Universidad del País Vasco
Euskal Herriko Unibertsitatea

DEPARTMENT OF COMMUNICATIONS ENGINEERING

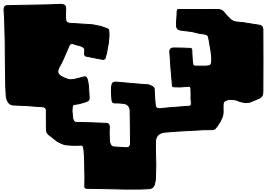
QOE ON MEDIA DELIVERY IN 5G ENVIRONMENTS

by:
Ángel Martín Navas

Supervised by:
Dr. Jon Montalbán Sánchez
&
Prof. Julián Flórez Esnal

Donostia – San Sebastian, Monday 7th May, 2018

eman ta zabal zazu



Universidad del País Vasco
Euskal Herriko Unibertsitatea

DEPARTMENT OF COMMUNICATIONS ENGINEERING

QOE ON MEDIA DELIVERY IN 5G ENVIRONMENTS

by:
Ángel Martín Navas

Supervised by:
Dr. Jon Montalbán Sánchez
&
Prof. Julián Flórez Esnal

Donostia – San Sebastian, Monday 7th May, 2018

This thesis is dedicated to my family, friends, workmates, classmates and anybody who made me become what I am today.

A special mention to my brothers for guiding me to the engineering field.

For their intentionally or subconscious support and encouragement.

Enara, Chloe, Leo, Papá y Mamá.

"Does that answer your questions, Doctor?" – *Rorschach*

Abstract

5G promises high-bandwidth, low latency, always-on and massive connectivity by expanding the possibilities and capabilities of mobile networks. The network revolution of 5G will be achievable with the introduction of new technologies, both in the access to and in the core of mobile networks, such as the flexible and scalable assignment of network resources. To this end, the digital transformation of networks, enabled by cloud architectures and technologies, and the advanced radio capabilities will make the networks agile and broader.

At the same time, the prominence of the video traffic in the world's mobile data turns efficiency of video delivery into a core application to be managed and optimized by network operators. Here, when it comes to media consumption and production, user habits and expectations are changing profoundly. First, media services need to cope with a vast volume of untagged media. It is fundamental to make media catalogue relevant, interest and personalized to expand audience capture. To this end, a deeper media tagging to discover underlying media relations is essential. Second, media services produce an increasing demand, in terms of data rates and number of simultaneous users connected, struggling to get more stable and accurate quality requirements. Therefore, the quality of experience plays a crucial role in maximizing audience retention. In this regard, guaranteeing a quality of service is crucial.

However, the networks work on a best-effort basis with a neutral position in terms of traffic delivery. This means that solutions which achieve a dynamic and efficient media delivery will make the difference. Going beyond, media delivery will turn into a critical area to be explored, catalysed by the

sensor explosion in vertical sectors such as IoT, Connected Cars, Health and Industry 4.0.

Beyond 4K or UHD resolutions, media users will experience a smooth and more attractive media consumption dynamically adapted to a user interest and mobility context. Essentially, the challenge is to take quality of experience on video delivery to a new level. 5G networks will meet quality of experience needs of next generation media services, as they produce a core traffic to be managed and optimized. Furthermore, they can be complemented with solutions from different corners of the media delivery workflow such as media servers, media players and the networking infrastructure. All of them can afford solutions to enhance the quality experience and shield from service degradation or outages.

Media services must be adapted differently to variations in radio network performance. HTTP adaptive streaming media delivery technologies meet those multimedia services demands by supporting a wide display ecosystem, different user preferences, languages and changeable mobility situations with a content delivery networks ready design. HTTP adaptive streaming enables media players to switch dynamically between different media qualities by tracking quick and sudden variations in the network conditions during the media playback. Moreover, HTTP adaptive streaming is a pull-based HTTP protocol that easily traverses middleboxes, such as firewalls and network address translation devices.

However, in dense client cells this client-driven approach could damage the overall quality of experience producing re-buffering times and potential image freezes along with quality fluctuations. This is caused by multiple media players sharing the available bandwidth where each one optimizes its individual quality based on instant decisions.

The telecommunication industry's proposal to empower the network edge in a more coordinated manner is based on multi-access edge computing technology. It turns a base station into a service catalyser, which dynamically improves network performance and user experience for a specific service. Operators can expose their edge radio access network through an

application program interface to authorized third parties to provide them with radio network information in real-time. Edge computing supplies new features, such as awareness of the radio status and close to zero delays, to monitor and to dynamically tune the traffic in a transparent manner. Furthermore, the decentralization of specific network functions to the edge of the network brings agility, adaptability and context awareness.

Beyond this, dynamically switching in real-time from one content delivery network to another is a relevant scenario. This switching decision must fit to a well-balanced trade-off between the quality of experience and the costs. This approach can become a reality by using video delivery analytics from the edge components, which tend to be proprietary solutions.

Concerning the network backhaul and core, the telecommunication industry envisions self-organising networks. Here, a centralized and autonomous network management system steers network operations towards deep and persistent dynamics. To this end, machine learning algorithms are applied. Machine learning techniques can engine a system of service demand prediction. This forecast can be exploited by other machine learning algorithms to evaluate the optimal network setup to serve a predicted traffic demand, using virtualisation to provision network resources. This is achieved while optimising performance, use of available network and virtual machine resources, overall energy requirements and operational costs.

The benefits from the 5G network in satisfying next generation media service needs are evident. However, 5G ships new parameters and technologies which can play a significant role in enhancing the quality of media services. They provide new challenges to deliver media services in 5G environments. First, the massive client connections volume where the 5G network handles a huge pool of devices spontaneously connected to media services. Second, the dense client cells where the media players strive to deliver the best performance when massive media sessions come from a specific area. Third, the edge video analytics by exploiting the interfaces of network components to dynamically and automatically optimize the media delivery. Fourth, the self-organising network ability of 5G, where scalable network

management systems on the network stack exploit the transformation of network functions into software and virtual entities to mutate the network.

To meet each of these challenges, this research proposes a four-tier complementary solution based on media delivery mechanisms for enhanced quality of experience of media services in 5G environments.

First, concerning quality of experience for audience capturing, media servers can improve the experience of a social media service through the media analysis of large volume catalogues produced by the users of social networks. Here, media servers dynamically orchestrate an elastic cloud of spontaneous workers populated with client devices to perform delay-tolerant media analysis.

Second, a client-side bitrate adaptation decision mechanism to make a dense client cell scenario steady, fair and efficient for all media players when quick and unforeseen changes in network conditions occur.

Third, when a client-side decision mechanism is not sufficient for guaranteeing the best performance since each client is unaware of the presence of the others, an in-network aware adaptation mechanism will provide adaptation decisions, in a distributed and zero latency manner, based on accurate, granular and geo-binned metrics. Furthermore, it will be able to switch to a healthy content delivery network in a transparent manner.

Fourth, a manager to dynamically allocate and set up the network to provide a cost-effective network topology that satisfies quality of experience operational constraints.

To implement, deploy, test, and evaluate the proposed solutions, we used a real long term evolution (LTE) infrastructure, based on the OpenAirInterface framework with an evolved packet core, an eNodeB, a set of LTE user equipments running Gstreamer media players acting as a dense cell, and an OpenStack-based software defined network operated by OpenDaylight. This experimental infrastructure is further enhanced with a multi-access edge computing proxy, able to parse and process MPEG-DASH media streams,

and a Docker container for machine learning components. The implementation of the media delivery solutions allows media services and mobile network operators to efficiently influence media players and efficiently manage network resources to maintain a target level of user satisfaction.

Each explored tier shows different abilities in enabling media services and network operators to engage, balance and ensure the quality of experience for 5G mobile networks based on novel media delivery techniques.

First, a mobile as an infrastructure provider platform, named Social at Work, creates an elastic cloud of massive and spontaneous connected resources running delay-tolerant tasks. The results of the experiment confirm the benefits when the number of devices is high and the tasks are independent and can be queued.

Second, a bitrate adaptation mechanism on the client-side, named LAMB-DASH, has been implemented with a low complexity design. Testing of LAMB-DASH for live and on-demand streams conclude its ability to provide a steady, consistent and unbiased quality of experience, with a low deviation of the estimated mean opinion score across all the media players in a dense client cell.

Third, the multi-access edge computing system, named MEC4FAIR, exploits zero-latency and geo-based video analytics granted by novel 5G multi-access edge computing architecture systems. The results show that it achieves a more coordinated delivery of media services with higher average bitrates.

Fourth, a network resource allocator provisions an efficient network topology and cardinality in order to shield quality of experience of a traffic demand forecast for media services. The accuracy of the results is better as the demands in bandwidth are higher. So, the wider the media service demand, the more confident this approach becomes.

Resumen

5G promete expandir las capacidades futuras de las redes móviles mediante un alto ancho de banda, una baja latencia y la capacidad de proveer conectividad de forma masiva y en un régimen perpetuo, sin fallos. Esta revolución en las redes supondrá la introducción de nuevas tecnologías para la asignación de forma escalable y flexible de recursos de red, tanto en la infraestructura de acceso como en el núcleo de la red móvil.

En este contexto, el inmenso volumen que constituye el tráfico de datos de vídeo, convierte la eficiencia con la que comunican los servicios multimedia en un aspecto crítico para los sistemas de gestión y optimización de operadores de red. Sin embargo, la red, adoptando una posición neutral de funcionamiento, no ayuda a fortalecer los parámetros que inciden en la calidad de experiencia. Este hecho se ve magnificado por la previsible criticidad de las comunicaciones multimedia alentado por la masiva llegada de sensores provenientes de otros sectores como *Internet de las cosas*, el *vehículo conectado*, salud o *industria 4.0*. En consecuencia, las soluciones diseñadas para realizar un envío de tráfico multimedia de forma dinámica y eficiente cobran un especial interés.

Aquí, los hábitos del usuario y sus expectativas cuando consume o produce contenidos multimedia han cambiado radicalmente. En primer lugar, los servicios multimedia gestionan un nutrido catálogo de contenidos sin etiquetar, del cual se necesita mejorar su relevancia para suscitar el interés que le permita llegar a la audiencia objetivo y captar un mayor público. En segundo lugar, la popularización de determinados servicios o contenidos trae consigo una mayor demanda que supone un número elevado de sesiones simultáneas tratando de obtener una calidad de la experiencia estable y

óptima. De este modo, garantizar la calidad de experiencia se vuelve un aspecto fundamental para mantener la audiencia.

Las redes 5G atesorarán las cotas exigidas de calidad de experiencia necesarias por la siguiente generación de servicios multimedia, dada su enorme presencia en la red. Para tal propósito, la trasformación digital de las redes, mediante arquitecturas y tecnologías *cloud*, y los avances en la capacidades radio resultarán en unas redes más agiles y robustas. Lejos de delegar toda responsabilidad de unas comunicaciones multimedia estables y eficientes a las redes 5G, los servidores multimedia y los reproductores de contenidos también pueden complementar a la infraestructura de red. Cada uno de ellos puede aportar soluciones para mejorar la calidad de la experiencia o prevenir degradaciones o cortes del servicio. Más allá de resoluciones 4K o UHD, los usuarios de servicios multimedia esperan una experiencia de reproducción multimedia fluida que se adapte de forma dinámica a los intereses del usuario y a su contexto de movilidad. Por ello, el reto es llevar la calidad de experiencia a un nuevo nivel.

Los servicios multimedia deben ser adaptados a las diferentes variaciones de las condiciones radio de la red. Nuevas tecnologías de envío multimedia sobre HTTP tienen un diseño que dota de estas capacidades de adaptación en movilidad a la vez que añade el soporte a un amplio ecosistema de dispositivos, preferencias lingüísticas e infraestructuras de distribución multimedia. Estas tecnologías permiten que el reproductor multimedia escoja dinámicamente entre diferentes calidades durante la reproducción para mitigar cambios repentinos en las condiciones de conectividad.

Sin embargo, en situaciones de alta densidad de usuarios accediendo a contenidos multimedia en un celda a través de una misma antena puede producir efectos que dañen la calidad de la experiencia, desde situaciones de parones a cambios constantes de calidad. Esto se debe a la presencia de múltiples reproductores tratando de optimizar de forma autónoma el uso de ancho de banda y su calidad, basándose únicamente en decisiones instantáneas.

La apuesta del sector de telecomunicaciones es capacitar las infraestructuras radio con sistemas que permitan a terceros mejorar o expandir sus servicios de un modo más coordinado. Esta solución convierte una estación base en un servicio donde mejorar dinámicamente la calidad de experiencia de un servicio específico. Los operadores de red pueden exponer las interfaces a sus infraestructuras radio y autorizar a aplicaciones de terceros el acceso a información de red en tiempo real. Esto trae consigo nuevas posibilidades para monitorizar el rendimiento de la red, procesar estadísticas y ajustar parámetros operativos del envío de datos. Todo ello sin latencia y de forma transparente a los servicios conmutados. Además, la descentralización de estos servicios a las infraestructuras radio incorporan agilidad y adaptabilidad a un contexto concreto.

En la misma línea de mejorar el envío de servicios multimedia, la capacidad de cambiar dinámicamente y en tiempo real de un proveedor de servicios de distribución de contenidos a otro es un escenario que cobra cada vez más importancia. La materialización de tal solución es factible a través de la utilización de analítica de datos de envío multimedia. Sin embargo, estas soluciones tienen a ser soluciones propietarias que requieren la integración de librerías en el servicio y la interpretación humana de los resultados.

Si nos centramos en el núcleo de la red, la industria de telecomunicaciones persigue la autonomía de la red para operar del modo más conveniente. En este caso, sistemas de gestión centralizados y autónomos configuran la red conforme a cambios persistentes y profundos en el tráfico. Para ello, algoritmos y técnicas de aprendizaje automático son aplicables para, por ejemplo, predecir la demanda de un determinado servicio. El pronóstico puede ser utilizado a su vez por otro algoritmo que evalúe la topología de red más adecuada para absorber dicha demanda bajo unas cotas de calidad de servicio, uso energético y costes operativos. La topología resultante puede ser provista a través de un sistema de virtualizado de redes.

Los beneficios de las redes 5G para satisfacer las necesidades de la siguiente generación de servicios multimedia es evidente. Sin embargo, 5G incorpora nuevos parámetros y tecnologías que pueden ser explotados en

pro de la calidad del servicio multimedia. Ellos proveen a su vez nuevos retos en entornos 5G. Conviene por tanto tener presente las características propias del envío de experiencias multimedia en redes 5G al suponer a la vez un reto y una oportunidad a la hora de abordar el diseño e implementación de nuevas soluciones. En primer lugar, la conexión masiva de clientes, donde la red 5G provee servicios multimedia a una gran cantidad de usuarios que espontáneamente acceden a los mismos. En segundo lugar, la alta densidad de clientes en celdas de redes 5G favorece la lucha de los diferentes dispositivos en un área por los recursos de red disponibles. En tercer lugar, los interfaces de los elementos radio de la red posibilitan el análisis de métricas de red para optimizar de forma dinámica, automática, distribuida y sin latencia el envío de datos multimedia. Por último, la habilidad de auto-gestión de redes 5G para implementar la escalabilidad de la red a través de sistemas de gestión que muten la topología de la red gracias a la transformación de los nodos de una red en entidades software virtualizadas.

Para mejorar la calidad de la experiencia de servicios multimedia en entornos 5G la investigación llevada a cabo en esta tesis ha diseñado un sistema múltiple, basado en cuatro mecanismos.

Primero, con vistas a mejorar la captura de la audiencia, los servidores multimedia deben mejorar la experiencia del usuario de una red social con contenidos mejor etiquetados. Para ello, el análisis multimedia del vasto catálogo de contenidos producidos y compartidos por los usuarios es vital. Para ello, los servidores multimedia deben ser capaces de coordinar una granja de recursos de computación espontáneamente conectados a sus sistemas. De tal modo que el servidos asigne dinámicamente tareas de procesado multimedia que puedan demorarse en función de las capacidades de cada dispositivo de los usuarios.

Segundo, para alcanzar una experiencia del servicio homogénea para todos los clientes que comparten una celda, es necesario un mecanismo que, alojado en el reproductor multimedia, permita elegir una calidad multimedia apropiada para cambios repentinos y inesperados en las condiciones de la red.

Tercero, cuando las decisiones tomadas de forma autónoma en cada cliente no son suficientes para garantizar una experiencia fluida, estable y uniforme en todos los clientes, dado el desconocimiento de la presencia de los demás, un mecanismo para coordinar la selección de la calidad para adaptarse a las condiciones de red es necesario. En este caso resulta necesario apoyar dicho mecanismo en la propia red y sus métricas. Además, dicho mecanismo podría explotar no sólo métricas a nivel de enlace (radio) sino a nivel de red, para determinar situaciones desfavorables en cuanto al rendimiento de los proveedores de infraestructuras de distribución multimedia, para cambiar a otro proveedor de forma transparente al servidor y al cliente del servicio.

Cuarto, un mecanismo que otorgue al sistema de gestión de la red de la capacidad de provisionar de forma dinámica nuevos recursos y elementos de red configurados para formar una topología que satisfaga necesidades del cliente de calidad del servicio a la vez que se mantengan controlados los costes operativos del operador de red.

Cada mecanismo explota diferentes habilidades para permitir a los servicios multimedia y operadores de red, atraer, equilibrar y asegurar una calidad de experiencia sobre redes 5G basándose en nuevas técnicas de envío multimedia. Destacar que para implementar, desplegar, probar y evaluar las contribuciones fruto de las actividades de investigación, se ha empleado una infraestructura real Long Term Evolution (LTE). Sobre dicha infraestructura los resultados obtenidos para cada mecanismo se resumen a continuación.

El primer mecanismo, llamado SaW, crea una granja elástica de recursos de computación que ejecutan tareas de análisis multimedia que no requieren un tiempo de ejecución concreto. Los resultados de los experimentos confirman la competitividad de este enfoque respecto a granjas de servidores especialmente cuando el número de dispositivos conectados al servidor es grande y cuando las tareas a repartir son independientes, atómicas y pueden ser encoladas.

El segundo mecanismo, llamado LAMB-DASH, para la selección de la calidad en el reproductor multimedia, ha sido diseñado e implementado para incurrir en una baja complejidad de procesamiento. Las pruebas realizadas para flujos bajo demanda y en vivo concluyen su habilidad para mejorar la estabilidad, consistencia y uniformidad de la calidad de experiencia en los clientes que comparten una celda de red. Para ello, se ha logrado una baja desviación de la calidad de experiencia desde su valor medio.

El tercer mecanismo, un sistema 5G para la parte radio de la red, llamado MEC4FAIR, explota las capacidades de nula latencia y procesamiento de datos en un contexto geo-localizado para dotar al servicio multimedia de la capacidad de analizar métricas del envío de los diferentes flujos. Los resultados muestran cómo habilita al servicio a coordinar a los diferentes clientes en la celda para mejorar la calidad del servicio.

El cuarto mecanismo sirve para provisionar recursos de red y configurar una topología capaz de conmutar una demanda estimada y garantizar unas cotas de calidad del servicio. En este caso, los resultados arrojan una mayor precisión cuando la demanda de un servicio es mayor.

Acknowledgements

First of all, I would like to thank my supervisors Jon Montalbán, Julián Florez and Pablo Angueira. They guided me in the Ph.D. process over the last years. More specifically in the appropriate way to describe ideas and key aspects of a research activity. This will for sure conduct my research publications and my projects' presentations in the future.

I would also like to thank all my colleagues in Vicomtech. Roberto Viola, thank you for sharing with me all the research, surveillance of the state of the art, development of algorithms, experiments, tests and writing process; Josu Gorostegui, thank you for your priceless skills for implementing libraries for HAS technologies and encryption; Mikel Zorrilla, thank you for your time and dedication, for providing formal policies and deep reviews for the manuscripts and for finding new opportunities with projects and funding for such projects; Iñigo Tamayo, thank you for your great support in the technical design; Felipe Mogollón, thank you for your excellent attitude and aptitude addressing problems; Ana Dominguez, thank you for sharing procedural information about the PhD process and your positive mood; Gorka Velez, thank you for show me that is possible the straight direction to chase opportunities; and Marco Quartulli for apparently crazy ideas that sometimes are turned into brilliant ones.

I would like to thank all my colleagues from the CogNet project, it has been a great adventure and a fantastic framework to develop part of the research of this Ph.D. The CogNet project has received funding from the European Union's H2020 and the 5GPPP Programme for research, technological development and demonstration under grant agreement 671625 (H2020-ICT-2014-2, Research and Innovation action). Futhermore, I would like to thank the NITOS testbed patron, Donatos Stavropoulos, from Fed4Fire+

H2020 project. Its open call for experimenting on top of 5G SDR open testbed made real experimentation possible.

I would also like to thank other people essential to this Ph.D. process. Fernando Díaz, from University Carlos III of Madrid, thank you for your mentoring when I started the Ph.D. process in Madrid and introducing me into research activities. Last but not least, special thank you to Gregory Maclair for all the time spent in conversations and time together to commute along a distance equivalent to go to the moon. It is a real pleasure to have worked with all of you.

Finally, I would like to express my gratitude to Vicomtech for providing me with a great environment to carry out my research and create this Ph.D. dissertation. Thank you to management, Julián Flórez, Jorge Posada and Edurne Loyarte, and to my director of department, Mikel Zorrilla, for helping me as much as necessary.

Gracias

Ángel Martín Navas

June 2018

Contents

List of Figures

List of Tables

Part I

Introduction

CHAPTER

1

Scope of the research

1.1 Motivation

5G promises to expand the possibilities and capabilities of mobile networks. This technology revolution will be achievable only with the introduction of new technologies, both in the access to and in the core of mobile networks, such as the flexible and scalable assignment of network resources. The scalable management framework will enable a reduction of the network management Operational Expenses (OPEX) by at least 20% compared to today.

5G shall provide an answer to new rates of: volume, both on the downlink and the uplink while taking benefit of advances in video compression and transmission solutions, and low-cost storage and caching; mobility, to deliver the best network connectivity of media services anywhere, regardless of from the user's location or nomadic movement; density, to deliver a steady and stringent network connectivity of media services anywhere, regardless of from the users' physical concurrency; security, to provide efficient access control of cached video content; and quality, to provide appropriate performance parameters (latency, bandwidth, security, connectivity...) to the business and operational requirements of the media service.

According to Cisco reports and forecasts, over three-quarters (78%) of the world's mobile data traffic will be video by 2021. This turns efficiency of video delivery into a

core application to manage and optimize. Media services and traffic represent the most engaging and crowded data consumption of Internet users in the entertainment sector nowadays.

User habits and expectations when it comes to media consumption and production are changing profoundly. Media services need to cope with an increasing demand in terms of data rates, number of simultaneous users connected and/or more steady and accurate quality requirements. High quality and high-resolution audio-visual services are important drivers for increased downlink data rates where 5G promises to provide cost-effective media delivery. At the same time, user generated content as well as the use of cellular technology for professional and semi-professional media production are key drivers for increased uplink data rates. 5G will enable this viable and immensely growing area of cellular and IP-based live media production as a business to grow further, supporting new business models, such as production in the cloud.

The benefits gained from the 5G network in satisfying next generation media service needs are evident. However, additional aspects from 5G such as new parameters and possibilities, key to the quality of the media service, can be exploited. 5G ships new parameters and technologies which can play a significant role in enhancing the quality of the media services. This research work is focused on some of these major changes. **First**, the massive client connections volume where the 5G network handles a huge pool of devices spontaneously connected to media services. **Second**, the dense client cells where the media players strive to deliver the best performance when massive media sessions originate from a specific area. **Third**, the edge video analytics by exploiting network components interfaces to dynamically and automatically optimize the media delivery based on accurate, granular and geo-binned metrics, in a distributed and zero latency manner. **Fourth**, the self-organising network ability of 5G, where scalable network management systems on the network stack exploit the transformation of network functions into software and virtual entities in order to mutate the network.

In the following paragraphs the motivation for each of them is explained.

First, the quality of the media service is an essential aspect in maximizing user loyalty, engagement and offering a compelling service. Service providers aim to engage the audience, eager for contents, by boosting the media relevance. Therefore, it is necessary to improve the matching of user interests with the huge content database and reveal

hidden connections between items through deeper tagging. In other words, the service is enhanced by improving the media content indexing.

The social media paradigm has led to a significant rise in the volume of user generated content managed by social networks with millions of users accessing services, each of them often using multiple devices at the same time. To enhance media relevance, *a deeper automatic tagging* system is needed. Media tags enable better matching of user interests with the content database and reveals underlying connections between items, such as applying face detection mechanisms or content-based indexing to find related videos. Image analysis algorithms empower automatic retrieval of salient features, but they also involve computing-intensive functions. Therefore, the processing requirements grow substantially when all the media items comprising the social network database need to be analysed. The OPEX of the required infrastructure to automatically tag media uploaded to a media service and apply new tagging campaigns over the full catalogue to expand the detected features or taxonomy, could be unaffordable for social media services.

Second, once the audience has been captured, the objective of the media service is to increase audience retention, where the Quality of Experience (QoE) plays a significant role. The goal of media services is to deliver a smooth and high-quality playback, with low video start times and high bitrates while reducing buffering.

The media delivery standard to satisfy the previously mentioned goals and universally adopted by media services is HTTP-based Adaptive Streaming (HAS). HAS responds to demands from multimedia services supporting heterogeneous display setups, different user preferences and languages and changeable mobility situations with a Content Delivery Network (CDN)-ready design. HAS is a pull-based protocol [Begen et al.11] that easily traverses middleboxes, such as firewalls and NAT devices. At the same time, it keeps minimal state information on the server side, making servers more scalable than conventional push-based streaming servers. Last but not least, concerning existing HTTP caching infrastructures, HAS allows distributed CDNs to enhance the scalability of media delivery, where an individual segment of any content is cacheable as a standard Web object.

HAS solutions provide a manifest file detailing a playlist of segments with the available media representations for different resolutions, languages, views and bitrates. The

essence of this approach is the transformation of the traditional push-mode to a pull-mode. This way, the service delegates the responsibility of operating the service in a proper and efficient manner to the players. The aim of the bitrate selection algorithm is to maximize the quality of the playback according to the bandwidth availability constraints. To this end, the *players autonomously take real-time decisions to request a specific segment tied to a nominal bitrate*.

This client-driven approach, where control is distributed over the various clients and each client strives to optimize its individual quality, brings up some issues that can damage the QoE. The issues span initial buffering delay, temporal interruptions or pauses, and visible video resolution switches during a video transmission [Seufert et al.15].

Furthermore, a client-side decision algorithm might not be sufficient for guaranteeing the best performance since each client is unaware of the presence of others in a dense client cells. This client-side decision approach is missing the in-network knowledge.

Third, from the network perspective, the quality of the network experience is an important element in customer satisfaction and retention. A key requirement of 5G will be to create a network that is highly optimised to make maximum use of available radio spectrum and bandwidth for Quality of Service (QoS). The goal is to provide the best possible QoS in order to get a live, fluent and continuous multimedia experience.

5G Multi-access Edge Computing (MEC) is a foundational network architecture concept integrated into the mobile network infrastructure in 5G. European Telecommunications Standards Institute (ETSI) envisions a video analytics use case where the 5G MEC technology guides the video server to apply the optimal bitrate to a particular video stream or user based on the radio conditions [ETSI18]. The idea is to use *Radio Access Network (RAN) analytics to determine/estimate the throughput* likely to be available at the radio downlink interface for a user, and then use packet headers to convey that information to the video server, so that it can adapt the stream accordingly. This way the streaming service achieves a noticeable performance improvement when operators communicate RAN conditions to the video server in this way.

However, the integration into a real mobile Software Defined Radio (SDR) network and validation performed on a real, rather than simulated setup, to check the feasibility and performance of an active component of the video delivery chain at the mobile

edge to get fair QoE in dense client cells and enforce CDN provision, has not yet been performed.

Fourth, going deeper into 5G systems, 5G must deal with fast, heterogeneous, multi-tier networks, which are also dynamic in nature. A digital transformation, enabled by cloud architectures and technologies, is taking place in 5G networks, disrupting the way in which the network works. Turning networking functions into software entities with common interfaces that can be remotely and dynamically operated is a significant breakthrough towards agile network management. This change allows the next step to be taken, based on the virtualization of those software-based network functions such as common 5G base technologies, thereby enabling innovation and network transformation. Unlike previous mobile network technologies that attempted to provide a "one size fits all" infrastructure, 5G mobile networks are designed to provide optimized setup for a variety of heterogeneous services and types of end users. Hence, Network Function Virtualization (NFV) [Foundation13] and Software Defined Networks (SDN) [Foundation12] are two key enabler technologies of 5G. NFV leads to cost efficiency, improvements in time-to-market and innovation in agile network infrastructure and applications. SDN enables network administrators to manage network services through the abstraction of lower-level functionality. This is achieved by decoupling the system that makes decisions about where traffic is sent (the control plane) from the underlying systems that forward traffic to the selected destination (the data plane). Implementation of SDN results in infrastructure savings, operational savings and flexibility [Kim and Feamster13]. On top of SDN technologies, it is possible to develop systems to autonomously improve network agility and flexibility to efficiently support the evolving demands of users. For example, *5G optimization tools can provide elements of control-path selection* and manage prioritization for different traffic types depending on their importance in a cost-effective way [Xu et al.13], and the paths are directly related with the topology of the network.

Today, the selection of the most efficient topology to assure an operational QoS and QoE of an incoming traffic demand of a media service has not yet been performed [Bizanis and Kuipers16]. There is not a reliable solution that addresses the problems for flexible creation by scaling an elastic network up/down or in/out in an automated manner [Szabo et al.15].

