1)$$

Y el tamaño del buffer (en segundos) al final de la descarga del n -ésimo segmento se calcula utilizando la siguiente ecuación:

$$B_n = \max(0, B_{n-1} + \tau - T_n), n \geq 1 \quad (3.2)$$

- En el segundo paso, el algoritmo de tasa adaptativa determina la tasa del vídeo para el siguiente segmento (U_{n+1}) de acuerdo con los resultados la estimación del throughput (R_n) y el tamaño del buffer (B_n), es decir,

$$U_{n+1} = F(B_n, \{R_m : m \leq n\}), n \geq 1 \quad (3.3)$$

Donde $F(\cdot)$ es una función de cuantización.

- El tercer paso consiste en que el algoritmo de tasa adaptativa determina el tiempo en que el próximo segmento de vídeo será descargado calculando el intervalo entre solicitudes *HTTP* de $\hat{T}_n$, tal que:

$$\hat{T}_n = G(B_n, \{R_m : m \leq n + 1\}), n \geq 1 \quad (3.4)$$

De acuerdo con las definiciones anteriores se pueden detectar tres posibles resultados del algoritmo de tasa adaptativa en el estándar *DASH*, los cuales se plasman en la figura 3.2. El primero (figura 3.2 (A)) es que la tasa de vídeo (V_n) es igual al throughput estimado (R_n), en donde debido a que las dos velocidades están perfectamente adaptadas, se tarda exactamente τ segundos en transmitir el segmento, y no hay cambio en el tamaño del buffer al final del n -ésimo período (mediante la ecuación 3.2). Este es el estado ideal para el algoritmo de tasa adaptativa porque aprovecha al máximo la capacidad de la red y el tamaño del buffer es constante.

El segundo caso (figura 3.2 (B)) se produce cuando la tasa de vídeo (V_n) es menor que el

Figura 3.2: Escenarios de descarga de segmentos en *DASH* [Varma15]

throughput estimado (R_n), de modo que la red llena el buffer más rápido que la velocidad constante a la que el decodificador lo está drenando. Esta condición debe ser detectada por el algoritmo para que pueda tomar una acción de control, que consiste en aumentar la tasa de bits de vídeo para que pueda aprovechar la mayor capacidad de la red. Si la velocidad de bits del vídeo ya es máxima, el algoritmo ABR evita el desbordamiento del buffer espaciando el intervalo de tiempo entre los segmentos (comportamiento ON-OFF).

El tercer caso (figura 3.2 (C)) es que la tasa de vídeo (V_n) sea mayor al throughput estimado (R_n). Entonces el decodificador drena el buffer a una velocidad mayor que la velocidad a la que la red lo está llenando, lo que puede resultar en un agotamiento del buffer. Esta condición también necesita ser detectada rápidamente por el algoritmo para que pueda tomar una acción de control, que consiste en reducir la tasa del vídeo a un valor inferior a la velocidad de la red.

Debido a la cuantización en las velocidades del vídeo, es imposible llegar a $U_n = R$, y esto puede hacer que la velocidad del vídeo fluctúe constantemente entre $V(m)$ y $V(m+1)$ con el fin de evitar que el buffer se agote o desborde. Para evitar el agotamiento del buffer, la mayoría de los algoritmos de tasa adaptativa dejan de aumentar la tasa del vídeo U_n

después de que llega a $V(m)$ y compensan la diferencia entre $V(m)$ y R aumentando la distancia entre descargas de los segmentos siguientes.

El problema respecto a las estimaciones basadas en la ecuación 3.1 definida en la sección anterior, es que causaría un efecto de constante cambio en la calidad de los segmentos (bitrate switching), dado que las medidas del throughput instantáneo fluctúan excesivamente [Lin14] de un segmento a otro. Para resolver el problema mencionado anteriormente se utilizan métodos matemáticos de suavizado estadístico de las medidas del throughput instantáneo.

Otro factor que afecta a los algoritmos de tasa adaptativa de acuerdo con Sani *et al.* [Sani17] es el tamaño que dura en segundos el segmento de contenido multimedia. En la tabla 3.1 se describe el efecto que tiene en los algoritmos de tasa adaptativa (ABR) según la duración del segmento. Un segmento se considera pequeño si el contenido multimedia que almacena es menor a 3 segundos y se considera grande cuando es mayor de 3 segundos [Sideris15].

Tabla 3.1: Tamaño del segmento y ABR [Sani17]

Efecto en ABR	Segmento Grande	Segmento Pequeño
Rebuffering	Más Eventos	Menos Eventos
Eficiencia de la Codificación	Sobrecarga de cabeceras bajo	Sobrecarga de cabeceras Alto
Adaptabilidad	Baja	Alta
Fluctuaciones de Calidad	Las reduce	Las incrementa
Utilización de Recursos	Mejora utilización	Empeora utilización
Retardo en el Inicio	Alto retardo	Bajo retardo
Equidad	La mejora	La reduce

En el siguiente apartado se presentan y discuten las principales propuestas y trabajos sobre algoritmos de adaptación para el estándar *DASH*.

3.3. Estado del Arte sobre Algoritmos de Adaptación

En esta sección se discuten las principales propuestas de algoritmos de adaptación utilizados en los reproductores *DASH*.

Sani *et al.* [Sani17] propone clasificar los algoritmos en dos grandes áreas. La primera área llamada interna, la cual cubre los algoritmos que proponen mejoras internas, es decir, nuevos métodos para mejorar la estimación del throughput o el control del buffer. El área externa se corresponde con las mejoras de los mecanismos externos al esquema de tasa adaptativa, que pueden tener un impacto en el rendimiento del algoritmo de tasa adaptativa

(ABR). Por ejemplo, los algoritmos propuestos por [Essaili15] [Lentisco17b] [Bouten14] implementan un proceso de adaptación asistido por nodos de red intermedios (proxy) que tienen información sobre cuánto ancho de banda se va a asignar a los clientes en redes móviles.

Kua *et al.* [Kua17] clasifica los algoritmos de adaptación del estándar *DASH* en tres grandes apartados que son: los algoritmos que fueron creados para mejorar las estimaciones del throughput, los que se crearon para mejorar el control del buffer y otros algoritmos que son híbridos (utilizan el throughput y buffer) o que utilizan teoría de control para su funcionamiento.

3.3.1. Algoritmos basados en el throughput

Esta clase de algoritmos se basan en el ancho de banda que ofrece el protocolo *TCP* (de acuerdo con la estimación que se obtenga por la capa de aplicación) como el elemento clave para seleccionar la calidad de los siguientes segmentos de contenido multimedia. Los algoritmos basados en el throughput difieren en términos de los métodos que se utilizan para estimar el throughput y cómo se aplican las estimaciones en el algoritmo.

3.3.1.1. Akhshabi *et al.*

Akhshabi *et al.* [Akhshabi11] fueron de los primeros en probar las distintas tecnologías para el servicio de streaming de vídeo. Akhshabi *et al.* compararon clientes propietarios y clientes de código abierto, como son: Microsoft Smooth Streaming, Reproductor de Netflix y Adobe *OSMF*. El cliente de Adobe está implementado en código abierto y fue modificado para introducir una nueva forma de estimar el throughput del reproductor de vídeo, utilizando el método matemático “*Exponential Weighted Moving Average (EWMA)*”. Este artículo fue una de las primeras investigaciones en incorporar métodos matemáticos para realizar la estimación del ancho de banda.

También fueron los primeros en analizar el comportamiento de varios clientes compitiendo por el ancho de banda en un “cuello de botella” [Paxson99]. Para realizar el estudio de los reproductores utilizaron una herramienta, llamada DummyNet, para controlar el ancho de banda en el enlace de bajada para los clientes.

El problema que presenta el algoritmo propuesto por Akhshabi *et al.* es que utiliza pesos fijos en el algoritmo basado el *EWMA*, lo que hace que funcione bien en redes móviles pero con muchos cambios de bitrate en redes fijas.

3.3.1.2. Liu *et al.*

Liu *et al.* [Liu11] proponen un método de adaptación del bitrate que utiliza un mecanismo similar al control de congestión utilizado por *TCP* llamado “*Additive Increase/Multiplicative Decrease (AIMD)*”. Es decir, realiza una conmutación creciente conservadora (aumento aditivo) y una conmutación decreciente agresiva (disminución multiplicativa). Este método utiliza una media móvil simple de los últimos 10 segundos de historia para suavizar las medidas del throughput instantáneo, que Liu *et al.* definen como: la duración del segmento multimedia (Media Segment Duration) dividido entre el tiempo que se tarda en recuperar el segmento (Segment Fetch Time).

Además, este método de adaptación implementa condicionantes diferentes dependiendo de si la conmutación en la calidad del vídeo es creciente o decreciente. Para el caso creciente implementa un factor de cambio que depende del ancho de banda de la representación. Para el caso decreciente, implementa un nivel de seguridad relacionado con la ocupación del buffer para detectar congestiones en la red antes de que el buffer de la aplicación se quede sin datos. Este método fue simulado utilizando el software ns-2.

Sin embargo, el método de adaptación propuesto por Liu *et al.* no toma en cuenta el comportamiento de su método en ambientes de redes inalámbricas. Suavizar las medidas de throughput instantáneo con 10 segundos de historia en redes inalámbricas provocaría que el método subestimara o sobreestimara el throughput real que ofrece la red debido a la naturaleza fluctuante en las redes móviles [Ma17], por lo que se podrían encontrar eventos de parada en la reproducción del contenido multimedia.

3.3.1.3. Thang *et al.* 2012

Thang *et al.* [Thang12] proponen el uso de un método para calcular el throughput del cliente utilizando *EWMA* y un factor que llaman desviación de la tasa normalizada. Para calcular los pesos de *EWMA* utilizan una función logística que relaciona la desviación de la tasa normalizada y el tipo de red seleccionada (móvil, inalámbrica o fija).

También proponen el uso de datos extendidos en el archivo de presentación del contenido multimedia *MPD*, con los cuales pretenden realizar una mejor cuantización entre las estimaciones del throughput a los valores fijos de la tasa de bits de las representaciones multimedia.

El problema que presenta el algoritmo propuesto por Thang *et al.* es que, cuando más de un cliente compite por el ancho de banda en un cuello de botella, el algoritmo actúa de manera egoísta, lo que hace que se reduzca la QoE percibida por el cliente.

3.3.1.4. Thang *et al.* 2013

Thang *et al.* [Thang13] proponen el uso de un método para calcular el throughput del cliente utilizando *EWMA* y el *RTT* de los segmentos multimedia, con el fin de suavizar las medidas del throughput que obtienen de usar pesos fijos en el método *EWMA*. Para calcular el *RTT*, propone usar la formula definida para *RTT* dentro del control de tráfico de *TCP*, la cual esta basado en el método matemático *EWMA*.

También realizan un análisis sobre dos métodos propuestos para estimar la tasa de bits correcta para los segmentos, en el primer método se realiza en el mismo intervalo (Inter-stream estimation) y el segundo método (intra-stream estimation) intenta calcular la tasa de bits para un segmento futuro dentro de la reproducción.

El problema que presenta el algoritmo propuesto por Thang *et al.* es que, las mediciones de *RTT* en ambientes móviles [Romirer09] puede presentar medidas ambiguas y, por lo tanto, invalidar los resultados de la muestra de *RTT*.

3.3.1.5. Kim *et al.*

Kim *et al.* [Kim13] propusieron el uso de un Índice de Fluctuación Dinámico (IFD) junto al método matemático *EWMA* para verificar si las condiciones de la red han cambiado en un rango predecible. Si el valor absoluto de *IFD* es mayor que un límite de error e , la condición de la red se desvía de la tendencia estimada y cambia su valor, lo cual logra que los pesos exponenciales para *EWMA* se adapten dinámicamente a las condiciones de la red. Kim *et al.* utilizaron medidas de red inalámbrica Wi-Fi (IEEE 802.11g) y de red móvil WiMAX (IEEE 802.16e). Estas mediciones fueron utilizadas dentro de un emulador de las condiciones de la red utilizando el software NetDisturb emulator. El vídeo utilizado contenía 3 calidades con diferente resolución para cada calidad.

El algoritmo presenta el problema de que el índice de fluctuación que calcula es un mecanismo lento que se actualiza cada vez que se descarga un segmento completo, lo cual puede ocasionar que se subestime o sobreestime el ancho de banda actual de la red móvil.

3.3.1.6. Jiang *et al.*

Jiang *et al.* [Jiang14] identificaron problemas cuando varios reproductores DASH comerciales comparten un enlace que actúa como un “cuello de botella” [Paxson99], con respecto a tres parámetros: equidad, eficiencia y estabilidad. Como solución propusieron el método “*Fair, Efficient, Stable, adaptive (FESTIVE)*”.

El método *FESTIVE* implementa un conjunto de técnicas que sistemáticamente guían las compensaciones entre la estabilidad, la equidad y la eficiencia, con el fin de proponer un marco general para la adaptación robusta de contenido multimedia. Este método basa la estimación del throughput en la media armónica de los últimos 20 segundos de contenido multimedia. El módulo de actualización del bitrate recibe estimaciones de rendimiento del estimador de ancho de banda y calcula una tasa de bits de referencia. El planificador aleatorio programa el siguiente segmento de contenido multimedia para ser descargado inmediatamente, si el buffer de reproducción es menor al tamaño del buffer fijado, en caso contrario el siguiente segmento se programa con un retardo aleatorio.

El método *FESTIVE* fue implementado utilizando *OSMF* de Adobe y se comparó con varios reproductores comerciales reales y emulados en una variedad de escenarios que varían el ancho de banda y la cantidad de usuarios. Sin embargo, en entornos inalámbricos, hay fluctuaciones del ancho de banda a corto plazo que pueden hacer que este método sobreestime o subestime el ancho de banda disponible lo que ocasionaría paradas en la reproducción del contenido multimedia.

3.3.1.7. Ramamurthi *et al.*

Ramamurthi *et al.* [Ramamurthi13] [Ramamurthi15] propusieron un enfoque de adaptación de la tasa del vídeo consciente de la capa física especialmente diseñado para la reproducción de vídeo en redes móviles 4G. Este algoritmo se basa en estimar el ancho de banda utilizando la información que se puede obtener de la capa física y utilizar esta información como un complemento a las estimaciones de la capa de aplicación. Para esto, se utiliza el cálculo del throughput instantáneo del último segmento descargado y se compara con el goodput (medida que considera únicamente los datos útiles transmitidos) obtenido de la capa física de la red.

El análisis de este enfoque se realizó utilizando un simulador de la red LTE en la plataforma Matlab con diferentes tipos y calidades de vídeo utilizando la métrica de calidad “*Peak Signal-to-Noise Ratio (PSNR)*” sobre el vídeo. El problema que presenta este tipo de estimaciones es que los datos de la capa física incluyen información de otros servicios que están funcionando en el dispositivo del usuario.

3.3.1.8. Sun *et al.*

Sun *et al.* [Sun16] desarrollaron el método “*Cross Session Stateful Predictor (CS2P)*” para mejorar la selección y adaptación del bitrate en reproductores de contenido multimedia

usando un algoritmo de predicción del throughput basado en un conjunto de datos recolectados anteriormente. Este método utilizó un conjunto de datos proporcionado por iQIYI (plataforma de vídeo bajo demanda China), que consistió en 20 millones de sesiones que cubren 3 millones de direcciones IP de clientes únicas y 18 direcciones IP de servidores.

Concluyeron que las sesiones de vídeo bajo demanda comparten características clave similares (por ejemplo, ISP, región geográfica, etc.) y tienen valores de throughput y patrones dinámicos similares, con lo cual el comportamiento de las sesiones de streaming presentan poca variabilidad en el throughput dentro de una sesión multimedia.

La aportación del artículo es el desarrollo de *CS2P*, un sistema de predicción de throughput que usa un enfoque basado en aprender datos de un grupo (“cluster”) de sesiones similares, predice el throughput inicial y modela la evolución del estado del throughput de la sesión media con un modelo oculto de Markov.

3.3.2. Algoritmos basados en el buffer

Los algoritmos basados en el buffer utilizan la información de la ocupación del buffer para intentar maximizar la elección de la tasa de bits del contenido multimedia. Estos algoritmos usualmente intentan evitar eventos en los que la reproducción se detiene debido al agotamiento del buffer.

3.3.2.1. Miller *et al.*

Miller *et al.* [Miller12] propuso un algoritmo que usa tres niveles de umbral para el buffer de reproducción, de modo que $0 < Buffer - min < Buffer - low < Buffer - high$. El objetivo $Buffer - target$ está entre $Buffer - low$ y $Buffer - high$, y el intervalo óptimo $Buffer - optimum$ es el centro del intervalo objetivo. El algoritmo mantiene el nivel de buffer cerca de $Buffer - optimum$. Esto permite al diseñador controlar explícitamente la compensación entre la variación en la ocupación del buffer y fluctuaciones en el bitrate del vídeo en respuesta a la variación del rendimiento de *TCP* mediante el control de los umbrales $Buffer - low$ y $Buffer - high$.

Este prototipo fue evaluado en escenarios reales, comprobándose que funciona bien en situaciones de baja capacidad en el ancho de banda de la red. También se ejecutó el algoritmo en un banco de pruebas en donde se restringió la capacidad del ancho de banda y se realizaron experimentos en un entorno WiFi doméstico. De acuerdo con los autores, el algoritmo logró exhibir un comportamiento estable y justo cuando varios clientes comparten una red de acceso común.

Uno de los problemas que presenta este algoritmo es que para las pruebas realizadas cuentan con 20 diferentes calidades del vídeo que van desde 45 hasta los 4220 kbit/s, pero no se analiza el impacto de esto en la *QoE*.

3.3.2.2. Claey's *et al.*

Claey's *et al.* [Claey's13] propuso un algoritmo para controlar el buffer basado en “*Q-Learning*”, que es un algoritmo de aprendizaje de refuerzo sin modelo. El objetivo de *Q-Learning* es aprender una política que le dice al agente qué acción debe tomar bajo qué circunstancia.

El algoritmo propuesto por Claey's *et al.* utiliza el algoritmo *Q-Learning* para aprender dinámicamente el comportamiento óptimo correspondiente al estado actual de la red. Esto se logra utilizando una función de recompensa que se enfoca en los diferentes aspectos que conforman la *QoE* del servicio de distribución multimedia. Para evaluar el algoritmo diseñado por Claey's *et al.* se han realizado simulaciones en NS-3.

El problema que presenta este algoritmo es que no considera los cambios de calidad y los eventos de parada.

3.3.2.3. Huang *et al.*

Huang *et al.* [Huang15] fueron los primeros en proponer, evaluar experimentalmente y probar un algoritmo de adaptación del bitrate basado únicamente en el estado del buffer. Demostraron los desafíos en la estimación de la capacidad en servicios comerciales de vídeo bajo demanda causados por las fluctuaciones en las redes “*Best Effort*”. Este algoritmo utiliza el estado del buffer únicamente, es decir, selecciona una tasa de bits de vídeo puramente basada en la ocupación de buffer actual durante la fase de reproducción estable y se realiza una estimación del throughput cuando el buffer es demasiado bajo (por ejemplo, en la fase inicial de la reproducción cuando no existe suficiente información para tomar la decisión).

El algoritmo fue implementado en los clientes de Netflix, con lo cual los autores demostraron que este enfoque reduce la tasa de rebuffer entre un 10 y 20 % en comparación con el algoritmo original implementado en los clientes de Netflix, mientras que ofrece una tasa de vídeo promedio similar y una tasa de vídeo más alta en la fase de reproducción estable.

3.3.2.4. *García et al.*

García *et al.* [García15] propuso utilizar una técnica para la toma de decisiones bajo incertidumbre denominada “*Stochastic Dynamic Programming (SDP)*”, en la cual la solicitud de nuevos segmentos se rige por políticas a las que se asigna un parámetro de control (nivel de calidad del bitrate del vídeo) en cada estado posible del reproductor. En el trabajo propuesto por los autores, se propone utilizar un modelo de cadenas de Markov para simular el throughput disponible para el cliente.

En el trabajo presentado por estos autores, su algoritmo se compara con el algoritmo creado por [Miller12] y el reproductor de Apple HLS. Los autores utilizan el trabajo de [Claeys13] en el que se definen métricas objetivas para calcular la *QoE* que percibe el usuario.

Un inconveniente que presenta el algoritmo *SDP* es que utilizan 14 calidades del vídeo distribuidas entre 0.1 y 4.5 Mbps, con lo que se presentan pocos eventos de parada, pero no se analiza el impacto de estas calidades en la *QoE* que percibe el usuario.

3.3.2.5. *Zhou et al. y Bokani et al.*

Zhou *et al.* [Zhou16] propuso un enfoque de adaptación del bitrate basado en “*Markov Decision Process (MDP)*” para el servicio de streaming dinámico basado en *HTTP*. El algoritmo toma en cuenta varios parámetros del sistema que tienen un impacto crítico en la calidad visual. Estos parámetros incluyen la calidad de la reproducción del vídeo, la frecuencia y amplitud de cambios en el bitrate del vídeo, el desbordamiento del buffer y la ocupación del buffer.

El algoritmo propone una función de recompensa efectiva en términos de los parámetros del sistema considerados para la decisión de cambio de bitrate del vídeo, para maximizar la *QoE* del usuario. Ya que la implementación del algoritmo requiere de una complejidad computacional alta, los autores Zhou *et al.* también han propuesto un algoritmo codicioso sub-óptimo de baja complejidad para soportar la reproducción de vídeo en tiempo real utilizando *DASH*.

Bokani *et al.* [Bokani15] también propuso un algoritmo que utiliza *Markov Decision Process (MDP)*. El objetivo principal del algoritmo es el de obtener la política óptima que maximice la toma de la decisión correcta. El algoritmo funciona de manera óptima cuando tiene el conocimiento previo del modelo del ancho de banda. Además, Bokani *et al.* también proponen un modelo basado en *MDP* que no necesita conocer previamente el modelo del ancho de banda, y utiliza la técnica *Q-Learning* para establecer las probabilidades de

transición que requiere el modelo *MDP*.

Utilizando contenido multimedia que está disponible públicamente [Lederer12], los autores han encontrado que el método *MDP* reduce entre un 4 y un 15 % la probabilidad de que la reproducción del contenido multimedia sufra un evento de parada.

El problema de estos algoritmos es que son de una complejidad computacional alta, lo cual complica su implementación a gran escala para soportar servicios de distribución de contenido multimedia.

3.3.2.6. Beben *et al.*

Beben *et al.* [Beben16] propusieron “*Adaptation and Buffer Management Algorithm plus (ABMA+)*” [Wiśniewski15] como mejora a *ABMA*. *ABMA+* selecciona las tasas de representación de vídeo en función de la probabilidad pronosticada de que sucederá un evento de parada de la reproducción por la inanición del buffer. El algoritmo continuamente estima el tiempo de descarga del segmento y utiliza un modelo del buffer de reproducción pre-calculado para seleccionar la representación de vídeo máxima que garantice la reproducción sin inconvenientes. De acuerdo con los autores, el uso de un modelo del buffer pre-calculado minimiza los costos de cómputo y simplifica el despliegue en diferentes terminales.

Los autores validaron *ABMA+* con simulaciones y ensayos experimentales utilizando el plugin DASH del reproductor VLC [Batalla16]. También compararon su algoritmo con otros enfoques basados en buffer y throughput. Según sus resultados, confirmaron que *ABMA+* ajusta de forma eficiente las representaciones de vídeo a condiciones de red variables, lo que minimiza la congelación de vídeo y evita los frecuentes cambios de representación.

Uno de los problemas que presenta este algoritmo es que para las pruebas realizadas cuentan con múltiples calidades del vídeo que van desde 45k kbps hasta los 15 mbps. Sin embargo, no se analiza cómo esa diversidad de calidades afecta la QoE que perciben los usuarios.

3.3.2.7. Spiteri *et al.*

Spiteri *et al.* [Spiteri16] formularon ABR como un problema de maximización de la utilidad y diseñó un algoritmo de control en línea, utilizando las técnicas de optimización de Lyapunov para minimizar el rebuffering y maximizar la calidad del vídeo. Su algoritmo “Buffer Occupancy based Lyapunov Algorithm (BOLA)” logró una utilidad promedio en el tiempo que está dentro de un factor aditivo del valor óptimo. Este algoritmo también es un

algoritmo basado en buffer que no requiere ninguna estimación o predicción del throughput que obtiene el cliente de la red.

El algoritmo *BOLA* incorpora dos métricas clave para medir la *QoE* de los usuarios: la tasa de bits de vídeo promedio y la duración del rebuffering. Un aumento en la velocidad de bits promedio aumenta la utilidad, mientras que el rebuffering lo disminuye. La ventaja de este marco es que proporciona una garantía teórica sobre la utilidad lograda y proporciona un control explícito para que los proveedores de vídeo establezcan la importancia relativa de una alta calidad de vídeo en relación con la probabilidad de rebuffering.

Los autores evaluaron *BOLA* utilizando 12 vídeos de prueba proporcionados por DASH-Industry Forum (DASH-IF) en 85 perfiles de ancho de banda móvil 3G disponibles públicamente. Compararon los resultados con modificaciones del algoritmo *BOLA* donde se alteran los parámetros de utilidad para evitar oscilaciones y otro para maximizar la utilidad. Los autores también compararon *BOLA* con *ELASTIC* [Cicco13] y *PANDA* [Li14] (algoritmos que se analizan más adelante), y concluyeron que *BOLA* es superior en términos de utilidad y consistencia. Este algoritmo es una de las opciones de adaptación *DASH* en el cliente *dash.js* desde la versión 2.0.

Un inconveniente que presenta *BOLA* es que los autores, en un nuevo artículo [Spiteri19], analizaron que el algoritmo no responde lo suficientemente rápido a los cambios del throughput de la red, aunque funciona de manera óptima cuando se tienen condiciones de buffer alto (más de 20 segundos).

3.3.2.8. Martín *et al.*

Martín *et al.* [Martín16a] propusieron un algoritmo basado en una técnica de aprendizaje de refuerzo llamada *Q-Learning*, que usa una función de recompensa. Esta función permite que un sistema dinámico aprenda cuál es la acción más apropiada para la siguiente etapa dependiendo de las condiciones específicas en cada momento. La función toma en cuenta las calidades solicitadas, los cambios de calidad y el número de eventos de parada.