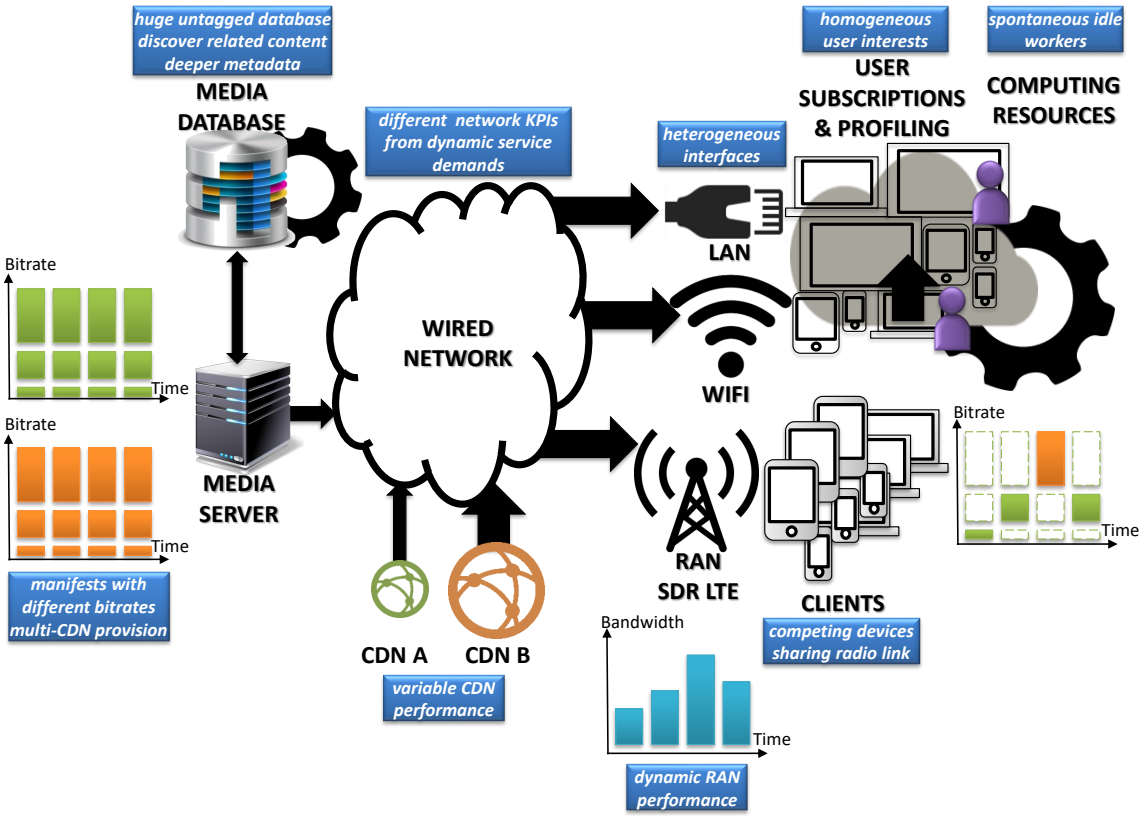


Figure 1.1: Overview of media delivery angles for improved QoE in 5G environments.

Figure 1.1 includes all the previously explained corners coming into play in a common communication schema. This diagram also includes media service actors and some media delivery aspects to improve the QoE. On the one hand, media servers manage huge volumes of untagged contents that need to be inter-related in order to suggest relevant or interesting contents to users. However, the processing capacity to analyse all the contents to improve the content connection could not be affordable. Likewise, the need to monetize content storage and delivery could place the analysis out of scope. A solution to create a pool of workers from the spontaneously connected clients, benefiting from idle hardware computing resources in the background when consuming streaming videos would make users' assets improve the service. Contents are tagged to capture audience making catalogue navigation, browsing and search easier. Furthermore, the audience retention also needs to be enhanced. In this regard, different actors come into play, first media servers and clients employ HAS streaming technologies shipping several bitrates to fit into heterogeneous display sizes and networking conditions. Then, the network needs to deal with the dynamic demands and different sets of Key Performance Indicators (KPIs) required from specific services for an optimal QoE. In respect of the mobility trends, the media delivery focus is placed at the network edge. The autonomous optimization of the QoE of independent players turns the utilization of a shared radio link into a competition for available bandwidth. In this scenario, a network element with RAN-awareness could control the bitrate selection of media players. Last but not least, media services can be delivered using multiple CDN providers. The URL of cached media segments is included in the MDP, this way the media server can switch to a specific CDN for a specific region or country. However, the performance of CDNs is not stable and can degrade QoE and produce service outages. Media services need to shield themselves from these situations to prevent QoE impacts. Accordingly, Figure 1.1 shows all the interconnected media actors, such as media server, database, network, CDNs and clients' devices accessing though different network interfaces. It also represents the environmental or technological conditions to be exploited. Finally, the icons on the right side show other homes and users *on the move*, sharing a multi-user experience.

Eventually, to fully understand the scope of our research and its challenges, the following list summarises the considered contextual factors regarding the demographic trends in media consumption habits of users together with technological alternatives

and business models. This is reinforced and aligned with the accumulated experience from first level telco operators, such as Telefónica I+D (http://www.tid.es/) and Orange (https://www.orange.fr/), widely participating in standardisation bodies such as ETSI and 5G Public Private Partnership (5GPPP), technology providers, such as Nokia (https://networks.nokia.com/), IBM (https://www.research.ibm.com/labs/) and Interoute (https://www.interoute.com/), universities, such as UPM (Universidad Politécnica de Madrid http://www.upm.es/), TUB (University of Berlin http://www.tu-berlin.de) and University of Trento (http://www.unitn.it/en), and research institutes like WIT (Waterford Institute Technology https://www.wit.ie/), Fhaunhofer Fokus (urlhttps://www.fokus.fraunhofer.de) and Vicomtech (Visual Interaction & Communication Technologies http://www.vicomtech.org/) working in the project CogNet [EC15]:

1. Multimedia consumption is gradually shifting from traditional TV to streaming video on mobile devices. Furthermore, according to demographic studies, the trend shows a sharp increase in streamed video viewing, particularly among younger generations [Ericsson15].

2. The combination of the increasing number of video streaming users heavily dominating the traffic over the Internet, the demanded high quality from the cutting edge displays of their devices and the required support for mobility is driving the evolution of media services. Fuelled by improved cameras with stunning picture quality [Saad et al.15] and the breakthroughs in display technology [Kathirgamanathan et al.15], the traffic for videos delivered over the Internet will reach 80% of the total Internet traffic by the end of 2019, according to the report issued by the world IT leader Cisco [Inc17b]. Meantime, reaching heterogeneous devices gains relevance thanks to the growth of mobile devices as an entry point to these services [Inc17a].

3. Nowadays video streaming services work on top of unmanaged delivery networks, where quality is not guaranteed, on a best-effort basis [Sodagar11].

4. The QoE degradation is tight in dense client cells, when considering a cellular network, the RAN, a Wi-Fi hotspot and the network edge. There, it becomes complex to provide video services to several users competing independently for the

available bandwidth when trying to maximize the used bitrate. This autonomous optimization makes the connection conditions highly changeable leading to continuous fluctuations of the target bitrate, artefacts, interruptions and disproportional shares of available bandwidth [Akhshabi et al.12]. Here, there is a trade-off between keeping a lower constant rate and dynamically adapting its rate with the risk of upsetting user experience.

5. The explosion in multimedia services brings about a higher consumption of networking resources needing sustained bandwidth and latency demands. This high-performance regime makes the capacity of the networks more critical to the user experience. Subsequently, usually the video services are usually employed to demonstrate distinctive advances and new features from a telco operator [Hagos16].

6. Service Level Agreement (SLA) is transforming the operational features of networking functions from reliability to agility. Traditionally telecoms equipment is expected to provide 99.999% availability [Liu et al.16a], however with many modern IT services requiring different levels of guaranteed bandwidth, latency and priority over other traffic, SLAs have become more important and more differentiated depending on the nature of the service.

7. International consortiums such as, the European Telecommunications Standards Institute (ETSI), and the International Telecoms Union (ITU) are driving the digital transformation of 4G networks towards 5G. Commoditization and virtualization of wireless networks will change the economics of mobile networks to help MNOs move from proprietary hardware vendors to virtualized software platforms through the abstraction of the execution environment. SDN is an architecture designed to enable more agile and cost-effective networks. SDN allows a dynamic reconfiguration of the network by taking a new approach to the network architecture. SDN permits the centralization of network management for different entities within a cellular network.

8. Scalability and granularity issues, such as the increased number of clients and size of the infrastructure, of 5G management systems are met by capillary SDR sys-

tems. On top of this, the MEC concept has evolved to draw on NFV technologies to allow VNFs to run as a distributed edge platform.

1.2 Hypothesis

The research developed in this PhD focuses on four scenarios where the common goal is to improve the QoE in 5G networks:

1. the engagement of users in social media services by deeper tagging resulting from downloading media analysis tasks from the service to the clients;
2. the fair and efficient utilization of a shared radio links in dense client cells when competing independently for the available bitrate;
3. the distributed control of the media sessions by the network edge exploiting network performance awareness and
4. the adaptation of the network topology to forecasted demands of media services where new topology guarantees a minimum QoS.

These scenarios supply some favourable conditions, specific environments, required features or applicable technologies which must be described to fully understand the research context.

According to the scenario depicted in Figure 1.1, a set of hypotheses has been compiled to serve as the basis of the PhD research. In this regard, hypotheses pivot around the media delivery chain. Here, four different corners such as media service, network core/backhaul, network edge and media players are considered. The working hypothesis is constructed as a statement of the following expectations:

- **Media Service**.
  - Needs to improve media engagement by means of better and deeper media tagging. The service aims to retrieve underlying connections in the contents that let recommendation systems improve their relevance and audience engagement.

  - Does not have enough computing resources to perform current and future analysis functions for the full catalogue of contents.
  - Can deal with the privacy and security concerns of outsourcing media analysis.
  - The analysis functions to be applied are:
    * Atomic. The data input and the function to be applied can be encapsulated for delivery. The volume of data representing each task is low and the retrieval of image dataset is costless, by employing keyframes of the encoded video, for example.
    * Lightweight. The function to be applied takes a short time compared to the average media catalogue duration when executed on a suitable device.
    * Autonomous. The analysis task does not depend on other contents of the full content, it can be applied to just one frame.
    * Delay-tolerant. The order of results is irrelevant and the time to complete a full batch of analysis tasks has no impact.
    * Partial. Not necessary to complete the full batch of tasks to all the images set to be exploited.
  - When Media Service providers require highly-demanding but delay-tolerant computing resources to improve their services, such as generating automatic tagging of the content. To this end, the service will make use of the client-devices as an infrastructure to distribute the processing tasks among all clients and reduce the workload in the service provider's cloud server.

- **Media Server**.
  - Schedules a queue with asynchronous media analysis tasks.
  - Has a timeout to send a task back to the queue.
  - Has a temporal black list to avoid dispatching tasks to unresponsive clients or to avoid over-utilization of a device's resources.
  - Is able to attach data to existing media streaming sessions.

  - Can assess the task to be done to match a target level of computing capacity from the devices connected and idle.
  - Provides a manifest of media contents including representations for a wide range of bitrates.
  - Employs multiple CDNs to serve the segments of the contents and defines the CDN endpoint in the manifest as a base URL for the segments of the content.
  - Uses technologies employing media encoding and streaming standards.
  - Expands the request-response transaction model into an asynchronous workflow dispatching to connected clients along the served media session. To this end, Media Server creates a queue to complete big volumes of image analysis tasks. To pop tasks from the queue, Media Server matches the task volume with the device computing profile.

- **Media Database**.
  - Stores all the contents available to be distributed across the CDNs.
  - Includes metadata employed by the media service to recommend other contents.
  - Includes new metadata coming from new features. Those processed employs and without a consistent result are marked in order to get off the queue of pending to be processed.

- **CDN**.
  - Caches requested contents from the database.
  - Provides media on a best-effort basis. CDN can be congested and consequently the service performance would be degraded. In the worst case scenario the service could suffer outages.

- **Core Network**.

  - Follows SDN and Virtualization paradigms from 5G to develop systems to autonomously improve network agility and flexibility to efficiently support the evolving demands of users.
  - The management systems have interfaces to get a representation from the topology deployed, i.e. YANG model by means of NETCONF protocol [Ietf10].
  - The network management systems such as OpenDaylight [Foundation17c] and OSM [TID17] allow an agile (seamless, costless and automatic) transition of the Network topology to another one which better deals with incoming traffic demands in terms of KPIs in a cost-effective way. Combining SDN and NFV concepts, the controller changes the network topology instantiating or removing VNFs to forward the incoming traffic in an efficient manner, removing the unused parts of a network to release these resources [Ismail et al.13].
  - Machine Learning algorithms are applied to develop a system of service demand prediction and provisioning which allows the network to resize and resource itself by using virtualisation to serve predicted demand according to parameters such as location, time and specific service demand from specific users or user groups.

- **SDR RAN**.
  - Brings wider possibilities towards distributed mechanisms of traffic coordination in a radio link.
  - Meets the scalability issues, thanks to the capillary nature of SDR systems.
  - Exposes edge API to authorized third parties to provide them with radio network information in real-time. This technology enables operators to better adapt traffic to the prevailing radio conditions, optimize service quality and improve network efficiency.
  - Authorized third parties can enforce their own service in real-time using applications hosted on the MEC servers, which are in the edge close to the end users and delivered through multiple CDNs.

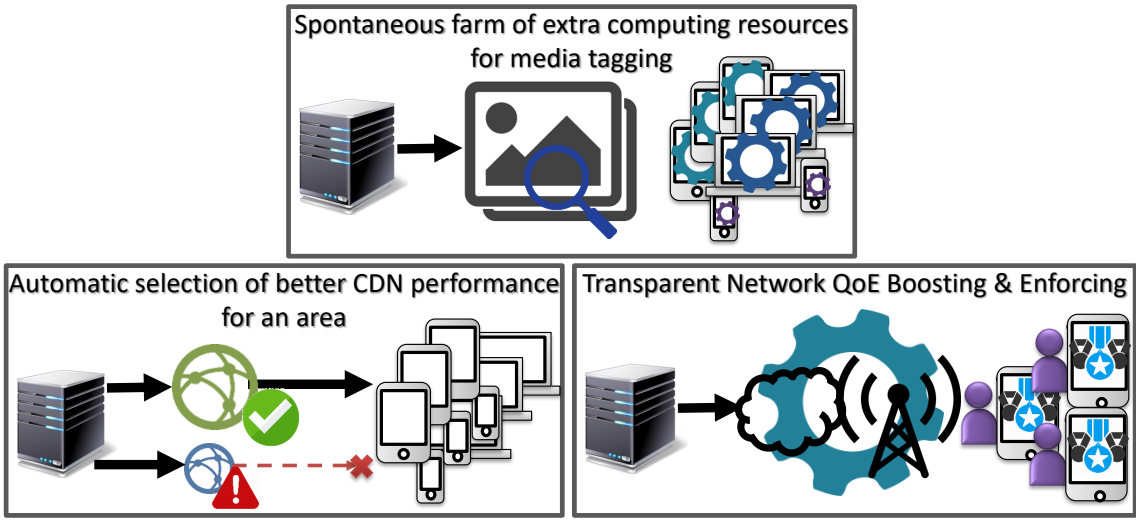
  - MEC turns a base station into a service catalyzer, which dynamically improves network performance and user experience for a specific service by exploiting media streaming analytics.

- **Media Player**.
  - Is able to asynchronously respond to assigned media analysis tasks.
  - Is able to process tasks attached to an ongoing media streaming session.
  - Can profile the computing capacity from the device and send the score to the server.
  - Tries to dynamically obtain the best QoE possible with the information coming from the MPD and the network performance measurements performed on the client side along the MPD and segments download.
  - Uses technologies employing media encoding and streaming standards.
  - When several Media Players share the same radio link to access to contents from the Media Service and they compete for available network resources to improve their QoE.

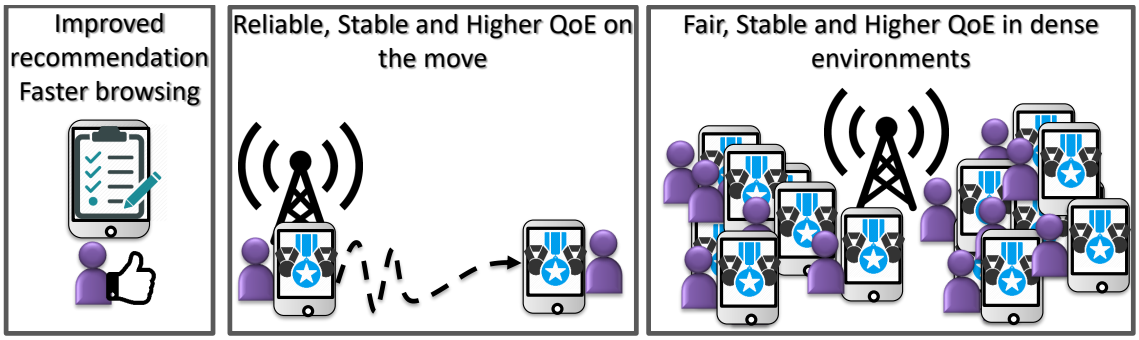
- **Users**.
  - Are actively watching/consuming a media content. The display is on.
  - With similar interests are likely interested in the same contents.
  - Connected in the social media service are likely interested in same contents.
  - Consent in the conditions and terms of media service to share spare processing resources to execute data processing and compilation tasks in order to improve the service.
  - QoE depends on content relevance/interest, average bitrate, frequency of quality switches, frequency and duration of freezes.

The expectations of the working hypotheses involve different stakeholders:

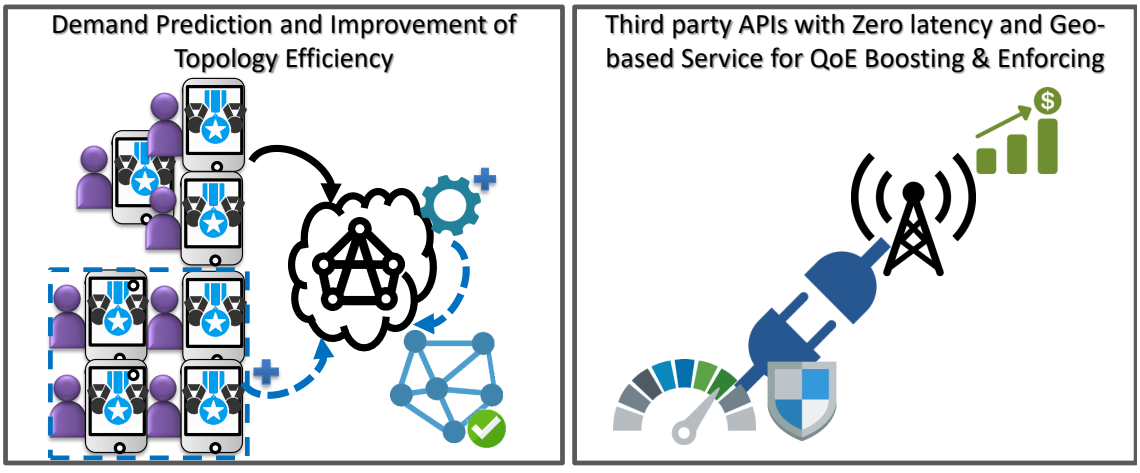
1. the *media service* gets an elastic cloud of computing resources spontaneously connected to the media service in order to perform batch image analysis (1.2a top);



(a)



(b)



(c)

Figure 1.2: Diagram of the hypothesis of the perception of **(a)** the media service, **(b)** the users and **(c)** the mobile network operator for an enhanced experience.

2. the *media service* becomes audience engaged for longer times;

3. the *media service* employs the healthier CDN for media delivery in each area transparently (1.2a bottom-left);

4. the *media service* gets better QoE for the clients transparently without any modification in the server side and communication overheads (1.2a bottom-right);

5. the *users* get more interesting and relevant contents from the media service (1.2b left);

6. the *users* get a steady QoE when they are on the move (1.2b center);

7. the *users* get an unbiased and homogeneous QoE compared to other surrounding users in dense client cells like concerts or sports events (1.2b right);

8. the *mobile network operator* deploys a well fitted network where topology optimises performance and the use of available network and VM resources while minimising overall energy requirements and costs (1.2c left);

9. the *mobile network operator* gets a new revenue flow by opening the SDR APIs to boost or enforce media services in real-time via applications hosted on the MEC servers (1.2c right).

1.3 Objectives

The main objective of this work is to improve the QoE of media services in 5G networks and environments by means of advanced media delivery architectures, solutions and algorithms. Furthermore, the main objective is decomposed into four individual objectives to overcome the orchestration of media delivery resources to enhance or enforce QoE:

1. Create an HTML-based standard and interoperable architecture to dispatch media processing transactions to spontaneous connected media players along media streaming sessions, matching resource profile and media task volume.

2. Provide technologies to enable the adaptation of selected content bitrate in dense client cells with the aim of improving the efficiency of radio link bandwidth utilization and the QoE fairness across all the media players.

3. Integrate a media service application to transparently boost and enforce media delivery applications in SDR solutions.

4. Empower network management systems using Machine Learning algorithms to allow the network to resize and provision itself, to serve a predicted media service demand.

According to this objective breakdown, it is necessary to address and provide solutions to overcome the four main challenges of media delivery for media services in 5G environments (see Figure 1.3):

- *Massive client connections*: ever increasing volume of connected users in *5G* can be exploited to dynamically build a processing infrastructure composed by thin devices to complement a cloud server. To this end, the media service enrols spontaneous connected users' device as computing resources of an elastic cloud to perform media analysis tasks in the background during the media session.

- *Dense client cells*: density of users in *5G* cells will introduce highly dynamic network conditions. Ensuring a steady and consistent *QoE* in dense client cells is complex, therefore, media players need to get a more accurate assessment of the effective bandwidth and awareness of the concurrency level in order to manage the efficiency and fairness radio-link utilization trade-off. This way, media players will leave behind autonomous bitrate adaptation in a best-effort basis.

- *Edge video analytics*: the *5G* MEC architecture exploits network edge awareness of connectivity performance to guide the media player in choosing the optimal bitrate to be used given the radio conditions. Thus, using RAN analytics at SDR components to estimate the throughput likely to be available at the radio downlink interface for a media player and influence transparently in their bitrate adaptation decisions minimizing impact on the *QoE*.

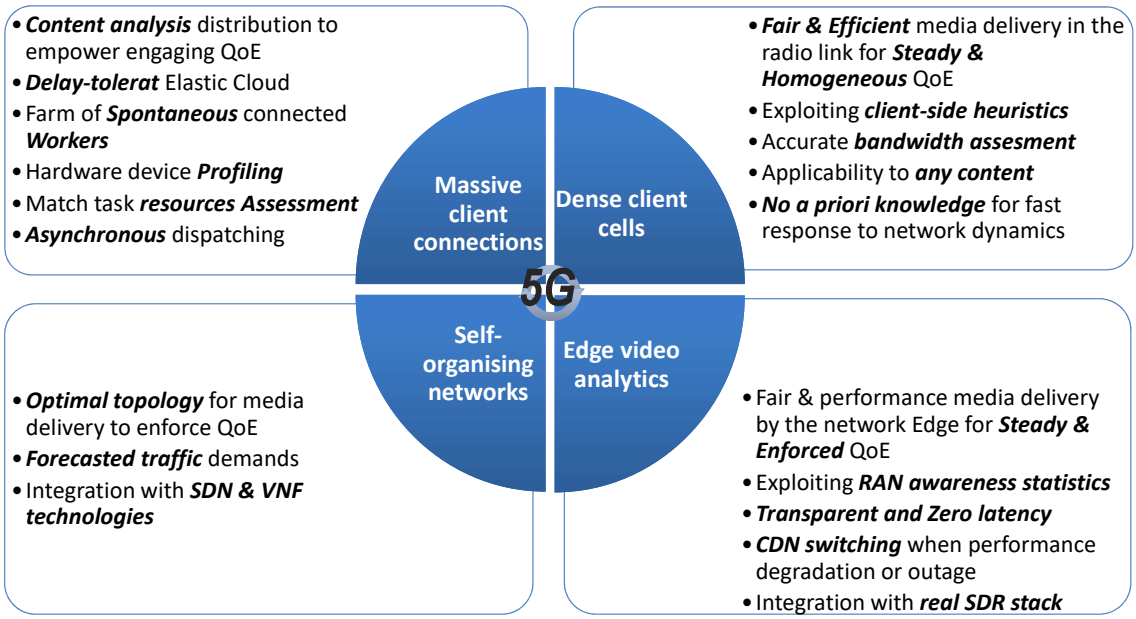


Figure 1.3: Main challenges to be addressed in order to achieve the main objectives.

- *Self-organising network*: application of Machine Learning technologies over *5G* technologies, such as SDN and VNFs, to forecast traffic demands and identify alternatives to automatically taking preventive actions to network degradation conditions such as congestion at both a network wide level to avoid overall *QoE* impact.

1.4 Contributions

The main contribution of this Ph.D. research is founded on the advances in media delivery technologies to provide an enhanced QoE of media services to different stakeholders including media services, media players and MNOs. These advances, based on standard solutions, enable context-sensitive, standard-compliant, fair-sensitive, CDN-aware and self-organising media delivery in new dynamic, agile, dense and capillary 5G environments.

More specifically, the main contribution can be translated into four specific outcomes. Figure 1.4 illustrates the four contributions of the research in a wider context

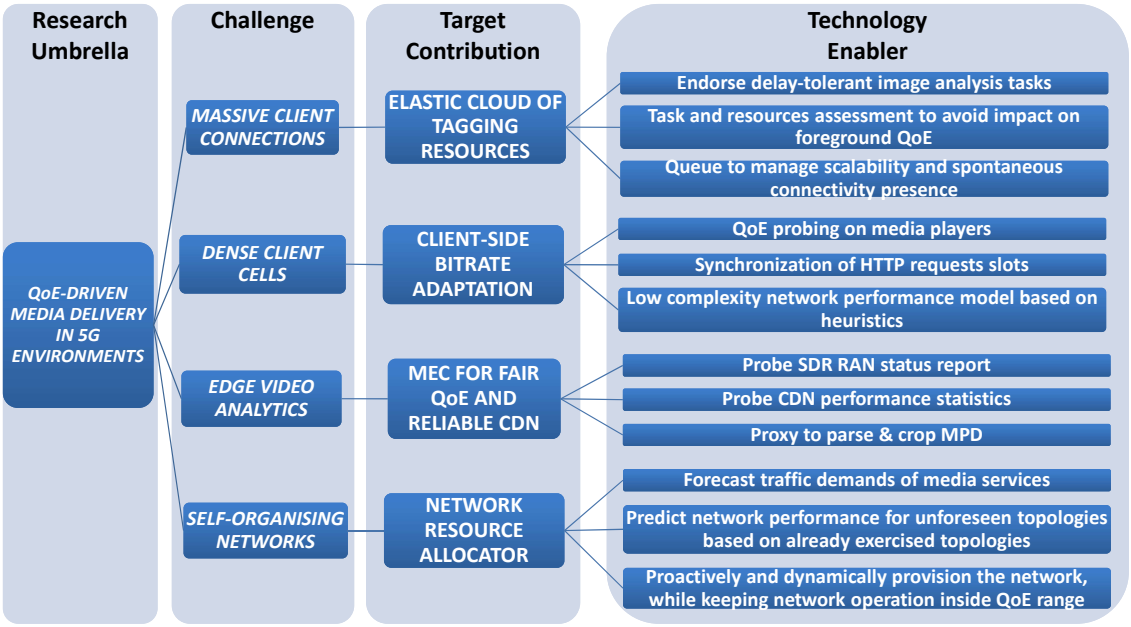


Figure 1.4: Diagram of the contributions of the research.

to address the creation, delivery and management challenges of multi-device media services.

1.4.1 Elastic Cloud of Tagging Resources

Media processing demands for different application domains have increased monotonically as the amount of information has exploded with advances on devices and network capacities. Cloud platforms are the solution for Big Data but they involve a significant cost.

This thesis has targeted Web-based social media content services, such as YouTube [Youtube17] or Vimeo [Vimeo17], as a dominant source of traffic for incoming 5G networks. In the case of social media services, they need to improve service monetization by means of higher user engagement. To this end, deeper media tagging is required. At the same time, the vast number of devices connected to the service means a significant amount of computing assets valid for delay-tolerant and independent tagging tasks. The common behaviour of the users of these services provides a convenient environment for the deployment of automatic tagging systems. On the one hand, a steady and continuous communication channel along the video consumption and on the other,

users are aware that media services are bandwidth demanding. Moreover, introduced communication and processing overheads should have a residual impact on battery life or expenses [Chen et al.13, Zhang et al.16].

This thesis has designed and implemented a solution to distribute delay-tolerant processing tasks to spontaneous connected computing resources for creating an elastic cloud infrastructure. Moreover, it saves the service provider tagging infrastructure costs, while matching the tasks' volume to computing resources to avoid an impact on foreground QoE.

The results show that the solution is able to exploit high user availability density from 5G networks based on the HTML stack. Furthermore, this work includes a performance-cost model to support service providers to determine suitable scenarios for this approach compared to the utilization of cloud computing servers.

Publication related to Contribution 1.4.1:

- *M. Zorrilla, J. Flórez, A. Lafuente, A. Martin, J. Montalbán, I.G. Olaizola and I. Tamayo, "SaW: Video Analysis in Social Media with Web-based Mobile Grid Computing," IEEE Transactions on Mobile Computing (TMC), vol. PP, no. 99, pp. 1-1. doi: 10.1109/TMC.2017.2766623* in Section 3.2

1.4.2 **Client-side Bitrate Adaptation**

The capacity of 5G cells will be significantly multiplied. Here, it becomes complex to provide video services to several users competing independently for the available bandwidth when trying to maximize the used bitrate. The issues span initial buffering delay, temporal interruptions or pauses, and visible video resolution switches during a video transmission [Seufert et al.15]. However, a service provider wants to provide a biased, consistent and uniform service experience removing circumstantial conditions that would turn experience unfair, heterogeneous and unstable.