Martín *et al.* [Martín16b] propusieron una mejora al algoritmo anterior en cuanto a la función utilizada para medir los eventos de parada, lo cual hace que el algoritmo mejore las prestaciones que se presentaban en el primer artículo. En este trabajo los autores proponen utilizar un modelo de cadenas de Markov para simular el throughput disponible para el cliente.

Los autores utilizan el trabajo de [Claeys13] en el que se definen métricas objetivas para calcular la *QoE* que percibe el usuario. El problema con este tipo de propuestas es que

requieren de una fase de entrenamiento, lo cual complica la implementación del algoritmo en un servicio de transmisión multimedia real.

3.3.3. Algoritmos basados en teoría de control o híbridos

Esta clase de algoritmos se basan en utilizar las estimaciones de throughput, así como la ocupación del buffer, como indicadores para la toma de decisiones, usando teoría de control o ecuaciones óptimas de control estocástico para maximizar la selección de la tasa óptima de representación.

3.3.3.1. De Cicco *et al.* 2013

El patrón de tráfico ON-OFF [Huang12] [Akhshabi12] del estado estacionario de *DASH* puede causar una utilización injusta del ancho de banda, la infrautilización del ancho de banda del servidor y frecuentes interrupciones en el throughput en presencia de otros flujos de vídeo o flujos de *TCP*. Por lo tanto, De Cicco *et al.* [Cicco13] propusieron el algoritmo “*Feedback Linearization Adaptive Streaming Controller (ELASTIC)*”, que utiliza una teoría de control de retroalimentación que no genera un patrón de tráfico ON-OFF.

A diferencia de los enfoques convencionales que usan dos controladores (uno para regular la tasa promedio de velocidad del bitrate, otro para regular el nivel del buffer), *ELASTIC* solo usa un controlador que calcula la tasa promedio de velocidad del bitrate para guiar el buffer de reproducción a un punto fijo. Este algoritmo utiliza el método matemático de la media armónica de los últimos 5 segmentos de vídeo para realizar la estimación del throughput.

ELASTIC genera un patrón de tráfico que es idéntico a cualquier flujo *TCP* de larga duración. Usando un banco de pruebas controlado, los autores investigaron si *ELASTIC* puede utilizar completamente el “cuello de botella” [Paxson99], compartir el “cuello de botella” de manera justa y obtener una participación equitativa cuando los flujos de *TCP* comparten el “cuello de botella” con los flujos de vídeo. Llegaron a la conclusión de que *ELASTIC* logra una gran equidad cuando coexiste con flujos *TCP*.

Los autores también compararon *ELASTIC* con *PANDA* [Li14] y *FESTIVE* [Jiang14] en escenarios en los que múltiples flujos *DASH* son comparados con y sin flujos de *TCP* de larga duración. Descubrieron que tanto *ELASTIC* como *PANDA* y *FESTIVE*, logran obtener una repartición equitativa del ancho de banda cuando solo hay transmisiones *DASH*, pero no cuando compiten con otro tipo de flujos *TCP*. *ELASTIC* puede lograr una repartición equitativa del ancho de banda mientras proporciona vídeo de alta calidad con poca

oscilación del bitrate del vídeo.

Usar la media armónica de los últimos 5 segmentos hace que el método funcione de manera óptima cuando se realizan mediciones de ancho de banda estables porque puede discriminar algunas de las mediciones atípicas. Sin embargo, en entornos inalámbricos, hay fluctuaciones de ancho de banda a corto plazo que pueden hacer que este método sobreestime o subestime el ancho de banda disponible.

3.3.3.2. Tian *et al.* y Zhou *et al.* 2013

El trabajo descrito en [Tian16] y [Zhou13] tiene como objetivo controlar la ocupación del buffer de reproducción, es decir, mantenerlo en un nivel de referencia y usar la diferencia entre el nivel de buffer actual y el nivel de referencia para controlar la adaptación del cliente *DASH*. Ambas investigaciones basan su algoritmo en el uso de controladores “*Proportional Integral Derivative (PID)*”, el cual es un mecanismo de retroalimentación de bucle de control ampliamente utilizado en aplicaciones que requieren un control modulado continuamente.

El algoritmo utiliza las ecuaciones del throughput del vídeo a partir de la derivada de la diferencia entre el tamaño del buffer y el throughput estimado, que a su vez determina el nivel de ocupación del buffer en combinación con el throughput real de *TCP*, cerrando así el bucle de control.

Sin embargo, mantener el buffer de reproducción en un nivel determinado de referencia no es la mejor forma de guiar el bucle de control. Desde el punto de vista de los usuarios, es más importante recibir la calidad de vídeo más alta (y más consistente) sostenible a la vez que se reducen las fluctuaciones del bitrate del vídeo. Los modelos descritos aquí no tienen en cuenta las fluctuaciones del bitrate del vídeo.

3.3.3.3. Mansy *et al.*

Mansy *et al.* [Mansy13] demostraron y cuantificaron el efecto bufferbloat [Gettys11] (se refiere a memorias intermedias grandes instaladas en la mayoría de los equipos de comunicación, que agregan *jitter* al tráfico cursado por la red.) en los flujos *DASH* junto con otras aplicaciones que comparten el mismo cuello de botella en una red doméstica. Para mitigar este problema desarrollaron “*Smooth Adaptive Bit Rate (SABRE)*” en el reproductor *VLC*. Mostraron que *SABRE* puede reducir los retrasos en las colas y reducir significativamente el efecto mencionado anteriormente.

SABRE ajusta dinámicamente la ventana de recepción (*rwnd*) de *TCP* en un cliente

DASH desde la capa de aplicación para que el tamaño de ráfaga del servidor al cliente se reduzca efectivamente al tamaño de cola promedio del router doméstico. Las técnicas clave son: canalización de *HTTP*, control de la velocidad de descarga de segmentos de vídeo y modo de operación de recarga o recarga dual.

La canalización de *HTTP* se utiliza para enviar múltiples solicitudes *HTTP* GET simultáneamente para mantener el buffer de recepción del cliente siempre lleno, por lo que no anuncia una ventana de recepción grande. *SABRE* calcula el número de solicitudes *HTTP* que el cliente debe canalizar utilizando información sobre el tamaño real del buffer de recepción, el tamaño del segmento de vídeo y la tasa de bits del segmento.

El algoritmo controla la velocidad de descarga en la aplicación al calcular la velocidad de la llamada en el *socket* del receptor de modo que la velocidad de descarga lograda en cualquier punto no sea mucho más alta que la tasa de bits de vídeo. El fundamento es controlar la velocidad a la que la aplicación lee desde el buffer de recepción, controlando así el crecimiento de la ventana de recepción. Como no es necesario leer datos del buffer en el *socket* de recepción a una velocidad superior a la velocidad del bitrate del vídeo, *SABRE* distribuye las llamadas *recv* del protocolo *TCP* uniformemente sobre el tamaño del segmento (en segundos).

El algoritmo *SABRE* utiliza el buffer de reproducción para el cambio del bitrate del vídeo hacia arriba o hacia abajo, operando en un modo de recarga y un modo de reducción. Cuando el nivel del buffer de reproducción cae por debajo de un umbral, opera en el modo de recuperación, una vez que excede otro umbral, ingresa en el modo de reducción para evitar el sobrellenado del buffer de vídeo.

Los autores también realizaron experimentos en los que varios clientes (*SABRE* y reproductores tradicionales) compiten por el ancho de banda y demostraron que *SABRE* puede coexistir bien con otros reproductores sin tener penalizaciones por el rendimiento.

3.3.3.4. De Cicco *et al.* 2014

De Cicco *et al.* [Cicco14] propusieron un algoritmo que utiliza teoría de control modelado como una máquina de estados finita para la toma de decisiones del algoritmo. El algoritmo propone dos métodos, uno para controlar la longitud del buffer del cliente y otro método para controlar el planificador para la petición de nuevos segmentos de contenido multimedia.

Los autores utilizan un entorno de pruebas emulado donde se utiliza las herramientas de Linux Traffic Control (TC) y NetEM para emular las condiciones de la red. Se utilizan

valores aleatorios dentro de 400kb/s a 4000 kb/s para emular la red de acceso y se realizan transferencias de datos simultáneas utilizando el software llamado *iperf*.

Un problema que presenta el algoritmo propuesto por De Cicco *et al.* es que, debido a que el algoritmo actúa de una forma conservadora, es posible que se obtengan bajas calidades del vídeo y que en caso de que ocurra un evento de parada en la reproducción el algoritmo tarde un tiempo largo en reaccionar.

3.3.3.5. Yin *et al.*

Yin *et al.* en las investigaciones [Yin14] y [Yin15], propusieron un algoritmo basado en un Modelo Predictivo de Control (MPC) que puede combinar de forma óptima las señales de respuesta del buffer de vídeo y de la tasa de throughput y formuló el problema de selección de velocidad como un problema de control óptimo estocástico. *MPC* intenta predecir variables de entorno clave sobre un horizonte de anticipación de movimientos y resolver un problema de optimización exacto basado en la predicción.

El algoritmo *MPC* predice el rendimiento esperado para los siguientes segmentos y usa esta medida para tomar una decisión de tasa de bits óptima para la maximización de la *QoE*. Yin *et al.* han implementado variantes del algoritmo *MPC* (*FastMPC*, *RobustMPC*) utilizando el reproductor *dash.js*, y que validan utilizando emulaciones basadas en trazas de tráfico reales. La teoría tiene en cuenta múltiples restricciones para derivar la tasa de bits óptima, y proporciona un marco para comparar el rendimiento emulado o real del algoritmo con el rendimiento “ideal” (el mejor posible) del algoritmo.

Yin *et al.* también proponen utilizar una métrica de *QoE* integral, que es una combinación ponderada de la calidad de vídeo promedio, las variaciones de calidad promedio, el tiempo de rebuffering y el retraso de inicio. El algoritmo *MPC* utiliza una optimización off-line para diferentes escenarios, y una sección en línea que usa búsquedas de tablas para apuntar a las soluciones calculadas.

En este algoritmo, la predicción del ancho de banda se utiliza para la selección de los segmentos de vídeo, pero su rendimiento depende en gran medida de la precisión de la predicción, que a su vez se basa en el cálculo de optimización off-line exhaustiva realizada fuera del cliente. Los autores compararon *MPC* con otros algoritmos (basados en throughput, basados en buffer, *dash.js* y *FESTIVE* [Jiang14]) y confirmaron las ventajas con un aumento insignificante en los requisitos de computación y memoria (dentro del cliente).

El problema que presenta el algoritmo *MPC* es que es necesario entrenar al algoritmo, lo que hace que los autores hayan desarrollado varios algoritmos dependiendo del escenario

en el que se va a implementar.

3.3.3.6. Li *et al.*

Li *et al.* [Li14] utilizaron experimentos de banco de pruebas para demostrar las limitaciones fundamentales de confiar únicamente en estimaciones de rendimiento. Demostraron que cuando varios clientes DASH compiten en un cuello de botella en la red de acceso, la naturaleza discreta de las velocidades de bits de vídeo resulta en una dificultad para que un cliente perciba correctamente su ancho de banda de cuota justa. Esto lleva a oscilaciones de velocidad de bits de vídeo y baja QoE. Por lo tanto, propusieron un mecanismo “*Probe and Adapt (PANDA)*” para la adaptación de la tasa de bits que es similar al control de la congestión *TCP*.

PANDA prueba la red estableciendo una tasa de datos promedio objetivo. Esta tasa de datos se usa posteriormente para determinar la siguiente tasa de bits del segmento de vídeo y el intervalo de solicitud posterior. El algoritmo *PANDA* utiliza para realizar las estimaciones del throughput de la red el mecanismo *EWMA* con un factor de peso fijo $\lambda = 0.2$, lo cual puede ocasionar estimaciones incorrectas del throughput.

Su mecanismo de sondeo AIMD es similar al control de congestión *TCP*. La diferencia fundamental es que el control de congestión *TCP* detecta congestión cuando hay pérdidas de paquetes o el aumento en RTT (Round Trip Time), mientras que *PANDA* detecta congestión con la reducción del rendimiento. Está propiedad garantiza que *PANDA* utilice de manera eficiente el ancho de banda de la red, y en presencia de múltiples clientes, el ancho de banda para cada cliente finalmente converge a un nivel de reparto equitativo.

PANDA planifica la próxima solicitud considerando la tasa objetivo promedio y el nivel de buffer. Presenta un algoritmo de adaptación basado en el llenado del buffer que resuelve el problema de optimización de la selección de calidad. El algoritmo *PANDA* fue analizado realizando experimentos con bancos de pruebas y estos mostraron que *PANDA* es capaz de reducir la inestabilidad del vídeo en más del 75 % en comparación con otros algoritmos convencionales sin eventos de “*rebuffering*”.

Un problema que presenta el algoritmo *PANDA* es que es necesario afinar seis parámetros (*probing additive increase bitrate, probing convergence rate, smoothing convergence rate, client buffer convergence rate, quantization margin and multiplicative safety margin*) dependiendo del escenario en el que se va a implementar el cliente, lo que complica el despliegue de los clientes para cada tipo de escenario donde se distribuirá el servicio de contenido multimedia.

3.3.3.7. Miller *et al.*

Miller *et al.* [Miller15] proponen un sistema eficiente que admite un gran número de sesiones de transmisión de contenido multimedia en una red de acceso inalámbrico. Esto se realiza considerando conjuntamente los problemas de la adaptación de la calidad en un servicio de transmisión de contenido multimedia y una red de acceso inalámbrica con muchos clientes (densa), utilizando la técnica de controlador proporcional integral y derivativo (PID), en donde el objetivo es estabilizar un sistema dinámico alrededor de un objetivo propuesto.

Miller *et al.* también implementan una técnica para evitar el error acumulado que pueden presentar los sistemas *PID* llamada “anti-windup” (esquema que evita que la salida del controlador se desvíe), la cual les permitió descentralizar el mecanismo diseñado para resolver los problemas en la reproducción que afectan la *QoE* que perciben los usuarios en la reproducción multimedia.

Para avalar el algoritmo de Miller *et al.* proponen tres tipos de experimento, en el primer experimento se evalúan usuarios de largo plazo sin rotación de estos (no churn), en el segundo, usuarios de corto plazo con alta rotación de usuarios (with churn) y por último, una combinación de usuarios de largo y corto plazo.

El problema que presenta el algoritmo de Miller *et al.* es que no cuenta con un mecanismo en bucle de control externo para el nivel del buffer, que además, se pueda ajustar en función de las estadísticas de calidad del enlace individual del usuario.

3.3.3.8. Le *et al.*

Le *et al.* [Le16] propusieron un nuevo algoritmo de adaptación llamado “*Adaptive VBR alGorithm (AVG)*”, el cual utiliza un método para suavizar la estimación del throughput utilizando *EWMA* y un método para controlar el estado del buffer que basa su funcionamiento en umbrales de la cantidad mínima y máxima (en segundos) de contenido del buffer. El método de buffer también implementa una función lógica que compara el throughput detectado por *EWMA* y el bitrate de enésimo segmento.

El algoritmo *AVG* fue comparado con un algoritmo que basa su funcionamiento utilizando solo medidas del throughput [Thang13] y con otro algoritmo que utiliza solo el control del buffer [Zhou14] como mecanismo de adaptación. El banco de pruebas usado para evaluar el algoritmo *AVG* utiliza perfiles de tráfico [Müller12] disponibles al público.

Un problema que presenta el algoritmo *AVG* es que en el artículo presentan tres versiones de *AVG* y en la mayoría de las pruebas el algoritmo *AVG* presenta peores métricas del

bitrate promedio que los algoritmos con los que se compara, lo cual afecta la QoE percibida por el usuario para el algoritmo *AVG*.

3.3.3.9. Wang *et al.*

Wang *et al.* [Wang16] propusieron “*Spectrum-based Quality Adaptation for DASH (SQUAD)*”, que utiliza tanto la retroalimentación a través de la estimación del throughput como el nivel del buffer para garantizar una *QoE* alta. Los autores pretenden maximizar la calidad promedio y minimizar el número de cambios de calidad (bitrate switches). Utilizaron el “espectro” (una medida centralizada para la variación de las tasas del bitrate en torno a la calidad promedio) como una métrica para capturar la *QoE*, y afirmaron que *SQUAD* resuelve las discrepancias de la estimación del ancho de banda de la capa de aplicación y el protocolo de transporte subyacente.

El algoritmo *SQUAD* fue probado con múltiples reproductores DASH en un banco de pruebas controlado y en un entorno de Internet transatlántico. En estas pruebas, se vio que *SQUAD* proporcionaba una *QoE* significativamente mejor en términos del número de cambios de calidad.

Un inconveniente que presenta el algoritmo *SQUAD* es que realizaron las pruebas con 14 calidades de vídeo que van desde 0.9 a 4.21 Mbps, lo que podría ocasionar métricas pobres de *QoE* según la calidad del vídeo descargado.

3.3.3.10. Zhou *et al.* 2017

Zhou *et al.* [Zhou17] estudiaron los problemas de adaptación en clientes *DASH* cuando estos compiten por el ancho de banda sobre enlaces con cuello de botella, especialmente en lo que se refiere a la eficiencia, estabilidad y equidad entre varios clientes. Los autores proponen el uso de un algoritmo basado en el control de congestión del protocolo *HTTP* llamado “*Logarithmic Increase Multiplicative Decrease (LIMD)*” como esquema de estimación del throughput para maximizar la selección de la tasa del siguiente segmento y que el ancho de banda se reparta justamente entre los clientes.

Los autores también proponen un esquema para el manejo del nivel del buffer estableciendo un doble umbral para garantizar la reproducción continua de vídeo con una alta calidad visual. Este esquema está basado en un control de probabilidades que controla al algoritmo de tasa adaptativa. Según sus resultados el algoritmo “*Throughput-Friendly DASH (TFDASH)*” mejora factores clave que incluyen la ocupación de buffer, la calidad de reproducción de vídeo y disminuye la frecuencia en la conmutación de las calidades del vídeo

cuando se compara con algoritmos actuales como *PANDA* [Li14] y *FESTIVE* [Jiang14].

El problema que presenta el algoritmo de Zhou *et al.* es que, al comparar dos clientes con el algoritmo *TFDASH* se puede observar que, a pesar de que en los resultados exponen que presenta una buena métrica de equidad (fairness) en los resultados se observa que un cliente consume más ancho de banda que el otro la mayor parte del tiempo.

3.3.3.11. Qin *et al.*

Qin *et al.* [Qin17] propusieron un algoritmo llamado “PID-control based ABR streaming (PIA)”, que se basa en un mecanismo de control proporcional integral y derivativo (PID). El mecanismo de control funciona supervisando continuamente un “valor de error” definido como la diferencia entre un punto de ajuste y la variable medida en el proceso.

En el algoritmo *PIA*, el controlador central define una política que hace que el sistema funcione como un sistema de control de circuito cerrado lineal, lo cual hace que sea sencillo de controlar y analizar. El algoritmo *PIA* fue implementado utilizando el reproductor *dash.js* en su versión 2.0, emulando un escenario de una red móvil LTE.

El algoritmo fue comparado con otros algoritmos actuales, estos algoritmos son *BufferBB* [Huang15] y *MPC* [Yin15]. Según sus resultados, el algoritmo *PIA* supera a los algoritmos de adaptación mencionados con anterioridad al equilibrar mejor las métricas claves de *QoE* en los clientes de reproducción para lograr una disminución del 40 % de cambios en el bitrate del vídeo y del 85 % de eventos de parada de reproducción.

Un problema que presenta el algoritmo propuesto por Qin *et al.* es que es necesario afinar seis parámetros (diferentes al algoritmo *PANDA*) dependiendo del escenario en el que se va a implementar el cliente, lo que complica el despliegue de los clientes para cada tipo de escenario donde se planea ofrecer el servicio de distribución del contenido multimedia.

3.3.3.12. Zahran *et al.*

Zahran *et al.* [Zahran18] propusieron el algoritmo *ARBITER+* utilizando como base un algoritmo creado anteriormente por ellos llamado “Adaptive Rate-Based Intelligent HTTP Streaming Algorithm (*ARBITER*)” [Zahran16]. El nuevo algoritmo propone el uso de un estimador del bitrate adaptable, un muestreo del throughput híbrido, conmutación del bitrate controlada y seguimiento de la tasa del vídeo real a corto plazo. Estos componentes están diseñados para garantizar una medición alta de la *QoE* para el usuario a la vez que se adapta a la variabilidad tanto del throughput de la red como de la tasa de reproducción del vídeo.

El algoritmo fue evaluado frente a otros algoritmos actuales como son: *MPC* [Yin15], *BufferBB* [Huang15] y *ELASTIC* [Cicco13]. El algoritmo *PIA* fue implementado en un cliente Android basado en la librería ExoPlayer de Google y ejecutado en un escenario con un servidor y un acceso inalámbrico (AP) simulando una red móvil 3G utilizando herramientas de Linux. También se simuló utilizando MATLAB un escenario donde el cliente va en un vehículo a una velocidad de 30 Km/h.

Un problema que presenta el algoritmo *ARBITER+* es que es necesario ajustar 8 parámetros dependiendo del escenario en donde se va a desplegar reproductor para el servicio de distribución multimedia, lo cual, hace que sea necesario tener varios reproductores afinados para los distintos escenarios en los que se implemente el servicio.

3.3.3.13. Huang *et al.*

Huang *et al.* [Huang18] propusieron el algoritmo de adaptación como un problema de optimización estocástico con el objetivo de maximizar la *QoE* del usuario. El algoritmo toma en cuenta factores importantes como son: el tiempo de congelación de la reproducción, los frecuentes cambios en la calidad del vídeo y el nivel del buffer promedio.

El algoritmo también implementa teoría de control, mediante un controlador dinámico basado en el buffer para determinar la tasa de bits del vídeo en cada segmento que se solicita mientras estabiliza el nivel del buffer. Para evaluar el algoritmo los autores realizaron un entorno de simulación donde se compara con otros algoritmos recientes como son *FESTIVE* [Jiang14] y *Smooth Rate Adaptation* [Tian16].

Un inconveniente que presenta este algoritmo es que, según los autores en general su algoritmo solicita la calidad media que es de 1050 kbps (diez calidades de vídeo entre 235 y 4500 kbps) lo cual degrada la calidad percibida por los usuarios, aunque en el artículo muestran que obtiene una medida de *QoE* alta.

Tabla 3.2: Algoritmos DASH y sus características I

Tipo de Algoritmo	Nombre del Algoritmo	Referencia	Tipo de Planificación	Técnica de Estimación	Técnica de buffer	Tipo de Evaluación	Código Disponible	Año
Throughput	EWMA	[Akshabhi11]	–	EWMA	–	Emulado	×	2011
	AIMD	[Liu11]	AIMD	Media Móvil	Umbrales	Simulado	✓	2011
	EWMA-NTD	[Thang12]	–	EWMA y TD	–	Emulado	✓	2012
	EWMA-DRE	[Thang13]	–	EWMA y RTT	–	Emulado	✓	2013
	EWMA-IF	[Kim13]	–	EWMA y FI	–	Emulado	×	2013
	FESTIVE	[Jiang14]	Aleatorio	Media Armónica	–	Emulado	×	2014
	PHY-LA	[Ramamurthi15]	–	Throughput Inst.	–	Simulado	×	2015
	CS2P	[Sun16]	Aleatorio	Predicción	Umbrales	Real	×	2016
	Miller	[Miller12]	–	Throughput Inst.	Umbrales	Emulado	×	2012
	Q-Lernmo	[Claeys13]	–	Throughput Inst.	Q-Learning	Simulado	×	2013
Buffer Level	BufferBB	[Huang15]	–	Throughput Inst.	Umbrales	Real	×	2015
	DASH-SDP	[Garcia15]	–	Throughput Inst.	Fun. de Costo	Simulado	×	2015
	Markov 1	[Bokani15]	–	Throughput Inst.	Markov	Simulado	×	2015
	Markov 2	[Zhou16]	–	Media Móvil	Markov	Emulado	×	2016
	ABMA+	[Beben16]	–	Throughput Inst.	Probabilidad	Emulado	✓	2016
	BOLA	[Spiteri16]	–	Throughput Inst.	Lyapunov	Emulado	✓	2016
	DASH-QL	[Martín16a]	–	Throughput Inst.	Q-Learning	Simulado	×	2016
	ELASTIC	[Cicco13]	–	Media Armónica	Linealización	Emulado	×	2013
	PIDC	[Zhou13]	–	Throughput Inst.	PD	Simulado	×	2013
	SABRE	[Mansy13]	–	EWMA	RED	Emulado	×	2013
Teoría de Control o Híbrido	SCS	[Cicco14]	–	Throughput Inst.	Estado Finito	Emulado	×	2014
	MPC	[Yin14]	–	Media Armónica	Teoría de Control	Simulado	✓	2014
	PANDA	[Li14]	Est. del Buffer	EWMA	AIMD	Emulado	×	2014
	Miller PID	[Miller15]	PID	Media Móvil	PID	Simulado	×	2015
	AVG	[Le16]	–	EWMA	Umbrales	Emulado	×	2016
	SQUAD	[Wang16]	–	Media Móvil	Espectro	Emulado	×	2016
	PIDJ	[Tian16]	–	Machine Learning	PID	Emulado	×	2016
	TFDASH	[Zhou17]	LIMD	EWMA	Umbrales	Emulado	×	2017
	PIA	[Qin17]		Media Armónica	PIA	Simulado	×	2017
	ARBITER+	[Zabran18]	Aleatorio	Media Armónica	Umbrales	Simulado	×	2018
	Buffer State	[Huang18]	Aleatorio	Throughput Inst.	PID	Simulado	×	2018

Tabla 3.3: Algoritmos DASH y sus características II

Tipo de Algoritmo	Nombre del Algoritmo	Escenario Ideal	Análisis QoE	Tipo de vídeo	Nombre del vídeo	Número de Calidades	Tasas de Codificación
Throughput	EWMA	WiFi	×	Animación	Big Buck Bunny	1	2040 kbps
	AIMD	Topología en NS-2	×	–	–	10	de 64 a 1408 kbps
	EWMA-NTD	WiFi	✓	Real	–	16	de 64 a 1024 kbps
	EWMA-DRE	Red móvil 3G	×	Real	Tokio Olympics	10	de 256 a 2560 kbps
	EWMA-IF	WiFi y WiMAX	×	Real	Vídeo Musical	3	de 1000 a 5000 kbps
	FESTIVE	Perfiles propios	×	Real	Propio	8	de 350 a 2750 kbps
	PHY-LA	Red móvil 4G	✓	Real	NBC News	–	de 50,08 a 519,82 kbps
	CS2P	Perfiles de iQIYI	✓	Real	Vídeo de Envivio	5	de 350 a 3000 kbps
	Miller	WiFi ocupada	×	Animación	Big Buck Bunny	20	de 45 a 4220 kbps
	Q-Learn	Perfil Propio	✓	Animación	Big Buck Bunny	7	de 300 a 2436 kbps
Buffer Level	BufferBB	WiFi	×	Real	Netflix	–	de 230 a 5000 kbps
	DASH-SDP	Red móvil 3G	✓	Real	RedBull Playstreet	14	de 100 a 4500 kbps
	Markov 1	Red móvil 3G (ambiente vehicular)	×	Animación	Big Buck Bunny	5	de 200 a 2000 kbps
	Markov 2	Perfiles de Planetlab	×	Animación	Big Buck Bunny	5	de 300 a 3500 kbps
	ABMA+	WiFi	×	Animación	Big Buck Bunny	–	de 45 a 15000 kbps
	BOLA	Perfiles DASH-IF	×	Animación	Big Buck Bunny	10	de 230 a 6000 kbps
	DASH-QL	Perfil propio	✓	Real	Valkama	14	de 123 a 4679 kbps
	ELASTIC	Perfil propio	×	Animación	Elephants Dream	5	de 300 a 3500 kbps
	PIDC	Perfil de Youku	×	–	–	5	de 400 a 2000 kbps
	SABRE	Perfil propio	×	Animación	Big Buck Bunny	6	de 2040 a 4100 kbps
Teoría de Control o Híbrido	SCS	Perfil propio	×	Animación	Elephants Dream	5	de 300 a 3500 kbps
	MPC	Perfil propio	✓	Real	Youtube bitrate levels	5	de 400 a 4500 kbps
	PANDA	Perfil propio	×	–	–	10	de 459 a 11321 kbps
	Miller PID	WiFi denso	×	Animación	Big Buck Bunny	6	de 500 a 4500 kbps
	AVG	Red móvil 3G	×	Real	Sony Demo	6	de 203 a 5180 kbps
	SQUAD	Perfiles propios	×	Animación	Big Buck Bunny	14	de 90 a 4210 kbps
	PIDJ	Perfil de Planetlab	×	–	–	51	de 100 a 5000 kbps
	TFDASH	Perfil de Planetlab	×	Animación	Big Buck Bunny	11	de 235 a 5800 kbps
	PIA	Red móvil 4G	×	Real	Youtube bitrate levels	8	de 200 a 8500 kbps
	ARBITER+	Red móvil 3G y 4G	✓	Animación	iVID DASH dataset	10	de 235 a 4300 kbps
	Buffer State	Red móvil 4G	✓	Animación	Big Buck Bunny	10	de 235 a 4500 kbps

3.3.4. Conclusiones sobre los algoritmos para DASH

Las tablas 3.2 y 3.3 resumen las principales características de cada uno de los algoritmos analizados anteriormente. La tabla 3.2 muestra el nombre del algoritmo, la referencia bibliográfica, el tipo de planificador para los nuevos segmentos, el método de estimación del throughput utilizado, el método de control del buffer, si el algoritmo fue evaluado utilizando simulación, emulación o en un ambiente real, si el código del algoritmo se encuentra en internet y el año en el que fue propuesto el algoritmo. La tabla 3.3 muestra el escenario ideal para implementar el algoritmo, si implementa algún método para medir la QoE que perciben los usuarios, el tipo de vídeo que utilizan en la evaluación del algoritmo propuesto, el nombre del vídeo utilizado, el número de representaciones que utilizan y por último las tasas de bits en las que están codificados los vídeos.

Tras realizar el análisis y crítica de los principales algoritmos de adaptación propuestos por la comunidad científica, se puede concluir que el estándar *DASH* ha emergido como un estándar potente para ofrecer el servicio de distribución multimedia a través de Internet. El componente central de *DASH* es el algoritmo *ABR* que permite la selección del bitrate adecuado a las condiciones ofrecidas por la red las cuales cambian de manera dinámica. Como se menciona en el capítulo 2, los algoritmos *ABR* no forman parte del estándar *DASH*, lo cual permite que investigadores y desarrolladores propongan algoritmos *ABR* para su análisis.

Los algoritmos *ABR* se dividen en tres tipos. Los algoritmos que basan su funcionamiento en la estimación del throughput, la cual se calcula en base en la descarga de los segmentos multimedia anteriores, los algoritmos que funcionan calculando únicamente la ocupación (pasada y presente) del buffer y, por último, los algoritmos que implementan la combinación de la estimación del throughput y el estado del buffer como parámetros de retroalimentación para realizar estimaciones más precisas.

Dentro de los algoritmos que funcionan realizando estimaciones del throughput, se puede concluir que el throughput instantáneo, *EWMA*, la media móvil y la media armónica son los principales métodos matemáticos utilizados. Sin embargo, estos métodos tienen el problema de que es necesario definir una serie de parámetros (tipo de red de acceso) al inicio del despliegue del servicio de distribución multimedia. Lo anterior puede provocar que los algoritmos muestren un desempeño pobre al ser utilizados en otra red diferente a la propuesta por los autores, ya que el algoritmo puede sobreestimar o subestimar las mediciones del ancho de banda que detecta resultando en interrupciones de la reproducción del servicio de distribución multimedia.

Otro problema que presentan los métodos matemáticos que efectúan estimaciones del throughput es que, es necesario establecer la cantidad de historia anterior a tomar en cuenta. Esto se debe a que, estos métodos funcionan mejor o peor dependiendo de la cantidad de historia que se considera para realizar la estimación.

Existen otros métodos de suavizado exponencial, que utilizan un suavizado doble (Holt-Winters) y hasta triple (Winter). El problema con estos métodos, como sucede con el método *EWMA*, es que es necesario establecer el valor de los pesos exponenciales dependiendo del escenario en donde se han de implementar. Por tanto, al igual que con el método *EWMA*, podría existir una sobreestimación o subestimación del ancho de banda.

Para los algoritmos que funcionan calculando el estado tanto anterior como presente del buffer, se puede concluir que por lo general no toman en cuenta los objetivos de *QoE* definidos por los usuarios. Por lo que se hace necesario utilizar algoritmos *ABR* híbridos, los cuales combinan las estimaciones del throughput, así como el estado del buffer.

Estos algoritmos híbridos normalmente utilizan teoría de control para suavizar los errores en la predicción del throughput y pueden maximizar las preferencias de *QoE* de los usuarios. Sin embargo, un problema que presentan es que la implementación de este tipo de algoritmos puede ser computacionalmente complejo, lo cual hace menos probable su implementación en ambientes reales.

Otro problema que se detectó al realizar el análisis de los algoritmos es que muchos fueron evaluados realizando simulaciones, lo cual hace que muchos factores que afectan la *QoE* percibida por el usuario no sean tomados en cuenta, ya que usualmente se simulan bajo condiciones ideales.

También se observó que muchos algoritmos no cuentan con una implementación de un planificador de nuevos segmentos, lo cual puede ocasionar que cuando varios reproductores compiten en un cuello de botella se sincronicen, agotando el ancho de banda para todos, lo cual provocaría eventos de parada y la reducción de la *QoE* percibida por el usuario.

3.4. Modelos de Calidad de la Experiencia (QoE) para DASH

La *QoE* es un concepto holístico, es decir, para medir la *QoE* es necesario entender la comprensión de los requisitos generales de la percepción humana. Por ello es necesario considerar varias disciplinas como son: ciencias cognitivas, psicología social, economía e ingeniería.

De acuerdo con la ITU-T, en el documento P.10/G.100 [ITU-T17b], la *QoE* se define

como “El grado de deleite o molestia del usuario de una aplicación o servicio”.

En el mismo documento [ITU-T17b] se define que la evaluación de la *QoE* es el proceso de medir o estimar la *QoE* para un conjunto de usuarios de una aplicación o un servicio con un procedimiento dedicado, y teniendo en cuenta los factores influyentes (posiblemente controlados, medidos o simplemente recopilados e informados).

El resultado del proceso puede ser un valor escalar, una representación multidimensional de los resultados y/o descripciones verbales. Todas las evaluaciones de *QoE* deben ir acompañadas de la descripción de los factores de influencia que se incluyen. La evaluación de *QoE* puede describirse como integral cuando incluye muchos de los factores específicos, es decir, la mayoría de los factores conocidos.

Existen dos maneras de calcular la *QoE*, lo cual se muestra a continuación:

- La primera es mediante una evaluación subjetiva (perceptual) de la calidad, lo cual requiere una gran cantidad de recursos tanto humanos, como de tiempo y económicos. Es la manera más precisa de calcular la experiencia del usuario.
- La segunda es utilizando una evaluación objetiva (técnica), la cual requiere modelos, mediciones y aparatos sofisticados. Sin embargo, puede ser una medición más o menos precisa dependiendo del número de parámetros utilizados.

Esto ha llevado a que la *QoE* en DASH sea un objetivo complejo ya que sus métricas a menudo son inter-dependientes y tienen relaciones contra-intuitivas y complejas [Kua17]. Estas nuevas métricas para medir la calidad en DASH han generado numerosos modelos objetivos para el cálculo de la *QoE*.

La medición de la *QoE* utilizando DASH ha cambiado el paradigma tradicional al calcular métricas de calidad más complejas que bajo esquemas tradicionales en la distribución de contenido multimedia, en los que se usa *PSNR* para obtener la calidad del servicio y *MOS* para medir la experiencia del usuario.

MOS es una métrica definida por la ITU-T en el documento P.800.1 con el nombre “Mean Opinion Score (MOS) Terminology” [ITU-T16]. El *MOS* se expresa como un único número racional, típicamente en el rango entre 1 y 5, donde 1 es la calidad percibida más baja, y 5 es la calidad percibida más alta.

Los modelos de *QoE* objetivos usualmente se clasifican en 3 tipos [Chen15]:

- Modelos con enfoque psicofísico.
- Modelos con enfoque en ingeniería.

- Modelos basados en referencia.

Los modelos con enfoque psicofísico normalmente se basan en mecanismos como “Human Visual System (HVS)”, los cuales miden el efecto de enmascaramiento, la sensibilidad al contraste, la adaptación al color y la iluminación del contenido multimedia.

Los modelos con enfoque en ingeniería se basan en modelos como “Structural Simetry (SSIM)”, los cuales analizan patrones de distorsión, características del vídeo o mecanismos de compresión.

Los modelos basados en referencia se clasifican en modelos de referencia completa, modelos de referencia reducida y modelos sin referencia. En estos, dependiendo del modelo, se necesita un mayor o menor acceso a la fuente del vídeo para realizar las comparaciones de calidad.

La distribución del contenido multimedia utilizando el estándar *DASH* requiere de un modelo de medición de la *QoE* que se efectúe en el momento (on the fly), y esto se considera como un nuevo paradigma de medición de la *QoE* para el tráfico que se distribuye por Internet llamado “Data-Driven *QoE* Assesment” (evaluación de *QoE* basada en datos).

La evaluación de la *QoE* basada en datos se enfoca en medir las siguientes métricas de *QoS*:

- Startup Delay: Se define como el tiempo que transcurre entre que el usuario solicita el contenido multimedia hasta que el vídeo se empieza a reproducir en el cliente.
- Rebuffering: Se define como el evento que sucede después de una interrupción en la reproducción del vídeo cuando el buffer de datos se vacía, para después esperar un momento a que el buffer recupere un nivel estable para reproducir el contenido multimedia de nuevo.
- Average Bitrate: Se define como la calidad a la que el contenido multimedia se decodifica en la pantalla del cliente.

En la tabla 3.4 se muestra como afecta a la calidad del vídeo percibida las características específicas de configuración de los distintos elementos del cliente que proveen la adaptación dinámica. Un ejemplo sería como afecta a la calidad percibida si se maneja un tamaño de buffer extenso o corto.

ITU-T, en el documento P.1203.1 [ITU-T19] define un módulo de estimación de la calidad del vídeo para servicios de distribución de contenido multimedia adaptativo como lo es el estándar *DASH*. El modelo define una serie de parámetros y coeficientes para calcular en términos de *MOS* la calidad de la experiencia percibida por el usuario.

Tabla 3.4: Efecto de los parámetros de la aplicación en la percepción [Seufert15]

Parámetro	Efecto	
	Pequeño	Grande
Tamaño del Buffer	Más stalling	Menos stalling
	Menos retardo inicial	Más retardo inicial
	Menos memoria	Más memoria
Tamaño del segmento	Pequeño	Grande
	Menos stalling	Más stalling
	Más archivos	Menos archivos
	Menor eficiencia en codificación	Mayor eficiencia en codificación
	Más cambios de calidad	Menos cambios de calidad
	Menos retardos	Más retardos
	Baja utilización de BW	Alta utilización de BW
	Menos equidad	Más equidad
Cambio de calidad	Baja	Alta
	Mejor que la peor calidad	Desagradable
	Menos degradación	Más degradación
Dirección/Amplitud cambio	Baja/Abajo	Alta/Arriba
	Empeora QoE	Mejora QoE
	Más impacto	Menos Impacto
	Cambio en calidad detectado	Cambio en calidad detectado
	Factor dominante	Factor no dominante

Oyman *et al.* [Oyman12] definen un modelo para medir la *QoE* en redes móviles, específicamente para redes *LTE*. Este modelo consiste en evaluar las métricas definidas tanto por el estándar ISO/IEC [ISO14b], como, por el estándar del 3GPP [3GPP18a]. Oyman *et al.* consideran el rebuffering como la principal causa de degradación de la *QoE* del usuario.

Thang *et al.* [Thang12] proponen definir nuevas etiquetas en el archivo de metadatos MPD en donde se guarden los datos de *QoE* de cada representación del contenido multimedia. De esta forma el cliente, con una ecuación con factores de peso, puede evaluar la calidad en la reproducción del contenido y con esto tomar decisiones en cuanto a la adaptación del contenido.

Claeys *et al.* [Claeys13] plantean utilizar tres métricas para calcular la *QoE* del usuario: la calidad promedio del segmento, la degradación asociada a los eventos de stalling y la influencia que tienen los cambios en el bitrate del contenido multimedia. Claeys *et al.* proponen calcular el valor de la *QoE* en un rango entre [0.0 a 5.30] lo cual no se puede traducir a la definición formal de *MOS*. En [García15] y [Martín16a] se utiliza este método para calcular la *QoE* de los usuarios.

Bouten *et al.* [Bouten14] proponen utilizar un índice de *QoE* que relaciona la tasa promedio del contenido multimedia, la *PSNR* y el método *SSIM* para evaluar la *QoE* del usuario. El objetivo de Bouten *et al.* es utilizar un proxy para informar al cliente sobre el

throughput disponible y poder repartir el ancho de banda de forma justa entre todos los clientes.

De Vriendt *et al.* [De Vriendt14] plantean utilizar tres métricas para calcular la *QoE* del usuario utilizando el método *MOS*, las cuales son: el promedio de la calidad, la desviación estándar de la calidad y la frecuencia de los cambios de calidad. Cada uno de estos parámetros es ajustado por una variable que afina las métricas propuestas.

Li *et al.* [Li14] proponen utilizar tres métricas para calcular la *QoE* del usuario: inestabilidad, ineficiencia e inequidad. La inestabilidad se refiere a los cambios en la calidad el bitrate del contenido multimedia. La ineficiencia se refiere al uso del ancho de banda cuando se hacen las peticiones de nuevos segmentos (periodos ON-OFF). La inequidad se refiere a lo que pasa cuando varios usuarios compiten por el mismo throughput. En [Zhou17] también se utilizan estas tres métricas para calcular la *QoE* del usuario.

Beben *et al.* [Beben16] renombran las métricas que se refieren a los cambios de calidad en el bitrate del contenido multimedia y los eventos de parada para calcular la *QoE* del usuario.

Cicalò *et al.* [Cicalò16] proponen utilizar un algoritmo para medir la *QoE* de *DASH* en redes *LTE*. El método para medir la *QoE* esta basado en *SSIM*, el cual arroja como resultado valores entre -1 y 1, en donde, si el valor es mayor a 0.94 indica que se ofrece una buena calidad visual y valores menores a 0.86 indican una pobre calidad visual.

Huang *et al.* [Huang18] plantean utilizar tres métricas para medir la *QoE* de los usuarios. La primera métrica, llamada “Freezing Score”, mide la cantidad de eventos de parada de la reproducción del contenido multimedia. La segunda métrica, llamada “Bitrate switch score”, evalúa la cantidad de cambios en la calidad del contenido multimedia. La tercera métrica, llamada “video bitrate score”, calcula la tasa promedio a la que se reproduce el contenido multimedia. Estas tres métricas se definen en una ecuación lineal en donde las métricas dos y tres tienen coeficientes de peso positivos para corregir errores en la evaluación de la *QoE*.

Una vez que se han analizado los distintos trabajos de métricas de *QoE* que existen, es necesario seleccionar cuales de estas métricas se pueden utilizar en este trabajo de investigación. A continuación, se describen las métricas de *QoE* seleccionadas que se utilizarán en el resto del documento y que serán utilizadas para evaluar la calidad percibida por los usuarios.

- *Rendimiento promedio* [Akhshabi11] es la tasa promedio que se ha calculado durante la reproducción del contenido multimedia. Normalmente, la tasa promedio se calcula

como la diferencia entre la cantidad de datos descargada y el tiempo transcurrido entre el inicio y el final de la descarga del segmento multimedia.

- *Número de cambios en el bitrate* [De Vriendt14] es el número de cambios en el bitrate que se producen durante la reproducción del contenido multimedia. La conmutación entre diferentes bitrate previene que el buffer del reproductor se agote en el servicio de streaming adaptativo basado en *DASH*.
- *Eventos de parada* [Claeys13] es el número de eventos en el que ha permanecido parada la reproducción como consecuencia de la ausencia de datos en el buffer del reproductor. Esto puede producirse porque los segmentos multimedia se han recibido más tarde del instante en que debían reproducirse. Un evento de parada afecta de manera significativa la *QoE* percibida por los usuarios.
- *Duración del Evento de parada* [Claeys13] es el tiempo que ha permanecido parada la reproducción como consecuencia de la ausencia de datos en el buffer del reproductor. Normalmente, esta métrica se mide en tiempo, es decir, cuánto tiempo ha estado detenida la reproducción del contenido multimedia. La duración de los eventos de parada afectan a la *QoE* percibida por los usuarios.
- *Equidad* [Li14] se define como el uso justo del ancho de banda entre todos los clientes o servicios. Normalmente, para medir la equidad se utiliza la métrica propuesta por Jain *et. al.* [Jain84]

3.5. Conclusiones

En este capítulo se han presentado y discutido las distintas propuestas y trabajos sobre los algoritmos de adaptación para el estándar *DASH*. El capítulo empieza describiendo los principios generales que rigen los algoritmos de tasa adaptativa y sus definiciones.

En la sección 3.2 se presentó el funcionamiento de los algoritmos de tasa adaptativa, así como los objetivos que cualquier algoritmo debe cumplir para ofrecer un buen servicio de distribución de contenido multimedia. También, se mostraron los eventos que se presentan en el reproductor al ofrecer el servicio de distribución de contenidos multimedia y el efecto que tiene en la percepción del usuario las características en la definición del tamaño (en segundos) de los segmentos del contenido multimedia.

En la sección 3.3 se presentaron las principales propuestas de algoritmos de tasa adaptativa propuestos por diferentes autores. Los algoritmos presentados se dividen en 3 tipos:

los algoritmos que proponen un nuevo método de estimación del throughput, los algoritmos que proponen un nuevo mecanismo de control del buffer y los algoritmos que utilizan teoría de control o un esquema híbrido entre el throughput y el control del buffer. Finalmente se presentan las principales conclusiones sobre los algoritmos de adaptación dinámica.

En la sección 3.4 se analizaron los principales modelos para medir la *QoE* del usuario utilizando las métricas definidas por el estándar *DASH*, según la sección 2.4 de este documento. Además, al final de esta sección, se presentan las métricas seleccionadas por este trabajo de investigación con las que se evaluará la calidad percibida por los usuarios en el servicio de distribución multimedia usando *DASH*.

Como conclusión final de este capítulo, al desarrollar el análisis de los distintos algoritmos de adaptación dinámica que se han propuesto para *DASH*, se puede concluir que aún existen áreas de oportunidad en los que se pueden proponer soluciones con las que se pueda mejorar el servicio de distribución multimedia, siendo este el objetivo principal de este trabajo de investigación. En el siguiente capítulo se muestra la definición de un nuevo algoritmo para la estimación del throughput que puede ser implementado en redes cableadas e inalámbricas sin necesidad de implementar distintos clientes para cada red.

CAPÍTULO 4

MÉTODO ADAPTIVE FORGETTING FACTOR (AFF)

4.1. Introducción

En este capítulo se propone un nuevo método para la estimación del throughput en los reproductores de un servicio de distribución multimedia. El método se llama “Adaptive Forgetting Factor (AFF)” y funciona estimando de forma adaptativa el rendimiento de la transmisión del vídeo. Este método se caracteriza por actuar frente a fluctuaciones a corto plazo, aumentando el peso de las observaciones más reciente de forma que “olvida” las mediciones antiguas más rápidamente.

En la sección 4.2 se presentan las definiciones necesarias para sustentar el método propuesto en esta Tesis Doctoral que utiliza la detección de puntos de cambio para identificar momentos en que la distribución de probabilidad de un proceso cambia. Después, se definen los conceptos necesarios para proponer un método matemático que utiliza “Adaptive Forgetting Factor (AFF)” para el cálculo de la estimación del throughput promedio utilizando las mediadas del throughput instantáneo dentro del algoritmo *AFF*. El siguiente paso es la definición de las ecuaciones necesarias para actualizar los valores de los pesos exponenciales cuando se ha terminado de descargar un segmento de vídeo. En la sección 4.2.3 se definen parámetros necesarios y sus valores óptimos para regular el comportamiento del método *AFF*. Además, en la sección 4.3, se presenta una discusión acerca del método *AFF* y las ventajas que presenta con respecto a otros métodos frecuentemente utilizados en los algoritmos de adaptación de los reproductores *DASH*. Después, en la sección 4.4, se muestra el pseudo-código que describe al método *AFF* con instrucciones para su posterior implementación en un lenguaje de programación. Por último, en el apartado 4.5 se

presentan las conclusiones de este capítulo.

4.2. Estimación del throughput en DASH

En muchas tareas de análisis de datos, estos no están disponibles todos a la vez, sino que llegan de forma secuencial. Ejemplos de este tipo de tareas pueden ser el análisis de sensores, las series temporales financieras o las aplicaciones de control adaptativo. En estos casos, es útil no reiniciar desde cero la estimación del modelo, sino simplemente actualizar el modelo sobre la base de los datos recién recopilados.

La estadística es la ciencia que se encarga de estudiar elementos o eventos con características determinadas por medio de la recolección, recopilación e interpretación de datos. Dentro de esta rama matemática el control del proceso estadístico o “*Statistical Process Control (SPC)*”, creado por W. Shewhart [Shewhart39] en 1939, es un método para monitorizar el desempeño de un proceso a lo largo del tiempo y distinguir variaciones de fondo de eventos de importancia. *SPC* utiliza gráficos de control para monitorizar el centro del proceso y su variación alrededor del centro con el fin de detectar la ocurrencia de cualquier evento que se salga de unos límites de control establecidos, “*Upper Control Limit (UCL)*” y “*Lower Control Limit (LCL)*”.

Dentro de *SPC* existen dos gráficos de control conocidos que son: “*Cumulative Sum (CUSUM) control chart*” [Page54] creado por E. Page en 1954 y “*EWMA control chart*” [Roberts59] (ver figura 4.1) creado por S. Roberts en 1959. Estos gráficos son usados normalmente para monitorizar la detección de cambios y funcionan en dos estados que son: en control (entre de *UCL* y *LCL*) y fuera de control (afuera *UCL* y *LCL*). El problema que presentan este tipo de gráficos de control es que dependen de la elección de los parámetros de control iniciales.

El método *AFF* se diseñó originalmente como un método *SPC* que utiliza un filtro adaptativo recursivo de mínimos cuadrados o “*Recursive Least-Square (RLS)*” [Widrow85] para recuperar datos de señales dañadas [Cooper00]. El algoritmo de filtro adaptativo *RLS* [Gauss23] [Plackett50] funciona encontrando recursivamente los coeficientes que minimizan una función de costo en relación con las señales de entrada. Con el filtro *RLS* se puede modelar cualquier sistema, lo cual se logra mediante el uso de coeficientes adaptativos, con los cuales se pueden compensar las perturbaciones que pudieran surgir durante la transmisión de señales en tiempo real. Se pueden encontrar métodos similares en varios campos como: astronomía [Bhaduri11], medicina [Burkom05] [Castagliola11], química [Rotstein92],

Figura 4.1: Gráfica de control EWMA

finanzas [Wu04], procesamiento de señales [Leung05] [Paleologu08] o monitorización de redes de computadoras [Lindsay00].

En esta tesis se propone el uso de un método basado en *AFF* originalmente diseñado por Dean A. Bodenham [Bodenham14] para analizar flujos de datos en ciber-seguridad con el fin de detectar anomalías (ataques) utilizando la media y la varianza de los datos.

Utilizando el método *AFF* creado por Bodenham [Bodenham14] se pretende mejorar la calidad en la reproducción del contenido multimedia en *DASH*. Para ello, el método *AFF* ha sido adaptado para calcular el promedio de las estimaciones del throughput que calcula el reproductor *DASH* en un servicio de streaming multimedia, tal como se explica a continuación.