When considering a cellular network, this thesis has targeted the potential QoE degradation in dense client cells, when considering a cellular network, the Radio Access Network (RAN), a Wi-Fi hotspot and the network edge.

This thesis has identified, designed, implemented and integrated in real media players a bitrate adaptation algorithm in setups in which multiple players share a connection link making real-time bitrate decisions to conduct a more steady and unbiased

media delivery. The design pivots around two key features. First within its flexibility to produce a fast response, valid for any kind of incoming content characteristics or connectivity status, meaning that the algorithm does not require *a priori* knowledge. Second, within its simplicity, with a low-complexity heuristic model, based on measurements and estimations from a current stream state. Hence, the mechanism's goal is efficient and fair QoE in dense client cells.

The results show that the solution is able to perform a live characterization of network performance including the concurrent traffic demands while the algorithm requires a reduced background computation on the client side. The solution is valid for balancing QoE and radio link utilization across the devices sharing a radio link in two different scenarios, since the clients tend to use the same representation bitrate. On the one hand, a scenario with clients synchronized to a common clock joining the live stream at once and on the other, an on-demand-like scenario where clients randomly request a stream. The scenario with a synchronized connectivity status assessment produces a more accurate and stable characterization.

Publication related to Contribution 1.4.2:

- *A. Martin, R. Viola, J. Gorostegui, M. Zorrilla, J. Flórez and J. Montalbán, "LAMB-DASH: a DASH-HEVC adaptive streaming algorithm in a sharing bandwidth environment for heterogeneous contents and dynamic connections in practice," Springer Journal of Real-Time Image Processing, Oct. 2017. doi: 10.1007/s11554-017-0728-x* in Section 4.2

1.4.3 MEC for Fair QoE and Reliable CDN

When a client-side decision algorithm is not sufficient in dense client cells for guaranteeing the best performance given that each client is unaware of the presence of others, the network must support the media service QoE.

5G architecture envisions MEC systems as a RAN-aware system that exploits radio link reports existing in the LTE stack. This thesis has targeted 5G systems to empower the network edge with MEC systems. MEC systems can provide RAN awareness in real-time for providing a bitrate adaptation in a distributed and transparent manner.

This thesis has identified, designed, implemented and integrated on a real SDR testbed an MEC system turning media delivery analytics into actionable data to shield

itself from content delivery degradation and outages in a zero-latency and fully capillary way. The RAN-aware system steers stable and unbiased network resources utilization, avoiding situations where media players trend towards radio-link capacity exhaustion, before they reach full utilization, and dynamically switching in real-time CDN. Thus, the goal is to get geo-based fair and reliable QoE in dense client cells.

The results show that the solution is capable of performing real-time updates in the manifest with the suitable qualities and CDN endpoints. This MEC system exploits L2 (link), L3 (network) and L7 (application) metrics to support switching decisions on HAS quality and CDN provider. Furthermore, the solution is integrated and validated on a real mobile LTE SDR network under two different scenarios. The solution plays a more significant role improving efficiency, in terms of network utilization and quality experienced in the stochastic scenario, where clients randomly join an on-demand stream. However, the synchronous scenario, with clients joining a live stream at discrete and synchronous times, obtains better scores than the stochastic one.

Publication related to Contribution 1.4.3:

- *A. Martin, R. Viola, M. Zorrilla, J. Flórez and J. Montalbán, "MEC for Fair, Reliable and Efficient Media Streaming in SDR Mobile Networks," submitted to IEEE Transactions on Network and Service Management (May 2018).* in Section 5.2

1.4.4 **Network Resource Allocator**

Machine learning techniques can make possible to develop systems to autonomously improve network agility and flexibility to efficiently support the evolving demands of users.

This thesis has targeted 5G agility to efficiently and dynamically respond to a variable pool of users producing a dynamic traffic demand with latency and bandwidth KPI constraints that must be addressed. It is favourable for the application of machine learning algorithms which are ideal for feeding SDN technologies with insights.

This thesis has identified, designed, implemented and integrated on a simulation testbed a real-time and autonomous Network Resource Allocator system to distribute the predicted traffic demand. This self-organising network management tool employs Machine Learning, SDN and NFV technologies to dynamically provision the network in a proactive way, while keeping the network operation within business ranges. This

means the system is able to scale the network topologies and assure the QoE, required for media services.

The results show an operational Machine Learning tool able to empower a SDN controller with abilities to forecast a service demand and to instantiate an efficient network topology accordingly in an automated manner. The Network Resource Allocator has been tested and validated for Netflix like, UStream like, and Skype like media services. Moreover, the results conclude that the accuracy of the results is better when the cardinality of the network is bigger and the demands in bandwidth are higher, while the fidelity drops for tiny setups and audiences. So, the more complex the infrastructure and wider is the media service demand, the more confident the approach becomes.

Publications related to Contribution 1.4.4:

- *A. Martin, J. Egaña, J. Flórez, M. Quartulli, J. Montalbán, R. Viola and M. Zorrilla, "Network Resource Allocation system for QoE-aware delivery of media services in 5G Networks," IEEE Transactions on Broadcasting (TBC), 2018. doi: 10.1109/TBC.2018.2828608* in Section 6.2

- *T.S. Buda, A. Martin et al., "Can machine learning aid in delivering new use cases and scenarios in 5G?," NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, 2016, pp. 1279-1284. doi: 10.1109/NOMS.2016.7503003* in Annex A.3

- *L. Xu, H. Assem, I.G.B. Yahia, T.S. Buda, A. Martin et al., "CogNet: A network management architecture featuring cognitive capabilities," 2016 European Conference on Networks and Communications (EuCNC), Athens, 2016, pp. 325-329. doi: 10.1109/EuCNC.2016.7561056* in Annex A.5

- M. Tolan, J. Tynan, A. Martin, F. Mogollon, "Dynamic Policy Based Actuation for Autonomic Management of Telecoms Networks," IEEE European Conference on Networks and Communications (EuCNC), 2017. in Annex A.2

1.5 Document structure

This thesis has been structured as follows. Part I presents an introduction to the research scope, focusing on the motivation for the research, the main objectives, the hypothesis,

the methodology and the main contributions of the Ph.D. work.

Part II overviews literature related to mobile cloud computing, self-organising networks and bitrate adaptation solutions, including client-side and network-supporting strategies, for the application domain of media delivery.

In Part III the research results are described in four main chapters:

- Chapter 3 describes the contributions to distribute media analysis tasks to an elastic cloud of spontaneous computing resources (Contribution 1.4.1). The goal is to enable more engaging contents in the media service.
- Chapter 4 describes the contributions to create a client-side mechanism that distributes media content in a steady and efficient manner in radio-links with dense client cells (Contribution 1.4.2). The goal is to avoid unfair QoE amongst the media players.
- Chapter 5 describes the contributions to create a network edge solution that distributes media content in a reliable and efficient manner in radio-links with dense client cells (Contribution 1.4.3). The goal is to shield the media service from outages and degradations and unfair QoE amongst the media players.
- Chapter 6 describes the contributions to create a self-organising network that automatically scales to distribute forecast traffic demands (Contribution 1.4.4). The goal is to satisfy operational QoE for the incoming media service demands.

In Part IV the main conclusions of the research can be found, including a discussion that enables future work.

Finally, Part V provides other publications of the author and his Curriculum Vitae as an appendix, while Part VI contains the bibliography.

Part II

State of the Art

CHAPTER

2

Related Work

2.1 Overview

As explained in Section 1, media delivery driven to fair, enhanced and enforced QoE in 5G environments can be addressed from four different tiers. The research activities in this thesis have been compiled around them:

1. the media service provider and its capacity to create a cloud of resources, detailed in section 3.2.2;

2. the media player to avoid eager behaviours and to get a steady and fair radio link utilization, detailed in section 4.2.2;

3. the network edge to exploit radio statistics to shield itself against service degradation and outages, detailed in section 5.2.2;

4. and the network core to arrange the necessary amount of resources and appropriate setup for the incoming traffic demand, detailed in section 6.2.2.

These four-tier mechanisms enable media services and network operators to engage, balance and ensure the quality of experience for 5G mobile networks. They exploit novel media delivery techniques in the four research areas as depicted in Figure 2.1.

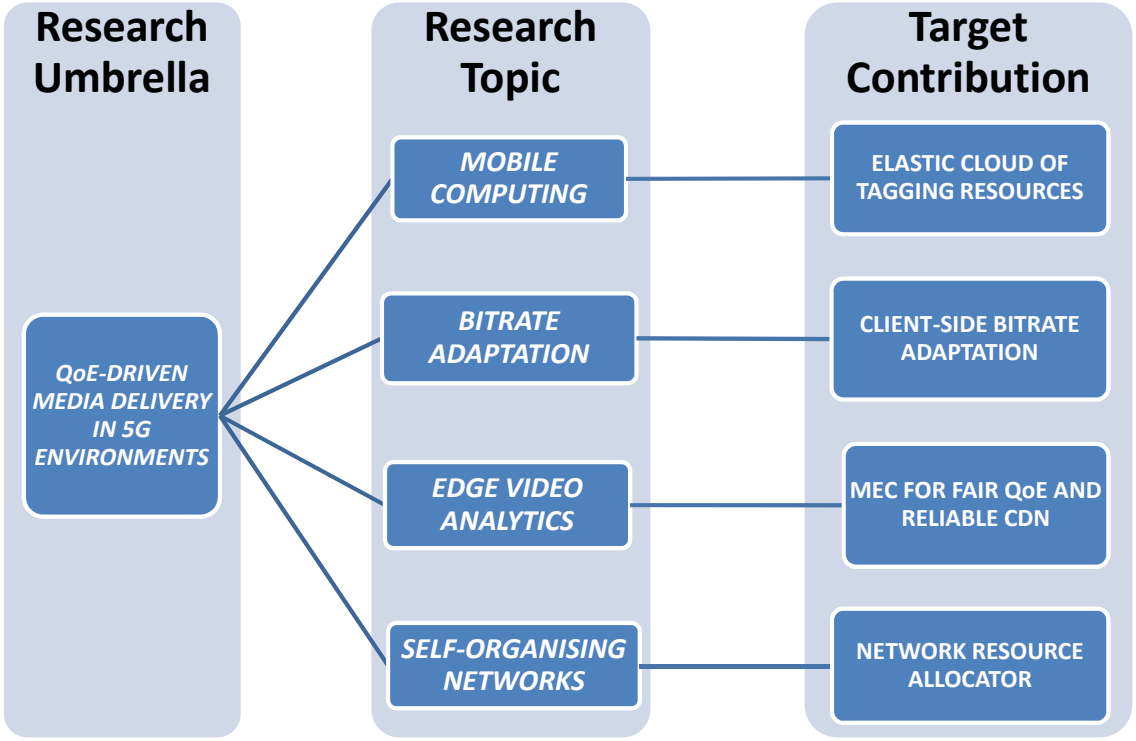


Figure 2.1: Main areas targeted by the research.

This chapter gathers the state of the art of mobile computing, bitrate adaptation, edge video analytics and self-organising networks, including activities, technology initiatives, market solutions and standardization groups which provide the basis of the research areas compiled in this document.

In order to provide a more holistic perspective the Table 2.1 comprises all the related work under the classification criteria.

2.2 Mobile Computing

The variety, volume and velocity of media contents being uploaded to media services have need of continuous and deeper tagging analysis. The goal is to enhance the content visibility and match user preferences to engage audience. At the same time, the mobile devices capabilities, in terms of CPU and GPU processing accompanied by empowered web stacks to leverage the full hardware capabilities, place mobile computing at the focus of distributed and massive computing infrastructures. The combination of

heavy processing needs and the availability of resources that can be unified to produce a synergetic condition favourable for media services.

Nowadays, the high potential of the abundant and frequently idle client hardware boosts the opportunistic and delay-tolerant [Conti and Kumar10] use of client resources on the grid. In this volunteer computing, SETI@home is the most popular example. This [of California99] approach has been the pioneer of big data grid infrastructures benefiting from the Internet-connected computers of volunteers.

Mobile Cloud Computing (MCC) [Huang et al.13] includes mobile devices as clients of the virtualised services, usually following the classical client-server asymmetric model, which involves a one-way communication direction produced by requests from mobile clients to cloud services. Nevertheless, symmetric MCC models have also been proposed [Neumann et al.11], where a crowd of mobile devices populate a cloud offloading the tasks to be performed by the service infrastructure.

The main drawbacks of these solutions are based on the heterogeneous computing on a variety of modern CPUs, GPUs, DSPs, and other microprocessor designs. The trend towards heterogeneous computing and highly parallel architectures has created a strong need for software development infrastructure in the form of parallel programming languages and subroutine libraries supporting heterogeneous computing on hardware platforms produced by multiple vendors [Stone et al.10].

The rapidly increasing use of the Web as a software platform [Anttonen et al.11] with truly interactive applications is boosted by emerging standards such as HTML5 and WebGL. They are removing limitations and transforming the Web into a real application platform middleware to address the interoperability problem.

The cross-entry point is bridged by WebGL and WebCL. WebGL allows communication between JavaScript applications and the OpenGL software libraries, which access the host's graphics processor. Thereby, it enables use of the hardware's full capabilities not only to perform advanced 3D objects and effects rendering but also for general purpose algorithms, such as image processing. WebCL is designed to enable Web applications with high performance and general purpose parallel processing on multi-core/many-core platforms with heterogeneous processing elements. It provides ease of development, application portability, platform independence, and efficient access through a standards-compliant solution [Jeon et al.12]. Thus, WebGL excels in

graphics applications while WebCL fares better when more flexibility is required in execution platform selection, load balancing, data formats, control flow, and memory access patterns [Aho et al.12].

Once the technology stack is able to grant the architectural blocks to distribute computing tasks, privacy must be analyzed. Social acquaintance is a valid policy in order to decentralize privacy, when social media processing is decentralized [Mohaisen et al.14]. Recent blockchain technology has demonstrated that trusted and auditable computing is possible using a decentralized network of peers accompanied by a public ledger [Zyskind et al.15]. Emerging smart contract systems over decentralized systems allow mutually distrustful parties to transact safely without trusted third parties [Kosba et al.16]. This report [Spectrum17b] provides a decision tree to replace a traditional database with a blockchain technology. Ethereum [Ethereum17] is the most representative technology for blockchains. It uses transactions that are miniprograms, called smart contracts, which can be written with an unlimited amount of complexity. Ethereum is utilized to build a decentralized platform that runs smart contracts, applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third-party interference. Miners can run more complex programs, like the software for a social media network [Spectrum17a]. In practice, this means that anyone can embed a software program into a transaction and know that it will remain there, unaltered and accessible for the life span of the blockchain.

Concerning the online digital advertising industry, a key source of income for media services, publishers face falling revenue, users feel increasingly violated, and advertisers' ability to assess effectiveness is diminished. The solution is a decentralized, transparent digital ad exchange based on blockchains. Here, a ledger system that measures user attention to reward publishers accordingly. Basic Attention Token (BAT) [BAT17] is a token based on user attention, which simply means a person's focused mental engagement. The BAT can be exchanged between publishers, advertisers, and users. So, BAT is a technology for blockhain-based digital advertising. It all happens on the Ethereum blockchain.

Recently, the opportunistic cloud of unaware computing resources gained relevance and was linked to crypto-currencies and blockchains. According to Kaspersky reports [Lab17], the number of attacked users in the first eight months of 2017 reached 1.65 million. The media services and content websites, such as CBS, Showtime or The Pirate

Bay contained JavaScript that secretly commandeered viewers' web browsers to mine cryptocurrency [MIT17].

2.3 Bitrate Adaptation

The bitrate adaptation algorithm inside a HAS media player allows the client to independently choose its playback quality. From the Mobile Network Operator (MNO) perspective, multiple bitrate streams are operated by adjusting the play-out rate to stay within the actual network throughput and device capability. Thus, adaptive encoding offers benefits to allow operators to plan the capacity of their delivery networks to match the average, rather than the peak, usage demands. This way, MNOs save considerable Capital Expenditure (CAPEX) maintaining an uninterrupted user experience by means of client-based switching decisions.

However, multiple clients competing for bandwidth across a bottleneck link can cause instability in the selected representation, link under-utilization, and disproportional shares of available bandwidth [Chen et al.16b]. Therefore, recent research in adaptive streaming is focusing on the development of client-side adaptation algorithms. The client monitors some key indicators to make the decision of switching to a representation bitrate that better fits the current state and maximizes the playback quality.

On the one hand, *connection-based* algorithms choose the representation bitrate considering server-client connection status (most common indicators are bandwidth and latency). Here, the *heuristic-based* algorithms take direct measurements and use decision rules based on the observations. These allow the most appropriate level to be dynamically requested, based on the current network conditions in multi-client scenarios [Petrangeli et al.15]. To track quick changes on networking conditions, the algorithm [Liu et al.11] explores step-wise increases and aggressive decreases in the adaptation algorithm in single-user scenarios. Some *heuristic-based* algorithms are Festive (Fair, Efficient, Stable, adaptIVE) [Jiang et al.14], Panda (Probe and Adapt) [Li et al.14b] and Lolypop (Low-Latency Prediction-Based Adaptation) [Miller et al.16]. On the other hand, the *optimization-based* algorithms perform mathematical modelling. They need a big dataset and a long learning time [Claeys et al.14b].

Content-based algorithms characterize the content, using Structural Similarity (SSIM), the human perception of the image, to adapt the representation bitrate accordingly [Chiariotti et al.16]. This *content-based* algorithm suffers from high implementation complexity and large overheads requiring reduced power consumption and prolonged battery life [Chen et al.16a, Zorrilla et al.17].

More complex solutions [Li et al.14c] explore both, the status of the connection-player and the features of the video content. However, the issue related to processing overheads persists.

Whatever the adopted solutions, the aim of each algorithm is to enhance the quality of the playback. A consolidated way to evaluate the QoE is the Mean Opinion Score (MOS), with five incrementing quality levels (from 1 to 5) [ITU]. This type of testing leads to long evaluation times. Therefore, for practical reasons, many objective models for evaluating an estimated MOS (eMOS) have been studied to profile the subjective human perception of the quality.

The work [Vriendt et al.13] investigates the most common models to verify the fit of each model. The models shown are: bitrate model, PSNR or SSIM based model, chunk-MOS based model and quality model. It concludes that the chunk-MOS model is the optimal one. Moreover, the works [Claeys et al.14a, Mok et al.11] conclude a *QL model* which limits the eMOS evaluation to a set of objective metrics from the connection heuristics, such as quality switches, frequency and duration of freezes. These parameters are the key metrics of HAS services. Work from [Claeys et al.14a] concludes that the operational range of the eMOS is [0; 5.84], in contrast to the discrete scale from 1 to 5 of the theoretical MOS [ITU].

Study Group 12 (SG12) of the ITU-T is currently working on the standardization of a new QoE assessment method, known as video Mean Opinion Score (vMOS) [Lentisco et al.17b]. Its goal is to provide a unified and user-centric standard that enables quantification of the quality of video streaming services over different networks, screens or scenarios, without a specific focus on mobile broadcast services.

2.4 Edge Video Analytics

5G will foster the media innovation ecosystem by opening interfaces to adapt the network capabilities to media application needs in real time. Thus, 5G will boost the

efficiency of media services and their businesses.

Specifically, the network must participate in coordinating media players to accurately estimate available bandwidth and shield itself from CDN performance degradation and outages. In this regard, it is important to design a scalable solution which captures metrics with zero latency, processes them and prevents QoE degradation situations in real-time. The solution must be transparent in different levels, from the media delivery protocol perspective, to be universally adopted, and from the networking perspective, to avoid overheads with extra messaging.

Scientific approaches to make data actionable in a coordinated way, with a network centric perspective, often consider SDN-enabled wireless networks [f. Lai et al.15]. Some schemes include in-network proxies [Petrangeli et al.15], a proxy manager and a resource controller at the eNodeB level [Rubin et al.15, Chang et al.15] to provide the clients with target quality suggestions. Other works [Vleeschauwer et al.13, Essaili et al.15] automatically and fairly adapt the video quality to react to congestion and data flow throughput starvation by overwriting client-side decisions.

The relevance of metrics in making decisions to enhance media services is evident. Media services using CDNs can enforce reliability by avoiding overloaded CDNs through the use of content delivery analytics. There are platforms to monitor client experience and benchmark the performance of every CDN and service. Focused on IP Video performance, Cedexis [Cedexis17] and Conviva [Conviva17] platforms sustain networking decisions in a centralized manner, via a cloud system highly coupled with the service provider and the player which has an agent to gather continuous quality telemetry, adding signalling overheads.

Operators can expose their RAN edge Application Programming Interface (API) to authorized third parties to provide them with radio network information in real-time. Hence, MEC opens the door for authorized third parties, such as content providers (CP), to develop their own applications hosted on the MEC servers. Here, ETSI envisions MEC as a system to exploit RAN awareness video analytics, and therefore MEC improves the user experience by managing the media delivery closer to video viewing screens.

Moreover, dynamically switching in real time from one CDN to another, fitting to a well-balanced trade-off between QoE and costs, is a relevant scenario. This can become a reality by using content delivery analytics from the MEC components, which tend to be proprietary solutions, creating open real-time analytic data of throughput

and ping speeds to caches to measure the speed and availability of different delivery paths over the Internet. To this end, a MEC system can collect and process geo-based analytic data in real-time to select a more reliable CDN by measuring the performance and availability of different delivery paths over the Internet.

MEC paradigm is the core of the systems in improving MPEG-DASH performance [Li et al.16]. This approach brings new features, such as close to zero delays and awareness of the radio status. Here, an HTTP proxy removes or adds back representations from the media presentation description (MPD) manifest according to Channel Quality Indicators (CQI) reports avoiding signalling overheads. Following the MEC vision, a hybrid edge and client adaption solution for HAS media services is applied to cellular links with shared bandwidth [Yan et al.17]. This work goes a step further by considering the cumulative viewing experience to tune the QoE continuum and fairness model, and two theoretical moving patterns. An alternative approach [Chen and Liu16] targets continuity of the viewing experience and efficient resource allocation. This hybrid MEC and client-side mechanism, orchestrates time slots to make HTTP requests, different for each media player and serving rates. A further MEC component prioritizes or drops different HTTP transactions tailored to H.264 Scalable Video Coding (SVC) streams [Fajardo et al.15]. So this work employs L2 (CQI reports) and L7 (H.264/SVC hierarchical dependencies) to achieve QoE-driven fair scheduling of radio resources.

The integration of MEC systems into SDR technologies is explored in [Wang et al.17] by employing an OpenAirInterface (OAI) SDR system to integrate on the eNodeB a MAC packet scheduler (L2).

2.5 Self-organising Networks

5G will deliver a 1000-fold gain in capacity per geographical area, a 10 to 100 times scale in connected devices, a 10 to 100-fold increase in the individual end-user data rate experience that is capable of extremely low end-to-end latency, under 1ms, promising 10 times lower energy consumption, and granting ubiquitous access even in low density remote areas. To this end, 5G must deal with fast, heterogeneous, multi-tier networks, which are also dynamic in nature.

The main advances of 5G focus on two directions [5GPPP16]. First, the radio access network (RAN), by means of additional spectrum bands and higher spectral efficiency,

to achieve higher capacity [Chávez-Santiago et al.15]. Second, the SDN solutions, to empower the core and the edge of the network [Nguyen et al.16].

SDN [Foundation12] and NFV [Foundation13] are two key enabler technologies of 5G. These technologies catalyse the transformation of operative switching and forwarding into programmable and configurable functions. SDN and NFV technologies lead to an agile network infrastructure enabling decisions about how to forward traffic [Kim and Feamster13].

SDN and NFV technologies-based solutions are proliferating and explored as common 5G base technologies as the standardization phase progresses. By combining SDN and NFV concepts the network management systems employ interfaces, virtualization frameworks and solutions such as OpenFlow [Foundation17e], OpenVSwitch [Foundation17d], OpenStack [OpenStack17], OpenDaylight [Foundation17c], OSM [TID17], OpNFV [SDxCentral17] to implement the ETSI 5G stack.

Machine learning algorithms applied to Self-Organising Networks (SON) attempt to address a fully autonomous and flexible network with robust and intelligent mechanisms. The explored techniques applied to the SON field are summarized in [Klaine et al.17]. These solutions span self-configuration, self-optimization and self-healing functions to add the required intelligence. Self-configuration deals with operational parameters. Self-optimization can be applied to backhaul optimization, including caching, load balancing, capacity and energy efficiency, and antenna parameter optimization, applied to interference management and handover. Whereas self-healing targets faults and failures. Figure 2.2 shows the most common algorithms in the literature of cellular SON.

The network manager and telco operator needs tools to improve QoS in a 5G environment. More specifically, tools to support the selection of the most efficient topology and setup to deliver the best QoS at the best cost. The SDN and NFV paradigms boost network adaptability and provide elasticity functions to make networks easily scalable. However, this brings up the need for mechanisms to manage the network due to the increase of the network complexity. Here, machine learning is applicable on self-managing networks by means of its ability to learn from historical data, make predictions, dynamically adapt to new situations while learning from new data [Mohri et al.12] and take decisions. Going further, in the network management area, machine learning could forecast resource demand and react appropriately. Combining machine learning, SDN

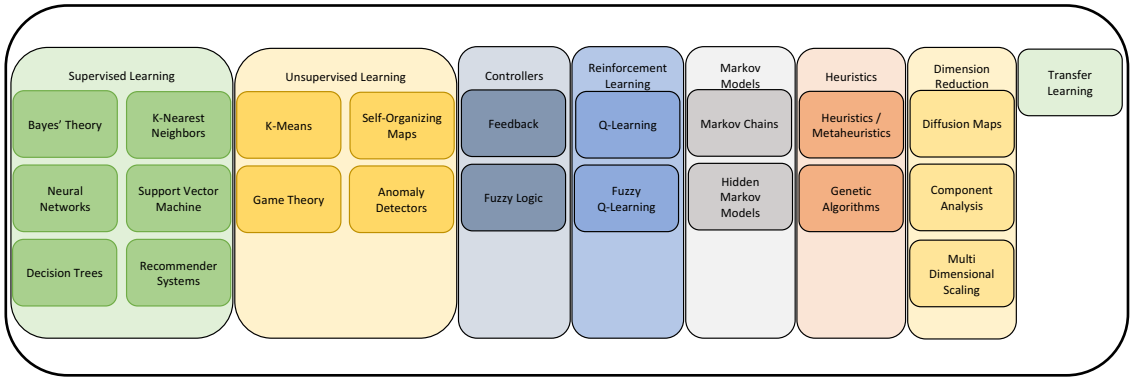


Figure 2.2: Overview of most common machine learning algorithms in the literature of cellular SON. Source: Survey by Klaine et al. [Klaine et al.17, Fig. 2].

and NFV concepts, a centralized view of the network can be exploited to automatically identify networking issues. Thus, enabling the controller to change the network topology by instantiating or removing VNFs to forward the incoming traffic in an efficient way, and removing the unused parts of a network to release these resources [Ismail et al.13].

With regard to media networking, the main volume of traffic delivered by current and next generation networks, in [Caglar and Gokhale14] the author optimizes the resource utilization and achieves a target QoS by finding correlations in the historical data and predicts future resource usage. However, the system is not automated.

On top of SDN technologies, it is possible to develop systems to autonomously improve network agility and flexibility to efficiently support the evolving demands of users. Machine learning technologies must be considered to meet the network resources allocation that dynamically meets changing demands, while achieving SLA network operation enforcement, and to keep the networking operation inside business ranges [Buda et al.16].

Media services are consuming more and more network resources. In order to optimise the use of the network assets, the network must allocate them on-demand as required by the delivered media service. An SDN controller could change the network topology through instantiating or removing Virtual Network Functions (VNF) to forward the incoming traffic in an efficient way, removing the unused parts of a network to release these resources [Ismail et al.13]. Failure to estimate the resource utilization of applications running on top of a virtualized infrastructure might lead to a severe performance degradation of those applications. So far, studies have shown that servers in

many existing data centres are often severely underutilized due to overprovisioning to avoid degradations [Armbrust et al.09]. This overloading can also have an impact on such degradations. In [Xiao et al.13], authors describe the trade-off between overload avoidance and cost-effective computing. This can be avoided by gathering insights into the source of performance degradations, detecting and anticipating these in advance through machine learning, and applying the corrective measures to avoid them.

Table 2.1: Related Work on QoE and media delivery.