4.2.1. Qué es el método de estimación *AFF*

En el análisis estadístico, la detección de cambios o la detección de puntos de cambio intenta identificar los momentos en que cambia la distribución de probabilidad de un proceso estocástico o de serie temporal (secuencia de datos medidos en determinado momento y ordenados cronológicamente). En general, el problema se refiere tanto a la detección de si un cambio (anomalía) ha ocurrido o no, o si se han producido varios cambios, así como a la identificación de los tiempos de dichos cambios.

En nuestro caso las anomalías se refieren a cambios súbitos en el comportamiento del throughput instantáneo que detecta el reproductor *DASH* al momento de descargar los seg-

mentos de contenido multimedia. Este tipo de cambios súbitos (short-term bandwidth fluctuations) suceden con mayor frecuencia en redes de acceso móviles [Alejos12]. Esto se debe a que el canal radio exhibe comportamientos temporales aleatorios y una dependencia de la frecuencia que puede producir, entre otros fenómenos de propagación, el multitrayecto (multipath).

Tabla 4.1: Nomenclatura del Algoritmo *AFF*

Notación	Descripción
THR_N	El valor del throughput instantáneo para el n -ésimo segmento
$\vec{\lambda}_i$	Vector que guarda los pesos calculados por el método <i>AFF</i> para cada segmento
$\overline{THR}_{N, \vec{\lambda}}$	Resultado hasta el n -ésimo segmento del método <i>AFF</i>
$m_{N, \vec{\lambda}}$	Calcula el promedio hasta el n -ésimo segmento multiplicado por el peso λ_i
$w_{N, \vec{\lambda}}$	Calcula el número de segmentos multiplicado por el peso λ_i
$L_{N+1, \vec{\lambda}}$	Define la función de costo del método descenso del gradiente de un paso
$\Delta_{N, \vec{\lambda}}$	Resultado de derivar la ecuación $m_{N, \vec{\lambda}}$
$\Omega_{N, \vec{\lambda}}$	Resultado de derivar la ecuación $w_{N, \vec{\lambda}}$

Supongamos que se desea implementar un nuevo método de estimación del throughput en un reproductor de contenido multimedia utilizando el estándar *DASH*. Además, supongamos que el reproductor cuenta con un método de estimación del ancho de banda básico que consiste en calcular el tiempo que toma en descargar un segmento *TCP* con respecto al tamaño de dicho segmento (Throughput Instantáneo). Asumamos que las variables $THR_1, THR_2, \dots, THR_N$ son el resultado del cálculo mencionado anteriormente y que estos valores se pueden considerar variables aleatorias independientes e idénticamente distribuidas. Asumamos además, que las variables THR_N calculan el throughput instantáneo R_n , y que se han almacenado N observaciones hasta el momento.

El objetivo es obtener una buena estimación de la media o promedio actual del throughput instantáneo $E[THR_N]$, con el fin de detectar puntos de cambio o anomalías en throughput (fluctuaciones a corto plazo). Se podría decir que $(E[THR_\tau] - E[THR_{\tau+1}])$ calcula el tamaño del cambio en la media, y con esto poder reiniciar el algoritmo después de detectar la anomalía (fluctuaciones a corto plazo del ancho de banda).

El objetivo del método *AFF* es seleccionar correctamente la representación del contenido multimedia cuya tasa de bits coincide con las condiciones de ancho de banda que ofrece la red para lograr la mayor calidad en la reproducción de vídeo. Esto se logra al detectar fluctuaciones a corto plazo en el ancho de banda, y cuando se encuentran, el método aumenta el peso de las observaciones más recientes de forma que “olvida” las mediciones antiguas más rápidamente.

Esto genera un problema: este algoritmo de detección de cambios requiere los valores de los parámetros de la distribución, pero es inusual conocer los valores de los parámetros posteriores al cambio, y no se pueden usar los valores de los parámetros previos al cambio (ya que se ha producido un cambio).

Para resolver este problema *AFF* utiliza parámetros de peso para disminuir el impacto de mediciones anteriores en las que se detectaron anomalías. Estos parámetros de peso se calculan exponencialmente en las estimaciones de la media. Existe una cierta similitud con el método matemático *EWMA* el cual utiliza un factor de pesos fijos. Sin embargo, el método de estimación *AFF* calcula dinámicamente los factores de peso.

La estimación media del throughput se muestra en la Ecuación 4.1. El mecanismo de la estimación adaptativa *AFF* es $\vec{\lambda}$, donde $\vec{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_N)$ y $\vec{\lambda}_i \in [0, 1]$.

$$\overline{THR}_{N, \vec{\lambda}} = \frac{m_{N, \vec{\lambda}}}{w_{N, \vec{\lambda}}}, \quad N \geq 1 \quad (4.1)$$

Una ventaja del método *AFF* es que cuenta con ecuaciones de actualización secuencial, como son las ecuaciones 4.2 y 4.3, que muestran que el cálculo de $\overline{THR}_{N, \vec{\lambda}}$ únicamente requiere un número finito de estadísticas para ser almacenadas, y muestra que el cálculo por dato es de una complejidad constante. Lo cual hace que el cliente implemente una programación que no sobrecarga el funcionamiento del reproductor *DASH* en el navegador web.

$$m_{N, \vec{\lambda}} = \lambda_{N-1} m_{N-1, \vec{\lambda}} + THR_N, \quad N \geq 1 \quad (4.2)$$

$$w_{N, \vec{\lambda}} = \lambda_{N-1} w_{N-1, \vec{\lambda}} + 1, \quad N \geq 1 \quad (4.3)$$

La ecuación 4.2 representa las medidas del throughput instantáneo acumulado. La ecuación 4.3 muestra la suma del número de segmentos multimedia acumulados hasta el momento. Donde $m_{0, \vec{\lambda}} = 0$, $w_{0, \vec{\lambda}} = 0$ y $\lambda_0 = 1$ respectivamente.

El método *AFF* desarrollado por Bodenham [Bodenham14] cuenta con un detector de cambios eficiente y completamente secuencial, con el cual se puede calcular la media de un flujo univariante, como es el servicio de distribución de contenido multimedia. El método matemático para actualizar λ_N , que es la parte esencial en este algoritmo de estimación media del throughput, se explica en la siguiente sección.

4.2.2. Actualización del factor adaptativo

La parte principal del método *AFF* es cómo se actualiza el factor de peso, es decir $\lambda_N \rightarrow \lambda_{N+1}$. Para obtener λ_N , dado que esto se aplicará en el contexto de un flujo de datos continuo, es necesario aplicar un método de optimización online. En esta investigación se utilizará, el método descenso de gradiente de un paso (one-step gradient descent).

El método descenso de gradientes de un paso es, con mucho, la estrategia de optimización más popular utilizada en “machine learning” y “deep learning” actualmente. Además, es un método que se puede combinar con todos los algoritmos y funciona mejor cuando los parámetros que se necesitan calcular no pueden obtenerse de forma analítica (por ejemplo, utilizando álgebra lineal).

El descenso de gradiente de un solo paso es un algoritmo de optimización iterativa de primer orden para encontrar el mínimo de una función. Este método de optimización online se muestra en la ecuación 4.4.

$$\lambda_{N+1} = \lambda_N - \eta \frac{\partial}{\partial \lambda} L_{N+1, \vec{\lambda}} \quad (4.4)$$

Este método de optimización necesita una función de coste, la cual consigue que el método se aproxime de mejor o peor manera al mínimo de la función. Existen múltiples opciones para seleccionar la función de coste, sin embargo, es preferible seleccionar aquellas en las que se mida la diferencia entre los datos originales y los resultados producidos por el método descenso de gradiente de un solo paso [Ruder16] [Sutton09]. Las ecuaciones de coste comúnmente utilizadas por el método de optimización online son:

- Desviación cuadrática media (MSE)
- Error cuadrático medio (RMS)
- Error absoluto medio (MAE)

Para poder utilizar el descenso de gradiente de un solo paso, es necesario seleccionar una función de coste $L_{N+1, \vec{\lambda}}$, que en nuestro caso será la desviación cuadrática media definida por la ecuación 4.5 para el algoritmo *AFF*. Esta ecuación se ha seleccionado para comparar los valores estimados medios o dispersión del throughput acumulado con respecto a la nueva medición del throughput instantáneo.

$$L_{N+1, \vec{\lambda}} = \left[\overline{THR}_{N, \vec{\lambda}} - THR_{N+1} \right]^2 \quad (4.5)$$

Otras funciones de coste podrían usarse para el método *AFF*. De hecho, la elección de la función de coste dependerá de la estadística que se monitoriza, ya sea la media o la varianza [Anagnostopoulos12].

Para resolver la derivada parcial que se presenta en la ecuación 4.4, se aplica la regla de la cadena (chain rule) en la ecuación 4.5, lo cual da como resultado la ecuación 4.6.

$$\frac{\partial}{\partial \vec{\lambda}} L_{N+1, \vec{\lambda}} = 2 \left[\overline{THR}_{N, \vec{\lambda}} - THR_{N+1} \right] \frac{\partial}{\partial \vec{\lambda}} \overline{THR}_{N, \vec{\lambda}} \quad (4.6)$$

Esta solución genera una nueva derivada parcial con respecto de λ_N en $\overline{THR}_{N, \vec{\lambda}}$, que sería equivalente a obtener la derivada parcial de la ecuación 4.1. En la ecuación 4.7 se muestra la solución a esta derivada parcial.

$$\frac{\partial}{\partial \vec{\lambda}} \overline{THR}_{N, \vec{\lambda}} = \frac{\partial}{\partial \vec{\lambda}} \left(\frac{m_{N, \vec{\lambda}}}{w_{N, \vec{\lambda}}} \right) = \frac{w_{N, \vec{\lambda}} \frac{\partial}{\partial \vec{\lambda}} m_{N, \vec{\lambda}} - m_{N, \vec{\lambda}} \frac{\partial}{\partial \vec{\lambda}} w_{N, \vec{\lambda}}}{(w_{N, \vec{\lambda}})^2} \quad (4.7)$$

La ecuación 4.7 muestra que es necesario calcular la derivada parcial con respecto de λ_N de las ecuaciones 4.2 y 4.3. Estas derivadas se resuelven aplicando el cociente de Newton o cociente diferencial alternativo, que es la derivada de f en x y que se describe como el límite conforme h se aproxima a cero del valor de cuando $f(x+h)$ se acerca más a $f(x)$ dividido entre h que es conocido como el cociente diferencial, como se muestra en la ecuación 4.8.

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (4.8)$$

Para esto es necesario adaptar el cociente de Newton a nuestras ecuaciones. Teniendo en cuenta que $\vec{\lambda}_N = (\lambda_1, \lambda_2, \dots, \lambda_N)$, para cualquier $h \in \mathbb{R}$, se tiene que $\vec{\lambda}_N + h = (\lambda_1 + h, \lambda_2 + h, \dots, \lambda_N + h)$. Con lo cual se redefinen las ecuaciones 4.2 y 4.3 segun las nuevas ecuaciones 4.9 y 4.10

$$m_{N, \vec{\lambda}+h} = (\lambda_{N-1} + h)m_{N-1, \vec{\lambda}+h} + THR_N, \quad (4.9)$$

$$w_{N, \vec{\lambda}+h} = (\lambda_{N-1} + h)w_{N-1, \vec{\lambda}+h} + 1, \quad (4.10)$$

Una vez definidas las ecuaciones se aplica el cociente de Newton, es decir se transforman las ecuaciones 4.2 y 4.9 utilizando la ecuación 4.8. Las ecuaciones 4.11 y 4.12

muestran el resultado de implementar el cociente de Newton.

$$\frac{\partial}{\partial \vec{\lambda}} m_{N, \vec{\lambda}} = \Delta_{N, \vec{\lambda}} = \lim_{h \rightarrow 0} \frac{1}{h} [m_{N, \vec{\lambda}+h} - m_{N, \vec{\lambda}}] \quad (4.11)$$

$$\frac{\partial}{\partial \vec{\lambda}} w_{N, \vec{\lambda}} = \Omega_{N, \vec{\lambda}} = \lim_{h \rightarrow 0} \frac{1}{h} [w_{N, \vec{\lambda}+h} - w_{N, \vec{\lambda}}] \quad (4.12)$$

Para poder calcular los límites mencionados en las ecuaciones 4.11 y 4.12 es necesario definir la forma no secuencial de las ecuaciones 4.2 y 4.3. Esta forma no secuencial se muestra en las ecuaciones 4.13 y 4.14.

$$m_{N, \vec{\lambda}} = \sum_{i=1}^N \left[\left(\prod_{p=i}^{N-1} (\lambda_p) \right) THR_i \right] \quad (4.13)$$

$$w_{N, \vec{\lambda}} = \sum_{i=1}^N \left[\left(\prod_{p=i}^{N-1} \lambda_p \right) \right] \quad (4.14)$$

También, es necesario definir la forma no secuencial para las ecuaciones 4.9 y 4.10, para lo cual se define la ecuación 4.15.

$$\prod_{p=i}^{N-1} (\lambda_p + h) = \prod_{p=i}^{N-1} \lambda_p + h \left(\sum_{j=i}^N \left(\prod_{\substack{p=i \\ p \neq j}}^N \lambda_p \right) \right) + O(h^2). \quad (4.15)$$

Se puede observar que, en la ecuación 4.15 se utiliza $x = x + O(h^2)$, que es una propiedad de los productorios llamada “producto vacío”. Un producto vacío es 1 y no 0, y se utiliza para reducir el número de casos a considerar en la ecuación matemática. Una vez que se cuentan con todos los elementos se procede a encontrar la solución para la derivada de la ecuación 4.11, la cual se muestra en la ecuación 4.16.

$$\begin{aligned}
\Delta_{N, \vec{\lambda}} &= \lim_{h \rightarrow 0} \frac{1}{h} \left[m_{N, \vec{\lambda}+h} - m_{N, \vec{\lambda}} \right], \\
&= \lim_{h \rightarrow 0} \frac{1}{h} \left[\sum_{i=1}^N \left[\left(\prod_{p=i}^{N-1} (\lambda_p + h) \right) THR_i \right] - \sum_{i=1}^N \left[\left(\prod_{p=i}^{N-1} (\lambda_p) \right) THR_i \right] \right] \\
&= \lim_{h \rightarrow 0} \frac{1}{h} \sum_{i=1}^N \left[\prod_{p=i}^{N-1} (\lambda_p + h) - \prod_{p=i}^{N-1} \lambda_p \right] THR_i \\
&= \lim_{h \rightarrow 0} \frac{1}{h} \sum_{i=1}^N \left[h \left(\sum_{j=i}^{N-1} \left(\prod_{p=i}^{N-1} \lambda_p \right) \right) + O(h^2) \right] x_i \\
&= \lim_{h \rightarrow 0} \sum_{i=1}^N \left[\frac{1}{h} \cdot h \left(\sum_{j=i}^{N-1} \left(\prod_{\substack{p=i \\ p \neq j}}^{N-1} \lambda_p \right) \right) + \frac{1}{h} \cdot O(h^2) \right] THR_i \\
&= \lim_{h \rightarrow 0} \sum_{i=1}^N \left[\sum_{j=i}^{N-1} \left(\prod_{\substack{p=i \\ p \neq j}}^{N-1} \lambda_p \right) + O(h) \right] THR_i \\
\Rightarrow \Delta_{N, \vec{\lambda}} &= \sum_{i=1}^N \left[\sum_{j=i}^{N-1} \left(\prod_{\substack{p=i \\ p \neq j}}^{N-1} \lambda_p \right) THR_i \right] \tag{4.16}
\end{aligned}$$

Y de igual manera se procede a encontrar la solución para la derivada de la ecuación 4.12, la cual se muestra en la ecuación 4.17.

$$\begin{aligned}
\Omega_{N,\vec{\lambda}} &= \lim_{h \rightarrow 0} \frac{1}{h} \left[w_{N,\vec{\lambda}+h} - w_{N,\vec{\lambda}} \right], \\
&= \lim_{h \rightarrow 0} \frac{1}{h} \left[\sum_{i=1}^N \left[\left(\prod_{p=i}^{N-1} (\lambda_p + h) \right) \right] - \sum_{i=1}^N \left[\left(\prod_{p=i}^{N-1} (\lambda_p) \right) \right] \right] \\
&= \lim_{h \rightarrow 0} \frac{1}{h} \sum_{i=1}^N \left[\prod_{p=i}^{N-1} (\lambda_p + h) - \prod_{p=i}^{N-1} \lambda_p \right] \\
&= \lim_{h \rightarrow 0} \frac{1}{h} \sum_{i=1}^N \left[h \left(\sum_{j=i}^{N-1} \left(\prod_{\substack{p=i \\ p \neq j}}^{N-1} \lambda_p \right) \right) + O(h^2) \right] \\
&= \lim_{h \rightarrow 0} \sum_{i=1}^N \left[\frac{1}{h} \cdot h \left(\sum_{j=i}^{N-1} \left(\prod_{\substack{p=i \\ p \neq j}}^{N-1} \lambda_p \right) \right) + \frac{1}{h} \cdot O(h^2) \right] \\
&= \lim_{h \rightarrow 0} \sum_{i=1}^N \left[\sum_{j=i}^{N-1} \left(\prod_{\substack{p=i \\ p \neq j}}^{N-1} \lambda_p \right) + O(h) \right] \\
\Rightarrow \Omega_{N,\vec{\lambda}} &= \sum_{i=1}^N \left[\sum_{j=i}^{N-1} \left(\prod_{\substack{p=i \\ p \neq j}}^{N-1} \lambda_p \right) \right] \tag{4.17}
\end{aligned}$$

Después, con las ecuaciones resueltas se procede a regresar las ecuaciones a la forma secuencial para ser implementadas en el algoritmo *AFF*. Esto se muestra en las ecuaciones 4.18 y 4.19.

$$\Delta_{N,\vec{\lambda}} = \lambda_{N-1} \Delta_{N-1,\vec{\lambda}} + m_{N-1,\vec{\lambda}} \tag{4.18}$$

$$\Omega_{N,\vec{\lambda}} = \lambda_{N-1} \Omega_{N-1,\vec{\lambda}} + w_{N-1,\vec{\lambda}} \tag{4.19}$$

Las medidas iniciales para las ecuaciones 4.18 y 4.19 se definen como: $\Delta_{1,\vec{\lambda}} = 0$ y $\Omega_{1,\vec{\lambda}} = 0$ respectivamente. La ecuación 4.20 es el resultado de aplicar las soluciones que se muestran anteriormente para resolver λ_{N+1} . Las ecuaciones aquí presentadas [Bodenham14] han sido modificadas para poder adaptarse al área de investigación de este

trabajo.

$$\lambda_{N+1} = \lambda_N - 2\eta \left[\overline{THR}_{N,\vec{\lambda}} - THR_{N+1} \right] \left(\frac{\Delta_{N,\vec{\lambda}} m_{N,\vec{\lambda}} - \Omega_{N,\vec{\lambda}} w_{N,\vec{\lambda}}}{(w_{N,\vec{\lambda}})^2} \right) \quad (4.20)$$

Una vez encontrada la solución para actualizar el factor adaptativo del método *AFF*, es decir $\lambda_N \rightarrow \lambda_{N+1}$, es necesario establecer una serie de límites para optimizar el funcionamiento del método *AFF*, estos límites se explican en el siguiente apartado.

4.2.3. Comportamiento óptimo del método *AFF*

El método *AFF* cuenta con dos maneras de regular su comportamiento. La primera manera sería con los valores que la variable λ_N puede tomar $[0, 1]$. La segunda manera sería seleccionar el valor óptimo de la variable η . A continuación, se diserta sobre estos mecanismos.

4.2.3.1. Comportamiento óptimo de *AFF* con λ_N

Es posible que utilizando la ecuación 4.20 se pudiera dar el caso en que $\vec{\lambda}$ pueda obtener valores que estén fuera del intervalo $[0, 1]$, por lo cual, por diseño el método matemático *AFF* implementa las siguientes reglas:

$$\lambda_{N+1} = \min(\lambda_{N+1}, \lambda_{\max}), \quad (4.21)$$

$$\lambda_{N+1} = \max(\lambda_{N+1}, \lambda_{\min}), \quad (4.22)$$

Donde $\lambda_{\min}$ y $\lambda_{\max}$ son los valores mínimos y máximos de $\vec{\lambda}$ respectivamente. Bodenham en [Bodenham14], realizando numerosas simulaciones (más de 1000 simulaciones), llegó a la conclusión de que truncar el rango de $\vec{\lambda}$ como $\lambda_N \in [0.6, 1]$ hace que el método *AFF* se recupere más rápido a valores estables después de encontrar anomalías o en nuestro caso fluctuaciones a corto plazo en el ancho de banda.

Se podrían usar otros valores además de $\lambda_{\min} = 0.6$, pero este valor parece ofrecer un buen equilibrio entre permitir la mayor caída posible de $\vec{\lambda}$, al tiempo que se evitan los problemas de recuperación y el aumento de la varianza de $\overline{THR}_{N,\vec{\lambda}}$.

4.2.3.2. Selección óptima de la variable η

La variable η se corresponde a la variable encargada de cuantificar el tamaño de los pasos del algoritmo de optimización descenso por gradiente. Es decir, el valor de η permite afinar el comportamiento del algoritmo de optimización del descenso del gradiente. De acuerdo con Bodenham en [Bodenham13] los valores de la variable η deberían estar entre $\eta \in (0.001, 0.1)$, y muestra que los valores cercanos a 0.1 requieren de menos pasos para regresar a valores óptimos cuando encuentran anomalías usando el método *AFF* mientras que valores cercanos a $\eta = 0.001$ hacen que el método *AFF* reaccione de manera lenta. En esta investigación se ha definido el valor de $\eta = 0.1$.

4.3. Pseudo-código para el Algoritmo *AFF*

Las ecuaciones explicadas anteriormente se implementaron como se muestra en el Algoritmo 1. El objetivo del método *AFF* es seleccionar correctamente la representación del contenido multimedia cuya tasa de bits coincide con las condiciones de ancho de banda que la red ofrece para lograr la mejor calidad en la reproducción de vídeo. Esto se logra al estimar con el método *AFF* de forma adaptativa el throughput de la transmisión de vídeo, y al encontrar fluctuaciones a corto plazo, el método *AFF* aplica menores pesos en las observaciones más recientes y, por lo tanto, “olvida” las mediciones antiguas más rápido.

El algoritmo compara si la medición de throughput es la primera, porque en ese caso el cálculo se realiza con diferentes variables de inicialización. Después, el algoritmo calcula el throughput estimado utilizando la siguiente medición de throughput instantáneo en las ecuaciones *AFF*.

El algoritmo *AFF* se ejecuta cada vez que se obtiene una nueva medición del throughput instantáneo, es decir, cada vez que se descarga un segmento de contenido multimedia. Una vez que se estima el throughput promedio, el algoritmo selecciona una nueva representación de vídeo realizando una búsqueda secuencial del bitrate de las diferentes representaciones del contenido multimedia, desde la más alta a la más baja, para seleccionar el bitrate más alto que esté por debajo del throughput promedio.

Algoritmo 1 Algoritmo de estimación del throughput AFF

Input: $maxbitrateindex, bitratebw(i),$
 $SegmentIndex = n, THR_n$
Output: $switchbitrate(i)$
Initialization : $m_0 = w_0 = 0; \lambda_0 = 0;$
 $\Omega_1 = \Delta_1 = 0; \eta = 0.1;$
1: **if** ($SegmentIndex == 1$) **then**
2: $m_1 = (\lambda_0 * m_0) + THR_1$
3: $w_1 = (\lambda_0 * w_0) + 1$
4: $\overline{THR}_1 = \frac{m_1}{w_1}$
5: $\lambda_1 = \lambda_0 - \eta * 2 * (\overline{THR}_1 - THR_1) * (\frac{\Delta_1 * w_1 - \Omega_1 * m_1}{w_1^2})$
6: **else**
7: $\Delta_n = (\lambda_{n-1} * \Delta_{n-1}) + m$
8: $\Omega_n = (\lambda_{n-1} * \Omega_{n-1}) + w$
9: $m_n = (\lambda_{n-1} * m_{n-1}) + THR_n$
10: $w_n = (\lambda_{n-1} * w_{n-1}) + 1$
11: $\overline{THR}_n = \frac{m_n}{w_n}$
12: $\lambda_n = \lambda_{n-1} - \eta * 2 * (\overline{THR}_n - THR_n) * (\frac{\Delta_n * w_n - \Omega_n * m_n}{w_n^2})$
13: **end if**
14: **for** $i = maxbitrateindex$ **to** 0 **do**
15: **if** ($\overline{THR}_n \geq bitratebw(i)$) **then**
16: **return** $switchbitrate(i)$
17: **end for**
18: **end if**
19: **end for**

Figura 4.2: Algoritmo de estimación del throughput AFF

4.4. Comparación con otros métodos de estimación

Como ya se mencionó en la sección 3.4 los métodos de estimación de throughput utilizados por diferentes algoritmos para la adaptación dinámica en los reproductores del estándar *DASH* son principalmente: el throughput instantáneo, la media armónica, la media móvil y *EWMA*.

El uso del throughput instantáneo presenta problemas al seguir las fluctuaciones del ancho de banda especialmente en redes móviles con un alto número de variaciones del ancho de banda, por ejemplo, el algoritmo de adaptación cambiaría constantemente la calidad del vídeo, lo cual afecta de manera negativa la calidad de la experiencia para el usuario.

Utilizar la media armónica para suavizar las mediciones del throughput instantáneo presenta el problema de que normalmente se acota a una cantidad pasada de segmentos para

realizar la estimación, por ejemplo, como el algoritmo *ELASTIC* [Cicco13] que utiliza las últimas 5 mediciones o el algoritmo *FESTIVE* [Jiang14] que utiliza las últimas 20 mediciones. Las últimas 20 mediciones con segmentos de 2 segundos de duración significa utilizar 40 segundos de historia para estimar la media armónica, lo que puede ocasionar que este método no reaccione adecuadamente a cambios súbitos en el ancho de banda, como sucede en redes móviles [Alejos12].

La media móvil se utiliza también como un método para suavizar la estimación del throughput instantáneo, y al igual que la media armónica, dependiendo de la cantidad de historia utilizada para realizar la estimación, funciona mejor o peor en los escenarios en los que se implementa. El método presenta el problema de que cuando se utilizan pocas mediciones, tiene un comportamiento parecido al del throughput instantáneo y cuando se utilizan mucha historia puede ocasionar que el método sobre-estime o sub-estime el ancho de banda disponible. El algoritmo para la estimación del throughput que utiliza el reproductor *dash.js* [DASH-IF18b] utiliza la media móvil de los últimos 3 segmentos.

Utilizar el método *EWMA* presenta el problema de que es necesario seleccionar de antemano el valor utilizado para los pesos exponenciales, dependiendo del escenario en donde se va a implementar. Usualmente se utilizan valores entre 0 y 0.1 para escenarios con redes fijas [Akhshabi11] y valores entre 0.2 y 0.3 para redes móviles [Thang12]. Sin embargo, en escenarios en donde puede suceder un evento de handover entre una red móvil 4G y una red inalámbrica doméstica, el ancho de banda puede cambiar drásticamente [Ahmed14] lo que ocasionaría que el método *EWMA* no funcionara de manera adecuada.

En la figura 4.3 se muestra el método *AFF* comparado con los métodos mencionados anteriormente. La figura muestra mediciones del ancho de banda y como realizan las estimaciones los métodos: la media armónica de 10 mediciones anteriores, *EWMA* con un peso exponencial de 0.2, la media móvil de 5 mediciones anteriores y el método *AFF* con un valor de $\eta = 0.1$.

Se puede observar que cuando existe un cambio drástico en el ancho de banda, el método *AFF* lo detecta primero y se adecua de manera rápida al cambio que se observa, mientras que, otros métodos pueden llegar a tardar unos segundos más. Este comportamiento hace que el método *AFF* pueda ser adecuado como único método de estimación en reproductores *DASH* para diferentes tipos de redes de acceso.

Figura 4.3: AFF contra otros métodos de estimación

4.5. Conclusiones

En este capítulo se ha propuesto un nuevo método denominado Adaptive Forgetting Factor (AFF) para la estimación del throughput en los reproductores de un servicio de distribución multimedia. *AFF* tiene como objetivo seleccionar correctamente la representación del contenido multimedia cuya tasa de bits coincide con las condiciones de ancho de banda que ofrece la red para lograr la mayor calidad en la reproducción de vídeo. Esto se logra estimando de forma adaptativa el rendimiento de la transmisión del vídeo y colocando mayores pesos en las observaciones más recientes cuando se encuentran fluctuaciones a corto plazo, lo cual resulta en un método que “olvida” las mediciones antiguas rápidamente.

En la sección 4.2 se presentaron las definiciones necesarias para sustentar un método que utiliza la detección de puntos de cambio para identificar momentos en que la distribución de probabilidad de un proceso cambia. Después se definieron los conceptos necesarios para proponer un método matemático que utiliza adaptive forgetting factor (*AFF*) para el cálculo de la estimación del throughput promedio utilizando las medidas del throughput instantáneo.

En la sección 4.3 se mostró el pseudo-código creado con el fin de ser implementado en el lenguaje de programación del reproductor de contenido multimedia basado en el estándar *DASH*. El algoritmo describe todos los pasos necesarios para el funcionamiento del método *AFF*.

En la sección 4.4 se presentó una comparativa del algoritmo *AFF* con otros métodos co-

munes de estimación usados por la comunidad científica para realizar cálculos suavizados del ancho de banda que detecta el reproductor. También en la sección 4.4 se presentaron las ventajas de utilizar el método *AFF* dentro de los reproductores *DASH*.

En el siguiente capítulo, se mostrarán las herramientas seleccionadas para la implementación del algoritmo descrito en este capítulo, así como los escenarios en que se evaluará el funcionamiento del algoritmo de adaptación comparándolo con otros algoritmos propuestos por la comunidad científica.

CAPÍTULO 5

ESCENARIOS Y MEDIDAS DE VALIDACIÓN

5.1. Introducción

En este capítulo se exponen los escenarios y procedimientos de validación con los cuales se evaluará el servicio de distribución de contenido multimedia, particularmente, el método *AFF* propuesto en el capítulo 4 de este documento. La plataforma para evaluar el servicio de distribución de contenidos multimedia que se presenta en este capítulo corresponde con trabajos realizados en el marco del proyecto “LTExtreme: Optimización de Servicios Multi-usuario y Multimedia sobre LTE y LTE–Advanced” (IPT-2012-0525-430000) del programa INNPACTO, financiado por el Ministerio de Economía y Competitividad del Gobierno de España.

En el apartado 5.2 se describen los requisitos necesarios para evaluar el servicio de distribución multimedia la elección y creación del contenido multimedia en el formato *DASH*, y la elección del reproductor *DASH*, en el que se implementará el algoritmo *AFF* para estimar el throughput. El apartado 5.3 describe el desarrollo de la plataforma virtualizada para la validación del algoritmo propuesto en capítulo 4. En el apartado 5.4 se definen los distintos escenarios propuestos para realizar las pruebas con las que se validará el algoritmo *AFF*. Por último, en el apartado 5.5 se presentan las conclusiones del capítulo.

5.2. Requisitos para evaluar el servicio de distribución multimedia

Para evaluar el servicio de distribución multimedia utilizando el estándar *DASH*, es necesario definir los requisitos básicos necesarios para implementar dicho servicio. Dentro del estándar *DASH* se definen dos elementos indispensables para ofrecer el servicio de distribución de contenido multimedia, que son: el archivo de metadatos *MPD* y las características de codificación soportadas por el estándar *DASH*.

Otro elemento indispensable es el reproductor del contenido multimedia que se ejecuta en el cliente. En el estándar *DASH*, como ya se explico en el capítulo 2, no se define el algoritmo de tasa adaptativa implementado en el reproductor, sino que se establecen guías del comportamiento esperado del algoritmo.

5.2.1. Contenido DASH

Como se menciona anteriormente, el contenido multimedia es una parte importante dentro del estándar *DASH*. En esta investigación fueron seleccionados dos vídeos elaborados por la “Blender Foundation”, los cuales son: “Big Buck Bunny” [Blender08] y “Sintel Movie Trailer” [Blender10]. Ambos vídeos son cortos animados por computadora y fueron realizados utilizando el software llamado “Blender” desarrollado por la fundación, que es de distribución libre.

El corto Big Buck Bunny tiene una duración de 10 minutos y el corto Sintel Trailer tiene una duración de 1 minuto. Ambos cortos fueron generados a 24 *Frames per second (FPS)* en varias resoluciones (1080p, 720p, etc.). Los dos vídeos fueron descargados en su versión “lossless” (sin perdida) con extensión Y4M. El vídeo Big Buck Bunny fue descargado a una resolución de 720p y el vídeo Sintel Trailer a una resolución de 480p.

Para preparar los vídeos se utilizó la herramienta X.264 [Merritt06], para codificar el vídeo del formato Y4M al formato H264 “*Variable Bitrate (VBR)*” utilizando la compresión “*Advanced Video Coding (AVC)*” del estándar H.264. La herramienta X.264 también sirvió para generar distintas representaciones en las cuales cambia la calidad del vídeo (bitrate). Para Big Buck Bunny se crearon 4 representaciones (250, 500, 1000 y 2000 Kbps) y para el vídeo Sintel Trailer se crearon 3 representaciones (300, 600 y 1200 Kbps).

Después de generar el vídeo con los distintos bitrates, se procedió a generar el contenido *DASH* utilizando la herramienta MP4Box [Feuvre17]. La herramienta MP4Box genera el

archivo de metadatos *MPD* y encapsula en segmentos el vídeo completo para cada calidad creada con la herramienta X.264. Para el desarrollo de esta investigación se seleccionaron segmentos de vídeo con duración de 2 segundos y el perfil para vídeo bajo demanda.

En la tabla 5.1 se muestran las características de codificación del contenido multimedia generado para la evaluación del servicio de distribución multimedia.

Tabla 5.1: Características del vídeo codificado para *DASH*

Nombre	Resolución	FPS	Bitrate(kbps)	Codec	Level	Preset	Profile
Big Buck Bunny	720p	24	2000	H.264/AVC	3.1	Medium	High
	720p	24	1000	H.264/AVC	3.1	Medium	High
	720p	24	500	H.264/AVC	3.1	Medium	High
	720p	24	250	H.264/AVC	3.1	Medium	Main
Sintel Trailer	480p	24	1200	H.264/AVC	3.1	Slow	Baseline
	480p	24	600	H.264/AVC	3.1	Slow	Baseline
	480p	24	300	H.264/AVC	3.1	Slow	Baseline

5.2.2. Reproductor DASH

El reproductor del contenido multimedia para el estándar *DASH* es otra parte importante para evaluar el servicio de distribución de contenido multimedia. En esta investigación se decidió utilizar el reproductor de referencia del foro DASH-IF [DASH-IF18b] llamado “*dash.js*”.

Este reproductor, está programado en el lenguaje JavaScript y utiliza el “Media Source Extension” (MSE) del W3C para reproducir el contenido multimedia en el navegador web. El reproductor *dash.js* se distribuye bajo la licencia BSD-3, la cual permite modificar el código fuente sin necesidad de contar con licencias. El reproductor *dash.js* fue seleccionado por la facilidad de modificar el código JavaScript y la portabilidad que presenta, ya que el MSE se encuentra disponible con casi todos los navegadores *HTTP* para computadoras de escritorio que existen en el mercado (Edge, Firefox, Chrome, Safari) y para algunas versiones de navegadores en el sistema operativo Android.

Para evaluar la propuesta, se ha modificado el reproductor *dash.js* para integrar el algoritmo *AFF* descrito en el capítulo 4. En la figura 5.1 se presenta el diagrama de bloques del mecanismo de adaptación implementada en el reproductor *dash.js*.

Este mecanismo de adaptación consiste en cuatro elementos que son: la entidad de reglas, la lógica de adaptación, el módulo de la estimación del throughput y el módulo de control del buffer. La entidad de las reglas es la encargada de definir todos los parámetros

Figura 5.1: Diagrama de bloques del mecanismo de adaptación de *dash.js*.

necesarios para iniciar la reproducción del contenido multimedia, por ejemplo, el nivel mínimo de buffer (*MinBufferLevel*). La lógica de adaptación es la encargada de planificar las peticiones de nuevos segmentos de contenido multimedia y se alimenta de los módulos del throughput, buffer y reglas.

El módulo del throughput es el encargado de estimar el ancho de banda que detecta el reproductor. Esto se realiza calculando el throughput instantáneo para cada segmento de contenido multimedia descargado. El algoritmo que implementa el reproductor *dash.js* para estimar el throughput es el método “media móvil simple”, evaluando los últimos 3 segmentos descargados. En este módulo se ha implementado el algoritmo *AFF* y también se ha implementado el método *EWMA* con un valor de peso exponencial $\lambda = 0.2$ como se recomienda en [Akhshabi11] y [Thang13] cuando se utilizan las redes de acceso móviles.

El módulo del buffer es el encargado de medir los niveles del buffer del reproductor. El algoritmo que implementa el reproductor *dash.js* para controlar el buffer es un método heurístico basado en el establecimiento de niveles como en [Sun16] y [Miller12], en el que cuando el nivel del buffer baja de 8 segundos, se empiezan a planificar bitrates con calidades más bajas independientemente de las medidas que se obtienen del algoritmo de estimación del throughput.

5.3. Diseño y desarrollo de la plataforma de validación

En esta sección se describe la plataforma creada para evaluar el servicio de distribución multimedia utilizando el estándar *DASH*.

5.3.1. Antecedentes

Este trabajo de tesis doctoral se enmarca en los trabajos de investigación realizados en el proyecto INNPACTO “LTeXtreme: Optimización de Servicios Multi-usuario y Multimedia sobre LTE y LTE-Advanced” (IPT-2012-0525-430000). Dentro del proyecto, el objetivo de trabajo (PT) 3 consistía en evaluar tecnologías y protocolos para la distribución de contenido multimedia mediante streaming.

Para poder lograr el objetivo mencionado anteriormente se establecieron cuatro tareas primordiales, las cuales fueron: evaluar las tecnologías y protocolos para la distribución mediante streaming, definir modelos de distribución, simular los modelos para su validación y realizar un análisis de los resultados y medidas de prestación del servicio. Para dar cumplimiento a estas dos últimas tareas, se creó una plataforma virtualizada, la cual se muestra en la figura 5.2. Esta plataforma virtualizada se basa en el artículo escrito por Lecompte *et. al.* [Lecompte12], en donde se proponen de manera general los componentes necesarios para proporcionar un servicio de difusión multimedia en redes móviles utilizando el servicio *eMBMS* de la red *LTE*, descrito en el capítulo 2.

Figura 5.2: Plataforma Virtualizada para Difusión

Esta plataforma, presentada en [Lentisco15], se creó para evaluar los servicios de difusión de contenido multimedia, es decir, el servicio en que el contenido multimedia se envía a todos los receptores conectados. En la plataforma se implementa un servidor y un cliente “*FLUTE*” basado en el software de código libre *MAD-FLUTE* [Peltotalo07]. El servicio *flute* se define mediante un documento *XML* llamado “*File Delivery Table (FDT)*” el cual asigna un “*Transport Object Identifier (TOI)*” a cada archivo enviado. Es por ello por lo que se desarrollaron scripts que automatizan el proceso de leer el archivo *MPD* del contenido *DASH* y convertirlo a un documento *FDT*.

En el cliente se implementó un servicio de cache basado en el servidor proxy Squid. Este servidor se encarga de validar si los segmentos de contenido multimedia transmitidos por *FLUTE* han llegado correctamente a través de la sesión multicast. En caso de que no se hayan recibido correctamente, el servidor proxy Squid se encarga de recuperar esos segmentos utilizando la conexión a Internet del cliente. Igualmente, para obtener este comportamiento se crearon scripts que modifican el comportamiento del servidor proxy Squid.

5.3.2. Plataforma virtualizada

Para evaluar el servicio de distribución multimedia utilizando el estándar *DASH*, fue necesario crear una plataforma virtualizada, que se basa en el modelo “cliente-servidor”. Esta plataforma se desarrolló de forma virtualizada para poder ejecutarse sobre sistemas operativos diferentes (portabilidad) utilizando una máquina virtual basada en “Open Virtual Appliance (OVA)”, formato que es acorde al estándar abierto “Open Virtual Format (OVF)” creado por la “Distributed Management Task Force (DMTF)”. La OVA utiliza el sistema operativo Linux en versión Light Ubuntu 14.04. Para ejecutar la OVA se ha seleccionado el software de Oracle llamado “VirtualBox” [Oracle17].

Dentro de la OVA, se ha creado un escenario utilizando una herramienta de código abierto llamada “*Virtual Networks over linuX (VNX)*” [Fernández16]. Esta herramienta se basa en el despliegue de escenarios virtuales definidos por un documento *XML* que se despliega mediante el uso de contenedores virtuales “*LinuX Container (LXC)*”. El archivo *XML* define el número de elementos de red, las interconexiones entre estos elementos y comandos que ejecutan tareas específicas.

Los contenedores *LXC* se consideran una virtualización del nivel del sistema operativo para ejecutar múltiples sistemas en una máquina (host) de control. La principal ventaja de los contenedores *LXC* es que se pueden implementar desde el sistema operativo del host. Un inconveniente de los contenedores *LXC* es que solo ejecutan sistemas operativos Linux.

Como se mencionó anteriormente una de las ventajas de utilizar la herramienta *VNX* es la de poder automatizar la ejecución de comandos y la copia de archivos, para ello se definen etiquetas en el archivo *XML*. Otra ventaja de utilizar la herramienta *VNX* es que define un método para agrupar la ejecución de secuencias de comandos con el fin de personalizar la máquina virtual al inicio del escenario.

En la figura 5.3 se muestran los elementos que componen al escenario propuesto para evaluar los servicios de distribución multimedia, específicamente el algoritmo de estimación del throughput *AFF* definido en el capítulo 4. El escenario se compone de tres elemen-

Figura 5.3: Plataforma Virtualizada para Distribución

tos virtuales, los cuales son: el servidor, un conformador de tráfico (shaper) y el cliente.

El servidor almacena el contenido multimedia definido anteriormente en la tabla 5.1. También se encarga de implementar un servidor *HTTP*, basado en la aplicación Apache 2 desarrollada por la “*Apache Software Foundation*”, para entregar el contenido multimedia al cliente de la plataforma virtualizada.

El shaper se comporta como un emulador de red que modifica las condiciones de la red en tres aspectos: ancho de banda disponible, retardo de red y pérdida de paquetes. El shaper se basa en un “shell script” de Linux que lee un archivo de perfil de tráfico de la red, el cual contiene valores separados por comas o “*Comma-Separated Values (CSV)*” que definen las condiciones de la red en diferentes períodos de tiempo. Este archivo se usa para cambiar la configuración de las interfaces de red durante los experimentos para evaluar los servicios de distribución multimedia. El shaper utiliza las herramientas de control de tráfico (TC) y emulación de red (NetEm) de Linux. Más específicamente, el shaper utiliza la disciplina de cola “Hierarchy Token Bucket (HTB)” para clasificar el tráfico *TCP* entrante y aplicar los parámetros de red leídos desde el archivo que define el perfil tráfico de la red [Fuentes09].

El cliente incluye un navegador web Chrome que descarga el reproductor *dash.js* del servidor, para después solicitar una sesión *DASH* y así acceder a los segmentos de contenido multimedia almacenados en el servidor. El reproductor también es responsable de recopilar los datos que se utilizan para analizar el comportamiento del algoritmo de estimación de rendimiento, es decir, el tamaño del buffer, el bitrate del vídeo y el throughput instantáneo calculado.

En la siguiente sección se explican los escenarios propuestos para evaluar el servicio de distribución de contenido multimedia, en particular para validar el algoritmo *AFF*.

5.4. Escenarios de evaluación

Una vez definida la plataforma es necesario proponer escenarios con los cuales evaluar el servicio de distribución de contenido multimedia, y en concreto el algoritmo *AFF* propuesto en el capítulo 4 de esta investigación.

Dado que las redes de acceso fijas (ADSL, fibra, etc.) poseen una elevada capacidad para transportar información (ancho de banda), se puede prestar el servicio de distribución de contenidos multimedia fácilmente. Sin embargo, el problema aparece cuando se utilizan redes de acceso móviles [Alejos12].

En las redes móviles el canal radio presenta un comportamiento temporal aleatorio y multitrayectos, lo que ocasiona interferencia con los símbolos de las modulaciones digitales. Estos fenómenos del canal radio provocan fluctuaciones a corto plazo en el ancho de banda, lo cual afecta a los algoritmos de tasa adaptativa implementados en el estándar *DASH*. Esto se debe a que, para el reproductor, al haber fluctuaciones a corto plazo (desvanecimientos) en el ancho de banda, es más difícil estimar de manera efectiva el throughput que ofrece la red de acceso para que el algoritmo de tasa adaptativa seleccione el bitrate del contenido multimedia más apropiado para el cliente.

Por lo tanto, en esta investigación, se proponen escenarios para evaluar el servicio de distribución de contenido multimedia utilizando perfiles de tráfico de una red de acceso móvil. A continuación, se describen los escenarios propuestos para evaluar al algoritmo de tasa adaptativa.

- Tres escenarios con perfiles de tráfico de una red móvil (*LTE*) real.
- Un escenario con un perfil de tráfico artificial creado para estudiar el handover vertical.
- Un escenario con un perfil de tráfico artificial para medir la equidad (fairness) de varios reproductores.

En los siguientes apartados se presentan los escenarios y los perfiles creados para evaluar el servicio de distribución de contenido multimedia.

5.4.1. Perfiles de tráfico de una red móvil (*LTE*) real.

Los primeros tres escenarios propuestos utilizan perfiles de tráfico de una red de acceso móvil real, obtenidos utilizando versiones anteriores de la plataforma descrita en la sección

anterior. Estos perfiles de tráfico fueron capturados durante una serie de visitas sucesivas a las instalaciones de la empresa Alcatel-Lucent (ALU) durante el desarrollo del proyecto “LTextreme”. Estas visitas tuvieron la finalidad de acceder a un piloto de demostración de la red *LTE* con equipo físico de la empresa ALU.

Para conectar cliente y servidor a través del piloto de demostración *LTE*, fue necesario separar la plataforma virtualizada (figura 5.2) en dos *OVA*s, una *OVA* en donde se ejecuta el cliente, y otra *OVA* en donde se ejecuta el servidor. De esta forma, es posible ejecutar cliente y servidor en máquinas físicas diferentes y así interconectarlas con el piloto de demostración de la red *LTE*, como muestra la figura 5.4. Esta plataforma separada en cliente-servidor fue otra de las contribuciones al proyecto “LTextreme: Optimización de Servicios Multi-usuario y Multimedia sobre *LTE* y *LTE-Advanced*” plasmada en el documento del proyecto como el entregable LTextreme 3.3.

Figura 5.4: Plataforma Virtualizada para Difusión Cliente-Servidor

La máquina que ejecuta el servidor se conecta a la “*Evolved Packet Core (EPC)*” del piloto de la red *LTE*, mientras que la máquina que ejecuta el cliente se conecta a la red *LTE* a través de un modem USB *LTE*. El piloto *LTE* se compone por dos “*Evolved Node B (eNodeB)*”, cada uno de ellos con dos antenas. Un *eNodeB* se ha utilizado para transmitir la señal de vídeo. El otro se ha utilizado para transmitir una señal interferente. Así, es posible simular diferentes condiciones del canal radio *LTE*.

Durante una de las visitas realizadas a la empresa ALU se definieron una serie de experimentos para evaluar el rendimiento de un reproductor *DASH* bajo diferentes condiciones en el canal radio *LTE*. Las diferentes condiciones de canal fueron simuladas modificando atenuadores de señal manuales sobre las antenas del *eNodeB* utilizado para transmitir la señal de vídeo.

Debido a que los experimentos se realizaron conectando un único usuario a la red *LTE*,

Tabla 5.2: Atenuación de las antenas para piloto de red *LTE*

Atenuación Antena 1 (dB)	Atenuación Antena 2 (dB)
15	15
20	15
22	20
25	20

ese usuario contó con todos los recursos disponibles del *eNodeB* por lo que el ancho de banda máximo era más alto que los bitrates del contenido multimedia que se creó (ver tabla 5.1). Con el objetivo de limitar el ancho de banda, se redujo el número de bloques de recursos físicos o *PRB* que un *eNodeB* puede utilizar para transmitir información. Limitar el número de *PRB* a 9 permite al cliente obtener anchos de banda cercanos a los rangos de los bitrates definidos para el contenido multimedia.

Los experimentos consistieron en establecer cuatro sesiones de tráfico *TCP* utilizando el software *IPERF* para cada prueba, mientras se simulaban diferentes condiciones del canal radio modificando el valor de las atenuaciones para las antenas, como se muestra en la tabla 5.2. Estos valores de atenuación permanecieron fijos durante la duración de cada experimento, la cual fue de 300 segundos para cada sesión.

De los cuatro experimentos realizados, se seleccionaron 3 perfiles de tráfico, los cuales muestran patrones adecuados para evaluar los servicios de distribución multimedia en redes móviles (*LTE*). Los perfiles de tráfico seleccionados se muestran en la figura 5.5.

Tabla 5.3: Estadísticas de los perfiles de tráfico de reales en Mbps

Escenario	Tipo de Perfil	Máx.	Mín.	Prom.	σ
(Mbps)					
Escenario 1	High bandwidth, low fluctuation.	2.40	0.80	2.17	0.2765
Escenario 2	Low bandwidth, High fluctuation.	4.57	0.01	1.23	0.6374
Escenario 3	High bandwidth, High fluctuation.	5.73	0.01	2.31	1.3317

En la tabla 5.3 se muestran las estadísticas principales para cada uno de los perfiles de tráfico utilizados para evaluar el rendimiento del reproductor *DASH* en redes móviles. Las estadísticas que se muestran son: máximo, mínimo, promedio y desviación estándar (σ) para cada perfil de tráfico.

Figura 5.5: Perfiles de tráfico de una red *LTE* real.

A continuación, se explican estos escenarios con los cuales se pretende evaluar y validar al algoritmo de estimación del throughput *AFF* utilizando perfiles de tráfico de una red móvil (*LTE*):

- El primer perfil de tráfico se seleccionó para representar el comportamiento de una conexión móvil estable, es decir, un escenario en el que se usa el máximo ancho de banda del perfil de tráfico, y en el que el reproductor *DASH* podría descargar el bitrate más alto del contenido multimedia casi todo el tiempo.
- El segundo perfil de tráfico tiene como objetivo estudiar el comportamiento de los algoritmos de tasa adaptativa del reproductor *DASH* para anchos de banda limitados y que presentan severas fluctuaciones a corto plazo. Con este perfil de tráfico se pretende evaluar un escenario en donde se busca saber si los mecanismos de estimación del throughput y el control del buffer son capaces de ofrecer un servicio continuado en la reproducción del contenido multimedia.
- El tercer perfil de tráfico presenta un patrón similar al perfil de prueba uno (TP1)

definido en el artículo de Kim *et al.* [Kim13], el cual representa a un usuario que camina durante el día en un entorno urbano. Con este perfil de tráfico se pretende evaluar un escenario que presenta anchos de banda altos y fluctuaciones severas a corto plazo. Una de las características principales de este perfil de tráfico es que por un periodo de tiempo largo (segundo 60) el ancho de banda se queda en el valor mínimo (0.1 Mbps).

5.4.2. Perfiles de tráfico artificiales

Los siguientes dos escenarios propuestos utilizan perfiles de tráfico creados artificialmente utilizando el software MATLAB. Esta herramienta es una plataforma de programación diseñada específicamente para ingenieros y científicos. Utilizando la herramienta MATLAB se puede analizar datos, desarrollar algoritmos, crear modelos y aplicaciones para resolver problemas matemáticos complejos. Estos perfiles de tráfico creados artificialmente se muestran en la figura 5.6.

(a) Low-High bandwidth, low fluctuation: Scenario 4

(b) Fairness Analysis: Scenario 5

Figura 5.6: Perfiles de tráfico artificiales.