Contribution	Research Field	Topic	Category	References
Elastic Cloud of Tagging Resources	**Mobile Computing**	Computing Models	Grid	[Foster et al.01, Conti and Kumar10, of California99, Ahuja and Myers06]
			Cloud	[Neumann et al.11, Ahuja and Myers06, Huang et al.13, Armbrust et al.10, Liu13, Williams et al.13, Pang et al.15]
		Interoperability	Stack	[Xamarin17, PhoneGap17, Foundation17a, Stone et al.10, Khronos12, Moreno-Vozmediano et al.13, Project17]
			HTML	[Anttonen et al.11, Chandra et al.13]
		Tasks Distribution	Best-effort	[Catak and Balaban13, Lin et al.10, Foundation05, De Francisci Morales et al.11, of Virginia Computer Graphics Lab12, Sweeney et al.11, Gum12, Kalooga12]
			Foreground sensitive	[Chandra et al.13, Chen et al.13, Zhang et al.17, Zhang et al.16, Zorrilla et al.15a]
		Parallel Processing	Stack	[Yang et al.11, Nvidia07, OpenMP13, Mathematics and Science96, Jarp et al.12]
			Web	[Jeon et al.12, Garrett et al.05, Fette and Melnikov11, Langhans et al.13, Cushing et al.13, Tilkov and Vinoski10, Aho et al.12, MacWilliam and Cecka13]
		Data Structures		[Tan et al.13, Han et al.11]
		Trusted Distribution	Social	[Mohaisen et al.14]
			Blockchain	[Zyskind et al.15, Kosba et al.16]
Client-side Bitrate Adaptation	**Bitrate Adaptation**	Heuristic-based	Client	[Petrangeli et al.15, Liu et al.11, Claeys et al.14b]
			Dense client RAN	[Jiang et al.14, Li et al.14b]
		Optimization-based	Client	[Miller et al.16, Chiariotti et al.16, Li et al.14c, Rainer and Timmerer14]
			Dense client RAN	[Chen and Liu16, Seufert et al.15, Chiariotti et al.16, Li et al.14c, Toni et al.15]
		QoE	MOS	[Vriendt et al.13, Chen et al.15, Orosz et al.14]
			Estimated MOS	[Claeys et al.14a, Mok et al.11, Lentisco et al.17a]
MEC for fair QoE and reliable CDN	**Edge Video Analytics**	Heuristic-based	Network	[Cedexis17, f. Lai et al.15, Petrangeli et al.15]
			Dense client RAN	[Rubin et al.15, Chang et al.15, Fajardo et al.15, Wang et al.17]
		Optimization-based	Network	[Vleeschauwer et al.13, Essaili et al.15]
			Dense client RAN	[Li et al.16, Yan et al.17, Kourtis et al.17]
		Multi-CDN switching		[Adhikari et al.12]
Network Resource Allocator	**Self-Organised Networks**	Machine Learning for self-	configuration	[Mohri et al.12, Klaine et al.17]
			optimization	[Ismail et al.13, Bizanis and Kuipers16, Szabo et al.15, Klaine et al.17, Wainio and Seppänen16, Sandhir and Mitchell08, Edwards et al.97]
			healing	[Klaine et al.17]
		OPEX & CAPEX-driven	RAN	[Chávez-Santiago et al.15]
			Core&Backhaul	[Kim and Feamster13, Nguyen et al.16, Sun et al.15, Hernandez-Valencia et al.15, Chávez-Santiago et al.15]
		QoS-driven networks	RAN	[Li et al.16, Yan et al.17, Chen and Liu16]
			Core&Backhaul	[Liu et al.16a, Rong et al.16, Xu et al.13, Serrano et al.16, Emeakaroha et al.10, Caglar and Gokhale14, Bendriss et al.17, Farzaneh and Moghaddam08]
		Content Centric Networks		[Liu and Wei16, Rhaiem et al.15, Park et al.14, Awiphan et al.13]

Part III

Research Results

CHAPTER

3

Elastic Cloud of Tagging Resources

3.1 Context

Most of the posts of social media networks are photos or videos. With huge volumes of contents uploaded to social networks, browse and search is traditionally based on text-driven technology. Hence, the captions and tags have a key role to discover the right contents.

Social services aim to engage audience, eager for contents, by boosting media relevance. To this end, a more precise automatic tagging enables better matching of user interests. Image analysis helps to better describe contents aiming better search results, but they also involve computing-intensive functions. Therefore, the processing requirements grow substantially when all the media items comprising the social network database are analysed. Thus, it is needed to build a scalable system to understand content.

At the same time, a vast number of devices are concurrently consuming media services. These client devices have often idle computing resources while playing media content in foreground. The long duration of sessions, when consuming video media services, decreases the volatility of connected devices. Hence, the sessions are long

and steady. This context is perfect to dispatch lightweight tasks to connected devices while any impact on the foreground experience is avoided and the communication overhead is reduced. So, client devices can contribute with part of their resources to perform atomic tasks, such as creating automatic tagging by image analysis mechanisms to enhance the media experience and save cloud resources.

An elastic cloud of resources for delay-tolerant media tagging could take benefit of massive client connections in 5G networks. To create such a system, some aspects must be overcome. First, the system must deal with a farm of spontaneous workers, available only while consuming media contents. Then, a mechanism to perform asynchronous tasks dispatching is needed. Finally, to avoid any impact on the foreground experience it is essential to match the task computing demand with the hardware processing capacity.

Drawing inspiration from volunteer computing initiatives for big data, Section 3.2 proposes a solution where thin devices can complement a cloud service for delay-tolerant computing tasks. The proposed system, named SaW, is a pure Web-based distributed solution which exploits both CPU and GPU resources of the client devices in an interoperable manner. To this end, a proof-of-concept implementation of SaW using WebGL and WebCL technologies is provided, to evaluate the SaW approach, supported by experimental results and an analysis of the performance based on a cost model for hardware-accelerated processing distribution.

3.2 SaW: Video Analysis in Social Media with Web-based Mobile Grid Computing

- **Title:** SaW: Video Analysis in Social Media with Web-based Mobile Grid Computing
- **Authors:** Mikel Zorrilla, Julián Flórez, Alberto Lafuente, Angel Martin, Jon Montalbán, Igor G. Olaizola and Iñigo Tamayo
- **Journal:** Transactions on Mobile Computing
- **Publisher:** IEEE
- **Year:** 2017
- **DOI:** `http://dx.doi.org/10.1109/TMC.2017.2766623`

Abstract. The burgeoning capabilities of Web browsers to exploit full-featured devices can turn the huge pool of social connected users into a powerful network of processing assets. HTML5 and JavaScript stacks support the deployment of social client-side processing infrastructure, while WebGL and WebCL fill the gap to gain full GPU and multi-CPU performance. Mobile Grid and Mobile Cloud Computing solutions leverage smart devices to relieve the processing tasks to be performed by the service infrastructure. Motivated to gain cost-efficiency, a social network service provider can outsource the video analysis to elements of a mobile grid as an infrastructure to complement an elastic cloud service. As long as users access to videos, batch image analysis tasks are dispatched from the server, executed in the background of the client-side hardware, and finally, results are consolidated by the server. This paper presents SaW (Social at Work) to provide a pure Web-based solution as a mobile grid to complement a cloud media service for image analysis on videos.

Keywords: Distributed computing, image analysis, multimedia databases, multimedia systems, social media, web-based architecture

3.2.1 Introduction

The social media paradigm has led to a significant rise in the volume of user generated content managed by social networks with millions of users accessing services, each of them often using multiple devices at the same time. Service providers aim to engage audience, eager for contents, by boosting the media relevance. To this end, a deeper automatic tagging enables better matching of user interests with the content database and reveals underlying connections between items, such as applying face detection mechanisms or content-based indexing to find related videos. Image analysis algorithms empower automatic retrieval of salience features but they also involve computing-intensive functions. Therefore, the processing requirements grow substantially when all the media items comprising the social network database are analysed. Here, on the one hand big data challenges arise when social services have continuously increasing databases, while on the other hand more and more processing resources are required to analyse all the content.

Grid and Cloud technologies provide High Performance Computing systems that aim to satisfy these requirements. However, as pointed in [Neumann et al.11], other

under-explored alternatives could enhance the trade-off between infrastructure cost, elapsed time and energy saving. It would depend on the number of available processing nodes, the inherent characteristics of the tasks to be performed in parallel and the data volume.

To deal with the aforementioned context, this paper introduces a new concept of Social at Work: SaW. It aims to complement a Web-based social media service with all the client devices, mostly mobiles, that usually have underexploited resources while accessing the service. SaW proposes a Mobile as an Infrastructure Provider (MaaIP) model, going beyond the Infrastructure as a Service (IaaS) model, and creating a system related to Mobile Grid Computing [Ahuja and Myers06] concept with the available CPU and GPU resources of the different client devices to complement a virtualised cloud server, which provides the social media service.

Inspired by the Mobile Grid Computing and the Mobile Cloud Computing (MCC) [Huang et al.13] research fields over a social network mainly based on video content, SaW aims to bring together the huge pool of users permanently connected to media services in social networks and the ever increasing processing capabilities of most of their devices. As a consequence, service providers will embrace the community assets building a device centric grid to improve the social service by means of media analysis. Thus, SaW concept enables service provider to recruit spare CPU/GPU cycles of client devices into an active gear of the social platform, saving cloud resources to the server when the connected clients can perform those tasks.

In order to achieve a SaW system some issues must be addressed, such as turning a Web client into a runtime application framework. The gap between native applications and Web-apps is shrinking by empowered Web engines. Moreover, Web stack can deploy a communication layer to distribute background analysis tasks. The remaining aspect is to manage a volume of spontaneous workers, tracking the status of the tasks, while dealing with the uncertainty of resource availability and heterogeneous processing capabilities.

First, the current device ecosystem is highly heterogeneous, with different operating systems and programming languages, resulting in complex software cross-platform development. In this context, SaW proposes a pure Web-based approach since Web technologies overcome the interoperability barriers. HTML5 is continuously empowering the browser turning the Web into a real application platform middleware able to

access hardware resources of the appliances through JavaScript [Anttonen et al.11]. Additionally, HTML5 introduced a number of features to enable offline Web applications, such the application cache and local storage. Compared to other mobile cross-platform native development frameworks such as Xamarin [Xamarin17], Web Apps based on Web-stack are bridging the gap between a typical mobile Web experience and a dedicated custom app on any device. At the same time, they inherit same-origin and permission security policies of the browser. In terms of development and updates, one codebase can serve many platforms, as long as it does responsive design, while they are versionless and backwards compatible. This means saving on developments costs. Going further, the frameworks to develop hybrid applications built with Web technologies and packaged as native apps, such as PhoneGap [PhoneGap17] or Apache Cordova [Foundation17a] could benefit from the SaW solution, running specific native features where HTML5 is not able yet. Anyway, most social media services do not need specific features unavailable from the browser.

Second, in order to exploit native GPU and multi-CPU potential of a device, WebGL and WebCL bindings to OpenGL and OpenCL run hardware-accelerated, parallel and cross-platform programs. So, they endow Web applications with parallel computing capabilities, accelerating Web applications for intensive image processing [Jeon et al.12].

Then, Ajax (Asynchronous JavaScript and XML) [Garrett et al.05] and Websockets [Fette and Melnikov11] are employed as a vehicle for establishing and maintaining mainstream communication between server and clients, transforming the classical synchronous request-response model into a full bidirectional one. This feature enables the server to send asynchronously updates to the client-side browser and to deliver background data.

Finally, the possibility to perform image processing tasks in parallel, such as feature extraction, segmentation, clustering and classification, eases to leap scalability. Due to the video stream nature, composed by individual frames, they can be easily split into independent tasks ready to be distributed. Beyond, the intrinsic presence of key frames in video coding, makes easier navigation and selection of representative images. Servers can dispatch the tasks to users' devices where they are run in the background. These background Web browser applications must balance the mechanism to leverage all available computing resources while provide the best possible user experience.

Thus, the validation of the approach can be explained in terms of the net benefit obtained in the server by delegating part of the tasks. The equation includes two relevant keys, which are based on certain parameters that are dependent on the technological state-of-the-art and the users' social media consumption habits: (1) the amount of work that can be distributed to the client devices, which depends on the number of available clients, their capabilities, and the fraction of resources they can dedicate to background tasks without disturbing the user experience, and (2) the extra work created in the server to manage the task scheduling, which should be residual.

As a summary, this paper presents SaW, a pure Web-based, interoperable distributed solution, which is deployed on top of the user appliances of a social media community, including the hardware-accelerated features for suitable devices. Thereby, the service provider leverages the huge processing ability of the social community. This allows the service provider to perform independent background hardware-accelerated image processing tasks, which are embedded to the different social media services accessed by the users. Operational thresholds where this approach is able to compete with traditional computing alternatives need to be defined. In order to do so, existing and validated cost models for parallel computing are shaped to the parameters of the SaW design, such as heterogeneity of devices or their sporadic availability.

3.2.1.1 **Contributions**

In this work, we introduce the concept of Mobile as an Infrastructure Provider (MaaIP), aimed to extend the resources of a cloud service by using mobile clients as a grid complement.

The MaaIP concept is demonstrated through the design of SaW, a system for the analysis of video content collection in media driven social services. In this context, cost-efficiency benefits can be found for SaW since the following favourable conditions are met: the service provider needs to perform a large volume of atomic tasks and there is a crowd of potentially under-exploited devices with a continuous session.

The work presented in this paper extends a previous evaluation model to include GPU usage. Finally, we provide a proof-of-concept implementation of SaW using WebGL and WebCL technologies.

3.2.1.2 Paper structure

This paper starts with the related work in Section 3.2.2, exploring the different Internet-based computing models and analysing their existing mechanisms for the interoperability, task distribution, support for parallel processing and different data structures. Section 3.2.3 presents the main contribution of the paper with the definition of the SaW concept. It describes the contributions of SaW to the aforementioned related work, presents a suitable scenario for SaW on a social media service, defines the design objectives of the SaW architecture, and presents a pure Web-based architectural design.

It follows with the evaluation of the SaW approach in Section 3.2.4. Subsection 3.2.4.1 describes the experimental results in terms of scalability over a proof-of-concept implementation of SaW using WebGL and WebCL, subsection 3.2.4.2 presents a performance analysis, extending the already published model in [Zorrilla et al.13] in terms of GPU, and in subsection 3.2.4.3 some remarks regarding the validation of the SaW hypothesis are presented. Finally, Section 3.2.5 presents the conclusions.

3.2.2 Related work

This section presents the related work, providing a definition of the Internet-based computing models and focusing on the different topics addressed by distributed computing: the interoperability, the task distribution managing, the parallel processing capabilities and the different data structures.

3.2.2.1 Computing Models

This section describes the main involved concepts in terms of Internet-based computing, where shared resources, data and information are provided to computers to reach a common goal.

Grid Computing

Grid Computing [Foster et al.01] has been an important paradigm in distributed systems for the last two decades. Basically, a grid is a network system where computing tasks are distributed to use non-dedicated computing resources, which may include servers or client computers. The high potential of the nowadays abundant and frequently idle client hardware boosts the opportunistic and delay-tolerant [Conti and Kumar10] use of client resources in the grid. In this volunteer computing

SETI@home is the most popular example. SETI@home [of California99] approach has been the pioneer of big data grid infrastructures taking benefit of Internet-connected computers of volunteers. SETI@home has spread the collaborative network model to other unselfish research in areas such as astronomy, climate, astrophysics, mathematics, genetics, molecular biology and cryptography where volunteers and donors share the computing time from personal devices.

Mobile Grid Computing

Grid Computing is characterised by the heterogeneity of the resources in both amount and nature, by the sporadic availability, churn and unreliability of the devices, and by their anonymity and lack of trust. These issues are more relevant in Mobile Grid Computing (MGC) [Ahuja and Myers06], where computing resources include mobile devices with wireless communications, and therefore prone to disconnections and other eventualities.

Cloud Computing

More recently, Cloud Computing [Armbrust et al.10], a new paradigm of distributed computing where virtualised computing resources are provided on-demand, has experienced a dramatic growth. Nowadays the cloud is a cost-saving opportunity for many enterprises [Liu13] and many cloud vendors [Williams et al.13]. Amazon is a popular cloud service provider with solutions like Amazon Simple Storage Service S3 and the Elastic Cloud Computing EC2 as an interface to them. Eucalyptus [Nurmi et al.09] is an open source cloud implementation on top of Amazon EC2.

Being not tied to a specific hardware model, Cloud Computing enables an improved time-to-market for services achieving: a reduced infrastructure deployment time thanks to an increased service availability and reliability; rapid creation of additional service instances; and cloud interoperability, which lets professionals deploy a service on multiple clouds. Thus, cloud computing provides theoretically unlimited scalability and optimised service performance.

Since the costs of cloud solutions are a key factor, new models are required to fit better with specific applications, infrastructure environments and business contexts. These new models are classified in three, according to the different virtualisation layers: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). An example of how client devices can be integrated into cloud services is

STACEE [Neumann et al.11], which proposes a peer-to-peer (P2P) cloud storage where different devices can contribute with storage to the cloud.

Mobile Cloud Computing

In Mobile Cloud Computing (MCC) [Huang et al.13] computing resources include mobile devices as clients of the virtualised services, usually following the classical client-server asymmetric model, which involves a one way communication direction produced by requests from mobile clients to cloud services. MCC addresses the resource scarcity problem of mobile devices by offloading computation and/or data from mobile devices into the cloud, such as in [Zhang et al.17], which provides an energy-efficient offloading in mobile cloud for video-based applications. In the converging progress of mobile computing and cloud computing, the cloudlet is an important complement to the client-cloud hierarchy [Pang et al.15]. Since the main purpose of cloudlets is to enable resource-intensive and interactive mobile applications by providing powerful computing resources to mobile devices with lower latency, it can be considered as an extension of cloud computing infrastructure. This vision is sustained by the limited resources on mobile devices. Nevertheless symmetric MCC models have been also proposed [Neumann et al.11], where a crowd of mobile devices populate a cloud offloading the tasks to be performed by the service infrastructure. This pushes a user-centric strategy to MCC solution shifting to new models: Mobile as a Service Consumer (MaaSC), Mobile as a Service Provider (MaaSP), and Mobile as a Service Broker (MaaSB).

MaaSC model is inherited from the traditional client-server design where mobile devices are mere service consumers. Here, mobile devices outsource their computation and storage functions onto the cloud. MaaSP switches the role of the device from a service consumer to a service provider. Last but not least, MaaSB can be considered as an extension of MaaSP, where the device gateways other handhelds or sensing nodes. Moreover, the proxy mobile device can also provide security and privacy protections to the data.

MaaSC is the most common MCC service model. Most of the MaaSC solutions, such as CloneCloud, MAUI, ThinkAir, Dropbox, GoogleDrive provide computation task offloading service for mobile devices, keeping the mobile device thin. However, with the recent advances, the features of handheld device are getting closer to regular laptops catalysing new opportunities for MaaSP deployments, like in STACEE.

3.2.2.2 **Interoperability**

The interoperability in heterogeneous networks implies two abstraction levels. On the one hand, it requires the deployment of solutions over specific architectures and operating systems. On the other hand, it also requires the provision of interfaces for remote operation and orchestration over a distributed system.

Despite the wide support of SETI@home, HTCondor or Eucaliptus to different architectures and operating systems, including GNU/Linux, Windows and some of the Mac OS platforms, the main drawbacks of these solutions lay on the heterogeneous computing on a variety of modern CPUs, GPUs, DSPs, and other microprocessor designs. The trend towards heterogeneous computing and highly parallel architectures has created a strong need for software development infrastructure in the form of parallel programming languages and subroutine libraries supporting heterogeneous computing on hardware platforms produced by multiple vendors [Stone et al.10]. In response to this completely new landscape, OpenCL [Khronos12] is a new industry standard adopted by Intel, AMD, Nvidia, Altera, Samsung, Qualcomm and ARM holdings.

Service interoperability between different cloud providers requires standard interfaces and formats for managing virtual appliances. Nowadays, due to the lack of standard way for cloud managing, each provider publishes its own APIs. In order to establish a universal connection, some proposals have been released [Moreno-Vozmediano et al.13]:

- OCCI [OGF17] defines a protocol and API specification for remotely managing of cloud computing infrastructures,
- CIMI [DMTF17] targets to set an interface and a logical model for managing resources within a cloud, and
- CDMI [SNIA17] establishes an interface for manipulating data elements from the cloud.

OpenNebula [Project17] and Eucalyptus have made important contributions in the deployment of interoperable cloud platforms. OpenNebula implements the OCCI and CDMI specifications to enable interoperability among heterogeneous cloud platforms, whereas Eucalyptus incorporates different well-known interfaces using Amazon Web Services (AWS) [Amazon17] as a de facto standard.

The rapidly increasing use of the Web as a software platform [Anttonen et al.11] with truly interactive applications is boosted by emerging standards such as HTML5 and WebGL that are removing limitations, and transforming the Web into a real application platform middleware to address the interoperability problem. HTML5 applications can be packed for the different execution environments providing interoperability with minor changes through independent OSs. That is why HTML5 is being strongly promoted by the standardisation bodies and a sector of the market to achieve a HTML5 marketplace instead of the available proprietary ones, such as Android Market, iOS App Store, etc. All the previously described technologies put aside new breakthroughs that turn the Web into a real interoperable application framework over the heterogeneous mobile platforms.

In this line, the ComputePool component of the Nebula cloud provides computation resources through a set of volunteer compute nodes [Chandra et al.13]. Compute nodes within a ComputePool are scheduled by a ComputePool master that coordinates their execution. The task is executed on a compute node inside a Google Chrome Web browser-based native client sandbox. Thus it provides a secure way to access local user device computational resources inheriting Web security policies to avoid compromising users' local data.

3.2.2.3 **Task Distribution**

In our application area, social media analysis, the batch processing to be executed can be easily split into independent tasks ready to be distributed. This way, servers can dispatch the work to different processing nodes.

Focusing on generic purpose massively collaborative computation with Web technologies, MapReduce [Catak and Balaban13] has a noticeable position. It has been employed by Google to generate its search engine's index of the World Wide Web. In [Lin et al.10] another solution is proposed to overcome server-side task dispatching over a set of nodes, based on open source Apache Hadoop [Foundation05] frameworks. The work proposed in [De Francisci Morales et al.11] highlights the ubiquitous nature of the image matching problems analysing some image processing algorithms specifically implemented for MapReduce technology. Another image processing projects hold by this technology are: HIPI [of Virginia Computer Graphics Lab12] [Sweeney et al.11]

that provides an API for performing image processing tasks in a distributed computing environment; and many more [Gum12] [Kalooga12]. Current research goes further aggregating client-side nodes to work together with the server ones. In this direction, JSMapReduce [Langhans et al.13] is an implementation of MapReduce which exploits the computing power available in the computers of the users of a Web platform by giving tasks to the JavaScript engines of any Web browser. JSMapReduce provides simple and unique frontend for Web developers that only have to focus in JavaScript code.

MapReduce defines a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. An alternative to transform the Web browser into a distributed computer middleware [Cushing et al.13] can be also created on top of Node.js [Tilkov and Vinoski10]. It provides more freedom to meet new requirements, to keep full code control and to ease third parties integration.

Nebula [Chandra et al.13], which uses volunteer edge resources for both computation and data storage, assigns tasks based on application-specific computation requirements and data location. Nebula also implements numerous services and optimisations to address these challenges, including location-aware data and computation placement, replication, and recovery. Nebula considers network bandwidth along with resources computation capabilities in the volunteer platform. Consequently, resource management decisions optimise computation time as well as data movement costs. In particular, computational resources can be selected based on their locality and proximity to the input data, whereas data might be staged closer to efficient computational resources. In addition, Nebula implements replication and task re-execution to provide fault tolerance.

Finally, concerning who launches the requests for task distributions, two approaches are possible: a push model where the service delegates a set of tasks over a set of available resources, and a pull model where idle computing nodes request for new jobs to be performed.

3.2.2.4 **Parallel Processing**

The definition of smaller tasks could bring finer granularity easing efficient processing strategies based on parallel execution in some scenarios. Hence, independent job

scheduling can produce significantly better performance. Here, the social media nature brings some computational benefits. First, media can be easily decomposed in independent frames or clips. Second, the possibility to perform tasks in parallel of the multimedia processing algorithms such as segmentation, clustering and classification eases to leap the scalability dimension. Third, the multimedia processing work fits with continuous advances on parallel processing of multimedia data over GPU architectures.

With the emerging hardware acceleration technologies to exploit GPU and multicore architectures, the parallel programming languages and the hardware computing platforms are getting closer. The most representative languages that aim to enable dramatic increases in computing performance by harnessing the power of the GPU are [Yang et al.11]: CUDA [Nvidia07] for NVIDIA devices provides a general purpose scalable parallel programming model for writing highly parallel algorithms; OpenMP [OpenMP13] has established a method and language extension for programming shared-memory parallel computers. OpenMP, combined with MPI [Mathematics and Science96] specification for message passing operations, is currently the de-facto standard for developing high-performance computing applications on distributed memory architecture. The underlying mechanism consists of partitioning loop iterations according to the performance weighting of multi-core nodes in a cluster. Another mainstream options are pthreads, Cilk, Ct/RapidMind/ArBB, TBB and Boost threads [Jarp et al.12]. These solutions remove barriers by providing abstraction layers for thread block managing, shared memory handling and synchronisation scaling.

MaaSP solutions must meet heterogeneity of browser ecosystem (Chrome, Firefox, Opera, Safari, Edge, IE). The SaW system targets all of them being able to exploit underlying hardware. The cross-entry point is bridged by WebGL and WebCL. In essence, WebGL allows communication between JavaScript applications and the OpenGL software libraries, which access the host's graphics processor. Thereby, it enables use of the hardware's full capabilities not only to perform advanced 3D objects and effects rendering but also for general purpose algorithms, such as image processing. WebCL is designed to enable Web applications with high performance and general purpose parallel processing on multi-core/many-core platforms with heterogeneous processing elements. It provides ease of development, application portability, platform independence, and efficient access through a standards-compliant solution [Jeon et al.12]. Thus, WebGL excels in graphics applications while WebCL fares better when more flexibility is

required in execution platform selection, load balancing, data formats, control flow, or memory access patterns [Aho et al.12].

An implementation example is CrowdCL [MacWilliam and Cecka13]. It presents an open source framework for volunteer computing with OpenCL applications on the Web.

3.2.2.5 **Data Structures**

The scale and diversity of big data problems has inspired many innovations in recent years. Different alternatives to Relational Database Management Systems (RDBMS) have emerged to fit different big data applications.

Not only Structured Query Language (NoSQL) systems, are rapidly gaining popularity and market traction overcoming limitations of relational databases [Tan et al.13]. The NoSQL databases were designed to offer high performance, in terms of speed and size, with a trade-off of full ACID (Atomic, Consistent, Isolated, Durable) features [Han et al.11]. These storing systems include commercial solutions such as Amazon DynamoDB, Google BigTable, and Yahoo PNUTS, as well as open source ones such as: Cassandra, used by Twitter, Facebook and some other corporations; HBase, as part of the Hadoop project; and MongoDB. All of them focus on scalability and elasticity on commodity hardware. Such platforms are particularly attractive for applications that perform relatively simple operations (create, read, update, and delete). They combine low-latency features with scaling capabilities to large sizes querying engine schedules and optimizing its execution.

NoSQL data stores offer various forms of data structures such as document, graph, row-column, and key-value pair enabling programmers to model the data closer to the format as used in their application.

3.2.3 **SaW: Social at Work**

Influenced by the underlying concepts and technologies, SaW system deploys an opportunistic and delay-tolerant distributed computing platform queuing media analysis tasks over a set of trusted devices. As said before, cloud services imply a cost-saving opportunity to service providers, but depending the requirements of the service, it could still be highly demanding. This kind of services usually have a huge pool of users

permanently connected to it. Moreover, second screen and multi-device media experiences are becoming very popular [NAPTE14] [Nielsen14b] [Nielsen14a]. In the social media scenarios considered in this paper, users access them usually from mobile devices, which have increasing processing capabilities that are usually under-exploited. This pushes service providers to go deeper in the cost-saving opportunity using the mobiles as an infrastructure, replacing partially cloud resources. Regarding the computing model, SaW extends the cloud computing Infrastructure as a Service (IaaS) concept to the MCC paradigm, coining a new term of Mobile as an Infrastructure Provider (MaaIP) working together with a cloud service. MaaIP uses client mobiles as a grid infrastructure that allows to extend the cloud resources and complement the cloud service. In SaW, requests match a two-way communication pattern, since servers request clients to hire resources from mobile devices while clients access the main service. However, the different scenarios to be performed on top of the SaW system do not require a symmetric model.

To address the heterogeneity of infrastructure, and sharing Nebula design, SaW system does not require to download or install any software in the client-side thanks to a fully Web-browser based execution stack. SaW goes beyond Nebula's ComputeTool performance by emphasising the Web stack that foster hardware-accelerated parallel programming for GPU and multi-CPU over the Web browser.

Regarding task distribution, in order to address the design objectives of elasticity, performance and security, SaW server-side schedules the queued tasks meeting computation requirements and processing availability of the client devices. This asynchronous execution model ensures control to avoid duplicities for a same task, but it does not provide support for intertask communication. Moreover, to exploit parallel processing capabilities in client devices, SaW brings hardware-accelerated performance through the JavaScript engine, WebGL and WebCL, leveraging full GPU and multi-CPU potential.

Finally, related to the data structures, SaW takes advantage of the document-oriented NoSQL technologies for media repositories fitting into one-to-many relationships of a social service.

3.2.3.1 **SaW Use Case**

Service providers aim to engage audience, eager for contents, by boosting the media relevance. Therefore, it is necessary to improve the matching of user interests with

the huge content database, and reveal hidden connections between items through a deeper tagging. In other words, the service is enhanced by improving the media content indexing.

The target scenario of SaW is a Web-based social media content service, such as YouTube [Youtube17] or Vimeo [Vimeo17]. This target scenario brings beneficial features to the SaW system. First, this scenario provides a continuous communication channel, since users are typically consuming video content for some minutes without interruption, with an active application that provides the content through an adequate bandwidth. Users are aware that media-driven services are bandwidth demanding so they will try to select a high speed network or an appropriate coverage of mobile network. On second place, even though the intrinsic bandwidth requirements of the foreground service, the SaW approach introduces a residual bandwidth overhead comparing with the video itself. SaW will add an extra frame with a processing code to the connection, but it will be residual comparing to the data volume of a progressive download or streaming of a video. On third place, users do not usually perform any other task on the device while consuming video content. This often ends to an under-exploited device with still spare computing resources when compared with more demanding applications [Chen et al.13] [Zhang et al.17].

In the SaW context, video streaming services, the background computation is not significant when compared to the foreground service in terms of computing resources devoted to video stream networking, decoding and visualization [Zhang et al.16]. Moreover, from the energy consumption perspective, the background computation on smartphones reveals negligible compared to the energy consumption of displays in multimedia applications [Chen et al.13].

Finally, the use of an additional screen (e.g., a smartphone) accessing related content while consuming the mainstream video on a first screen (e.g., a TV set) [Zorrilla et al.15a] boosts a very favourable scenario for the SaW approach, since a single user provides multiple devices at the same time connected to a single service that could end to more exploitable resources.

3.2.3.2 **SaW Design Objectives**

SaW targets a MaaIP model where a mobile device provides a computing component within a cloud resource system. This concept holds a key driving force moving the pro-

visioning of processing core assets to harvesting huge amounts of available devices. Nevertheless, MaaIP, as an extension of MaaSP and Mobile Grid Computing, opens some challenges such as elasticity, performance, security and privacy, that are design objectives for the SaW architecture proposed in this article.

Elasticity

In terms of service elasticity, it is mandatory to gain cloud ability to automatically scale services and infrastructures for cost reduction when infrastructure and platform sizes are adapted to service demands. This needs of rapid and dynamic provisioning mechanisms to provide efficient service virtualisation. This factor is even more critical in SaW, when mobiles devices come into action as an available infrastructure which is unstable, sporadic, and with specific features such as limited battery autonomy. This means dealing with uncertainty of the resource availability managed by a notification mechanism providing presence awareness and performance information. This aspect turns task independence into a major condition. SaW assures elasticity through Performance Evaluation and Performance Filtering modules (see Section 3.2.3.3). These modules profile the capabilities of the client devices in terms of CPU and GPU, but also regarding other features such as the level of battery. This profile is captured by the elasticity parameter, as described in Section 3.2.4.2.

Performance

In big data a residual inefficiency is multiplied by the dataset dimension with severe impact on the global system. This means that scheduling and dispatching mechanisms must be implemented to orchestrate all the elements keeping efficiency for data transmission overhead related to work delivery. Furthermore, it is important to match processing needs with device capabilities and minimise re-execution of uncompleted tasks. Thus, the Web client must perform a benchmarking test in order to assess the processing capabilities and the hardware assets disposal of the user's device. Section 3.2.4.2 presents a performance modeling that demonstrates the cost-saving of the MaaIP approach.

Security and Privacy keys

Security and privacy are major concerns for cloud infrastructures even when data is hosted on a corporative data center. However, it turns into a severe issue once the data leaves the corporative firewall. The media managed in social networks consists of images, audio and video elements shared with friends. Such information is highly

privacy-sensitive, and malicious attackers may access a target user's obtaining private information. Additional issues comes when dealing with security and privacy of the node provider, the owner of the device. However, to meet both dimensions, content and device owner, a combination of policies should be applied over the data transmission and storage.

SaW takes into consideration the security aspects regarding confidentiality and integrity. Those are assured by the well-known standard mechanisms of authentication, authorisation, encryption and auditory. As mentioned above, in SaW a client has to commit the hiring of its device resources in order to access the social media service. Thus, reciprocity conditions concerning privacy and security should be observed by the registered client and the server.

It is mandatory to verify social identity of the computation node to check its rights and permissions. The use of a centralised mechanism eases handling frequent user access privilege updates (such as invitation or revocation of access rights) in large dynamic systems like social networks. For this purpose, SaW considers three types of media scopes with different set up implications: public, widening the media analysis to any available device; shared with friends, limiting the trusted area to the devices inside the social acquaintance circle; private sharing, constrained to a specific list of computing nodes from the cloud to manipulate data.

Once the trustworthy handshake has been done, the data must be encrypted to prevent man-in-the-middle attacks. SaW deploys a temporal token based solution to limit access permissions and encrypts the data flows, with TLS protocols, for the Web communication layer.

Concerning the security threats, a push design lets SaW to prevent search over the database and DoS attacks. Note that in a pull model, an attacker, a malicious computation node, could get a promiscuous mode by notifying a permanent idle status to retrieve a set of processing tasks and associated data (crawling for later search) or to capture all the queued tasks (turning the uniformly distributed dispatching management into a burst one for a DoS attack). To prevent this behavior, in the push model of SaW, the cloud broker employs the queue of batch jobs to delegate them. This way, it is difficult for the nodes to search a specific data or claim a particular task. Moreover, server authentication could be addressed using a Synchronised Token Pattern (CSRF Token) that prevents against Cross-Site Request Forgery (CSFR) attacks [Barth et al.08].

Concerning the potential security threats introduced by WebGL and WebCL, the Khronos Working Group is continuously addressing raised issues [Khronos17] and adopted by the browsers continuously.

3.2.3.3 SaW Architecture

The deployed SaW solution works over a client-server architecture (see Figure 3.1). It improves the architecture presented on [Zorrilla et al.13] towards a hardware accelerated approach, considering all the new aspects introduced by usage of GPU resources within SaW concept. On the server-side there is a SaW Scalable Cloud Server (SSCS) which manages server resources in order to provide a consistent, scalable and a single service front-end to the clients. It deals with balancing the load through the different available servers. The SaW client-side is completely Web browser oriented. Hence, emerging technologies such as HTML5, JavaScript, WebGL or WebCL play a crucial role by providing interoperability to cope with hardware and software heterogeneity. Algorithms 1 and 2 provide an example of SaW with the client device benchmarking process and the SSCS workflow respectively.

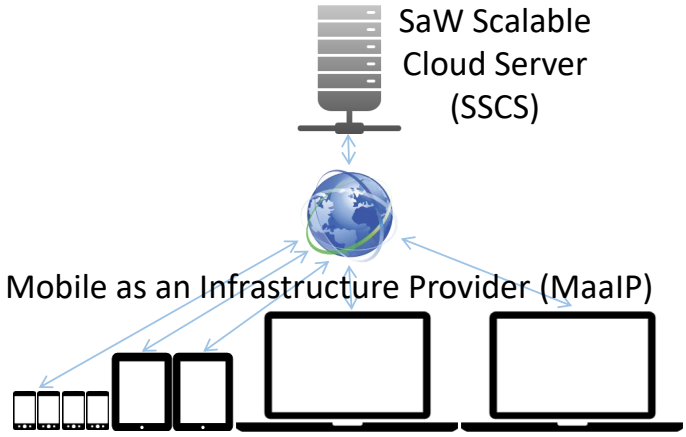


Figure 3.1: General SaW system architecture diagram

SSCS executes two concurrent tasks in Algorithm 2. First, by *EnrollThread*, SSCS continuously performs a recruitment loop which orders the new devices connected to the social media service to self-assess their performance scores. Thus, SSCS enrolls the devices in the appropriate queue based on the reported type following Algorithm 1, which is executed in client devices remotely and provides, as an outcome, a normalised

device type i according to the classification in Table 3.2. Second, by *TaskDistribution*, SSCS matches the queued image processing tasks to suitable devices in terms of workload and elasticity factors. To this end, the image size and algorithm complexity are considered. This decision program of SSCS is shown in Algorithm 2.

Algorithm 1 Device benchmark example

procedure BENCHMARK(d_{id}) ▷ assessed at each device
Input: d_{id} ▷ device ID from social media session
$\hat{b_d}$ ▷ estimated bandwidth for device
$\hat{F}_{cd}$ ▷ estimated CPU processing capability
$\hat{F}_{gd}$ ▷ estimated GPU processing capability
$i \leftarrow$ getDeviceType($\hat{b_d}$, $\hat{F}_{cd}$, $\hat{F}_{gd}$)
report i ▷ send normalised device type to the SSCS following the classification in Table 3.2

All the computing and data transmission overhead in the client-side cannot affect the experience of the consumed content. Hence, on a first step, the SaW system has to create a device capabilities profile. To this end, the server adds in the first response to the client a benchmarking test in order to assess the processing capabilities and hardware assets of the client device (*Benchmark* function from Algorithm 2). The score is sent to the SSCS task distribution manager, which decides the complexity of the background image analysis tasks that fit into that client following the global task distribution strategies.

On a second step, once the SSCS has set specific tasks to be run on a suitable client device, a data transfer is initiated from the server with the image frame and the image processing JavaScript script or scripts (*Dispatch* function from Algorithm 2). These are classified by complexity and invoke different technologies such as WebGL or WebCL to exploit the GPU and/or multi-core assets of the device. The client applies the scripts over the images as a background process, avoiding any user experience damage through elasticity factor considerations. The computed results are sent back to the SSCS and it harvests, formats and stores all the incoming image computing outcome to be mined later by the service provider. While a user is enjoying a social service similar to YouTube or Vimeo, SaW allows the service to deliver independent image analysis tasks queued to the different clients until each session finishes. In case the server does not receive a result in an elapsed maximum time from a specific client (*waitTTL* call from Algorithm

Algorithm 2 SSCS workflow example

```
   procedure EnrollThread( )                                   ▷ SSCS recruitment loop
Data: qD                                                       ▷ N queues by type of available devices
      for each newSession d_id do                              ▷ new connection
          i ← Benchmark(d_id)                                  ▷ get type from the device
          queue(qD_i, d_id)                                    ▷ queue device based on type

   function Transfer(Σ_j, Ω_k, d_id, i)                        ▷ send task to a device of type i
Input: Σ_j                                                     ▷ image to be processed
Input: Ω_k                                                     ▷ programmed image processing algorithm
Input: d_id                                                    ▷ device ID from social media session
Input: i                                                       ▷ normalised device type
Data: qD_i                                                     ▷ queue with available devices of type i
      deliverTask (Σ_j, Ω_k) to d_id                           ▷ deliver task to resource
      waitTTL(i)                                               ▷ wait estimated TTL for device type i
      if error then
          return error_msg                                     ▷ error
      if timeout then
          dequeue(qD_i, d_id)                                  ▷ remove unresponsive device
          return error_msg                                     ▷ timeout
      return ok                                                ▷ ok

   function TaskDistribution(Ω_k)                              ▷ Tasks dispatching loop
Input: Ω_k                                                     ▷ programmed image processing algorithm
Data: qΣ                                                       ▷ images queue
Data: qD                                                       ▷ N queues with available devices based on type
      while !empty(qΣ) do                                      ▷ more images in the queue qΣ
          Σ_j ← dequeue(qΣ)                                    ▷ next image to be processed
          z ← getTargetDevice(Σ_j, Ω_k)                        ▷ target device type under elasticity
                                                                   factors
          for i = N to z do                                    ▷ start from more powerful devices N, stop at
                                                                   target devices z
              if !empty(qD_i) then
                  d_id ← dequeue(qD_i)
                  Transfer(Σ_j, Ω_k, d_id, i)                  ▷ assign task to resource
                  if !ok then
                      queue(qΣ, Σ_j)                           ▷ re-queue image
      return completed                                         ▷ completed

   procedure MainLoop( )                                       ▷ SSCS main loop
      EnrollThread()                                           ▷ recruitment loop
      while !empty(qΩ) do                                      ▷ more algorithms in the queue qΩ
          Ω_k ← dequeue(qΩ)                                    ▷ next batch processing
          TaskDistribution(Ω_k)                                ▷ tasks dispatching loop
```

2), it considers that device is not available anymore and queues it to another one. Figure 3.2 depicts a more detailed client-server SaW architecture and the communication between them.

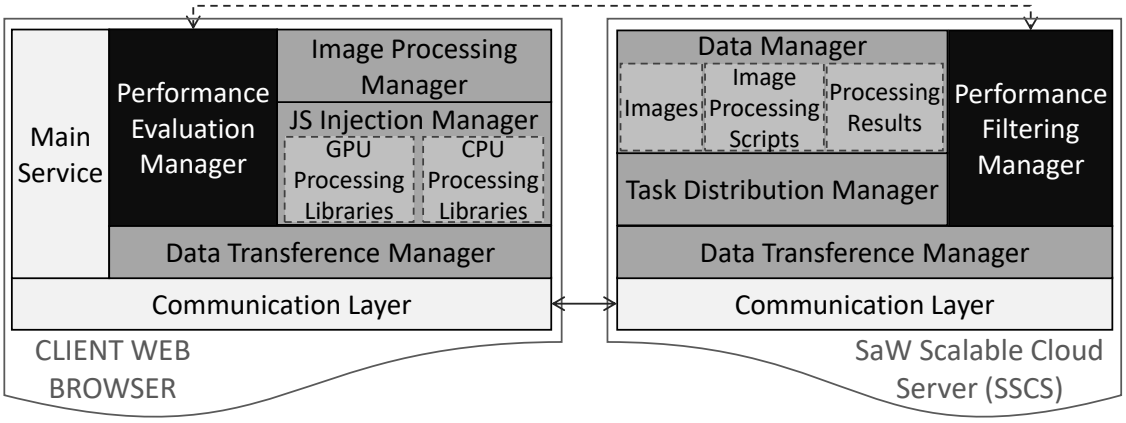


Figure 3.2: SaW Client-server block diagram and its communication

Client Web Browser SaW Architecture

The SaW approach is designed to run the client-side application over a standard Web browser composed by the following modules (see Figure 3.2):

Main Service: This is the main social media application of the service provider and gates what the user wants to consume. Note that a client using the Main Service has committed to join SaW by allowing service providers to gain idle resources in the user's device to add background activities while preserving a good Quality of Experience (QoE).

Communication Layer: This module enables the communication between the client and the server with widely supported Web communication protocols: Websocket and AJAX. The WebSocket Protocol enables two-way communication between a client and the server. Here the security model used is origin-based that is widely used by Web browsers. The protocol consists of an opening handshake followed by basic message framing, layered over TCP. The goal of this technology is to provide a mechanism for browser-based services that need two-way communication with servers that does not rely on opening multiple HTTP connections [Fette and Melnikov11]. Even if the implementation of the WebSocket protocol is widely implemented and on the roadmap of all the Web browsers, nowadays there are some restrictions to use Websockets over some

devices, such as mobiles. Anyway, a less efficient polling approach with Ajax is a feasible alternative to WebSocket. Ajax is a group of interrelated Web development techniques used on the client-side to create asynchronous Web applications. With Ajax, Web applications can send data to, and retrieve data from, a server asynchronously keeping visual fluidity and behavior of the foreground Web application [Garrett et al.05].

Performance Evaluation Manager: This module launches a performance test at the beginning of the application runtime in order to profile the capabilities of the device in terms of CPU, GPU, and the available bandwidth (see Algorithm 1). Then, according to the benchmark results, a normalised device type estimation is sent to the SSCS and it settles the complexity threshold for image processing that this device can deal with considering all the discovered aspects. This evaluation test is not repeated during an active session but it could be performed again to tune the background tasks to a new context, specially if the main service is very changeable from a processing requirement perspective.

Data Transference Manager: This module works hand in hand with the Communication Layer, dealing with the data transference between the client and the SSCS. HTML5 Web Storage facilities are used to create and maintain the incoming data. On the one hand, the client receives a new image for each image processing task and one or various scripts to be applied for that image. These are stored locally and once the processing is over, this module fits the format of the results to transfer them to the SSCS. It is important to highlight that SaW runs entirely in the memory of a Web browser.

JS Injection Manager: It takes charge of handling the scripts received from the server. This module injects and deletes the scripts on runtime without interfering on the user experience and prepares them to execute the background tasks in an optimal way. It also manages the libraries to be used on each case to take advantage of the GPU and CPU resources of an appliance depending the performance ranking. This module will have available some *CPU Processing Libraries* oriented to exploit CPU resources enabling multi-core tasks through WebCL, and some *GPU Processing Libraries* in order to foster hardware acceleration by the GPU of the client device using WebGL and WebCL technologies.

Image Processing Manager: This core layer provides the Web application an API to perform the management of the image processing scripts on the client side. It runs im-

age analysis tasks in the background on top of the JS Injection Manager and using the CPU and GPU processing libraries.

SaW Scalable Cloud Server Architecture

The SSCS has different modules to manage all the SaW service infrastructure on the server side. These elements are presented on Figure 3.2 and briefly explained below:

Communication Layer: It manages the communication between the SSCS and the client. With the same functionality mentioned in the client side, this module is deployed on top of WebSocket and AJAX protocols.

Data Transference Manager: It is supported by the Communication Layer and it is the responsible for exchanging the data with the client (e.g. the images and the scripts for each processing task).

Task Distribution Manager: This block has the global view of the SSCS to categorise and dispatch all the image analysis tasks that the service provider wants to perform through the client device community. It splits and queues the processing jobs by connecting an image with some specific scripts and estimating the complexity of each computing work. It collaborates with the Performance Filtering Manager module and requests and exchanges data with the Data Manager module.

Data Manager: This block manages all the data involved in the SSCS interrelated from different data-bases containing the *Images* to be processed, the *Image Processing Scripts* (some of them to be run over CPU architectures and others over GPU ones), and the *Processing Results* obtained by the clients.

Performance Filtering Manager: This element receives the performance assessment from the clients and analyses the capabilities of the devices to inform the Task Distribution Manager module, who assigns a specific task to that device gaining specific hardware (GPU or CPU) acceleration according to its assets disposal (see Algorithm 2).

3.2.4 Evaluation

In this Section a proof-of-concept implementation of SaW architecture is described, providing experimental evaluation results and an analysis of the performance based on a previous model. The evaluation is focused in two different aspects:

- The scalability of the SaW approach, with a specific comparison between the involved Web technologies: WebGL and WebCL.

- The performance behaviour of the system when considering different types of client devices, according to the target SaW scenarios. This includes a model of the computational cost that considers CPU, GPU and communication resources, as well as some performance figures obtained for different scenarios with realistic combination of device types.

3.2.4.1 **WebGL and WebCL scalability comparison**

This subsection presents the experimental results of using WebGL and WebCL technologies in order to explore the scalability of current Web browsers to exploit GPU resources. A proof-of-concept implementation of the SaW testbed has been developed to distribute a queue of 100 tasks over a different number of clients with identical capabilities. Using homogeneous devices enables to measure the scalability without loss of generality. The heterogeneity of the devices is addressed in next subsection.

For that purpose, a SSCS implementation has been built with a combination of Node.js and MongoDB to obtain a low latency server and to be able to deal with high concurrency requests. Both technologies provide event-driven systems that enables a non-blocking I/O model that makes it lightweight and efficient in high concurrency environments with a NoSQL data structure. Three different instances of the server have been deployed in order to avoid bottlenecks and provide sufficient resources for the different clients.

In the client side, according to the proposed architectural design, our implementation works over Web standards to cover a wide set of devices and follows a modular design. These modules enable a real-time communication with the server and are able to inject JavaScript libraries in runtime for the background tasks.

As clients, we used a different number of identical PCs with the following capabilities: Windows 8.1 Intel(R) Core(TM) i5-3330 CPU @ 3.00GHz with Intel(R) HD Graphics 2500. To test WebGL, Firefox 42.0 Web browser has been used, that enables WebGL by default [CanIUse17]. Currently, there is no native support for WebCL in Web browsers. Thus, an experimental extension [NokiaResearch17] has been used on top of a portable Firefox 22.0 Web browser.

The server creates a queue of 100 tasks with an image and an associated algorithm for each image. The sever dispatches the tasks to the available clients. Since all the

clients have the same capabilities, the tasks are homogeneously distributed through all of them.

The performed algorithm computes the DITEC method (Trace transform based method for color image domain identification) [Olaizola et al.14] by means of algebraic operations such as matrix dot products that can be highly parallelised at different states (per frame, per angle during the Radon Transform operation, etc.).

We have evaluated two different SaW implementations with the aforementioned workload, the first one using WebGL and the second one using WebCL. In the performed experiments the workload has been distributed among a number of workers going from 1 to 20. The same queue of tasks has been also performed on a single PC with the same capabilities using OpenGL and OpenCL instead of doing it from the Web browser. The results obtained are shown in Table 3.1.

Comparing the values described in Table 3.1, obtained over the same PCs, of using a local server with OpenGL and OpenCL, with a single worker in the distributed approach with WebGL and WebCL respectively, it can be said that the local approach obtains better results. The reasons are mainly two: (1) the latency introduced by the delivery time of the image, the script and the results between the SSCS and the client in the distributed approach, and (2) the performance gap of the bindings of WebGL and WebCL to exploit the hardware resources in comparison with OpenGL and OpenCL.

From the obtained values in Table 3.1 regarding the distributed approach with different number of clients, it can be inferred that (1) the speedup is very high for both implementations, which denotes that the parallelisable fraction of the workload is very big, as expected for the described SaW use case; (2) WebCL implementation performs better than WebGL, and (3) the speedup obtained for the WebCL version is not as high as the one obtained for WebGL.

Using Amdahl's Law [Amdahl67] we have calculated the parallelisable fractions of the workload for both WebGL and WebCL versions, which have resulted 98.87% and 94.36% respectively. To obtain these values, we have excluded the measures obtained with one single worker, since it does not reflect the task scheduling effects of managing different workers and consolidating the results. As shown in Figure 3.3, Amdahl's Law interpolates real measures with reasonable fidelity for the range [2, 20] of workers. Accordingly, we have used this approximations to predict results for 50 and 100 workers

Table 3.1: Computational cost in terms of time for the same workload for a local sever (OpenGL & OpenCL) and for a number of distributed workers from 1 to 20 (WebGL & WebCL)

LOCAL SERVER

Number of workers	Computational time in ms with OpenGL	Computational time in ms with OpenCL
1	132,570	61,780

DISTRIBUTED APPROACH

Number of workers	Computational time in ms with WebGL	Computational time in ms with WebCL
1	275,295	111,300
2	161,820	73,765
5	63,260	34,290
10	34,476	19,740
20	17,825	11,120

(see Figure 3.3). As appreciated in the Figure, WebCL would outperform WebGL until near 100 workers.

3.2.4.2 **Performance Modeling with heterogeneous devices**

In this subsection we present a performance evaluation model of the SaW approach based on the model published in [Zorrilla et al.13] taking into account different type of devices. More specifically, here we take into consideration both CPU and GPU resources, while in [Zorrilla et al.13] only CPU resources were contemplated.

The performance of SaW can be analysed by following the Bulk Synchronous Parallel (BSP) model [Valiant90]. The BSP model is a generalisation of the classical PRAM model [Fortune and Wyllie78] for shared memory.

As presented in [Zorrilla et al.13], we will consider 3 different kind of devices as processing units for the distributed approach: smartphones, tablets and PCs. Table 3.2 shows different connectivity and processing power values for each one of the device types based on market surveys [TomsHardware11] [LegitReviews12]. The table includes information about a server in order to give a comparative estimation of the computational cost. According to market surveys [Analytics17], 50% of the PCs and 30% of tablets have a GPU available, while it decreases until 10% in the case of the smartphones.

Equation 3.1 extends the Equation 6 of [Zorrilla et al.13], which considers the total computational cost (C_T) as the time to perform a workload unit distributing it across all the different type of devices in parallel.

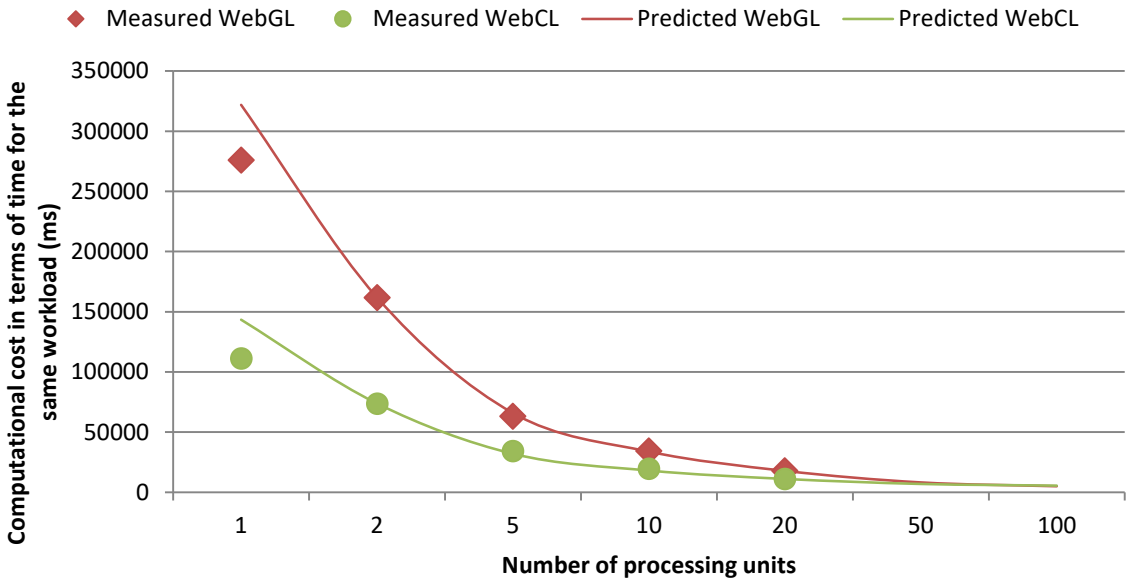


Figure 3.3: Computational cost in terms of time for a different number of workers in terms of processing units in the distributed approach for WebGL and WebCL. The real measured values, presented in table 3.1, are shown for a range of workers from 1 to 20, while predicted values, following Amdahl's Law, are shown for a range of workers from 1 to 100.

Table 3.2: Estimated processing and communication properties for different type of devices (i)

ID	Device	Connect.	Bandwidth $\hat{b}_i$	CPU GFlops $\hat{F}_{ci}$	GPU GFlops $\hat{F}_{gi}$
(m)	Mobile phone	UMTS	3Mbit/s	0.05	0.2
(t)	Tablet	Wifi	8Mbit/s	0.08	0.32
(p)	PC	DSL	20Mbit/s	2.5	10
(s)	Server	SATA	6Gbit/s	p_sx82.8	–

$$C_T = \left(\sum_{i=1}^{n} \frac{1}{C_{ci}} + \sum_{i=1}^{n} \frac{1}{C_{gi}} \right)^{-1} \tag{3.1}$$

The equation divides the partial computation time cost for each type of device to perform their part of the workload (C_i) in two:

- C_{ci}: the partial computation time cost for each type of device that only have CPU processing capabilities to perform their part of workload.

- C_{gi}: the partial computation time cost for each type of device that have GPU and CPU processing capabilities to perform their part of the workload.

- n: the number of different type of devices. Note that according to the information of Table 3.2, n will be 3 since the table defines three type of devices for the distributed approach: smartphones, tablets and PCs.

Equations 3.2 and 3.3 represent C_{ci} and C_{gi} respectively, extending the equation 7 of [Zorrilla et al.13] and assuming sufficient resources in the server side,

$$C_{ci} = \frac{W_i}{f_{pci} \cdot \hat{F}_{ci} \cdot p_{ci}} + \frac{g_i}{f_{bi} \cdot \hat{b}_i \cdot p_{ti}} + \hat{m} \cdot p_{ci} \tag{3.2}$$

$$C_{gi} = \frac{W_i}{f_{pgi} \cdot \hat{F}_{gi} \cdot p_{gi} + f_{pci} \cdot \hat{F}_{ci} \cdot p_{ci}} + \frac{g_i}{f_{bi} \cdot \hat{b}_i \cdot p_{ti}} + \hat{m} \cdot p_{gi} \tag{3.3}$$

where:

- W_i is the computational workload in terms of the computation time assigned for device type i, to be distributed among all the different available processing units of device type i.

- g_i is the communication workload in terms of number of bytes of information to be transmitted from the server to the devices of type i.

- $\hat{m}$ is the estimated cost in terms of computation time to establish a new task to a processing unit and its management.

- $\hat{b_i}$ is the average estimated bandwidth for device type i (see Table 3.2).

- $\hat{F}_{ci}$, $\hat{F}_{gi}$ are the average estimated processing capability in terms of CPU, GPU flops, respectively, for device type i (see Table 3.2).

- p_{ci} is the number of different processing units of device type i with only CPU capability.

- p_{gi} is the number of different processing units of device type i with both GPU and CPU capabilities. Note that following the assumption that all the devices with GPU capability will also have CPU capabilities, in Equation 3.3 p_{ci} and p_{gi} will be the same number of processing units.

- p_{ti} is the number of messages exchanged between the processing units of type i and the server.

- f_{bi} is the elasticity factor introduced to determine the percentage of the bandwidth to be used from the available bandwidth for device type i.

- f_{pci}, f_{pgi} are the elasticity factor introduced to determine the percentage of the CPU, GPU, respectively, to be used from the available HW resources of device type i.