A continuación, se explican estos escenarios artificiales, con los cuales se pretende evaluar y validar al algoritmo de estimación del throughput *AFF*:

- El cuarto perfil de tráfico fue creado para evaluar el comportamiento del reproductor *DASH* cuando se presenta un escenario de cambio repentino en el ancho de banda, como podría suceder en un evento de handover vertical. Esto permite evaluar cómo un cambio brusco en el ancho de banda afecta a los algoritmos que utilizan el promedio de mediciones pasadas para estimar el throughput.

- El quinto perfil de tráfico fue creado para evaluar el comportamiento del reproductor *DASH* cuando se presenta un escenario en el que varios reproductores *DASH* compiten por el ancho de banda en un cuello de botella y con esto evaluar la equidad del algoritmo de estimación del throughput *AFF* con respecto al ancho de banda disponible.

En la tabla 5.4 se muestran las estadísticas principales para cada uno de los perfiles de tráfico utilizados para evaluar el rendimiento del reproductor *DASH* en los escenarios definidos anteriormente, máximo, mínimo, promedio y desviación estándar.

Tabla 5.4: Estadísticas de los perfiles de tráfico artificiales en Mbps

Escenario	Tipo de Perfil	Max.	Min.	Prom.	σ
(Mbps)					
Escenario 4	High-low bandwidth, low fluctuation.	2.39	0.60	1.50	0.8063
Escenario 5	Fairness Analysis.	22	6	14.77	6.8453

Para evaluar el algoritmo *AFF* en el escenario 5 se creó una plataforma utilizando equipos reales. Esta plataforma contiene los mismos elementos que la plataforma virtualizada definida en la figura 5.3. Para implementar el servidor *HTTP* se utilizó una máquina con el sistema operativo Linux Ubuntu versión 17, se instaló el software Apache 2.0 y se copió el mismo contenido creado para la plataforma virtualizada, el cual se muestra en la tabla 5.1. El shaper se implementó en una Raspberry Pi 3 con dos interfaces de red. El shaper presenta los mismos elementos que se utilizan en la plataforma virtualizada, solo que se usa el escenario 5 como perfil de tráfico. En esta plataforma real se han utilizado 10 máquinas físicas con Linux Ubuntu versión 17 utilizando el navegador Chrome, el cual descarga el reproductor *dash.js* del servidor *HTTP*, para empezar la sesión *DASH*.

Figura 5.7: Escenario real para evaluar la equidad.

En la figura 5.7 se muestra el esquema de esta plataforma real para evaluar la equidad del algoritmo de estimación *AFF* cuando varios reproductores compiten por el mismo

ancho de banda en un “cuello de botella”.

5.5. Conclusiones

En este capítulo se propusieron diferentes escenarios y una plataforma virtualizada para evaluar los servicios de distribución de contenido multimedia. Además, se describen las características del contenido multimedia creado en el formato *DASH* y, se ha elegido un reproductor *DASH* en el cual se ha implementado el método propuesto en el capítulo 4 de esta investigación. Los escenarios propuestos para evaluar los servicios de distribución multimedia basados en *DASH* constan de perfiles de tráfico que fueron obtenidos usando una red móvil *LTE* real. También, se crearon otros dos perfiles de tráfico artificiales para evaluar, por un lado, un evento de handover vertical, y por el otro, un evento en el que varios reproductores compiten por el mismo ancho de banda, para medir la equidad en la repartición de este.

En el apartado 5.2, se definieron tanto el contenido multimedia como el reproductor elegido para evaluar los servicios de distribución multimedia. En esta sección se da una breve descripción de las herramientas utilizadas para crear el contenido multimedia, así como del proceso. También se explica el funcionamiento general del reproductor *dash.js*, en el que se integra el algoritmo *AFF* propuesto en el capítulo 4.

En el apartado 5.3, se detalló la plataforma virtualizada que se creó para evaluar los servicios de distribución multimedia, así como las tecnologías que posibilitaron su implementación y las diferentes partes que la integran, explicando las funciones que desempeñan dentro de la plataforma virtualizada.

En el apartado 5.4, se definieron los escenarios elegidos para evaluar el servicio de distribución multimedia. Estos escenarios, se basan en tres perfiles de tráfico que provienen de mediciones reales capturadas en una prueba piloto de una red móvil *LTE* real y dos perfiles de tráfico creados artificialmente, uno para evaluar que sucede en un evento de handover y el otro para analizar la equidad de varios reproductores *DASH*.

En el siguiente capítulo, se presentan los resultados obtenidos al evaluar los servicios de distribución multimedia utilizando las plataformas descritas en este capítulo. Además, dicho capítulo incluye las pruebas de rendimiento, las pruebas de equidad (fairness) y los resultados de *QoE* obtenidos en este trabajo de investigación.

6.1. Introducción

En este capítulo se presentan los resultados de la evaluación del método *AFF* para la estimación del throughput en los reproductores de un servicio de distribución multimedia. Para validar al método *AFF* es necesario compararlo con otros métodos de estimación comúnmente utilizados, como son el método “media móvil simple” y el método *EWMA*. El método “media móvil simple”, utiliza un número de mediciones anteriores para realizar la estimación del throughput. El método *EWMA*, se utiliza comúnmente en implementaciones de algoritmos *DASH* definiendo valores fijos para los pesos exponenciales que aplica. Los resultados que se muestran en este capítulo se obtuvieron al evaluar el método *AFF* en los escenarios propuestos en el capítulo 5, utilizando las plataformas virtualizadas y reales con los perfiles de tráfico de redes móviles definidos en el mismo capítulo.

En la sección 6.2 se presentan las pruebas de rendimiento de los métodos evaluadas en los tres primeros escenarios propuestos en el capítulo 5, los cuales se obtuvieron al realizar pruebas en una red móvil 4G *LTE* real, y también, se muestra la prueba de rendimiento en el escenario 4, creado de manera artificial para evaluar un evento de handover vertical. La sección 6.3, presenta las pruebas de equidad (fairness) entre varios reproductores *DASH* al competir por el mismo ancho de banda en un “cuello de botella” (bottleneck). En la sección 6.4, se presentan los resultados de las métricas de QoE de los tres métodos de estimación del throughput evaluados en los cuatro primeros perfiles de tráfico (escenarios). Por último, en la sección 6.5 se presentan las conclusiones de este capítulo.

6.2. Evaluación del Rendimiento

Para validar el método de estimación del throughput *AFF* es necesario realizar pruebas de rendimiento con otros métodos de estimación comúnmente utilizados por algoritmos *DASH*. En el capítulo 3, se han presentado diversos algoritmos de adaptación para el estándar *DASH*, en ellos se implementan varios métodos de estimación del throughput siendo los más comunes *EWMA* y la “media móvil simple”.

El reproductor *DASH* seleccionado en esta investigación para realizar las pruebas de rendimiento implementa el método “media móvil simple” de los últimos $N = 3$ segmentos y cuyo funcionamiento se define en la ecuación 6.1. De aquí en adelante, este método se denominará *avg-last-3*.

$$\overline{THR}_N = \frac{THR_N + THR_{N-1} + \dots + THR_{N-(N-1)}}{N} = \frac{1}{N} \left(\sum_{i=1}^{N-1} THR_{N-i} \right) \quad (6.1)$$

El método *EWMA* utiliza pesos exponenciales para realizar la estimación del throughput, como se muestra en la ecuación 6.2. En esta investigación se utilizará el peso exponencial $\alpha = 0.2$ como se propone en [Akhshabi11], [Thang12] y [Kim13].

$$EWMA_N = (1 - \alpha)\overline{EWMA}_{N-1} + \alpha \cdot THR_N. \quad (6.2)$$

Las primeras tres pruebas consistieron en evaluar cada método de estimación por separado utilizando el perfil de tráfico para cada escenario dentro de la plataforma virtualizada definida en el capítulo 5. Cada prueba tuvo una duración de 100 segundos utilizando el perfil de tráfico correspondiente. Al final de la prueba se recuperaron las estadísticas medidas por el reproductor *dash.js*.

La cuarta prueba consistió en evaluar cada método de estimación por separado utilizando el perfil de tráfico correspondiente dentro de la plataforma virtualizada definida en el capítulo 5. Esta prueba tuvo una duración de 200 segundos utilizando el perfil de tráfico correspondiente al escenario cuatro. Al final de esta prueba se recuperaron las estadísticas medidas por el reproductor *dash.js*.

En esta sección se presenta el análisis de los tres métodos de estimación del throughput utilizando los primeros cuatro perfiles de tráfico. El análisis para cada escenario incluye dos figuras, una en la que se muestra el comportamiento de los algoritmos de estimación del throughput y el peso exponencial adaptativo del algoritmo *AFF* llamado λ y una segunda

figura que muestra los resultados del algoritmo de tasa adaptativa (throughput y buffer) en la forma de una Función de Distribución Acumulada o “*Cumulative Distribution Function (CDF)*”. Esta segunda figura se divide en dos partes, la primera, muestra las estadísticas de elección de la calidad del vídeo (bitrate) en forma de porcentajes y la segunda, muestra las estadísticas en términos de porcentaje de la cantidad de tiempo que acumuló el buffer.

A continuación, se presentan las pruebas de rendimiento realizadas para evaluar a los distintos métodos de estimación utilizando los cuatro primeros escenarios definidos en el capítulo 5.

6.2.1. Escenario 1

El escenario 1 se creó para evaluar los algoritmos de estimación del throughput cuando un usuario utiliza como red de acceso una red móvil LTE en la que existe una conexión estable con un ancho de banda alto. En esta sesión del estándar *DASH* el reproductor puede recuperar la mejor calidad (bitrate) del vídeo la mayor parte del tiempo, ya que presenta durante los 100 segundos de la prueba un ancho de banda máximo de 2.4 Mbps casi todo el tiempo.

La figura 6.1 presenta las medidas del throughput instantáneo que calcula el reproductor *DASH*. Además, presenta los valores de estimación para cada método y el valor de la variable λ del método *AFF*. Dado que *AFF* y *EWMA* comparte un mecanismo de pesos exponenciales, los métodos presentan un comportamiento similar, diferente al método *avg-last-3*, el cual calcula el valor promedio de las últimas tres mediciones.

La figura 6.1 también muestra la evolución del factor λ del método *AFF* durante la sesión de reproducción del vídeo usando el estándar *DASH*. Cuando el throughput instantáneo se comporta de manera constante toma el valor $\lambda = 1$, el valor más alto. Pero cuando hay una fluctuación a corto plazo del throughput instantáneo, λ cae al valor más bajo de $\lambda = 0.6$, lo que hace que el método *AFF* olvide las últimas mediciones más rápido.

La figura 6.2 presenta la CDF de la selección en la calidad (bitrate) del vídeo para cada algoritmo. En ella se observa que el método de estimación del throughput *AFF* muestra un comportamiento más estable que los otros algoritmos, al haber obtenido la calidad (bitrate) del vídeo más alta durante la mayor parte de la reproducción.

La figura 6.2 también presenta la CDF del nivel del buffer del algoritmo de tasa adaptativa para cada uno de los métodos de estimación. Esta figura muestra que el método de estimación del throughput *AFF* presenta un comportamiento más estable que los otros algoritmos dado que el nivel del buffer se mantiene por encima de 10 segundos casi el 90 %

Figura 6.1: Throughput estimado de los métodos y lambda en el Escenario 1.

Figura 6.2: (a) CDF del bitrate y (b) CDF del nivel de buffer en el Escenario 1.

del tiempo. En esta misma figura se puede ver que los métodos *EWMA* y *avg-last-3* están casi un 40 % del tiempo con un nivel de buffer por debajo de 8 segundos, lo cual hace al

algoritmo de tasa adaptativa del reproductor *dash.js* solicitar calidades del vídeo más bajas independientemente del resultado del método de estimación del throughput.

6.2.2. Escenario 2

El escenario 2 se creó para estudiar el comportamiento de los algoritmos de tasa adaptativa del reproductor *DASH* para anchos de banda limitados y que presentan severas fluctuaciones a corto plazo. Con este perfil de tráfico se evaluó si los mecanismos de estimación del throughput y el control del buffer fueron capaces de ofrecer un servicio continuado en la reproducción del contenido multimedia.

La figura 6.3 presenta, al igual que la figura 6.1, las medidas del throughput instantáneo que calcula el reproductor *DASH*, además, presenta los valores de estimación para cada método y el valor de la variable λ del método *AFF*. En esta figura se puede observar que el método *AFF* y el método *EWMA* presentan un comportamiento parecido frente al método *avg-last-3*, el cual presenta mediciones menos suavizadas del throughput estimado.

Figura 6.3: Throughput estimado de los métodos y lambda en el Escenario 2.

La figura 6.3 también muestra la evolución el factor λ del método *AFF* durante la sesión de reproducción del vídeo usando el estándar *DASH*. Con respecto a la figura 6.1 se puede observar que el throughput instantáneo muestra un comportamiento más inestable debido

a las fluctuaciones a corto plazo que presenta el perfil de tráfico para el escenario 2. Por lo tanto, el método *AFF* muestra más tiempo el valor bajo del factor λ , que es $\lambda = 0.6$.

Figura 6.4: (a) CDF del bitrate y (b) CDF del nivel de buffer en el Escenario 2.

La figura 6.4 presenta la CDF de la selección en la calidad (bitrate) del vídeo para cada algoritmo. Se puede observar que el método de estimación del throughput *EWMA* muestra un mejor comportamiento debido a que casi un 40 % del tiempo solicitó una calidad más alta con respecto a los métodos *AFF* y *avg-last-3*. Sin embargo, al realizar el análisis final se puede observar que para el escenario 2 el throughput promedio del método *EWMA* es similar al obtenido con los métodos *AFF* y *avg-last-3*.

La figura 6.4 también presenta la CDF del nivel del buffer del algoritmo de tasa adaptativa para cada uno de los métodos de estimación. Esta figura muestra que el método de estimación del throughput *EWMA* presenta un comportamiento más estable, dado que casi el 70 % del tiempo el nivel del buffer estuvo por encima de los 10 segundos. También en esta misma figura se puede ver que casi un 60 % del tiempo el nivel del buffer está por debajo de 8 segundos en los métodos *AFF* y *avg-last-3*, lo cual hace al algoritmo de tasa adaptativa del reproductor *dash.js* solicitar calidades del vídeo más bajas independientemente del resultado del método de estimación del throughput.

6.2.3. Escenario 3

El escenario 3 se creó para estudiar un patrón similar al perfil de prueba uno (TP1) definido en el artículo de Kim *et al.* [Kim13], el cual representa a un usuario que camina durante el día en un entorno urbano. Con este perfil de tráfico se pretende evaluar un escenario que presenta anchos de banda altos y severas fluctuaciones a corto plazo. Una de las características principales del perfil de tráfico es que por un periodo de tiempo largo (del segundo 54 al segundo 63) el ancho de banda se queda en el valor mínimo (0.1 Mbps).

La figura 6.5 presenta al igual que las figuras anteriores las medidas del throughput instantáneo que calcula el reproductor *DASH*, los valores de estimación para cada método y el valor de la variable λ del método *AFF*. En esta figura se puede observar que el método *AFF* y el método *EWMA* presentan un comportamiento parecido, mientras que el método *avg-last-3* presenta mediciones menos suavizadas del throughput estimado sobre todo cuando existen variaciones severas del ancho de banda.

Figura 6.5: Throughput estimado de los métodos y lambda en el escenario 3.

La figura 6.5 también muestra la evolución el factor λ del método *AFF* durante la sesión de reproducción del vídeo usando el estándar *DASH*. En la figura se puede observar que cuando el ancho de banda se comporta de manera estable, el valor del factor $\lambda = 1$ se mantiene, pero cuando hay varias fluctuaciones a corto plazo del throughput, λ cae al valor más bajo $\lambda = 0.6$, lo que hace que el método *AFF* asigne pesos mucho menores a las

Figura 6.6: (a) CDF del bitrate y (b) CDF del nivel de buffer en el Escenario 3.

mediciones anteriores.

La figura 6.6 presenta la CDF de la selección en la calidad (bitrate) del vídeo para cada algoritmo. Se puede observar que el método de estimación del throughput *AFF* muestra un comportamiento más estable al solicitar el bitrate más alto casi el 65 % del tiempo, mientras que el algoritmo *avg-last-3* lo hace solo el 55 % del tiempo y el método *EWMA* el 45 % del tiempo.

La figura 6.6 también presenta la CDF del nivel del buffer del algoritmo de tasa adaptativa para cada uno de los métodos de estimación. Esta figura muestra que el método de estimación del throughput *avg-last-3* presenta un comportamiento más estable, dado que casi el 90 % del tiempo el nivel del buffer estuvo por encima de los 10 segundos. También, en esta misma figura se puede ver que el método *AFF* tiene más de 10 segundos de nivel de buffer casi un 80 % del tiempo. El método *EWMA* está por debajo de 8 segundos de nivel del buffer casi un 40 % del tiempo, lo que significa que en esos momentos solicitó calidades más bajas del vídeo ya que el control de buffer del reproductor *dash.js* realizaba la elección del bitrate sin importar la medición del algoritmo de estimación del throughput.

6.2.4. Escenario 4

El escenario 4 se creó para evaluar el comportamiento del reproductor *DASH* cuando se presentan cambios repentinos en el ancho de banda, como podría suceder en un evento de handover vertical. Esto permite evaluar cómo un cambio brusco en el ancho de banda afecta a los algoritmos que utilizan el promedio de mediciones pasadas para estimar el throughput.

La figura 6.7 presenta las medidas del throughput instantáneo que calcula el reproductor *DASH*, los valores de estimación para cada método y el valor de la variable λ del método *AFF*. En esta figura se puede observar que el método *AFF* y el método *EWMA* presentan una diferencia en los valores sobre todo cuando el ancho de banda fluctúa severamente. El método *avg-last-3* presenta un comportamiento parecido a las medidas del throughput instantáneo lo que hace que el reproductor *DASH* cambie constantemente la calidad (bitrate) del vídeo, lo cual degrada la *QoE* del servicio de distribución de contenido multimedia.

Figura 6.7: Throughput estimado de los métodos y lambda en el Escenario 4.

La figura 6.7 también muestra la evolución el factor λ del método *AFF* durante la sesión de reproducción del vídeo usando el estándar *DASH*. En la figura se puede observar que el método *AFF* pocas veces utilizó el valor del factor $\lambda = 0.6$, con lo que este perfil de tráfico presenta un comportamiento estable la mayor parte de la prueba.

La figura 6.8 presenta la CDF de la selección en la calidad (bitrate) del vídeo para cada

Figura 6.8: (a) CDF del bitrate y (b) CDF del nivel de buffer en el Escenario 4.

algoritmo. Se puede observar, que el método de estimación del throughput *AFF* muestra un comportamiento más estable al solicitar el bitrate con la tercera calidad casi el 40 % del tiempo, mientras que el algoritmo *avg-last-3* solicita la calidad más alta el 35 % del tiempo y el método *EWMA* solicita la calidad más alta el 25 % del tiempo.

La figura 6.8 también presenta la CDF del nivel del buffer del algoritmo de tasa adaptativa para cada uno de los métodos de estimación. Esta figura muestra que los tres métodos de estimación del throughput presentan un comportamiento muy parecido, dado que casi el 60 % del tiempo el nivel del buffer estuvo por encima de los 10 segundos. También, en esta misma figura se puede ver que los tres métodos están por debajo del nivel de 8 segundos casi un 20 % del tiempo lo que hace que se soliciten calidades más bajas del vídeo.

6.3. Métricas de QoE de los métodos de estimación

Dentro de las métricas de *QoE* que se presentan en este apartado, se seleccionaron las métricas que tienen que ver con los eventos de parada (su duración en segundos) y la cantidad de eventos de cambio en la calidad (bitrate) del vídeo como proponen Seufert *et al.* [Seufert15]. En este artículo también se define la métrica del retardo inicial, pero en el

reproductor *dash.js* es constante, ya que espera a obtener 5 segmentos de la calidad más baja, lo que, utilizando segmentos de dos segundos, se traduce en un retardo inicial de 10 segundos.

En este apartado se presentan las métricas de *QoE* del algoritmo de tasa adaptativa implementado en el reproductor *dash.js*. Dentro del reproductor *DASH* se implementan dos algoritmos para controlar la adaptación, que son: el algoritmo de estimación del throughput y el algoritmo de control del buffer. Dentro del algoritmo de estimación del throughput se implementaron los métodos *AFF* y *EWMA*; el algoritmo *avg-last-3* es el método de estimación implementado por defecto dentro del reproductor *dash.js*. El algoritmo de control del buffer implementa un método heurístico basado en el cálculo de niveles para tomar las decisiones en la selección de la calidad del vídeo.

El evento de parada es el evento que más degrada la *QoE* de la reproducción del contenido multimedia [Lentisco17a]. Otro evento que degrada la *QoE* son los cambios en la calidad (bitrate) del vídeo, sobre todo cuando los cambios se realizan a peores calidades del vídeo.

Las métricas de *QoE* que se presentan en la tabla 6.1 únicamente consideran los primeros cuatro escenarios propuestos en el capítulo 5 ya que el escenario 5 está diseñado para evaluar la equidad en el ancho de banda cuando varios reproductores compiten entre ellos y por lo tanto esa prueba está diseñada para que no existan eventos de parada en la reproducción del contenido multimedia.

Tabla 6.1: Resultado de las métricas de *QoE* de los algoritmos de estimación

Escenario	Método	Cambios del Bitrate	Eventos de Parada	Duración del Evento de parada (s)	Throughput Promedio (Kbps)
Escenario 1	AFF	7	0	–	1387.00
	avg-last-3	41	0	–	1002.10
	EWMA	21	2	1.28, 0.9	822.12
Escenario 2	AFF	34	0	–	492.00
	avg-last-3	35	1	5.01	471.83
	EWMA	14	0	–	563.00
Escenario 3	AFF	8	0	–	1216.80
	avg-last-3	15	0	–	1174.00
	EWMA	25	0	–	775.00
Escenario 4	AFF	26	0	–	730.00
	avg-last-3	33	2	0.3, 2.23	880.00
	EWMA	30	0	–	814.00

La tabla 6.1 muestra, para cada uno de los métodos, un conjunto de indicadores de

calidad de experiencia: cantidad de cambios en el bitrate, cantidad de eventos de parada en la reproducción debido al agotamiento del buffer (stalling), la duración de cada evento de parada en segundos y el throughput promedio en Kbps.

En el primer escenario, el método *AFF* presentó un mejor rendimiento, ya que hubo menos cambios en el bitrate del contenido multimedia y el throughput promedio fue el más alto comparado con los otros métodos. También, muestra que el método *EWMA* presentó dos eventos de parada que duraron 1,28 y 0,9 segundos respectivamente, lo que tuvo un gran impacto en la *QoE* de la reproducción.

En el segundo perfil de ancho de banda, el método *EWMA* presentó un comportamiento más estable que los otros dos métodos; sin embargo, la tasa de bits promedio de cada método no está lejos la una de la otra. También muestra que el método *avg-last-3* presentó un evento de parada que duró 5 segundos, lo cual tiene un gran impacto en la *QoE* del servicio de distribución multimedia.

Para el tercer perfil de ancho de banda, los resultados muestran que el método *AFF* presenta menos cambios en la calidad del vídeo (bitrate) y una mejor métrica del throughput promedio. Ninguno de los métodos de estimación del throughput presenta eventos de parada, con lo que se demuestra que con un ancho de banda alto, aunque existan fluctuaciones severas, los algoritmos de estimación del throughput y de control del buffer pueden absorber las fluctuaciones.

En la cuarta prueba, el método *AFF* presenta menos cambios de velocidad de bits. Sin embargo, el método *avg-last-3* tuvo el mejor throughput promedio. El problema con el método *avg-last-3* es que presenta dos eventos de parada, el primero duró tres segundos y el segundo duró dos segundos. Esto hace que el método *avg-last-3* no sea adecuado para entornos en donde se presenta un evento de handover vertical si el ancho de banda crece o decrece de manera pronunciada.

Después de analizar las pruebas de rendimiento se puede concluir que el método de estimación *AFF* presenta los mejores resultados con respecto al throughput promedio en los escenarios 1 y 3. También, presenta el menor número de cambios en el bitrate en los escenarios 1, 3 y 4. Además, el método *AFF* no presentó eventos de parada para ninguno de los escenarios, con lo que se puede concluir que el método *AFF* es válido para ser utilizado como método de estimación del throughput en los algoritmos de adaptación dinámica para el estándar *DASH*.

6.4. Prueba de Equidad (Fairness)

En la transmisión de vídeo adaptativa, las métricas de equidad se usan para determinar si el algoritmo de adaptación es capaz de entregar una parte equitativa del ancho de banda de la red a diferentes clientes. Para analizar la equidad del método *AFF*, se creó un escenario experimental con máquinas reales, el cual se explica en el capítulo 5, para medir el ancho de banda obtenido por varios reproductores simultáneos utilizando el estándar *DASH* que comparten un enlace con un “cuello de botella”.

Los experimentos consisten en repetir cinco veces la prueba con diez reproductores *DASH* que inician la reproducción multimedia al azar dentro de los primeros 15 segundos del experimento. Los resultados se obtienen midiendo el ancho de banda promedio de cada cliente en el intervalo de 50 a 350 segundos y después promediando los resultados de las cinco pruebas. Estos valores se utilizan para calcular el “*Jain Fairness Index (JFI)*” [Jain84] y el rendimiento promedio total, como se muestra en la Tabla 6.2.

Tabla 6.2: Resultados del análisis de la equidad.

Escenario	Método	JFI	Throughput Promedio (Kbps)
Escenario 5	AFF	.9977	1267.6
	avg-last-3	.9988	1175.4
	EWMA	.9975	1182.7

Los resultados muestran que el algoritmo de estimación *AFF* ofrece resultados de equidad que son similares a los algoritmos que se analizaron en el capítulo 3, con valores de *JFI* cercanos a 1. Por tanto, el algoritmo *AFF* es adecuado para ser implementado como el mecanismo de estimación del throughput en combinación con diferentes algoritmos de adaptación del buffer.