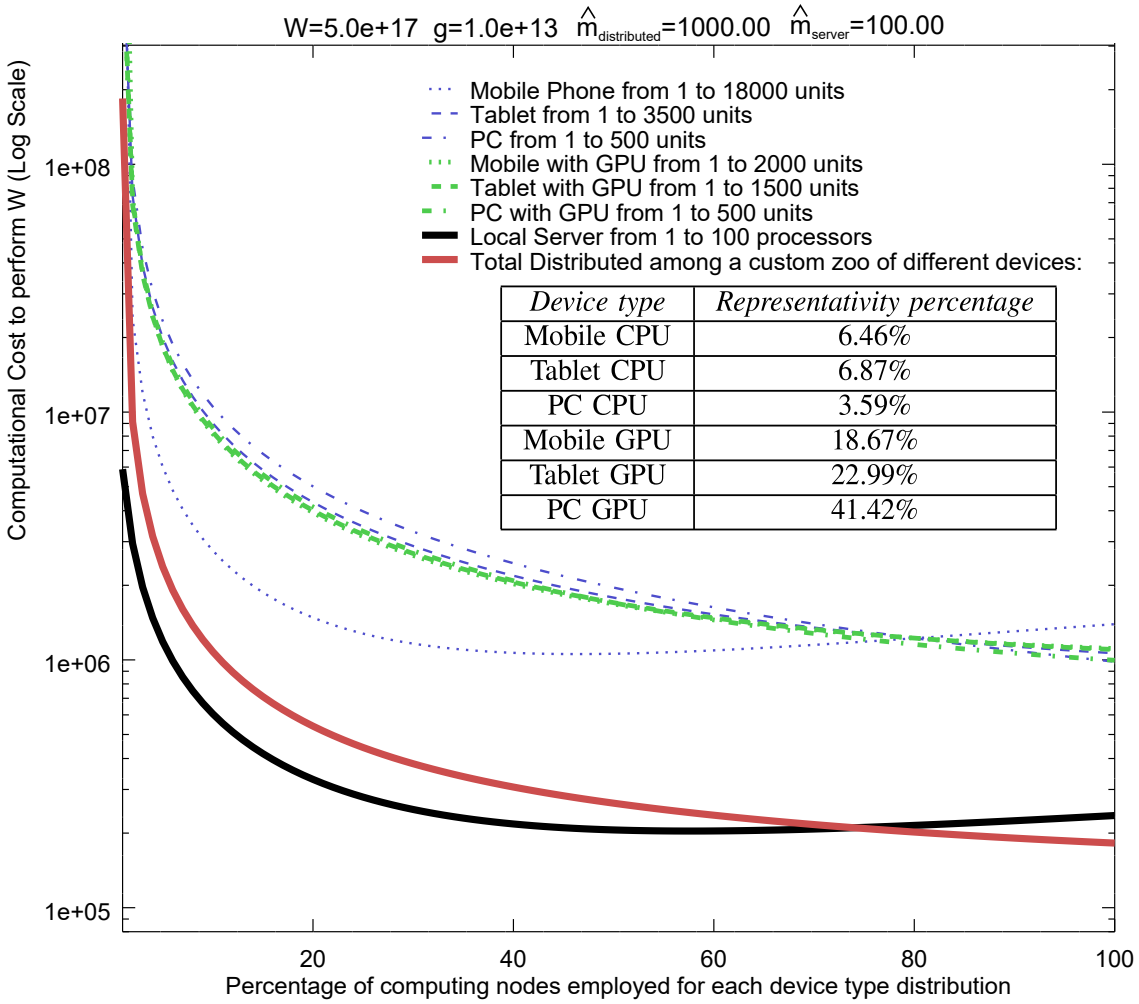
As presented in [Zorrilla et al.13], the same model can be used for a multi-core server approach in order to give a comparative estimation with the distributed approach. Equation 3.4 shows the cost of a multi-core server in terms of time (C_s) with p_s processing units sharing a single memory:

$$C_s = \frac{W}{f_{ps} \cdot \hat{F}_s \cdot p_s} + \frac{g}{f_{bs} \cdot \frac{\hat{b}_s}{p_s}} + \hat{m} \cdot p_s \tag{3.4}$$

In order to compare a distributed computing approach with a dedicated local server, different C_T and C_s have been calculated applying the aforementioned model. The elasticity factor has been set to 0.15 both for bandwidth (f_b) and for CPU processing power (f_{pc}) for the background activities of the distributed approach. However, the elasticity factor for GPU processing has been set to 0.3 for mobiles and tablets (f_{pgm} and f_{pgt}), and 0.5 for PCs (f_{pgp}), since using the GPU will have a lower impact on the user experience than adding background tasks to the CPU. Finally, we considered that the local server is exclusively dedicated to these tasks (f_{ps}=f_{bs}=1).

Figure 3.4 shows the performance behaviour for operational values of model parameters. A lineal increment of processing units is compared for specific values of W, g and $\hat{m}$. In order to understand the graph that serves to compare the cost for heterogeneous population of computing assets, it is important to highlight that the maximum number

of computing units for each device type is different, providing a collection of heterogeneous devices with all the curves inside the same picture. This collection deals with the heterogeneity of the device types from the performance capacity perspective, and focuses on the full distributed cost model with all the parameters coming into play. For example, Figure 3.4 shows that all device types have the same cost at 80% of the computing nodes, but this cost is given by 14400 mobiles with CPU, while the same cost is obtained with 1600 mobiles with GPU and CPU available.



Device type	*Representativity percentage*
Mobile CPU	6.46%
Tablet CPU	6.87%
PC CPU	3.59%
Mobile GPU	18.67%
Tablet GPU	22.99%
PC GPU	41.42%

Figure 3.4: Computational cost estimation for different volumes of device types for a constant work load (W) communication cost (g) and task management cost ($\hat{m}$: $\hat{m}_{distributed}$ for the client devices and $\hat{m}_{server}$ for the server). The table presents the load balance between the different devices in the distributed approach to have the same computational cost at 80% of the X axis.

As it can be observed in Figure 3.4, while the total distributed cost decreases for more devices, the local server starts to increase after reaching a minimum with 58 workers. This reflects that the data communication bus on the server side is a bottleneck while, in the distributed solution, the servers should theoretically have enough bandwidth to provide the required bandwidth for each device. This trend becomes more evident as the communication cost (g) increases. This, and other parameters from the cost model, can be simulated and evaluated by the reader using the following numerical computation program [Zorrilla17] with Octave [Eaton17].

From the cost model, the efficient management of all the created tasks becomes a critical factor as well when $\hat{m}$ is increased. The distributed approach has to manage thousands of devices while the local server goes from 1 to 100 workers so the management cost has a bigger impact in the SaW-based distributed solution.

In the case the global workload (W) increases, maintaining the same communication cost, a bigger gap between the number of workers in the local server and the volume of devices in the distributed approach is needed to push the distributed solution performance ahead. To sum up, with a enough size of user's devices partially dedicated to social service improvement it is possible for a SaW system to lead traditional server based performance. Moreover, the elasticity factor, which has been set in a conservative way, can be increased considerably in many scenarios without damaging the user experience.

3.2.4.3 **Remarks**

In this section some remarks are reflected regarding the validation of the SaW hypothesis that has been introduced in Section 3.2.1. Recall first that the SaW approach is oriented to complement a cloud server, and not intended to beat it in computational performance. In this regard, the delay-tolerant nature of the tasks to be distributed to the clients, such as in a video tagging scenario, plays in favour of the SaW approach.

As an example, consider a task that will require a time t to be executed in the server, and, upon the results obtained in Section 3.2.4.1, assume for the workload a parallelisable fraction of about 99%. This means that the server has to spend about 1% of t for the distribution overhead, represented for the third term in Equations 3.2 and 3.3. In

other words, the server would be able to manage the distribution of about 100 of these tasks in the computational time of t.

Continuing with the same example, assume now that the server is dedicated exclusively to task distribution, and that the SaW ecosystem is composed only of smartphones with about 1/50 of the processing power of the server according to Table 3.2. Note also that to preserve QoE, the smartphones will work with an elasticity factor, say it 0.15 to be conservative, which leads to a processing capability for each smartphone of 3/1000 of the server power capacity. For a set of 100 smartphones, the server will take a processing time of t to distribute the queue of 100 tasks, and will obtain all the results back in a time lapse of hundreds of t from the smartphones. This concludes that the server consumes only resources for time t, equivalent to perform a single task, and will asynchronously obtain the results for 100 tasks instead. However, the time period will be of hundreds of t.

Nevertheless, note that the example above reflects a worst-case scenario, according to the current use cases and user habits already mentioned. In a more realistic scenario, like the ones presented in the former subsection, the SaW approach would elastically distribute the workload among the different devices according their features (such as smartphones, tablets and PCs with CPU and/or GPU processing capabilities).

To summarise, the exchange of "time-for-resources" or, from the service provider perspective, "time-for-money" explained through the above example is in the core of the SaW approach, since delay-tolerance and elasticity provides sufficient freedom degrees in usual scenarios. Finally, the technological evolution also plays in favour of the approach. The performance gap between the different type of devices has been continuously reduced in the past and still continues. Besides, improvements in the bindings of Web browsers to support WebGL and WebCL can also be expected, which will result in a more efficient use of the hardware resources.

3.2.5 Conclusions

In this paper we have introduced the concept of Social at Work, SaW, which aims to complement a Web-based social media service with all the idle devices, mostly mobiles, that usually have underexploited resources while accessing the service. SaW proposes a Mobile as an Infrastructure Provider (MaaIP) model, creating a system related to Mobile

Grid Computing concept with the available CPU and GPU resources of the different client devices, to complement a virtualised cloud server providing the social media service.

Aimed to achieve enhanced and automatic media tagging over social media datasets, SaW fosters background dispatching of media analysis over connected clients, providing a high elasticity and dealing with the availability of the resources related to the spontaneous presence of users. Then, SaW copes with hardware-accelerated image processing tasks execution in background, according to the capabilities of each device. The computing tasks are embedded in the foreground social content without draining the users' bandwidth or affecting to the perceived Quality of Experience. In harmony with the presented scenario, delay-tolerant background tasks enable the SaW approach to exchange "time-for-resources" or "time-for-money". This means that mobile devices, instead of being as resource intensive as servers, can dedicate the sufficient time to perform the task, preserving the QoE according to their capabilities, and saving cloud costs to service providers.

SaW deploys a powerful pure Web platform for video analysis by means of exploiting high user availability density, and the explained capability to run scripts in background threads of Web browsers. Therefore, the SaW concept targets a device community as a processing grid removing the need for install client applications, adding a delivering computing layer to the stack of the HTML5-based main service instead.

In order to evaluate the approach, we have developed a proof-of-concept implementation of SaW, including versions for existing WebGL and WebCL technologies. Results of the experiments show the high speedup obtained by parallelisation, which confirm the scalability of the approach exploiting GPU resources from Web browsers with both WebGL and WebCL technologies. The scheduling elasticity in the server side has been designed to take advantage from the delay-tolerant target scenarios, with a heterogeneous community of client devices characterised by the assorted availability of resources.

This paper has extended a previous performance model that was focused only in CPU resources, to consider both CPU and GPU capabilities. The model allows to predict the performance of a distributed system including diverse client devices, which have been illustrated through a set of example configurations, in comparison with a local

server solution. The maximum benefit is obtained for higher delay-tolerant computational load, with independent tasks able to be distributed to idle devices, being able to compensate the task scheduling management and consolidation overload of the server. The technological evolution, with a clear trend to reduce the performance gap between laptops and mobile devices, as well as to improve the efficient exploitation of hardware resources from a Web browser, favours the SaW approach.

As a summary, SaW deploys a social distributed computing infrastructure on top of pure Web-based technologies, building a grid of resources to perform background media analysis tasks leveraging hardware-acceleration for a social media service.

Acknowledgment

The authors wish to thank the editor and reviewers for their constructive comments and suggestions to improve the manuscript. This work was supported by the European Commission project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action). The research in the UPV/EHU has been partially supported by the Spanish Research Council, Grant TIN2016-79897-P, and the Basque Government, Grant IT980-16.

CHAPTER

4

Client-side Bitrate Adaptation

4.1 Context

HAS solutions enables dynamic and efficient adaptation from media players to specific display features and changeable connectivity performance, by publishing segments with the different media that can be played. Thus, the players autonomously take the real-time decisions to request a specific segment tied to a nominal bitrate according to the connectivity performance to maximize the quality of the playback.

This client-driven approach, where control is distributed over the various clients and each client strives to optimize its individual quality, has some issues that can damage the QoE. The issues span initial buffering delay, temporal interruptions or pauses, and visible video resolution switches during a video transmission. This QoE degradation is even tighter in dense client environments, when considering a cellular network, the radio access network (RAN), a Wi-fi hotspot, and the network edge. Therefore, it becomes complex to provide video services to several users competing independently for the available bandwidth when trying to maximize the used bitrate.

Traffic shaping of HAS streams, when considering fairness, efficiency and quality, can reduce the number of stalls and quality switches for clients sharing a bottleneck link [Quinlan et al.15].

A client-side bitrate adaptation mechanism must deal with dynamics from dense client environments to coordinate QoE for dense client cells in 5G networks. To create such a system, some aspects must be overcome. First, exploit client-side heuristics to assess the available bandwidth accurately. Then, a mechanism for bitrate adaptation without a priori knowledge is needed. Thus, the algorithm can be applied to any content and its response is fast, tracking sudden changes in network dynamics. Finally, it is necessary to distribute radio resources fairly to get a steady, homogeneous and efficient radio link utilization.

To meet this scenario, Section 4.2 describes a heuristic based bitrate decision algorithm, called LAMB-DASH. Compared to literature alternatives the algorithm does not require a priori knowledge, so it produces a fast response, valid for any kind of incoming content characteristics or connectivity status. Furthermore, is based on a lightweight processing model, based on measurements and estimations from a current stream state. LAMB-DASH goals to improve the Quality Level (QL) chunk Mean Opinion Score (c-MOS). This QoE model limits the quality evaluation to a set of objective metrics from the connection heuristics, such as quality switches, frequency and duration of freezes. These parameters are the key metrics of HAS services.

To validate the results from LAMB-DASH, it has been implemented and deployed in a real, not simulated, setup where several clients compete for the available network resources.

4.2 LAMB-DASH: A DASH-HEVC adaptive streaming algorithm in a sharing bandwidth environment for heterogeneous contents and dynamic connections in practice

- **Title:** LAMB-DASH: A DASH-HEVC adaptive streaming algorithm in a sharing bandwidth environment for heterogeneous contents and dynamic connections in practice
- **Authors:** Angel Martin, Roberto Viola, Josu Gorostegui, Mikel Zorrilla, Julian Florez and Jon Montalbán
- **Journal:** Journal Real-Time Image Processing
- **Publisher:** Springer

- **Year:** 2017
- **DOI:** http://dx.doi.org/10.1007/s11554-017-0728-x

Abstract. HTTP Adaptive Streaming (HAS) offers media players the possibility to dynamically select the most appropriate bitrate according to the connectivity performance. A best effort strategy to take instant decisions could dramatically damage the overall Quality of Experience (QoE) with re-buffering times and potential image freezes along with quality fluctuations. This is more critical in environments where multiple clients share the available bandwidth. Here clients compete for the best connectivity. To address this issue we propose LAMB-DASH, an online algorithm that, based on the historical probability of the playout session, improves the Quality Level (QL) chunk Mean Opinion Score (c-MOS). LAMB-DASH is designed for heterogeneous contents and changeable connectivity performance. It removes the need to access a probability distribution to specific parameters and conditions in advance. This way, LAMB-DASH focuses on the fast response and on the reduced computing overhead to provide a universal bitrate selection criteria. This paper validates the proposed solution in a real environment which considers live and on-demand Dynamic Adaptive Streaming over HTTP (DASH) and High Efficiency Video Coding (HEVC) services implemented on top of Gstreamer clients.

Keywords: Adaptive Streaming, DASH, HEVC, QoE, dense client environments

4.2.1 Introduction

The combination of increasing video streaming users heavily dominating the traffic over the Internet, the demanded high quality from the cutting edge displays of their devices and the required support for mobility is driving the evolution of media services. Fueled by improved cameras with stunning picture quality [Saad et al.15] and the breakthroughs in display technology [Kathirgamanathan et al.15], the traffic for videos delivered over the Internet will reach 80% of the total Internet traffic by the end of 2019, according to the report issued by the world IT leader Cisco [Inc17b]. Meantime, reaching heterogeneous devices gains relevance thanks to the growth of mobile devices as an entry point to these services [Inc17a].

From an industry perspective, solutions for video distribution need to allow video traffic to cross delivery networks and middleboxes without the need for a specific setup.

Moreover, video streaming services must work on top of unmanaged delivery networks, where quality is not guaranteed, on a best-effort basis [Sodagar11]. Furthermore, they have to facilitate the development of new business models and personalized advertising [Maillé and Schwartz16].

All the described requirements have led to the creation of new efficient video streaming techniques over Hypertext Transmission Protocol (HTTP). HAS responds to demands from multimedia services supporting heterogeneous display setups, different user preferences and languages and changeable mobility situations with a Content Delivery Network (CDN)-ready design. It benefits from the ubiquitous connectivity because practically any connected device supports HTTP. HAS is a pull-based protocol [Begen et al.11] that easily traverses middleboxes, such as firewalls and NAT devices. At the same time, it keeps minimal state information on the server side, making servers more scalable than conventional push-based streaming servers. Last but not least, concerning existing HTTP caching infrastructures, the protocol stack of HAS is not different compared with any other HTTP application. This allows distributed CDNs to enhance the scalability of content distribution, where individual segment of any content is cacheable as a regular Web object.

HAS solutions provide a manifest file detailing a playlist of segments with the different media that can be played. The essence of this approach is the transformation of the traditional push-mode to a pull-mode. This way, the service delegates the responsibility of operating the service in a proper and efficient manner to the players. To this end, the players autonomously take the real-time decisions to request a specific segment tied to a nominal bitrate. The aim of the bitrate selection algorithm is to maximize the quality of the playback.

This client-driven approach, where control is distributed over the various clients and each client strives to optimize its individual quality, has some issues that can damage the QoE. The issues span initial buffering delay, temporal interruptions or pauses, and visible video resolution switches during a video transmission [Seufert et al.15]. This QoE degradation is even tighter in dense client environments, when considering a cellular network, the Radio Access Network (RAN), a Wi-Fi hotspot and the network edge. There, it becomes complex to provide video services to several users competing independently for the available bandwidth when trying to maximize the used bitrate.

Rate control is a core tool for video coding. Most of existing rate control algorithms are based on the bitrate (R) - quantization (Q) model [Wan et al.11], which characterizes the relationship between R and Q. The Q parameter is the critical factor for rate control as Q directly reflects on the resulting Quality. Moreover, the R - Q model is usually governed by the λ Lagrange multiplier to achieve the target bitrates accurately [Li et al.14a]. Likewise, our implemented LAMB-DASH algorithm deals with the selection of the decoded bitrate online for DASH streams to improve the QoE.

In order to get higher QoE, this paper targets an adaptation algorithm embedded in multiple players sharing a connection link which makes real-time bitrate decisions to conduct a more efficient and fair video transmission. To this end, we propose a bit rate decision algorithm to get a low-complexity adaptation mechanism that improves the QL c-MOS by controlling the bitrate selection criteria of a player, based on the historical probability of the playout session.

The novelty of LAMB-DASH lies, firstly within its flexibility to produce a fast response, valid for any kind of incoming content characteristics or connectivity status, meaning that the algorithm does not require *a priori* knowledge. Secondly, within its design, with a low-complexity heuristic model, based on measurements and estimations from a current stream state. And lastly, within the implementation of the algorithm and its deployment in a real, not simulated, setup in a scenario where several clients compete for the available network resources.

The paper is structured as follows. First, section 4.2.2 contains a review of the related work in terms of Adaptive Streaming over HTTP, the quality decision algorithms, the QoE models and how LAMB-DASH goes beyond this. Then, in section 4.2.3, we introduce LAMB-DASH, including the target scenario for this research work, problem statement and the notation employed, the decision algorithm and the shortcuts adopted in order to implement it in a practical manner. After describing our approach, we describe the implementation using MPEG-DASH [Sodagar11] and HEVC formats [Sullivan et al.12] in section 4.2.4. To assess the outcomes, section 4.2.5 on validation describes the set of experiments carried out and the results achieved compared with the ones from the literature. Finally, we present our conclusions in section 4.2.6.

4.2.2 Related Work

4.2.2.1 Adaptive Streaming over HTTP

HAS imitates traditional streaming via short downloads using a HTTP client, which downloads small video chunks. In an adaptive streaming system, the video content is stored in the server by encoding it in several representations and splitting the resulting streams into many temporal segments. The duration of the segments typically ranges from 2 to 15 seconds depending on the latency constraints of the streaming service. Each representation is characterized by a specific codec, language, resolution, bandwidth, view and framerate.

The client requests segments in chronological order to restore the original content, the chosen representation for each segment can vary in order to adapt the stream to the capabilities of the connection and the player. The bitrate adaptation algorithm inside the client player allows the client to independently choose its playback quality and prevents the need for intelligent components inside the network. The decision is conditioned by the particular decision logic implemented in the client, since it can be based on several adaptation algorithms. This mechanism is depicted in Figure 4.1.

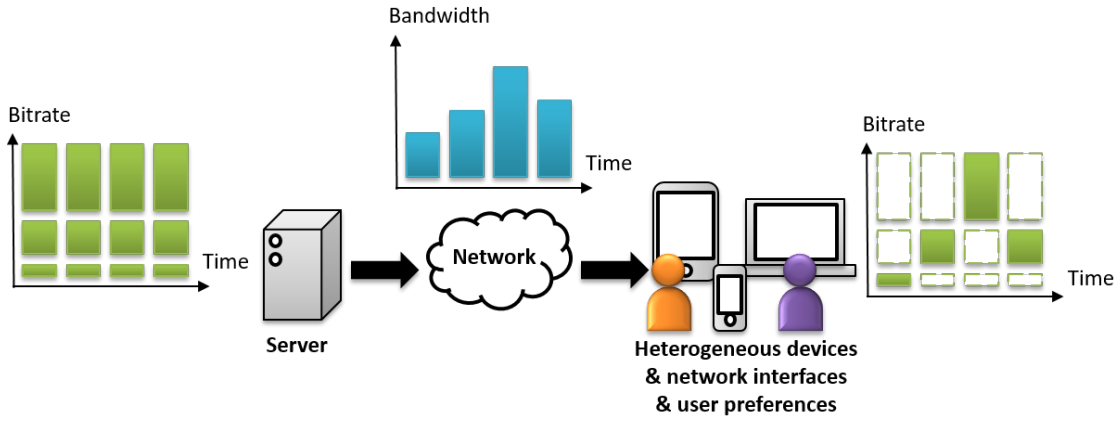


Figure 4.1: Adaptive streaming optimization depending on network performance, device features and user preferences.

4.2.2.2 **Bitrate decision algorithms**

Streaming services have to rely on the experience that derives from the network stability, efficient utilization, and fairness. Recent research in adaptive streaming, such as Low-Latency Prediction-Based Adaptation (Lolypop) by Miller et al. [Miller et al.16] and Chiariotti et al. [Chiariotti et al.16], is focusing on the development of client-side adaptation algorithms. To this end, the client monitors some key indicators in order to perform the decision that better fits with the current state and maximizes the playback quality. Key indicators are not unique, since many factors can be taken into account; in this sense, according to the ones chosen, the algorithms are grouped into connection-based and content-based.

Connection-based algorithms are focused on choosing the bitrate taking into account server-client connection status and the streaming session. Some common indicators are connection bandwidth and latency. Algorithms in this category are Fair, Efficient, Stable, adaptIVE (Festive) [Jiang et al.14], Probe and Adapt (Panda) [Li et al.14b] and Lolypop [Miller et al.16].

The aim of content-based algorithms is to characterize the content in order to adapt the representation bitrate with the scene, i.e. a high-motion scene is more complex than a static one, and then the representations can be improved by choosing a higher level of representations. Typical values to process in this case are Peak Signal-to-Noise Ratio (PSNR) or Structural Similarity (SSIM). The SSIM parameter is usually preferred because the PSNR is a purely mathematical value, while SSIM tends to adapt to the human perception of the image. Content-based algorithms are not as common as the connection-based ones, an example of an SSIM based algorithm is provided by Chiariotti et al. [Chiariotti et al.16]. Unfortunately, the existing research in content-based selection of bitrate suffers from high implementation complexity and large overhead.

More complex solutions are being explored in order address both aspects, the status of the connection-player and the feature of the video content. An attempt at integrating the knowledge of the quality into the Panda algorithm is explained by Zhi Li et al. [Li et al.14c]. However, the issue related to heavy model processing persists.

Another way to classify the adaptation logic is to divide them according to the decision rules of the algorithm. In this sense there are two categories, heuristic-based and optimization-based. The algorithms that belong to the first group are more common in the literature and are based on direct measurements and decision rules based

on the observations. The latter are based on mathematical modelling. Optimization-based algorithms are more precise and potentially generate a higher quality playback than the heuristic-based ones, but they require a big dataset and a long processing time. Going deeper, non-exhaustive mathematical modelling in order to get reasonable trade-off between learning speed and accuracy may even lead to suboptimal solutions. In [Claeys et al.14b] it is compared with a simplified state characterization (to gain efficiency) and more complete controllers (more complex and slower) of the representation selection. It concludes that the modelling is usually not fully representative in practice. Among the already cited algorithms, Panda is an example of a heuristic-based solution, while the SSIM based algorithm by Chiariotti et al. [Chiariotti et al.16] is optimization-based.

The bitrate decision gets more complex in the scenarios where several clients compete for the available bandwidth and in which different video flows traverse the same path in the network. This competition leads to instability in the bitrate decisions, causing frequent oscillations among different bitrate representations, bandwidth under-utilization and unfairness between players [Chen et al.16b]. Here, rate adaptation heuristics, based on the current network conditions captured at the video player, are the most appropriate parameters to dynamically request the appropriate bitrate representation [Petrangeli et al.15]. This work continues evaluating the algorithm through simulations, under highly variable bandwidth conditions and several multi-client scenarios.

4.2.2.3 **QoE models**

With regard to quality, QoE is adopted in order to address human perception. The common way to evaluate QoE consists of submitting the content to a highly diversified audience and reporting their subjective evaluations on a precise evaluation scale. A commonly used scale is the MOS which consists of five increasing levels of quality (from 1 to 5) [ITU]. The disadvantage of this type of testing is that it results in long evaluation times. Subsequently, and for practical reasons, many objective models for MOS estimation have been studied in order to profile the subjective human perception of the quality.

De Vriendt et al. [Vriendt et al.13] investigate the most common models in order to verify the fit of each model. In particular the models shown are: bitrate model, PSNR or SSIM based model, chunk-MOS based model and quality model. It concludes that chunk-MOS model is the optimal one. From here onwards this paper uses this quality model which is a particular configuration of the chunk quality model. Moreover, thanks to the work of Claeys et al. [Claeys et al.14a] the required parameters are limited to a number of objective metrics.

4.2.2.4 **Overview and Outlook**

Related work solves many of the problems in bitrate decision for improved QoE. Most of the algorithms perform characterization of the content and the network conditions resulting in tailored-specific models. The analysis of the performance is done from experiments based on simulations which range from client decisions to network profiling, while others just consider one HTTP client accessing the content [Seufert et al.15, Miller et al.16, Chiariotti et al.16, Li et al.14c, Toni et al.15]. However, in highly dynamic network scenarios arise some issues that are important to tackle. They often need high computing overhead that does not fit with the constrained processing capacities of the mobile devices and the required real-time response.

4.2.3 **LAMB-DASH for adaptive streaming**

4.2.3.1 **Highly dynamic network scenario**

LAMB-DASH is a client-driven approach, where control is distributed over the various clients and each client strives to optimize its individual quality. This situation happens in the provision of media services around a location where an event takes place, which requires the network having to cope with a high peak in multimedia consumption (e.g., a sports event or a concert in a stadium, with users accessing video contents across the network).

In HAS, a video is temporally split into segments which are encoded at different quality rates. Therefore, it allows the clients to independently choose the playback quality, removing the need for intelligent components inside the network to manage the session. The client can then autonomously decide, based on user preferences, display features, the current buffer filling and network conditions, the quality representation to

be requested. This way, control is distributed over the various clients. Thus, adaptive streaming offers a fluent and uninterrupted user experience by means of client-based switching decisions to get continuous viewing.

This autonomous optimization makes the connection conditions highly changeable, especially when considering dense client environments such as a cellular network, the RAN, a Wi-Fi hotspot and the network edge. Several clients or sessions for which the video stream flows traverse the same path in the network therefore compete for the available bandwidth. This competition leads to instability in bitrate selection algorithms, causing oscillations among available quality representations, bandwidth under-utilization and disproportional shares of available bandwidth between players [Chen et al.16b].

This scenario, where different clients influence each other as they compete for shared network resources, is more challenging for bitrate selection algorithms. Furthermore, this scenario offers a more realistic stochastic traffic environment, rather than the synthetic background noise widely employed in many of the simulations carried out in other research works [Seufert et al.15, Miller et al.16, Chiariotti et al.16, Li et al.14c, Toni et al.15].

4.2.3.2 **Adaptive Streaming model**

First, we will introduce the notation used throughout the paper. We consider a video content that is encoded at M representations bitrates and split into N segments of fixed duration τ, such that the total duration is $T = N*\tau$. The indexes $i \in \{0,1,...,N-1\}$ and $j \in \{0,1,...,M-1\}$ identify a particular segment and a particular representation respectively. Each representation j is associated to a particular bitrate R_j.

According to the notation from Miller et al. [Miller et al.16], time related variables are continuous with starting time t=0. t_i^r, t_i^c and t_i^p denote request time for the segment i, its downloaded time and its playback deadline respectively. Consequently, the playout buffer level at time t, denoted $\beta(t)$, is defined: $\beta(t) = \max(t_i^p \mid t_i^c \leq t) + \tau - t$. The buffer level should always be positive, otherwise some frames are skipped and it causes a consequent degradation of the playback quality. On the contrary, if the buffer level reaches the buffer size, some frames are dropped because there is no more space for storing them.