Los resultados también muestran que el uso del método *AFF* logra alcanzar un rendimiento promedio ligeramente más alto para los clientes, ya que resulta en una selección de calidades (bitrate) del vídeo más altas que los métodos *avg-last-3* y *EWMA*.

En la figura 6.9 se muestra el valor máximo, mínimo y el promedio del *JFI* que obtuvieron los tres métodos en las cinco pruebas realizadas. Se puede observar que el método *avg-last-3* obtiene mejores valores del *JFI*, sin embargo, los métodos *AFF* y *EWMA*, también se encuentran cercanos al valor más alto del índice *JFI*, es decir, los tres proporcionan muy buenas características en cuanto a la equidad en el consumo del ancho de banda.

Figura 6.9: Diagrama de caja y bigotes para el análisis de la equidad de los tres métodos.

6.5. Conclusiones

En este capítulo se presentaron los resultados de la evaluación del método *AFF* para la estimación del throughput en los reproductores de un servicio de distribución multimedia. Para validar al método *AFF* fue necesario compararlo con otros métodos de estimación comúnmente utilizados, como son el método de media móvil simple y el método *EWMA*. Los resultados que se muestran en este capítulo se obtuvieron al evaluar el método *AFF* en los escenarios definidos en el capítulo 5 utilizando las plataformas virtualizadas y reales con los perfiles de tráfico de redes móviles y artificiales definidos en el mismo capítulo.

En el apartado 6.2, se presentaron los resultados de los tres escenarios de tráfico real de la red *LTE* y el perfil de tráfico artificial creado para evaluar el comportamiento de los métodos en un evento de handover. Estos cuatro escenarios muestran que el método *AFF* es valido para estimar el throughtput en los reproductores *DASH*. Además, de este apartado se puede concluir que el método *AFF* presenta un buen desempeño en los escenarios uno y tres, en los cuales, se analiza el comportamiento del método en el escenario donde el ancho de banda se mantiene estable y en el escenario en donde un usuario camina durante el día en un entorno urbano.

También, en el apartado 6.3, se analizaron las métricas de *QoE* de los métodos de estimación evaluados, en las que se puede mostrar que el método *AFF* presenta las mejores métricas en cuanto al menor número de cambios de bitrate. Además, se puede observar que el método *AFF* no presenta eventos de parada en ninguno de los cuatro escenarios

evaluados. También, se puede resaltar que el método *AFF* obtiene los mejores throughput promedio en dos de los escenarios evaluados.

Finalmente, en el apartado 6.4 se evaluaron las métricas de equidad, con las cuales se puede determinar si el algoritmo de adaptación puede entregar de manera justa el ancho de banda entre varios clientes reproduciendo contenido multimedia al mismo tiempo. De este apartado se puede concluir que los tres métodos analizados obtienen valores cercanos a 1, lo que significa que el ancho de banda se reparte de manera justa.

CAPÍTULO 7

CONCLUSIONES Y TRABAJOS FUTUROS

7.1. Introducción

En este capítulo se resume como se han abordado los objetivos propuestos en el trabajo de investigación. También se exponen posibles líneas futuras de investigación que surgen a raíz de este trabajo.

En el apartado 7.2 se presenta el análisis de los objetivos planteados en el capítulo 1, en donde se relacionan los objetivos propuestos y el trabajo realizado para resolverlos. En el apartado 7.3 se exponen las conclusiones generales del trabajo de investigación. El apartado 7.4 describe la divulgación de los resultados obtenidos en el trabajo de investigación en forma de artículos aceptados de revista y de congreso, como también las contribuciones a proyectos de investigación nacionales. Finalmente, el apartado 7.5 enumera las líneas futuras de investigación.

7.2. Objetivos y Resultados

Como se propuso en el capítulo 1, el objetivo general de la tesis doctoral ha consistido en mejorar la calidad percibida por los usuarios mediante la propuesta de soluciones para los servicios de distribución multimedia en redes móviles.

Para lograr el objetivo general mencionado anteriormente se propusieron dos objetivos específicos. A continuación, se resume cómo se ha abordado cada uno de los objetivos específicos y los resultados obtenidos:

1. Proponer un algoritmo de tasa adaptativa

Este objetivo específico fue abordado en el capítulo 4. Sin embargo, primero fue necesario establecer los conceptos básicos sobre el servicio de distribución de contenido multimedia, haciendo especial énfasis en los conceptos que definen el servicio basado en el estándar *DASH*, mediante la identificación de los problemas del servicio, el análisis de la tecnología y los distintos escenarios en que se distribuye el contenido multimedia a los usuarios, lo cual fue abordado en el capítulo 2. También, se analizó el funcionamiento de los mecanismos de adaptación del estándar *DASH*, las principales propuestas de los algoritmos de tasa adaptativa y los modelos de *QoE* que han sido propuestos por la comunidad científica, lo cual se plasma en el capítulo 3. Finalmente, en el capítulo 4, se presentó el método *Adaptive Forgetting Factor (AFF)*, el cual tiene como objetivo seleccionar correctamente la representación del contenido multimedia cuya tasa de bits coincide con las condiciones de ancho de banda que ofrece la red para lograr la mayor calidad en la reproducción de vídeo. Este método estima de forma adaptativa el rendimiento de la transmisión del vídeo, y al encontrar fluctuaciones a corto plazo del ancho de banda, el método aumenta el peso de las observaciones más recientes de forma que “olvida” las mediciones antiguas más rápidamente. Al comparar el funcionamiento del método *AFF* con otros métodos de pesos exponenciales como *EWMA*, se puede concluir que el método *AFF* presenta la ventaja de que no es necesario establecer parámetros fijos según las características de la red de acceso que se utiliza para el servicio de streaming multimedia.

2. Validar el algoritmo propuesto

Este objetivo específico fue resuelto en el capítulo 6. Sin embargo, primero fue necesario proponer una plataforma virtualizada para evaluar los servicios de distribución de contenido multimedia utilizando el estándar *DASH*. También, se definieron distintos escenarios con los que evaluar al algoritmo propuesto en esta investigación. Además, se seleccionó un reproductor *DASH* en el cual se implementó el método *AFF* propuesto en el capítulo 4 de esta investigación. Finalmente, en el capítulo 6, se describen los resultados de las pruebas de rendimiento efectuadas utilizando perfiles de tráfico de una red móvil y un perfil artificial para evaluar un evento de handover vertical. En estas pruebas, se ha comparado el algoritmo de estimación del throughput *AFF* con otros dos mé-

todos comúnmente utilizados en los algoritmos de estimación del throughput en *DASH*, que son: *EWMA* y la media móvil. También, se han realizado pruebas de equidad cuando un número de clientes compiten por el ancho de banda disponible en un cuello de botella y se han presentado las métricas de *QoE* obtenidas al analizar las pruebas de rendimiento.

3. Difusión de resultados

Este objetivo específico fue abordado mediante la publicación de un artículo en la revista “*IEEE Transactions on Multimedia*” [Aguayo18], la cual esta indexada en el “*Journal Citation Reports (JCR)*”. Este artículo resume la investigación realizada en la tesis doctoral, la sección dos del artículo reseña el capítulo 3, el cual analiza y critica los distintos algoritmos de adaptación dinámica propuestos por la comunidad científica para *DASH*. La sección tres del artículo resume el capítulo 4, que describe el método *AFF*, el cual es utilizado para estimar el throughput en el reproductor *DASH*. La sección cuatro sintetiza el capítulo 5, en el que se describen los escenarios y las plataformas utilizadas para evaluar el método *AFF*. La sección cinco y seis del artículo resume el capítulo 6, que presenta los resultados de la evaluación del método *AFF*. Además, se presentaron dos artículos en congresos internacionales, uno fue presentado en la “*European Conference on Networks and Communications (EuCNC)*” [Lentisco15] y el otro en “*International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*” [Bellido15]. El primer artículo de congreso analiza la plataforma virtualizada que se describe en la sección 2.4.4 y 5.3.1 del documento de tesis. Por otro lado, el segundo artículo propone un mecanismo para mejorar los eventos de handover en el servicio de difusión de contenido multimedia en una red *LTE* y comprende la sección 2.4.5 y el escenario 4 de validación de la tesis.

7.3. Conclusiones Generales

Una vez analizados los objetivos específicos se procede a plasmar las conclusiones generales del trabajo de investigación desde una visión más amplia.

En este documento de investigación se ha realizado un estado del arte extenso desde el punto de vista tecnológico del servicio de distribución de contenido multimedia y también de los algoritmos de tasa adaptativa creados por la comunidad científica para el estándar

DASH. En este estado del arte se ha presentado la historia de los servicios de distribución multimedia hasta llegar al estándar *DASH* y se explica el funcionamiento, las distintas herramientas comerciales existentes y los tipos de entrega del servicio. También, se presenta una visión general de los modelos de *QoE* que se han propuesto para el estándar *DASH*.

En cuanto a los algoritmos de tasa adaptativa, se han analizado 31 de ellos con fechas de propuesta que van desde el 2011 hasta el 2018. Aunque existen más algoritmos, la mayoría se basan en alguno de los 31 algoritmos presentados en el documento. Como conclusión se extrae que los algoritmos de estimación del throughput usados como parte de los mecanismos de tasa adaptativa pueden necesitar ajustes precisos según las características de la red de acceso y en algoritmos de control del buffer.

La contribución principal se resume en el método (algoritmo) propuesto de estimación del throughput *AFF*, el cual, detecta fluctuaciones a corto plazo en el ancho de banda para aplicar pesos adaptativos sobre las mediciones siguientes con el fin de “olvidar” medidas anteriores. Es decir, es una especie de “reinicio” para el método de estimación que utiliza los valores históricos.

Durante el desarrollo de la investigación, se propusieron tres plataformas virtualizadas y dos de ellas se entregaron como contribución al proyecto INNPACTO LTextreme [LTextreme12] del Ministerio de Ciencia e Innovación de España. Una de las plataformas virtualizadas permitió obtener perfiles de tráfico reales, que fueron utilizados como escenarios para validar la propuesta del método de estimación del throughput *AFF*.

Con los escenarios mencionados anteriormente se realizaron experimentos para validar el método de estimación del throughput *AFF*, comparándolo con el método *avg-last-3* que utiliza la “media móvil simple” de las últimas 3 mediciones y el método *EWMA*. También, se realizaron pruebas de equidad cuando varios reproductores compiten por el ancho de banda y métricas de *QoE* con las que se validó al algoritmo propuesto por esta investigación.

Los experimentos realizados confirman que al usar el método de estimación del throughput *AFF*, es posible mejorar la *QoE* obtenida por un reproductor *DASH* sobre diferentes tipos de redes, evitando el ajuste fino de los parámetros de otros algoritmos de estimación del throughput.

Los resultados muestran que el método *AFF* puede obtener tasas del bitrate promedio ligeramente más bajas que los métodos alternativos. Sin embargo, presenta un mejor comportamiento en lo que respecta al número de interrupciones de la reproducción y al número de cambios en la calidad (bitrate) del vídeo, que son dos de los parámetros clave

para analizar la *QoE* percibida por el usuario.

Finalmente, los resultados de los experimentos de equidad nos permiten comprobar que el uso del método *AFF* muestra un comportamiento equitativo en el consumo del ancho de banda de la red cuando existen diferentes clientes reproduciendo contenido con el estándar *DASH*.

7.4. Divulgación de Resultados

El trabajo de investigación realizado en esta tesis doctoral ha permitido contribuir al estado del arte mediante la publicación en una revista con factor de impacto *JCR* y en conferencias, así como al desarrollo de proyectos nacionales.

A continuación, se muestran las publicaciones realizadas y se describen los proyectos relacionados.

7.4.1. Artículo de Revista

- **M. Aguayo**, L. Bellido, C. M. Lentisco and E. Pastor. *DASH Adaptation Algorithm Based on Adaptive Forgetting Factor Estimation*. IEEE Transactions on Multimedia, vol. 20, no. 5, pages 1224-1232, May 2018
Cuartil JCR: Q1, 4/107 (2018)
Factor de Impacto: 5.452 (2018)

7.4.2. Artículos de Congresos

- C. M. Lentisco, **M. Aguayo**, L. Bellido, E. Pastor, D. De-Antonio- Monte and A. G. Bolívar. *A virtualized platform for analyzing LTE broadcast services*. In 2015 European Conference on Networks and Communications (EuCNC), pages 512-516, Jun. 2015
Lugar y Fecha de conferencia: Paris, Francia, 29 Junio – 2 Julio de 2015
- L. Bellido, C. M. Lentisco, **M. Aguayo** and E. Pastor. *Supporting Handover between LTE video Broadcasting and Unicast Streaming*. In 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies, pages 329-334, Sep. 2015
Lugar y Fecha de conferencia: Cambridge, United Kingdom, 9–11 de Septiembre de 2015

7.4.3. Proyectos

Las propuestas presentadas en esta tesis doctoral están relacionadas con un proyecto nacional, el cual se describe a continuación:

INNPACTO LTeXtreme. (Optimización de Servicios Multiusuario y Multimedia sobre LTE y LTE-Advanced)

Este proyecto fue aprobado por el Ministerio de Ciencia e Innovación de España (IPT-2012-0525-430000) y financiado por el Ministerio de Economía y Competitividad. El proyecto inició en enero de 2013 y se extendió hasta el primer trimestre de 2015. El proyecto LTeXtreme consistió en el diseño y desarrollo de una arquitectura de sistema que optimizara los servicios multiusuario y multimedia utilizando los estándares del 3GPP, LTE y LTE-Advanced. Esto con el fin de mejorar las aplicaciones interactivas, permitiendo medir en tiempo real la calidad, lo que conllevará que aplicaciones colaborativas como la ofimática, los videojuegos y la distribución de contenidos multimedia puedan reaccionar ante la información obtenida de estas mediciones, optimizando su funcionamiento de cara al usuario.

El consorcio del proyecto fue formado por la empresa privada Alcatel–Lucent, y dos Universidades públicas, la Universidad Carlos III de Madrid y la Universidad Politécnica de Madrid.

En el marco del proyecto INNPACTO LTeXtreme el trabajo realizado en este documento de investigación se definió en el paquete de trabajo (PT) 3. Dicho PT tiene como objetivo evaluar las tecnologías y protocolos previos para la distribución de contenidos multimedia mediante streaming, así como definir los modelos que se van a utilizar en el proyecto. Después de esto, se procederá a la validación de los modelos a través de tareas de simulación. Este PT 3 consta de 4 tareas que fueron: evaluar las tecnologías y protocolos para distribución mediante streaming, definición de modelos, simulación de los modelos para validarlos y análisis de los resultados y medidas de prestación.

Las contribuciones al proyecto que forman parte de este documento de investigación se resumen en los documentos entregables E.3.3 y E.3.4. En este primer documento entregable la contribución fue la plataforma virtualizada que se definió en el capítulo 5, la cual se creó para evaluar servicios de difusión (broadcast) de contenido multimedia. En el segundo documento entregable, la contribución fue la creación de la plataforma virtualizada cliente-servidor la cual sirvió para evaluar medidas de extremo a extremo en una red LTE.

7.5. Trabajos Futuros

El objetivo de proponer trabajos a futuro es profundizar en algunas de las líneas de investigación ya abiertas, es decir, se debe centrar en aumentar el impacto de la contribución que realiza esta tesis a los mecanismos de adaptación dinámica para el servicio de distribución multimedia sobre redes móviles. Estas líneas de investigación futuras se dividen en tres apartados y se presentan a continuación:

- **Mecanismos de adaptación dinámica**

Durante el desarrollo de este trabajo de investigación, después de que fueron analizados los distintos algoritmos basados en el buffer, se puede proponer un algoritmo en el que la medida del buffer mínimo se adapte dinámicamente, con el fin de mejorar las prestaciones del método *AFF*. Además, otro factor importante que muchos de los algoritmos analizados no implementan es la necesidad de contar con un planificador de nuevos segmentos, que por un lado, introduzca aleatoriedad para evitar la sincronización de los reproductores y que, por otro lado, cuando el ancho de banda lo permita, pueda descargar segmentos en paralelo. Otra parte importante que tiene que ver con los mecanismos de adaptación, es la de mejorar la comunicación entre el estimador del throughput, el control del buffer y el planificador de nuevos segmentos para poder mejorar el desempeño del reproductor *DASH*.

- **Calidad de la Experiencia (QoE)**

En el apartado de *QoE*, es necesario proponer una metodología la cual permita analizar con eficacia la calidad que perciben los usuarios del servicio de distribución multimedia sobre redes móviles. Esto se puede lograr mediante un modelo que considere las métricas de *QoS* del servicio para obtener resultados *MOS*. Este nuevo modelo debe considerar no solamente las métricas de *QoE* presentadas en los resultados de este trabajo de investigación, sino también, considerar otras métricas mencionadas en el capítulo 3, en el apartado que muestra el estado del arte de modelos *QoE* para el estándar *DASH*.

- **Pruebas de rendimiento**

Para validar las propuestas surgidas durante el desarrollo de la tesis doctoral se ha desarrollado una plataforma virtualizada que realiza algunas asunciones que simplifican el problema. Aunque esta plataforma virtualizada fue útil para validar la propuesta del método *AFF*, resulta conveniente abordar su revisión y plantear nuevos

escenarios de redes móviles [van der Hooft16]. Además, es necesario mejorar los escenarios de emulación incrementando la complejidad de la red (más nodos) y tráfico agregado (descarga ftp, descarga web, tráfico de email, etc.). También, es necesario crear diferente contenido multimedia (vídeos reales, noticias, etc.) con mayores resoluciones (1080p, 1440p, 2160p, etc.). Por último, es necesario utilizar nuevas herramientas de codificación del contenido multimedia que utilicen otros estándares como puede ser H.265, también llamado *High Efficiency Video Coding (HEVC)*, el cual puede presentar ahorro en el ancho de banda con respecto al estándar H.264.

REFERENCIAS BIBLIOGRÁFICAS

- [3GPP17a] 3GPP. *Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description*. 3GPP TS 26.246 v15.0.0, 2017.
- [3GPP17b] 3GPP. *Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs*. 3GPP TS 26.346 v15.0.0, 2017.
- [3GPP18a] 3GPP. *Transparent End-to-End Packet-Switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP*. 3GPP TS 26.247 v15.2.0, 2018.
- [3GPP18b] 3GPP. *Transparent End-to-End Packet-Switched Streaming Service (PSS); Protocols and codecs*. 3GPP TS 26.234 v15.0.0, 2018.
- [Adhikari15] V. K. Adhikari, Y. Guo, F. Hao, V. Hilt, Z. Zhang, M. Varvello and M. Steiner. *Measurement Study of Netflix, Hulu, and a Tale of Three CDNs*. IEEE/ACM Transactions on Networking, vol. 23, no. 6, pp. 1984–1997, Dec 2015.
- [Aguayo18] M. Aguayo, L. Bellido, C. M. Lentisco and E. Pastor. *DASH Adaptation Algorithm Based on Adaptive Forgetting Factor Estimation*. IEEE Transactions on Multimedia, vol. 20, no. 5, pp. 1224–1232, May 2018.
- [Ahmed14] A. Ahmed, L. M. Boulahia and D. Gaiti. *Enabling Vertical Handover Decisions in Heterogeneous Wireless Networks: A State-of-the-Art and A Classification*. IEEE Communications Surveys Tutorials, vol. 16, no. 2, pp. 776–811, Second 2014.

- [Akhshabi11] S. Akhshabi, A. C. Begen and C. Dovrolis. *An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP*. In Proceedings of the 2nd annual ACM conference on Multimedia systems, pp. 157–168. ACM, Feb. 2011.
- [Akhshabi12] S. Akhshabi, L. Anantakrishnan, A. C. Begen and C. Dovrolis. *What happens when HTTP adaptive streaming players compete for bandwidth?* In Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video, pp. 9–14. ACM, 2012.
- [Alejos12] A.V. Alejos, M.G. Sanchez and I. Cuinas. *Experimental distribution functions for analysis of coherence bandwidth fluctuations for a fixed broadband wireless access system*. IET Microwaves, Antennas Propagation, vol. 6, no. 4, pp. 450–460, Mar. 2012.
- [Almashor15] M. Almashor, I. Khalil, Z. Tari, A. Y. Zomaya and S. Sahni. *Enhancing Availability in Content Delivery Networks for Mobile Platforms*. IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 8, pp. 2247–2257, Aug 2015.
- [Anagnostopoulos12] C. Anagnostopoulos, D. K. Tasoulis, N. M. Adams, N. G. Pavlidis and D. J. Hand. *Online linear and quadratic discriminant analysis with adaptive forgetting for streaming classification*. Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 5, no. 2, pp. 139–166, 2012.
- [Apple02] Apple. *QuickTime Streaming Server 4.1*, 2002. Apple Computer, Inc. Accesado: Nov. 2018.
- [Batalla16] J. Mongay Batalla, P. Krawiec, A. Beben, P. Wisniewski and A. Chydzinski. *Adaptive Video Streaming: Rate and Buffer on the Track of Minimum Rebuffering*. IEEE Journal on Selected Areas in Communications, vol. 34, no. 8, pp. 2154–2167, Ago. 2016.
- [Beben16] A. Beben, P. Wiśniewski, J. Mongay Batalla and P. Krawiec. *AB-MA+: Lightweight and Efficient Algorithm for HTTP Adaptive Streaming*. In Proceedings of the 7th International Conference on

- Multimedia Systems, MMSys '16, pp. 2:1–2:11, New York, NY, USA, 2016. ACM.
- [Bellido15] L. Bellido, C.M. Lentisco, M. Aguayo and E. Pastor. *Supporting Handover between LTE Video Broadcasting and Unicast Streaming*. In 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies, pp. 329–334, Sep. 2015.
- [Bhaduri11] K. Bhaduri, K. Das, K. Borne, C. Giannella, T. Mahule and H. Kargupta. *Scalable, asynchronous, distributed eigen monitoring of astronomy data streams*. Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 4, no. 3, pp. 336–352, 2011.
- [Blender08] F. Blender. *Big Buck Bunny Movie*, 2008. Accesado: Feb. 2017.
- [Blender10] F. Blender. *Sintel Movie Trailer*, 2010. Accesado: Sep. 2016.
- [Bodenham13] D.A. Bodenham and N.M. Adams. *Continuous Monitoring of a Computer Network Using Multivariate Adaptive Estimation*. In 2013 IEEE 13th International Conference on Data Mining Workshops, pp. 311–318, Dic. 2013.
- [Bodenham14] D.A. Bodenham. *Adaptive estimation with change detection for streaming data*. PhD thesis, Imperial College London, Lond., UK, 2014.
- [Bokani15] A. Bokani, M. Hassan, S. Kanhere and X. Zhu. *Optimizing HTTP-Based Adaptive Streaming in Vehicular Environment Using Markov Decision Process*. IEEE Transactions on Multimedia, vol. 17, no. 12, pp. 2297–2309, Dec. 2015.
- [Borg13] L. Borg, K. Streeter and G. Eguchi. *HTTP dynamic streaming specification version 3.0 final*. Adobe Systems Incorporated, 2013.
- [Bouten14] N. Bouten, S. Latré, J. Famaey, W. Van Leekwijck and F. De Turck. *In-Network Quality Optimization for Adaptive Video Streaming Services*. IEEE Transactions on Multimedia, vol. 16, no. 8, pp. 2281–2293, Dec. 2014.

- [Brueck10] D.F. Brueck and M.B. Hurst. *Apparatus, system, and method for multi-bitrate content streaming*, October 19 2010. US Patent 7,818,444 B2.
- [Burkom05] H. Burkom, S. Murphy, J. Coberly and K. Hurt-Mullen. *Public health monitoring tools for multiple data streams*. Morbidity and Mortality Weekly Report, vol. 54, no. Supplement on Syndromic Surveillance, pp. 55–62, 2005.
- [Buyya08] R. Buyya, M. Pathan and A. Vakali. *Content delivery networks*, volume 9. Springer Science & Business Media, 2008.
- [Castagliola11] P. Castagliola and P. Maravelakis. *A CUSUM control chart for monitoring the variance when parameters are estimated*. Journal of Statistical Planning and Inference, vol. 141, no. 4, pp. 1463–1478, 2011.
- [Chen15] Y. Chen, K. Wu and Q. Zhang. *From QoS to QoE: A Tutorial on Video Quality Assessment*. IEEE Communications Surveys Tutorials, vol. 17, no. 2, pp. 1126–1165, Secondquarter 2015.
- [Choi12] J. Choi, A. S. Reaz and B. Mukherjee. *A Survey of User Behavior in VoD Service and Bandwidth-Saving Multicast Streaming Schemes*. IEEE Communications Surveys Tutorials, vol. 14, no. 1, pp. 156–169, First 2012.
- [Cicalò16] S. Cicalò, N. Changuel, V. Tralli, B. Sayadi, F. Faucheux and S. Kerboeuf. *Improving QoE and Fairness in HTTP Adaptive Streaming Over LTE Network*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 12, pp. 2284–2298, Dec 2016.
- [Cicco13] L. De Cicco, V. Caldaralo, V. Palmisano and S. Mascolo. *ELASTIC: a client-side controller for dynamic adaptive streaming over HTTP (DASH)*. In Packet Video Workshop (PV), 2013 20th Int., pp. 1–8. IEEE, 2013.
- [Cicco14] L. De Cicco and S. Mascolo. *An Adaptive Video Streaming Control System: Modeling, Validation, and Performance Evaluation*.

- IEEE/ACM Transactions on Networking, vol. 22, no. 2, pp. 526–539, Abr. 2014.
- [Cisco18] Cisco. *Visual networking index (VNI): Global mobile data traffic forecast update 2017-2022*. Cisco Public Information, Dec. 2018.
- [Claeys13] M. Claeys, S. Latré, J. Famaey, T. Wu, V. Leekwijck and F. Turck. *Design of a Q-learning-based client quality selection algorithm for HTTP adaptive video streaming*. In Proceedings of the 2013 Workshop on Adaptive and Learning Agents (ALA), Saint Paul (Minn.), USA, pp. 30–37, 2013.
- [Cooper00] J.E. Cooper and K. Worden. *On-line physical parameter estimation with adaptive forgetting factors*. Mechanical Systems and Signal Processing, vol. 14, no. 5, pp. 705–730, May. 2000.
- [DASH-IF18a] DASH-IF. *Guidelines for Implementation: DASH-IF Interoperability Points*. V4.2, Guideline, DASH Industry Forum, April 2018.
- [DASH-IF18b] DASH-IF. *MPEG-DASH reference player dash.js*, 2018. Accesado: May. 2018.
- [De Vriendt14] J. De Vriendt, D. De Vleeschauwer and D. Robinson. *QoE model for video delivered over an LTE network using HTTP adaptive streaming*. Bell Labs Technical Journal, vol. 18, no. 4, pp. 45–62, 2014.
- [Dilley02] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman and B. Weihl. *Globally distributed content delivery*. IEEE Internet Computing, vol. 6, no. 5, pp. 50–58, Sep. 2002.
- [Erman11] J. Erman, A. Gerber, K. K. Ramadrishnan, S. Sen and O. Spatscheck. *Over the Top Video: The Gorilla in Cellular Networks*. In Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, IMC '11, pp. 127–136, New York, NY, USA, 2011. ACM.
- [Essaili15] A.E. Essaili, D. Schroeder, E. Steinbach, D. Staehle and M. Shehada. *QoE-Based Traffic and Resource Management for Adaptive*

- HTTP Video Delivery in LTE*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 6, pp. 988–1001, Jun. 2015.
- [Fernandes12] S. Fernandes and A. Karmouch. *Vertical Mobility Management Architectures in Wireless Networks: A Comprehensive Survey and Future Directions*. IEEE Communications Surveys Tutorials, vol. 14, no. 1, pp. 45–63, First 2012.
- [Fernández16] D. Fernández, F. J. Ruiz, L. Bellido, E. Pastor, O. Walid and V. Mateos. *Enhancing Learning Experience in Computer Networking through a Virtualization-Based Laboratory Model*. International Journal of Engineering Education, vol. 32, no. 6, pp. 2569–2584, Dec. 2016.
- [Feuvre17] J. Le Feuvre. *GPAC Multimedia Open Source Project*, 2017. Accedido: Mar. 2017.
- [Fuertes09] W. Fuertes and J. López. *An emulation of VoD services using virtual network environments*. Proc. GI/ITG Workshop on Overlay and Network Virtualization, Kassel, Germany, 2-6 March, 2009, published in Electronic Communications of the EASST, vol. 17, 2009.
- [García15] S. García, J. Cabrera and N. García. *Quality-Control Algorithm for Adaptive Streaming Services Over Wireless Channels*. IEEE Journal of Selected Topics in Signal Processing, vol. 9, no. 1, pp. 50–59, Feb. 2015.
- [Gauss23] C.F. Gauss. *Theoria combinationis observationum erroribus minimis obnoxiae*, volume 1. Henricus Dieterich, 1823.
- [Gellens11] R. Gellens, D. Singer and P. Frojdh. *The ‘Codecs’ and ‘Profiles’ Parameters for “Bucket” Media Types*. RFC 6381, Agto. 2011.
- [Gettys11] J. Gettys and K. Nichols. *Bufferbloat: Dark buffers in the internet*. Queue, vol. 9, no. 11, pp. 40, 2011.
- [Hamer12] G. Hamer. *Open Source Media Framework 2.0: Developer Documentation*. Adobe Systems Incorporated, 2012. Accedido: May. 2018.

- [Handley06] M. Handley, V. Jacobson and C. Perkins. *Session Description Protocol (SDP)*. RFC 2326, Jul. 2006.
- [Huang12] T. Huang, N. Handigol, B. Heller, N. McKeown and R. Johari. *Confused, timid, and unstable: picking a video streaming rate is hard*. In Proceedings of the 2012 ACM conference on Internet measurement conference, pp. 225–238. ACM, 2012.
- [Huang15] T. Huang, R. Johari, N. McKeown, M. Trunnell and M. Watson. *A buffer-based approach to rate adaptation: Evidence from a large video streaming service*. ACM SIGCOMM Computer Communication Review, vol. 44, no. 4, pp. 187–198, 2015.
- [Huang18] W. Huang, Y. Zhou, X. Xie, D. Wu, M. Chen and E. Ngai. *Buffer State is Enough: Simplifying the Design of QoE-Aware HTTP Adaptive Video Streaming*. IEEE Transactions on Broadcasting, pp. 1–12, 2018.
- [ISO13a] ISO. Information technology – Coding of audio-visual objects – Part 3: Audio, ISO/IEC 14496-3, 2013.
- [ISO13b] ISO. Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 4: Segment encryption and authentication, ISO/IEC 23009-4, 2013.
- [ISO14a] ISO. Information technology – Coding of audio-visual objects – Part 10: Advanced video coding, ISO/IEC 14496-10, 2014.
- [ISO14b] ISO. Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segments formats, ISO/IEC 23009-1, 2014.
- [ISO15a] ISO. Information technology – Coding of audio-visual objects – Part 12: ISO base media file format, ISO/IEC 14496-12, 2015.
- [ISO15b] ISO. Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 3: Implementation guidelines, ISO/IEC NP TR 23009-3, 2015.

- [ISO17a] ISO. Information technology – High high efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency video coding, ISO/IEC 23008-2, 2017.
- [ISO17b] ISO. Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 2: Conformance and reference software, ISO/IEC 23009-2, 2017.
- [ISO17c] ISO. Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 5: Server and network assisted DASH (SAND), ISO/IEC 23009-5, 2017.
- [ISO17d] ISO. Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 6: DASH with server push and WebSockets, ISO/IEC 23009-6, 2017.
- [ITU-T16] ITU-T. *Recommendation P. 800.1: Methods for objective and subjective assessment of speech and video quality; Mean opinion score (MOS) terminology*. Int. Telecomm. Union, Geneva, Jul. 2016.
- [ITU-T17a] ITU-T. *Recommendation E. 800: Definitions of terms related to quality of service*. Int. Telecomm. Union, Geneva, Nov. 2017.
- [ITU-T17b] ITU-T. *Recommendation P. 10/G.100: Vocabulary for performance, quality of service and quality of experience*. Int. Telecomm. Union, Geneva, Nov. 2017.
- [ITU-T19] ITU-T. *Recommendation P. 1203.1: Models and tools for quality assessment of streamed media; Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport Video quality estimation module*. Int. Telecomm. Union, Geneva, Ene. 2019.
- [Jain84] R. Jain, D. Chiu and W. Hawe. *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System*. Research report, Digital Equipment Corporation, DEC-TR-301, September 1984.

- [Jiang14] J. Jiang, V. Sekar and H. Zhang. *Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive*. IEEE/ACM Transactions on Networking, vol. 22, no. 1, pp. 326–340, Feb. 2014.
- [Kim13] Y. Kim, J. Shin and J. Park. *Design and implementation of a network-adaptive mechanism for HTTP video streaming*. ETRI Journal, vol. 35, no. 1, pp. 27–34, Feb. 2013.
- [Kua17] J. Kua, G. Armitage and P. Branch. *A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP*. IEEE Communications Surveys Tutorials, vol. 19, no. 3, pp. 1842–1866, thirdquarter 2017.
- [Kupka12] T. Kupka, P. Halvorsen and C. Griwodz. *Performance of on-off traffic stemming from live adaptive segmented HTTP video streaming*. In 37th Annual IEEE Conference on Local Computer Networks, pp. 401–409, Oct. 2012.
- [Le16] H.T. Le, H.N. Nguyen, N. Pham Ngoc, A.T. Pham and T.C. Thang. *A Novel Adaptation Method for HTTP Streaming of VBR Videos over Mobile Networks*. Mobile Information Systems, vol. 2016, Jun. 2016.
- [Lecompte12] D. Lecompte and F. Gabin. *Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: overview and Rel-11 enhancements*. IEEE Communications Magazine, vol. 50, no. 11, pp. 68–74, November 2012.
- [Lederer12] S. Lederer, C. Müller and C. Timmerer. *Dynamic Adaptive Streaming over HTTP Dataset*. In Proceedings of the 3rd Multimedia Systems Conference, MMSys '12, pp. 89–94, New York, NY, USA, 2012. ACM.
- [Lentisco15] C.M. Lentisco, M. Aguayo, L. Bellido, E. Pastor, D. De-Antonio-Monte and A. G. Bolívar. *A virtualized platform for analyzing LTE broadcast services*. In 2015 European Conference on Networks and Communications (EuCNC), pp. 512–516, Jun. 2015.

- [Lentisco17a] C.M. Lentisco, L. Bellido, J. C. Cuellar Q., E. Pastor and J.L. Arciniegas H. *QoE-Based Analysis of DASH Streaming Parameters Over Mobile Broadcast Networks*. IEEE Access, vol. 5, pp. 20684–20694, 2017.
- [Lentisco17b] C.M. Lentisco, L. Bellido and E. Pastor. *Seamless Mobile Multimedia Broadcasting Using Adaptive Error Recovery*. Mobile Information System, vol. 2017, Feb. 2017.
- [Leung05] S. Leung and C.F. So. *Gradient-based variable forgetting factor RLS algorithm in time-varying environments*. IEEE Transactions on Signal Processing, vol. 53, no. 8, pp. 3141–3150, 2005.
- [Li14] 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, no. 4, pp. 719–733, Abr. 2014.
- [Lin14] Q. Lin, Y. Liu, Y. Shen, H. Shen, L. Sang and D. Yang. *Bandwidth estimation of rate adaption algorithm in DASH*. In 2014 IEEE Globecom Workshops (GC Workshops), pp. 243–247, Dec. 2014.
- [Lindsay00] B. Lindsay, R. Pilla and P. Basak. *Moment-based approximations of distributions using mixtures: Theory and applications*. Annals of the Institute of Statistical Mathematics, vol. 52, no. 2, pp. 215–230, 2000.
- [Liu11] C. Liu, I. Bouazizi and M. Gabbouj. *Rate adaptation for adaptive HTTP streaming*. In Proceedings of the second annual ACM conference on Multimedia systems, pp. 169–174. ACM, 2011.
- [LTextreme12] Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (INNPACTO) LTextreme. *LTextreme (Optimización de servicios multiusuario y multimedia sobre LTE y LTE-Advanced)*. IPT-2012-0525-430000, Ministerio de Economía y Competitividad, 2012.
- [Ma17] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen and W. Zhu. *Understanding Performance of Edge Content Caching for Mobile Video*

- Streaming*. IEEE Journal on Selected Areas in Communications, vol. 35, no. 5, pp. 1076–1089, May 2017.
- [Mansy13] A. Mansy, B. Ver Steeg and M. Ammar. *Sabre: A client based technique for mitigating the buffer bloat effect of adaptive video flows*. In Proceedings of the 4th ACM Multimedia Systems Conference, pp. 214–225. ACM, 2013.
- [Martín16a] V. Martín, J. Cabrera and N. García. *Design, optimization and evaluation of a q-learning HTTP adaptive streaming client*. IEEE Transactions on Consumer Electronics, vol. 62, no. 4, pp. 380–388, Nov. 2016.
- [Martín16b] V. Martín, J. Cabrera and N. García. *Evaluation of Q-Learning approach for HTTP adaptive streaming*. In 2016 IEEE International Conference on Consumer Electronics (ICCE), pp. 293–294, Ene. 2016.
- [Merritt06] L. Merritt and R. Vanam. *x264: A high performance H. 264/AVC encoder*. [Online] http://akuvian.org/src/x264/overview_x264_v8_5.pdf, 2006.
- [Migues89] S. Migues and R. Housley. *Designing a trusted client-server distributed network*. In Proceedings of the Fifth Annual Computer Security Applications Conference, pp. 91–94, Dec. 1989.
- [Miller12] K. Miller, E. Quacchio, G. Gennari and A. Wolisz. *Adaptation algorithm for adaptive streaming over HTTP*. In 2012 19th International Packet Video Workshop (PV), pp. 173–178, May. 2012.
- [Miller15] K. Miller, D. Bethanabhotla, G. Caire and A. Wolisz. *A Control-Theoretic Approach to Adaptive Video Streaming in Dense Wireless Networks*. IEEE Transactions on Multimedia, vol. 17, no. 8, pp. 1309–1322, Aug. 2015.
- [Müller12] C. Müller, S. Lederer and . Timmerer. *An evaluation of dynamic adaptive streaming over HTTP in vehicular environments*. In Proceedings of the 4th Workshop on Mobile Video, pp. 37–42. ACM, 2012.

- [Negru04] D. Negru, A. Mehaoua, A. Kourtis and E. LeBars. *Large-Scale Mobile Multimedia Service Management in Next Generation Networks*. volume 3271, pp. 308–319, 10 2004.
- [Nussbaumer95] J.P. Nussbaumer, B.V. Patel, F. Schaffa and J. P.G. Sterbenz. *Networking requirements for interactive video on demand*. IEEE Journal on Selected Areas in Communications, vol. 13, no. 5, pp. 779–787, Jun 1995.
- [Oracle17] Oracle. *VirtualBox*, 2017. Accesado: May. 2017.
- [Oyman12] O. Oyman and S. Singh. *Quality of experience for HTTP adaptive streaming services*. IEEE Communications Magazine, vol. 50, no. 4, pp. 20–27, April 2012.
- [Page54] E.S. Page. *Continuous inspection schemes*. Biometrika, vol. 41, no. 1/2, pp. 100–115, 1954.
- [Paila04] T Paila, M Lubyet *al.* *FLUTE - File Delivery over Unidirectional Transport*. RFC 3926, Oct. 2004.
- [Paleologu08] C. Paleologu, J. Benesty and S. Ciochina. *A Robust Variable Forgetting Factor Recursive Least-Squares Algorithm for System Identification*. IEEE Signal Processing Letters, vol. 15, pp. 597–600, 2008.
- [Pantos17] R. Pantos and W. May. *HTTP live streaming*. 2017.
- [Pathan14] M. Pathan, R. Sitaraman and D. Robinson. *Cloudbased content delivery and streaming*. IEEE, 2014.
- [Paxson99] V. Paxson. *End-to-end Internet packet dynamics*. IEEE/ACM Transactions on Networking, vol. 7, no. 3, pp. 277–292, Jun 1999.
- [Peltotalo07] J. Peltotalo, S. Peltotalo, J. Harju and R. Walsh. *Performance analysis of a file delivery system based on the FLUTE protocol*. International Journal of Communication Systems, vol. 20, no. 6, pp. 633–659, 2007.

- [Plackett50] R.L. Plackett. *Some theorems in least squares*. Biometrika, vol. 37, no. 1/2, pp. 149–157, 1950.
- [Qin17] Y. Qin, R. Jin, S. Hao, K. R. Pattipati, F. Qian, S. Sen, B. Wang and C. Yue. *A control theoretic approach to ABR video streaming: A fresh look at PID-based rate adaptation*. In IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp. 1–9, May. 2017.
- [Raca19] D. Raca, Y. Sani, C. Sreenan and J. Quinlan. *DASHbed: a testbed framework for large scale empirical evaluation of real-time DASH in wireless scenarios*. In Proceedings of the 10th ACM Multimedia Systems Conference, pp. 285–290. ACM, 2019.
- [Ramamurthi13] V. Ramamurthi and O. Oyman. *Link aware HTTP Adaptive Streaming for enhanced quality of experience*. In 2013 IEEE Global Communications Conference (GLOBECOM), pp. 1675–1680, Dec. 2013.
- [Ramamurthi15] V. Ramamurthi, O. Oyman and J. Foerster. *Using link awareness for HTTP Adaptive Streaming over changing wireless conditions*. In 2015 International Conference on Computing, Networking and Communications (ICNC), pp. 727–731, Feb. 2015.
- [Rao11] A. Rao, A. Legout, Y. Lim, D. Towsley, C. Barakat and W. Dabbous. *Network Characteristics of Video Streaming Traffic*. In Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies, CoNEXT '11, pp. 25:1–25:12, New York, NY, USA, 2011. ACM.
- [Roberts59] S.W. Roberts. *Control chart tests based on geometric moving averages*. Technometrics, vol. 1, no. 3, pp. 239–250, 1959.
- [Romirer09] P. Romirer, F. Ricciato, A. DAlconzo, R. Franzan and W. Karner. *Network-wide measurements of TCP RTT in 3G*. In International Workshop on Traffic Monitoring and Analysis, pp. 17–25. Springer, 2009.

- [Rotstein92] G.E. Rotstein and D.R. Lewin. *Control of an unstable batch chemical reactor*. Computers & chemical engineering, vol. 16, no. 1, pp. 27–49, 1992.
- [Ruder16] S. Ruder. *An overview of gradient descent optimization algorithms*. Computing Research Repository, vol. abs/1609.04747, 2016.
- [Sandvine18] Sandvine. *Sandvine Global Internet Phenomena Report Oct.*, 2018. Accesado: Nov. 2018.
- [Sani17] Y. Sani, A. Mauthe and C. Edwards. *Adaptive Bitrate Selection: A Survey*. IEEE Communications Surveys Tutorials, vol. 19, no. 4, pp. 2985–3014, Fourthquarter 2017.
- [Schulzrinne98] H. Schulzrinne, A. Rao and R. Lanphier. *Real Time Streaming Protocol (RTSP)*. RFC 2326, Apr. 1998.
- [Schulzrinne03] H Schulzrinne, S Casner, S Frederick and V Jacobson. *Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP)*. RFC 2326, Jul. 2003.
- [Schulzrinne16] H. Schulzrinne, A. Rao, R. Lanphier, M. Westerlund and M. Stiemerling. *Real Time Streaming Protocol (RTSP) Version 2.0*. RFC 2326, Dec. 2016.
- [Seufert15] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld and P. Tran-Gia. *A Survey on Quality of Experience of HTTP Adaptive Streaming*. IEEE Communications Surveys Tutorials, vol. 17, no. 1, pp. 469–492, Firstquarter 2015.
- [Shewhart39] W. Shewhart and W. Deming. *Statistical method from the viewpoint of quality control*. Courier Corporation, 1939.
- [Sideris15] A. Sideris, E. Markakis, N. Zotos, E. Pallis and C. Skianis. *MPEG-DASH users’ QoE: The segment duration effect*. In 2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX), pp. 1–6, 2015.
- [Spiteri16] K. Spiteri, R. Urgaonkar and R. Sitaraman. *BOLA: Near-optimal bitrate adaptation for online videos*. In IEEE INFOCOM 2016 - The

- 35th Annual IEEE International Conference on Computer Communications, pp. 1–9, Abr. 2016.
- [Spiteri19] K. Spiteri, R. Sitaraman and D. Sparacio. *From theory to practice: Improving bitrate adaptation in the DASH reference player*. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 15, no. 2s, pp. 67, 2019.
- [Sun16] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu and B. Sinopoli. *CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction*. In Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference, pp. 272–285. ACM, 2016.
- [Sutton09] R. Sutton, H. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári and E. Wiewiora. *Fast gradient-descent methods for temporal-difference learning with linear function approximation*. In Proceedings of the 26th Annual International Conference on Machine Learning, pp. 993–1000. ACM, 2009.
- [Thang12] T.C. Thang, Q.D. Ho, J.W. Kang and A.T. Pham. *Adaptive streaming of audiovisual content using MPEG DASH*. IEEE Transactions on Consumer Electronics, vol. 58, no. 1, pp. 78–85, Feb. 2012.
- [Thang13] T.C. Thang, H.T. Le, H.X. Nguyen, A.T. Pham, J.W. Kang and Y.M. Ro. *Adaptive video streaming over HTTP with dynamic resource estimation*. Journal of Communications and Networks, vol. 15, no. 6, pp. 635–644, Dic. 2013.
- [Thang14] T.C. Thang, H.T. Le, A.T. Pham and Y.M. Ro. *An Evaluation of Bitrate Adaptation Methods for HTTP Live Streaming*. IEEE Journal on Selected Areas in Communications, vol. 32, no. 4, pp. 693–705, April 2014.
- [Tian16] G. Tian and Y. Liu. *Towards Agile and Smooth Video Adaptation in HTTP Adaptive Streaming*. IEEE/ACM Transactions on Networking, vol. 24, no. 4, pp. 2386–2399, Ago. 2016.

- [van der Hooft16] J. van der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen and F. De Turck. *HTTP/2-Based Adaptive Streaming of HEVC Video Over 4G/LTE Networks*. IEEE Communications Letters, vol. 20, no. 11, pp. 2177–2180, Nov. 2016.
- [Varma15] S. Varma. *Flow control for video applications*. In Internet congestion control, chapter 6, pp. 173–203. Morgan Kaufmann, Waltham, MA, USA, 2015.
- [Wang08] B. Wang, J. Kurose, P. Shenoy and D. Towsley. *Multimedia streaming via TCP: An analytic performance study*. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 4, no. 2, pp. 1–22, 2008.
- [Wang16] C. Wang, A. Rizk and M. Zink. *SQUAD: A spectrum-based quality adaptation for dynamic adaptive streaming over HTTP*. In Proceedings of the 7th International Conference on Multimedia Systems, pp. 1. ACM, 2016.
- [Widrow85] B. Widrow and S. D. Stearns. Adaptive signal processing, volume 15. Prentice-hall Upper Saddle River, NJ, USA, 1985.
- [Wiśniewski15] P. Wiśniewski, A. Beben, J. M. Batalla and P. Krawiec. *On delimiting video rebuffering for stream-switching adaptive applications*. In 2015 IEEE International Conference on Communications (ICC), pp. 6867–6873, Jun. 2015.
- [Wu00] D. Wu, Y. T. Hou and Y. Zhang. *Transporting real-time video over the Internet: challenges and approaches*. Proceedings of the IEEE, vol. 88, no. 12, pp. 1855–1877, Dec. 2000.
- [Wu04] H. Wu, B. Salzberg and D. Zhang. *Online event-driven subsequence matching over financial data streams*. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pp. 23–34. ACM, 2004.
- [Yin14] X. Yin, V.s Sekar and B. Sinopoli. *Toward a principled framework to design dynamic adaptive streaming algorithms over http*. In Pro-

- ceedings of the 13th ACM Workshop on Hot Topics in Networks, pp. 9. ACM, 2014.
- [Yin15] X. Yin, A. Jindal, V. Sekar and B. Sinopoli. *A control-theoretic approach for dynamic adaptive video streaming over http*. ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, pp. 325–338, 2015.
- [Zabrovskiy17] A. Zabrovskiy, E. Kuzmin, E. Petrov, C. Timmerer and C. Mueller. *AdViSE: Adaptive Video Streaming Evaluation Framework for the Automated Testing of Media Players*. In Proceedings of the 8th ACM on Multimedia Systems Conference, pp. 217–220. ACM, 2017.
- [Zahran16] A.H. Zahran and C.J. Sreenan. *ARBITER: Adaptive rate-based intelligent HTTP streaming algorithm*. In 2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW), pp. 1–6, July 2016.
- [Zahran18] A.H. Zahran, D. Raca and C. Sreenan. *ARBITER+: Adaptive Rate-Based InTElligent HTTP StReaming Algorithm for Mobile Networks*. IEEE Transactions on Mobile Computing, pp. 1–1, 2018.
- [Zambelli09] A. Zambelli. *IIS smooth streaming technical overview*. Microsoft Corporation, vol. 3, pp. 40, 2009.
- [Zhou13] C. Zhou, C.W. Lin, X. Zhang and Z. Guo. *Buffer-based smooth rate adaptation for dynamic HTTP streaming*. In 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, pp. 1–9, Oct. 2013.
- [Zhou14] Y. Zhou, Y. Duan, J. Sun and Z. Guo. *Towards simple and smooth rate adaption for VBR video in DASH*. In 2014 IEEE Visual Communications and Image Processing Conference, pp. 9–12, Dec 2014.
- [Zhou16] C. Zhou, C.W. Lin and Z. Guo. *mDASH: A Markov Decision-Based Rate Adaptation Approach for Dynamic HTTP Streaming*. IEEE Transactions on Multimedia, vol. 18, no. 4, pp. 738–751, Apr. 2016.

- [Zhou17] C. Zhou, C.W. Lin, X. Zhang and Z. Guo. *TFDASH: A Fairness, Stability, and Efficiency Aware Rate Control Approach for Multiple Clients over DASH*. IEEE Transactions on Circuits and Systems for Video Technology, vol. PP, no. 99, pp. 1–1, 2017.

LISTA DE ABREVIATURAS

3GPP 3rd Generation Partnership Project.

ABR Adaptive Bitrate.

ADSL Asymmetric digital subscriber line.

AIMD Additive Increase/Multiplicative Decrease.

AVC Advanced Video Coding.

CDN Content Delivery Network.

CUSUM Cumulative Sum.

DASH Dynamic Adaptive Streaming over HTTP.

eMBMS Evolved Multimedia Broadcast and Multicast Service.

eNodeB Evolved Node B.

EPC Evolved Packet Core.

EWMA Exponential Weighted Moving Average.

FDT File Delivery Table.

FEC Forward Error Correction.

FLUTE File Delivery over Unidirectional Transport.

FPS Frames per second.

GSM Global System for Mobile Communications.

HEVC High Efficiency Video Coding.

HTTP Hypertext Transfer Protocol.

IEC International Electrotechnical Commission.

IETF Internet Engineering Task Force.

ISO International Organization for Standardization.

ISP Internet Service Provider.

JFI Jain Fairness Index.

LCL Lower Control Limit.

LTE Long Term Evolution.

LXC Linux Container.

MBMS Multimedia Broadcast and Multicast Service.

MMS Microsoft Media Server.

MOS Mean Opinion Score.

MPD Media Presentation Description.

MPEG Moving Picture Experts Group.

NAT Network Address Translation.

OSMF Open Source Media Framework.

OTT Over-The-Top.

PRB Physical Resource Block.

PSNR Peak Signal-to-Noise Ratio.

QoE Quality of Experience.

QoS Quality of Service.

RLS Recursive Least-Square.

RTCP Real-Time Control Protocol.

RTP Real-Time Transport Protocol.

RTSP Real Time Streaming Protocol.

RTT Round Trip Time.

SDP Session Description Protocol.

SPC Statistical Process Control.

SWF Small Web Format.

TCP Transmission Control Protocol.

TOI Transport Object Identifier.

UCL Upper Control Limit.

UDP User Datagram Protocol.

URL Uniform Resource Locator.

VBR Variable Bitrate.

VNX Virtual Networks over linuX.

XML eXtensible Markup Language.