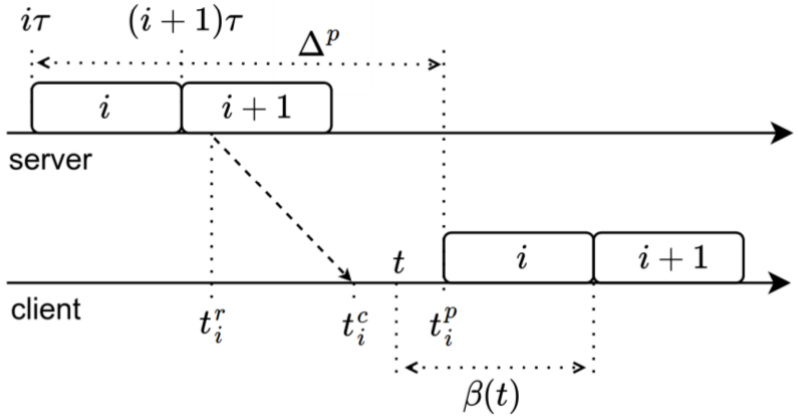


Figure 4.2: Illustration of notation used. Source: Lolypop by Miller et al. [Miller et al.16, Fig. 3.1].

The buffer size is denoted by B, then $0 \leq \beta(t) \leq B \; \forall t \in [0, N^*\tau]$. This notation is visually represented in Figure 5.3.

The LAMB-DASH algorithm is based on heuristic rules selecting the bitrate by addressing the current stream state:

- buffer level in seconds, $\beta(t)$,
- available bandwidth in Mbps, denoted as $\rho(t)$,
- and the frequency of the representation switches, $\Omega(t)$, which is defined as the ratio between the number of switches to higher bitrates and the number of downloaded segments.

Due to the fact that numerous switches affect the QoE, a configuration parameter, Ω^*, is used in order to limit the frequency switches, i.e. $\Omega(t) \leq \Omega^* \; \forall t \in [0, N^*\tau]$.

The decision for the representation j of the next segment i is performed during its request, then the values of all the above variables need to be known at request time t_i^r.

The QoE is affected by two factors, the switching frequency and the skipped frames. Therefore the probability of being downloaded before its deadline playback is evaluated for each representation, P_{ij}. Thus, it is important to note that a configuration parameter must be imposed in order to limit the probability of skipped segments, Σ^*, such that each representation whose probability of being downloaded out of time is too high is avoided.

4.2.3.3 **Online bitrate selection algorithm**

The final outcome of LAMB-DASH is to improve the video quality by selecting the representation bitrate that better fits with the status of the network and the player. The inputs of the algorithm are the network bandwidth, the playout buffer level, the segment duration and the configuration parameters Σ^* and Ω^*. The output is the representation index of the next segment. The decision program of LAMB-DASH is described in Algorithm 3.

Algorithm 3 LAMB-DASH algorithm

Σ^*, Ω^*, τ ▷ configuration parameters
t_i^r, t_{i-1}^r ▷ current and previous request time
$\rho(t_i^r)$ ▷ measured bandwidth at request time
$\beta(t_{i-1}^r)$ ▷ buffer level at previous request time
$\Omega(t_i^r)$ ▷ current value of relative quality transitions
j_{-1}^* ▷ representation of the last segment
if $i = 0$ **then**
  $\beta(t_0^r) = 0$
  $j^* = \max(j \mid R_j < \rho(t_0^r))$
else
  $\beta(t_i^r) = \beta(t_{i-1}^r) + \tau$ - $(t_i^r + t_{i-1}^r)$
  for $j = 0$ to N-1 **do**
    $P_{ij} = \text{function}(\beta(t_i^r), \rho(t_i^r), R_j)$
  $j^1 = \max(j \mid 1 - P_{ij} \leq \Sigma^*)$
  if $\Omega(t_i^r) \leq \Omega^*$ **then**
    $j^* = j^1$
  else
    $j^* = \min(j_{-1}^*, j^1)$
return j^*

The algorithm estimates the probabilities of each segment being correctly downloaded before its playback deadline. Since such probabilities are not available in the initial phase, the first segment is selected by estimating the initial bandwidth while downloading the Media Presentation Description (MPD) manifest. Here, the maximum bitrate that fits with the gauged bandwidth is selected. The selection of the maximum bitrate is an aggressive approach, but it helps to improve the overall perceived quality especially in cases of short duration video sequences, where a single segment has a high impact. On the contrary, a potential negative feature of such a decision is to allow

higher initial delay, because the client could need a higher buffering time in the initial phase [Rainer and Timmerer14].

From the second segment on, the algorithm has a characterization of the network bandwidth and it can evaluate the probabilities of correctly playing each representation. A segment *i* is correctly playable at representation *j* if its download finishes earlier than its playback deadline. The maximum admissible download time is equal to the buffer level, then the minimum download bitrate consists of the ratio between the segment size $s_{ij} = R_j * \tau$ and the buffer level. Here, R_j is the nominal bitrate of a representation *j*. Therefore, the probability of being correctly played can be written as:

$$P_{ij} = P[t_i^c - t_i^r \leq \beta(t_i^r)] = P\Big[\frac{R_j * \tau}{\beta(t_i^r)} \leq \bar{\rho}_i\Big] \tag{4.1}$$

The right expression is found by multiplying by $R_j * \tau$ and dividing by $\beta(ti^r) * (t_i^c - t_i^r)$. The value $\bar{\rho}_i = \frac{R_j * \tau}{t_i^c - t_i^r}$ is the actual average bitrate that the client will experience when downloading the segment *i*. However, this value is unknown until the download is completed. In order to solve this problem, the current measured value of the bitrate is used as a prediction for the future value of the bitrate; this strategy of approximation is taken from the alternatives explored in [Miller et al.16], as this results in a better performance. A relative prediction error is estimated by:

$$\epsilon_i = \frac{\hat{\rho}_i - \bar{\rho}_i}{\bar{\rho}_i} \tag{4.2}$$

where $\hat{\rho}_i$ and $\bar{\rho}_i$ represent the estimation and the real value of the average bitrate respectively. We can find $\bar{\rho}_i$ from the above equation and substitute it in the preceding one.

$$P_{ij} = P\Big[\frac{R_j * \tau}{\beta(t_i^r)} \leq \frac{\hat{\rho}_i}{1 + \epsilon_i}\Big] \tag{4.3}$$

ϵ_i is still an unknown value, but it is characterized by sampling instant measurements of bitrate and correlating the corresponding values of ϵ_i and their distribution. The notation can be simplified by noting that ϵ_i only depends on the network which is a stochastic environment, affected by the concurrent players competing for the available bandwidth. It follows that it is independent from the segment, then we simply use ϵ instead of ϵ_i.

Before explaining how to evaluate ϵ, it is important to note that the LAMB-DASH algorithm is based on estimating a Cumulative Distribution Function (CDF) of the relative error probability. This estimated CDF is achieved by means of two steps executed along the playback:

1. Acquire available bitrate samples and evaluate the Empirical Cumulative Distribution Function (ECDF) of the relative error probability ϵ.

2. Calculate the estimated CDF from the ECDF.

LAMB-DASH uses a heuristic approach, removing the need to perform a long processing stage to find an optimal CDF, which fits with the measured values. Thus, LAMB-DASH offers some design decisions from the implementation perspective:

- LAMB-DASH does not need *a priori* knowledge of the network condition.

- The evaluation of the ECDF is continuously executed using a few measurements taken during the stream session, while downloading the segments instead of at the start time. This way, no bandwidth overhead is introduced, since LAMB-DASH takes measurements from the data received from the stream.

- It is not necessary to fit a known reference CDF for all the session, but rather a piecewise linear approximation with a complexity $O(n)$, which provides an estimated CDF from the ECDF. This design saves heavy processing from the L2 distance minimization in Lolypop, with a complexity $O(n^2)$ [Pardalos93].

- The ECDF and the estimated CDF are periodically updated, as updated measures are loaded continuously. This mechanism makes the algorithm resilient to radical environment changes.

- Instead of having a significant initial computational overhead, it introduces a low computational overhead for the measurements during the stream session.

The differences across the CDF, ECDF and estimated CDF, are shown in the example in Figure 4.3. The goal is to characterize an unknown CDF curve to choose the appropriate bitrate representation accordingly. The ECDF is a step function sampling the

target CDF, which is assessed from samples of network performance measures. In order to approximate the CDF from the ECDF, there are two options. First, that employed by Lolypop, to minimize the distance of the ECDF curve to a set of known CDF curves. Second, that implemented in LAMB-DASH, to make a piecewise linear approximation of the ECDF. The ECDF is the function evaluated in the first step of both Lolypop and LAMB-DASH. While the selected CDF is the function that Lolypop evaluates at starting time and keeps unaltered during the session. This does not take part of the LAMB-DASH algorithm, which employs an estimated CDF that is periodically evaluated and updated. This approach makes LAMB-DASH able to provide a universal bitrate selection criteria for heterogeneous contents and changeable connectivity performance with a reduced computing overhead. This is achieved by means of removing the previous characterization stage to optimize the model to specific network conditions and content features in advance.

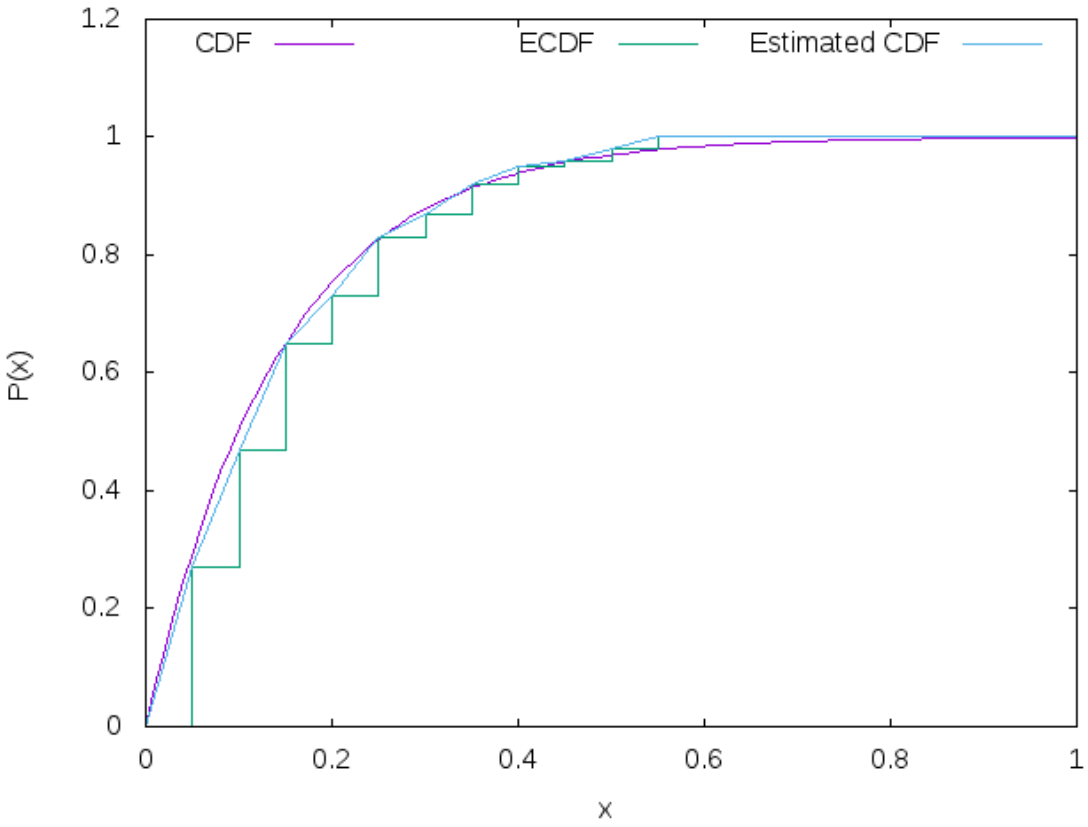


Figure 4.3: Example of CDF, ECDF ($\hat{F}_e(x)$) and estimated CDF ($F_e(x)$).

In order to explain the LAMB-DASH approach, it should be noted that the full range of values for ϵ is [-1;+∞), therefore, two cases should be distinguished:

- a negative value for ϵ, i.e. $\epsilon\in$[-1;0), means that the predicted bandwidth was an underestimation;

- a positive value for ϵ, i.e. $\epsilon\in$[0;+∞), means it was an overestimation.

Since they correspond to different situations, for each sub-range we construct a distribution function. In the following explanation we consider the case of overestimated values, in the same way as with the underestimation cases, by changing the measured values with their modulus.

We consider executing U measurements, i.e. we have U samples, during which we observe V distinct values for epsilon, ϵ_0, ϵ_1, ..., ϵ_{V-1} which have respectively q_0, q_1, ..., q_{V-1} occurrences. The sum of the occurrences is of course equal to U, the number of samples, i.e. $\sum_{n=0}^{V-1} q_n = U$. Then, for each ϵ_n we can define its probability: $P_n = \frac{q_n}{U}$. ECDF is then defined:

$$\hat{F}_\epsilon(x) = \frac{1}{U}\sum_{n=0}^{V-1} q_n * \mathbf{1}_{\epsilon_n \leq x} = \sum_{n=0}^{V-1} P_n * \mathbf{1}_{\epsilon_n \leq x} \tag{4.4}$$

where $\mathbf{1}_{\epsilon_n \leq x}$ is the unit step function which takes a value equal to one $\forall$n$\in$[0,V-1]:$\epsilon_n \leq$x.

Using the ECDF in (4), LAMB-DASH then constructs the estimated CDF though a piecewise linear approximation:

$$\begin{aligned} F_\epsilon(x) &= P(\epsilon \leq x) = \\ &= \sum_{n=0}^{V-1} P_n * \mathbf{1}_{\epsilon_n \leq x} + P_k \frac{x-\epsilon_{k-1}}{\epsilon_k - \epsilon_{k-1}}\Big|_{k=\min(m|\epsilon_m > x)} \end{aligned} \tag{4.5}$$

By joining the expressions (4.3) and (4.5), we find that the probability of the segment *i* being played without error at representation *j* is given by:

$$P_{ij} = P\Big[\frac{R_j * \tau}{\beta(t_i^r)} \leq \frac{\hat{\rho}_i}{1+\epsilon}\Big] = P\Big[\epsilon \leq \frac{\hat{\rho}_i * \beta(t_i^r)}{R_j * \tau} - 1\Big] = \\ = F\Big(\frac{\hat{\rho}_i * \beta(t_i^r)}{R_j * \tau} - 1\Big) \tag{4.6}$$

4.2.4 Implementation

4.2.4.1 DASH/HEVC services

In order to verify the proposed algorithm, we have deployed it in an environment where the adaptation logic in the clients will take action. To make the media content available to the clients, we use an Apache Server serving MPEG-DASH manifest and segments files (ISO / IEC 23009-1:2012). This way, they will be requested though HTTP GET.

In order to create the test sequences, we employ raw videos which are encoded in HEVC format (ISO / IEC 23008-2:2015). They are multiplexed in ISO MPEG4 files (ISO / IEC 14496-12 - MPEG-4 Part 12) and split into segments. The HEVC encoding and parsing capacity is already provided by Gstreamer[1] (*x265enc* and *265parse*). However, the current implementation (v1.12) does not support the configuration to introduce periodic or on-demand key frames and header information, as required to generate playable segments without inter-dependencies. Key frames are essential for HAS segments because they do not refer to other frames, i.e. it is always possible to start decoding from a key frame. In terms of header information, i.e. Sequence Parameter Set (SPS) and Picture Parameter Set (PPS), some fields are mandatory for playing the stream, as they provide basic parameters like the frame size. This way, the solution provided by Gstreamer only creates HAS contents to be played when starting from the first segment, because it is the only one that contains a key frame and header information. For the on-demand streaming mode, such limitation has no effect because the playback has to start from the beginning. On the contrary, in live streaming mode, the stream should start from the segment containing the current time. Thus, if the segment does not contain a key frame and headers, it is not possible to play it. This issue has been fixed by forcing the encoder to create a stream containing several key frames (at least

[1] Gstreamer website: `https://gstreamer.freedesktop.org`

one at the beginning of each segment) and sending header information each time that a key frame is encoded.

The encoded stream has to be multiplexed and split, but the official release of Gstreamer does not provide such operations at the moment (v1.12). It is possible thanks to the work of Thiago Santos who provides a multiplexer called *mp4dashmux*[2] and a file sink called *dashsink*[3] (v1.5). Both plugins are published under LGPL license conditions. The two plugins are highly related because they need to exchange information with each other, in order to properly create a manifest and segments. In the original release, they were not spanning all the possibilities considered in our experiments, and then two main improvements have been required:

- Extend support for HEVC because they are only meant for H.264/AVC.
- Add support for live streaming mode templates, as it is only able to generate on-demand streams.

Firstly, the encoded data is managed by *mp4dashmux* whose role is to recognize key frames and header information, previously inserted in the stream, and to use them to create consistent segments of a fixed duration. When the segment is ready, *mp4dashmux* sends a key unit event to *dashsink*. Then, *dashsink* writes the segment in the Apache server folder. Each time *dashsink* writes a segment, it returns a key unit event to *mp4dashmux* asking for a new one. The last function to be performed by *dashsink* is to recognize the content in the segments and periodically update the manifest.

On the client side, the main components for DASH playout are:

- The MPD parser, which receives and parses the XML-based media presentation description (MPD).
- The segment handler, which requests the segments for the selected representation, based on the decisions taken by the adaptation logic, and downloads them via an HTTP client.

[2]Git repository for *mp4dashmux* plugin: `https://cgit.freedesktop.org/~thiagoss/gst-plugins-good/?h=dashsink`

[3]Git repository for *dashsink* plugin: `https://cgit.freedesktop.org/~thiagoss/gst-plugins-bad/?h=dashsink`

- The adaptation logic, which decides the media representation that shall be selected for a given content, based on the network parameters, display characteristics and user preferences, in order to maximize the QoE.

The proposed online algorithm has been implemented in a Gstreamer client. Coincidentally, Gstreamer does not provide support for MPEG-DASH playback because there are no plugins that correctly parse on-demand and live manifests. Thiago Santos provides a plugin called *dashdemux*[4] (v1.9), which parses the manifest, provided by the source, and requests the segments for filling the buffer. The decision algorithm has been implemented inside *dashdemux*. To this end, we introduced a measurement process. This process runs in background and lets the algorithm discover the current state of the available bandwidth. Then the algorithm evaluates the distribution of the relative prediction error ϵ. Such measurement process is arranged to take samples every 200ms. This sampling ratio keeps the processing overhead low to avoid affecting the playout experience of the client device. The algorithm has to decide the next bitrate representation to immediately download a new segment, once the last downloaded segment starts playing. The algorithm is executed in order to evaluate the probability for each representation. Then, the algorithm chooses the bitrate representation that suits the measured bandwidth, the buffer level and the configuration parameters Σ^* and Ω^*, as already explained in Algorithm 3. Afterwards, in the *dashdemux* element, the playout buffer of the pipeline gets more seconds to reflect the new stored segment. As the segment is decoded and played, the buffer is drained by the following plugin in the pipeline of the player.

4.2.4.2 **QoE model**

From the work of De Vriendt et al. [Vriendt et al.13], we express our results in terms of MOS by means of this *QL model*. Our results are validated by following the conclusions of Claeys et al. [Claeys et al.14a], with a *QL model*, and Mok et al. [Mok et al.11], limiting the MOS evaluation to a set of objective metrics from the connection heuristics. The employed set of objective metrics perfectly fits the HAS environment, including

[4]Git repository for *dashdemux* plugin: `https://cgit.freedesktop.org/~thiagoss/gst-plugins-bad/?h=dashsink`

quality switches, frequency and duration of freezes. The final equation, as seen in [Claeys et al.14a, eq. (6)], is the following:

$$eMOS = \max(5.67 * \mu - 6.72 * \sigma - 4.95 * \phi + 0.17, 0) \tag{4.7}$$

In the equation, μ and σ are the normalized mean value and standard deviation of the QL assigned to the representations, respectively. So, they are inherently related to the quality switches. The values are calculated through the formulas presented in [Claeys et al.14a]:

$$\mu = \frac{\sum_{i=1}^{N} \frac{Q_i}{M}}{N} \tag{4.8}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} \left(\frac{Q_i}{M} - \mu\right)^2}{N-1}} \tag{4.9}$$

N and M represent the number of segments and the representations; while Q_i is the QL chosen for the segment *i*.

ϕ is the value that takes into account freeze events, since it compiles both duration and quantity, the resulting formula is presented in [Claeys et al.14a, eq. (5)]:

$$\phi = \frac{7 * \max\left(\frac{\ln(F_{freq})}{6} + 1, 0\right) + \left(\frac{\min(F_{avg}, 15)}{15}\right)}{8} \tag{4.10}$$

F_{freq} and F_{avg} represent the frequency of freezes events and the average duration.

Work from [Claeys et al.14a] concludes that the operational range of the estimated MOS is [0; 5.84], in contrast to the discrete scale from 1 to 5 of the theoretical MOS.

4.2.5 **Validation**

The total amount of time is set to 9 minutes and 50 seconds because it is the duration of the chosen video test (Big Buck Bunny). Its raw version is provided by Xiph.Org Foundation[5]. The chosen duration for each segment is fixed to 5 seconds, granting a balanced live delay and window time for successful segment download trade-off.

[5]Xiph.Org Foundation Video Test Media website: `http://media.xiph.org/video/derf`

The generated representations are useful for testing our algorithm on the client side. The considered networks and devices are translated into six representations for the generated content[6], as presented in Table 6.1. Here, the Group Of Pictures (GOP) size sets the number of frames between key frames.

Table 4.1: Set of MPEG-DASH representations employed in the experiments.

profile	bitrate	resolution	GOP size	framerate
3G	420kbps	288P	72frames	15fps
HDSPA	1000kbps	360P	90frames	30fps
LTE	1400kbps	432P	90frames	30fps
LO-Wi-Fi	2000kbps	480P	90frames	30fps
MID-Wi-Fi	2600kbps	576P	90frames	30fps
HI-Wi-Fi	3400kbps	720P	90frames	30fps

The algorithm is tested by setting the internal parameters Σ^* and Ω^* to 0.5 and 0.1 respectively. This means an error probability of less than 50% and a switching rate of less than $\frac{1}{10}$. Such values are chosen according to the results of Miller et al. [Miller et al.16], where such configuration provides the higher representation bitrate among all the different tests carried out.

The testbed is configured as in Figure 6.3. This way, the testbed can be easily setup and the networking conditions better controlled, thus avoiding interferences from other clients or networks typically present on cellular and Wi-Fi infrastructures. The download bandwidth limitation of 25Mbps would theoretically produce bottlenecks when 10 players try to access to highest bitrate option (3.4Mbps) listed in Table to specific network conditions and content features in advance.

In this environment, two different scenarios are presented:

- Scenario 1: the clients are synchronized to a common clock joining the live stream at once. This means clients are concurrently sharing common resources, as they are measuring the same available bandwidth value at once. This is shown on the left-hand panel;

[6]Encoding.com guide for HLS services: `https://www.encoding.com/http-live-streaming-hls`

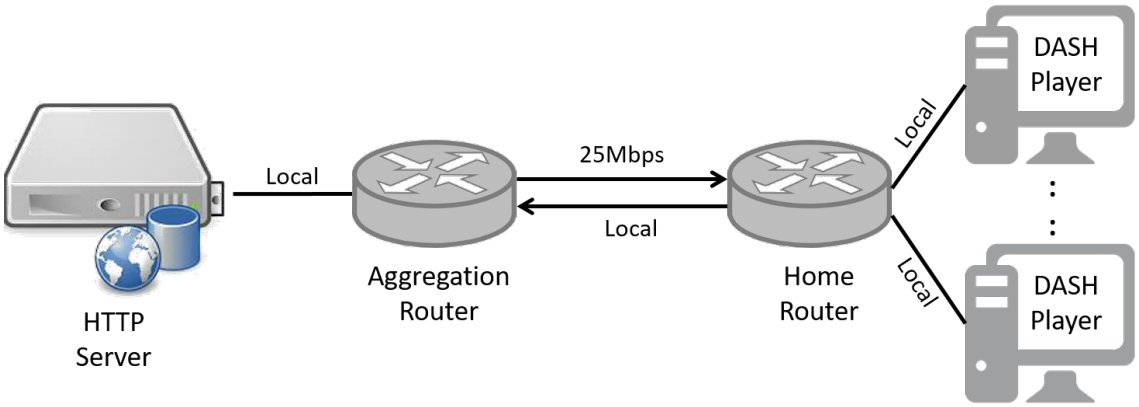


Figure 4.4: The network topology of the testbed. Local indicates that the bitrate is effectively unbounded and the link delay is 0 ms.

- Scenario 2: the clients are randomly joining the live stream. This means clients are measuring different bandwidth values, since they do not download at the same time, then they experience network bandwidth fluctuations. This is on the right-hand panel.

In scenario 1, the clock employed is based on Network Time Protocol (NTP). The clients employ the ability of Gstreamer to become synchronized to a NTP clock in order to synchronize the bootstrapping of the playout. The clock is no longer synchronized to follow the playback time afterwards.

Figure 4.5 and 5.6 show the behavior of the proposed algorithm executed on 10 competing clients that are sharing a wired network with an available bandwidth limited to 25Mbps.

Under the described conditions, the available bandwidth graphs show that, in the first scenario (Figure 4.5a), the clients tend to measure 2.5Mbps, which is the effective amount of bandwidth per client. Few peaks rise over 5Mbps due to extra available bandwidth when some clients are not accurately synchronized in their requests. In the second scenario (Figure 4.5b), the measured values span a range from 2.5 to 25Mbps. This goes from a fair utilization of the shared bandwidth to an unfair utilization, with players taking the total amount of bandwidth of the channel, as simultaneity is stochastic.

The above observation becomes more evident when comparing the measured available bandwidths and the selected representation bitrates, in terms of average value and deviation per client (Figure 4.5c and 4.5d). Again, it is clear, from the figures, that

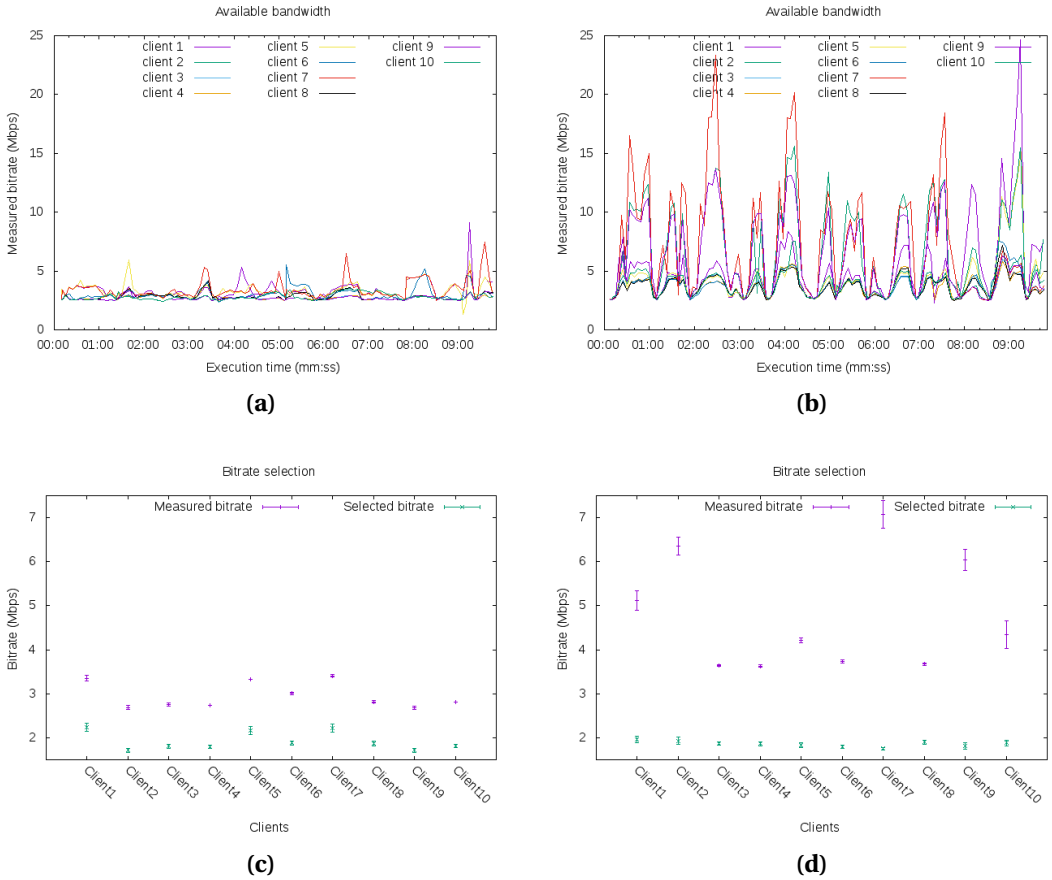


Figure 4.5: Ten clients sharing a 25Mbps down-link: scenario 1 for synchronous clients startup on **a** and **c** plots, and scenario 2 for stochastic clients startup on **b** and **d** plots. Plots **a** and **b** represent the available bandwidth over the execution time. Plots **c** and **d** compare the mean value and deviation of available measured bandwidths and the selected representation bitrates.

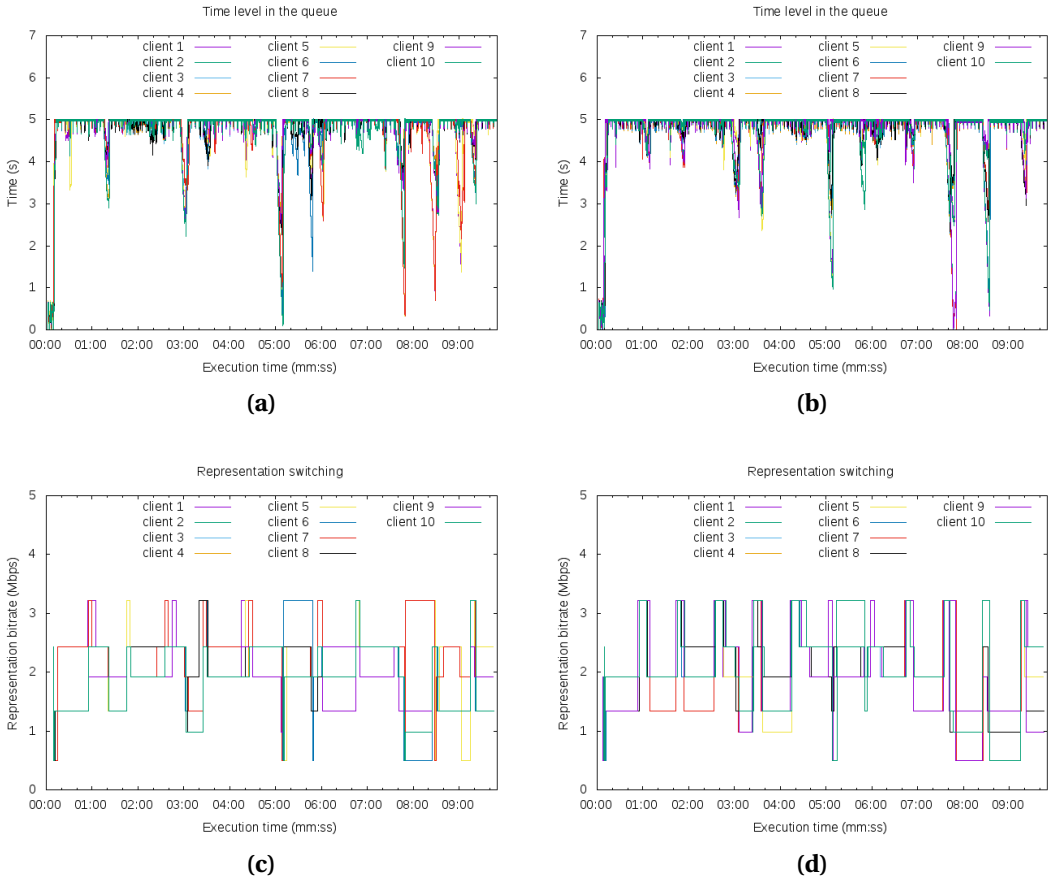


Figure 4.6: Ten clients sharing a 25Mb/s down-link: scenario 1 for synchronous clients startup on **a** and **c** plots, and scenario 2 for stochastic clients startup on **b** and **d** plots. Plots **a** and **b** show the playout buffer lengths. Plots **c** and **d** display the selected representation bitrates.

the second scenario is more variable because higher values for deviation are present. Moreover, in the first scenario (Figure 4.5c), clients tend to have an average selected representation bitrate that is proportional to the measured one. Here, there is an offset in favor of the measured one. On the contrary, the second scenario (Figure 4.5d) shows that a variable measured bitrate provides a worse behavior, as the average value of the selected representation bitrate does not follow the measured one.

Despite this different behavior, the curves of the playout buffer graphs (Figure 4.6a and 4.6b) and the selected representation bitrate (Figure 5.6a and 5.6b) look similar. In both scenarios the playout buffer level leans towards 5 seconds, which is the maximum amount of data queued. In our tests the buffer size has been defined to accommodate the duration of the segments. Therefore, sometimes the buffer level dramatically falls down and affects the playback with freezes. Such events occur when clients demands a bandwidth higher than the effective one. They switch to a representation with a higher bitrate which needs a higher download time causing buffer emptying.

Table 4.2: Number of switches (S_{Nb}), number of freezes (F_{Nb}) and average freeze duration (F_{avg}) evaluated for each scenario and client.

	Test 1			Test 2		
	S_{Nb}	F_{Nb}	F_{avg}[ms]	S_{Nb}	F_{Nb}	F_{avg}[ms]
client 1	27	3	57.1	26	2	55.7
client 2	23	3	59.2	26	3	53.8
client 3	22	3	79.5	29	3	55.7
client 4	23	3	49.0	28	3	59.4
client 5	29	3	69.5	27	2	52.5
client 6	25	3	68.4	29	3	61.4
client 7	30	2	52.4	27	3	51.5
client 8	22	2	52.5	28	3	46.1
client 9	23	3	57.5	27	2	41.0
client 10	24	3	56.4	30	2	52.6

With regard to the representation bitrate graphs (Figure 5.6a and 5.6b), the two scenarios encompass all the possible levels, as their choices span from the lowest representation bitrate, 420kbps, to the highest one, 3.4Mbps. We observe that the selection is effectively influenced by the estimated bandwidth and buffer level. In scenario 1, with a more stable bandwidth experienced, the algorithm reacts to buffer empty in a conservative mode by switching to a representation with lower bitrate. The aim is to get buffer

refill and avoid freezes. On the contrary, in scenario 2, with a stochastic measured bandwidth, a high peak in the measured bandwidth drives the algorithm towards greedy behavior. This means the algorithm switches to a higher bitrate in order to improve the quality. Such adaptability, at buffer and bandwidth level, is brought about by the live measurements allowing the algorithm to discover state changes. It means that the algorithm can be exploited in heterogeneous environments by tuning the conservative and greedy ratios using the internal parameters Σ^* and Ω^*.

Coincidentally, Table 4.2 points out the stability of scenario 1, in terms of segment quality switches, because the clients tend to change less than in scenario 2. The average value in the first case is 24.8, while in the second it is 27.8 for a total duration of 118 segments. Again, the result is definitively reasonable due to a less variable measurement of the available bandwidth.

Table 4.2 also shows the quantity and average duration of freezes. The behavior of the two cases is similar in terms of switches, since all the clients experience no more than 3 freezes with an average duration around 53ms.

Table 4.3: Average bitrate (R_{avg}) and eMOS evaluated for each scenario and client.

	Test 1		Test 2	
	R_{avg}[Mbps]	eMOS	R_{avg}[Mbps]	eMOS
client 1	2.24	2.77	1.97	2.57
client 2	1.71	1.94	1.93	2.01
client 3	1.81	2.14	1.87	2.11
client 4	1.80	2.02	1.86	2.09
client 5	2.17	2.63	1.83	2.36
client 6	1.89	2.12	1.80	1.99
client 7	2.23	2.74	1.75	1.87
client 8	1.86	2.24	1.90	2.15
client 9	1.72	1.94	1.82	2.12
client 10	1.81	1.99	1.88	2.38

The numerical results of the quality evaluation are presented in Table 5.6. As already explained, the evaluation has been done following the MOS model, because it gives us a human-like evaluation. MOS is evaluated for each scenario and client according to expression (7). In the scenario 1, the range for MOS spans from 1.94 (client 2) to 2.77 (client 1) with an average value of 2.25. While, in the scenario 2, MOS spans from 1.87 (client 7) to 2.57 (client 1) with an average value of 2.17. Such values correspond to a

variation of +3.7%, +7.7% and +4.1% respectively for the minimum, maximum and average value in favor of the scenario 1. Therefore this means that a situation where the bandwidth is equally distributed is favorable, since it provides the best overall quality.

In order to complete the evaluation, we also include the average bitrate in Table 5.6. We can observe from the resulting values how the algorithm is able to guarantee fairness in representation bitrate among the clients. In scenario 1, the average bitrate spans from 1.71 (client 2) to 2.24Mbps (client 1), then the difference is 0.53Mbps corresponding to +31% from the lowest to the highest. In scenario 2, the average bitrate ranges from 1.75 (client 7) to 1.97Mbps (client 1). Here, the difference is 0.22Mbps corresponding to +12.6%. So scenario 2 is not the best in terms of overall quality, but is fairer, because the variation between the lowest and highest average bitrates is smaller than scenario 1. The higher variability of the measured bandwidth during the playout time and among the clients provides fairness in representation selection. This is because, in scenario 2, the algorithm tends to under-utilize the network due to more frequent conflicts (Figure 4.5b) caused by erratic bandwidth assessment when autonomous clients compete for the available bandwidth. This lower average bitrate makes the operational range narrower.

Finally, it should be noted that scenario 1 represents a very singular case where the quality is improved by simply synchronizing all the clients for the initial HTTP requests. The strategy of employing a common clock for all the clients to constrain the discrete times to perform the first request is simple and obtains an improvement in quality (4%) and average bitrate (3.4%). This strategy results in a more accurate and stable characterization of the connectivity status (Figure 4.5a). The results of scenario 2 evaluate the performance of the algorithm when this synchronization is not possible.

4.2.6 **Conclusion and Future work**

In this work, we have presented a bitrate adaptation algorithm, named LAMB-DASH, whose aim is to maximize the video quality by means of a client-driven selection. LAMB-DASH allows the client to take the network conditions during the bitrate adaptation process into account, while still maintaining the ability to react to sudden bandwidth fluctuations in the local network.

LAMB-DASH is ahead of the existing solutions in two different aspects. First, it can be universally applied to different content types and changeable networking conditions. To this end, LAMB-DASH performs live assessment instead of preliminary processing for network featuring. Second, when considering the computational overhead over the video streaming playout, the required background computation is reduced when compared to heavier and less flexible alternative computing and optimizing models.

The algorithm has been implemented and validated on top of a GStreamer client and tested in a setup where multiple clients share the same path in the network therefore competing for the available bandwidth. Two different scenarios have been explored. Scenario 1 runs clients synchronized to a common clock joining the live stream at once. Scenario 2 arranges clients randomly joining the live stream. Here, they experience stochastic network bandwidth fluctuations.

The results of on both scenarios show that the algorithm achieves fairness, since the clients tend to the same representation bitrate. However, scenario 2 offers less quality that scenario 1, in which the average efficiency in terms of network utilization and quality experienced is higher. In scenario 1, a synchronized connectivity status assessment produces a more accurate and stable characterization. The strategy of employing a common clock for all the clients, to constrain the discrete times to perform the first request, is affordable and reliable with an out of band clock, maintaining the integrity of the DASH protocol.

Future work to LAMB-DASH algorithm will provide dynamic solutions while downloading a segment, in case of detection of sudden changes of network conditions, featuring a multi-pass reactive approach.

Acknowledgment

This work was supported by the European Commission project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action).

CHAPTER

5

MEC for Fair QoE and Reliable CDN

5.1 Context

The objective of the media services is to increase audience engagement and retention, where the QoE plays a significant role. Thus, the goal of the network for media services is to deliver a smooth and high-quality playback, with low video start times and high bitrates while reducing buffering.

Agile networks based on SDN technologies lacks scalability, as the number of clients and size of the infrastructure increase. Here, capillary Software-Defined Radio (SDR) systems, where the entire radio function is running on a general-propose processor, meet the scalability issues. Under this technology umbrella MEC is a foundational network architecture concept integrated on the mobile network infrastructure bringing new opportunities to improve the performance of HAS streams. MEC turns a base station into a service catalyser, which dynamically improves network performance and user experience for a specific service. Thus, a media service can exploit media delivery analytics from the MEC components to measure the speed and availability of different delivery paths over the Internet.

A solution for fair QoE in dense client environments and reliable CDN provision must take benefit of MEC position for exploiting edge video analytics in 5G networks. To create such a system, some aspects must be overcome. First, capture RAN awareness statistics. Then, a mechanism to control bitrate adaptation in a transparent manner with zero latency is needed. Moreover, the mechanisms can be applied to steer the CDN switching in response to performance degradation or provision outage. Finally, to check that the system can be operated, it is necessary to integrate with a real SDR setup which achieves a fair, steady and enforced QoE.

The solution described in Section 5.2 provides a novel solution based on a hybrid MEC and client adaptation for fair and efficient media streaming delivery in a mobile SDR network. The solution extends the role of the MEC component for QoE improvement by means of media delivery optimization. Our approach empowers the MEC with abilities to perform real-time updates in the manifest with the available qualities and CDN endpoints. Hence, our RAN-aware mechanism is transparent to the service provider and to media players enabling DRM/encryption support.

MEC system goals to improve the Quality Level (QL) chunk Mean Opinion Score (c-MOS). The employed QoE model limits the quality evaluation to a set of objective metrics from the connection heuristics, such as quality switches, frequency and duration of freezes. These parameters are the key metrics of HAS services.

To validate the results, the system has been integrated and validated into a real mobile SDR network performed on a real setup, not simulated, checking the feasibility and performance of an active component of the video delivery chain at the mobile edge. This includes a real Long-Term Evolution (LTE) RAN infrastructure of an operational Mobile Network stack, the radio base station (eNodeB) and the Evolved Packet Core (EPC).

5.2 **Hybrid MEC and Client Adaptation for Fair and Efficient Media Streaming in SDR Mobile Networks**

- **Title:** MEC for Fair, Reliable and Efficient Media Streaming in SDR Mobile Networks

- **Authors:** Angel Martin, Roberto Viola, Mikel Zorrilla, Julian Florez and Jon Montalbán
- **Journal:** IEEE Transactions on Network and Service Management
- **Publisher:** IEEE
- **Year:** (Submitted May 21, 2018)

Abstract: Radio access links, shared by users in wireless and mobile access networks, may turn into bottlenecks in cases of congestion, causing user experience to degrade. HTTP Adaptive Streaming (HAS) technology offers media players the possibility to dynamically select the most appropriate bitrate according to the connectivity performance. High dynamics of network performance in dense client cells can drive this client-driven approach to continuous re-buffering time and potential image freezes along with quality fluctuations damaging the overall Quality of Experience (QoE). Efficient and fair bandwidth utilization represents a core problem of current and future Packet Core and Radio Access Network (RAN) infrastructures. To address this issue we propose a hybrid Multi-access Edge Computing (MEC) and client-side quality adaptation mechanism. The MEC system limits transparently and dynamically the highest available quality for each player and the client-side mechanism governs individual player adaptation. This hybrid approach is designed for changeable connectivity performance to enhance the bitrate selection criteria of multiple clients sharing the available bandwidth in a common radio link. This paper presents a mechanism to improve the Quality Level (QL) chunk Mean Opinion Score (c-MOS) in a dense client cell. Furthermore, our solution is deployed and tested on top of a Software-defined Radio (SDR) 5G infrastructure. To this end, live and on-demand Dynamic Adaptive Streaming over HTTP (MPEG-DASH) streams are delivered, representing low-latency and ultra-broadband services. Results show that the hybrid system makes the media players tend to a common and high quality representation bitrate.

Keywords: content delivery network, fairness, multi-access edge computing, quality of experience, radio access network, software-defined radio.

5.2.1 Introduction

The evolution of mobile communication is leading an overall process towards agile networking with higher performance to meet increasing traffic demands. The volume of

video traffic over the Internet will reach 80% of the total Internet traffic by the end of 2019 [Inc17b]. Meantime, the growth of mobile devices as the entry point to services [Inc17a] is prominent. The change of Internet traffic in this sense makes the capacity of the networks even more critical to user experience.

Media services must be adapted differently to variations in radio network information. HTTP Adaptive Streaming (HAS) meets those multimedia services demands by supporting heterogeneous display setups, different user preferences and languages and changeable mobility situations with a Content Delivery Networks (CDN) ready design [Maillé and Schwartz16]. Moreover, HAS is a pull-based HTTP protocol [Begen et al.11] that easily traverses middleboxes, such as firewalls and Network Address Translation (NAT) devices. HAS enables players to switch dynamically between different media qualities tracking the variations in the network conditions during the media playback. Here, there is a trade-off between instantly offering the best video quality that will exploit the available connectivity resources, and minimizing quality fluctuations due to the risk of upsetting the user experience.

This client-driven approach, where control is distributed over the various media players and each one strives to optimize its individual quality, makes network edge and the Radio Access Network (RAN) highly dynamic. This makes dense client cells during live events (sport matches, concerts, etc) especially challenging. A media Content Provider (CP) will find it complex to ensure a level of quality to end-users that are massively accessing through the same access point and competing for the available bandwidth independently [Akhshabi et al.12]. Here, some issues, such as, initial buffering delay, temporal interruptions or pauses, and video resolution changes during a video transmission can damage the Quality of Experience (QoE), which is highly correlated to these features [Seufert et al.15].

Traffic shaping of HAS streams, when considering fairness, efficiency and quality, can reduce the number of stalls and quality switches for clients sharing a bottleneck link [Quinlan et al.15]. The telecommunication industry proposal is based on Multi-access Edge Computing (MEC) [ETSI17a]. MEC is a network architecture concept integrated on the mobile network infrastructure. MEC provides new opportunities to improve the performance of HAS, by moving Information Technology (IT) and cloud computing capabilities to the edge of the mobile network, closer to the user. Therefore, MEC can boost the delivery of content and applications to end users in 4G and 5G contexts.

Operators can expose their RAN Application Program Interface (API) to authorized third parties to provide them with radio network information in real-time. This technology enables operators to better adapt traffic to the prevailing radio conditions, optimize service quality and improve network efficiency. MEC turns a base station into a service catalyzer, which dynamically improves network performance and user experience for a specific service. The target features span ultra-low latency and round trip time (RTT), optimized bitrates, extra physical security and efficient caching. The decentralization of specific network functions to the edge of the network brings agility and adaptability, and context awareness. Hence, MEC opens the door for authorized third parties, such as CPs, to develop their own applications hosted on the MEC servers.

5G networks promise high-bandwidth, low latency, always-on, massive connectivity. International consortiums such as, the European Telecommunications Standards Institute (ETSI), and the International Telecoms Union (ITU) are the driving force behind the design of standard frameworks. The MEC concept has evolved to draw on Network Functions Virtualization (NFV) technologies to allow Virtual Network Functions (VNFs) to run on this distributed MEC platform. Furthermore, commoditization and virtualization of wireless networks are changing the economics of mobile networks to help Mobile Network Operators (MNOs) move from proprietary hardware vendors to virtualized software platforms through the abstraction of the execution environment. Software Defined Networking (SDN) is an architecture designed to enable more agile and cost-effective networks. SDN allows the dynamic reconfiguration of the network by taking a new approach to the network architecture. SDN enables centralization of network management for different entities within a cellular network. However, issues of scalability, as the number of clients and size of the infrastructure increase, are raised. Here, capillary Software-defined Radio (SDR) systems, where the entire radio function is running on a general-propose processor, meet the scalability issues. SDR systems bring wider possibilities to distributed mechanisms of traffic coordination in a radio link. This inserts MEC into a broader, more strategic discussion about network architecture evolution and distributed cloud in 5G.

The use case defined by ETSI for video analytics [ETSI17a] envisions MEC technology to guide the video server. There, the MEC system chooses the optimal bitrate given the radio conditions for a particular video stream or user. The idea is to use a RAN analytics application to determine/estimate the throughput likely to be available at the

radio downlink interface for a user, and then use packet headers to convey that information to the video server, so that it can adapt the stream accordingly. This way the streaming service achieves a noticeable performance improvement when operators communicate RAN conditions to the video server in this way.

5.2.1.1 **Contribution**

This paper provides a novel solution based on a hybrid MEC and client adaptation for fair and efficient media streaming delivery in a mobile SDR network. This solution has been achieved by providing three relevant contributions:

- A novel MEC component (MEC4FAIR) to perform real-time updates in the manifest with the available qualities. This vision empowers the role of MEC from ETSI for transparent QoE improvement.
- A combination of MEC4FAIR with a client-side algorithm as a novel hybrid MEC and client adaptation solution.
- Integration into a real mobile SDR network and validation performed on a real setup, not simulated, checking the feasibility and performance of an active component of the video delivery chain at the mobile edge.

This paper goes beyond the related work for fair and efficient utilization of a shared link among mobile users concurrently consuming media streaming services in the following aspects:

- Setup a real Long-Term Evolution (LTE) RAN infrastructure of an operational Mobile Network stack including the radio base station (eNodeB) and the Evolved Packet Core (EPC) [Liu et al.16b].
- The capillary of the MEC architecture makes the system highly scalable with a low response time.
- The exploitation of L2 (link), L3 (network) and L7 (application) information to support switching decisions on HAS quality.
- RAN-aware mechanism transparent to the service provider and to media players enabling DRM/encryption support.

- The fast convergence of the quality decision algorithm on the client-side.

5.2.1.2 Paper Structure

The paper is structured as follows. First, section 5.2.2 contains a review of the related work in terms of quality selection for media streaming services. Then, in section 5.2.3, we introduce a novel MEC component called MEC4FAIR to filter the quality representation on the HAS manifest as the main focus of the article and integrate it with a client adaptation algorithm as a hybrid solution. Section 5.2.4 describes the implemented testbed using an SDR platform, while section 5.2.5 presents the results of the validation experiments on the aforementioned testbed. Finally, we assert our conclusions and future work in section 5.2.6.

5.2.2 Related Work

5.2.2.1 Client-side adaptation

The QoE of streaming services relies on the experience derived from network stability, efficient utilization, and fairness. However, multiple clients (or sessions) competing for bandwidth across a bottleneck link can cause instability in the selected representation, link under-utilization, and disproportional shares of available bandwidth [Chen et al.16b]. Therefore, recent research in adaptive streaming is focusing on the development of such client-side adaptation algorithms. The client monitors some key indicators in order to make the decision of switching to a representation bitrate that better fits the current state and maximizes the playback quality.

Connection-based algorithms choose the representation bitrate taking into account server-client connection status (most common indicators are bandwidth and latency). Here, the *heuristic-based* algorithms take direct measurements and use decision rules based on the observations. These allow the most appropriate level to be dynamically requested, based on the current network conditions in multi-client scenarios [Petrangeli et al.15]. In order to track quick changes on networking conditions, the algorithm [Liu et al.11] explores step-wise increases and aggressive decreases of the adaptation algorithm in single-user scenarios. Whereas, the *optimization-based* algorithms perform mathematical modelling. They potentially generate a higher quality

playback than the *heuristic-based* ones, but they need a big dataset and a long learning time [Claeys et al.14b]. Some *heuristic-based* algorithms are Festive (Fair, Efficient, Stable, adaptIVE) [Jiang et al.14], Panda (Probe and Adapt) [Li et al.14b] Lolypop (Low-Latency Prediction-Based Adaptation) [Miller et al.16].

Content-based algorithms characterize the content, using Structural Similarity (SSIM), the human perception of the image, to adapt the representation bitrate accordingly [Chiariotti et al.16]. This *content-based* algorithm suffers from high implementation complexity and large overhead needing reduced power consumption and prolonged battery life [Chen et al.16a, Zorrilla et al.17].

More complex solutions [Li et al.14c] explore both, the status of the connection-player and the features of the video content. However, the issue related to processing overheads persists as the heuristic-based algorithms does not need previous characterization or training, gaining advantage when applied to previously unseen contexts and heterogeneous environments.

5.2.2.2 **QoE models**

Whatever the adopted solutions, the aim of each algorithm is to enhance the quality of the playback. A consolidated way to evaluate the QoE is the Mean Opinion Score (MOS), with five quality increasing levels (from 1 to 5) [ITU]. This type of testing leads to long evaluation times. Therefore, for practical reasons, many objective models for evaluating an estimated MOS (eMOS) have been studied in order to profile the subjective human perception of the quality.

The work [Vriendt et al.13] investigates the most common models in order to verify the fit of each model. In particular the models shown are: bitrate model, PSNR or SSIM based model, chunk-MOS based model and quality model. It concludes that chunk-MOS model is the optimal one. From here onwards this paper uses this quality model which is a specific configuration of the chunk-MOS based model.

Moreover, the works [Claeys et al.14a, Mok et al.11] conclude a *QL model* which limits the eMOS evaluation to a set of objective metrics from the connection heuristics, such as quality switches, frequency and duration of freezes. These parameters are the key metrics of HAS services.

Hence, the work [Claeys et al.14a] combine different parameters in the following equation:

$$eMOS = \max(5.67 * \mu - 6.72 * \sigma - 4.95 * \phi + 0.17, 0) \tag{5.1}$$

The parameter μ means of the normalized mean value of the quality level assigned to the selected representation:

$$\mu = \frac{\sum_{i=1}^{N} \frac{Q_i}{M}}{N} \tag{5.2}$$

where N and M represents the number of segments and the representations; while Q_i is the quality level chosen for the segment *i*.

The parameter σ means the standard deviation of the quality level assigned to the selected representation, complementing the assessment of the quality switches:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} \left(\frac{Q_i}{M} - \mu\right)^2}{N-1}} \tag{5.3}$$

Finally, the parameter ϕ means the freezes impact, formulated by:

$$\phi = \frac{7 * \max\left(\frac{\ln(F_{freq})}{6} + 1, 0\right) + \left(\frac{\min(F_{avg}, 15)}{15}\right)}{8} \tag{5.4}$$

where the frequency of freezes is represented by F_{freq} and their average duration by F_{avg}.

Recently, the work [Lentisco et al.17a] investigates a new model for MOS, called Ubiquitous-Mean Opinion Score for Video (U-vMOS), which makes initial buffering more relevant than [Claeys et al.14a].

5.2.2.3 **Network Management Function**

As described in subsection 5.2.2.1, it becomes complex to provide video services to several users autonomously competing for the available bandwidth. Therefore, a more coordinated approach for the users in a local radio link is needed, while maintaining scalability and response time to be able to capture metrics, process them and prevent QoE degradation situations in real-time. The solution must be transparent in different levels, from the media delivery protocol perspective, to be universally adopted, and

from the networking efficiency, to avoid overheads with extra messaging. This means that the network must participate.

Networks are migrating towards an agile, open and cost-effective traffic delivery system to dynamically adapt the resources to the traffic demands. Here, ETSI and ITU are defining a network architecture to provide greater flexibility to scale the actual performance in a more dynamic way and with finer granularity. Under the SDN umbrella, the NFV, VNF and SDR technologies enable the dynamic configuration, management and optimization of mobile networks based on changing traffic demands. Going further, MEC opens the MNO infrastructure to tune up a specific service or user through an exposed API. This way, authorized third parties can boost or enforce their own service in real-time through applications hosted on the MEC servers, which are on the edge close to the end users.

Fog and Mist Computing architectures [Chiang and Zhang16] define computing, storage, and networking resources provisioned in a cloud basis to host servers located at the edge of the network. Fog Computing is often related to the context of the Internet of Things (IoT), where host servers, routers, access points and computing assets are co-located with sensors and actuators. On the contrary, MEC is mainly exercised in the context of mobile networks, where host servers are integrated with the mobile network infrastructure.

Cloud RAN (C-RAN) [Checko et al.15] is another approach to empower RAN. C-RAN focuses on RAN functions commoditization and virtualization. In this case, RAN functionality is implemented in centralized data centre resources, instead of being distributed in the base stations. Centralized RAN brings easier software upgrade and higher performance by means of multi-cell coordination. However, this places RAN far from the user position, where zero delays are needed for our media scenario.

MEC steers a more efficient use of the network by exploiting information from different levels, such as L2, L3 and L7. The key is to make data actionable in order to setup stable and efficient resource utilization, avoiding situations where media players trend towards radio-link capacity exhaustion, before they reach full utilization. The relevance of metrics in order to make decisions to enhance media services is evident. There are platforms to monitor client experience and benchmark the performance of every CDN and service. Focused on IP Video performance, Cedexis [Cedexis17] and Conviva [Conviva17] platforms sustain networking decisions in a centralized manner,

via a cloud system highly coupled with the service provider and the player who has an agent to gather continuous quality telemetry, adding signalling overheads.

Scientific approaches to make data actionable in a coordinated way, with a network centric perspective, often consider SDN-enabled wireless networks [f. Lai et al.15]. Here, the centralized controller to manage the network is difficult to scale. Some schemes include in-network proxies [Petrangeli et al.15], proxy manager and resource controller at the eNodeB level [Rubin et al.15, Chang et al.15] to provide the clients with target quality suggestions. However, the bandwidth allocation scheme distributes the channel quality reports, which may significantly increase the signalling overhead. Other works [Vleeschauwer et al.13, Essaili et al.15] automatically and fairly adapt the video quality to react to congestion and data flow throughput starvation by overwriting client-side decisions. Therefore, they introduce limitations to track quick connectivity status changes.

MEC paradigm is the core of systems to improve HAS performance [Li et al.16]. This approach brings new features, such as close to zero delays and awareness of the radio status. Here, an HTTP proxy removes or adds back representations from the Media Presentation Description (MPD) manifest according to Channel Quality Indicators (CQI) reports avoiding signalling overheads. However, instead of prevention, it focuses on the reaction mechanism to fix the identified congestion situations. Moreover, its step-wise strategy, implemented by a scheme to gradually remove representations when congestion persists, brings slow convergence ability. Following the MEC vision, hybrid edge and client adaption solution for HAS media services is applied to cellular links with shared bandwidth [Yan et al.17]. This work goes a step further by considering the cumulative viewing experience, in order to tune the QoE continuum and fairness model, and two theoretical moving patterns. Here, the memory factor is another flavour of a mechanism that drives a slow convergence. An alternative approach [Chen and Liu16] targets continuity of the viewing experience and efficient resource allocation. This hybrid MEC and client-side mechanism, orchestrates time slots to make HTTP requests, different for each media player and serving rates. The required queues make this approach complex to scale for a big volume of User Equipment (UE). Further MEC component prioritizes or drops different HTTP transactions tailored to H.264 Scalable Video Coding (SVC) streams [Fajardo et al.15]. So this work employs L2 (CQI reports) and L7 (H.264/SVC hierarchical dependencies) to achieve QoE-driven fair scheduling of radio resources.

A common aspect, to all these scientific approaches, is that they are exercised and validated through simulations over LTE configurations, ignoring key RAN aspects, such as multiple frequency sub-carriers, time sub-frames structures and OFDM constellation symbols which isolate radio utilization among a volume of UEs [ETSI10]. Beyond that, they do not explore the integration of MEC systems into SDR technologies. The authors in [Wang et al.17] employ an OpenAirInterface (OAI) SDR system for the testbed, by means of a deep integration on the eNodeB operations building a MAC packet scheduler (L2).

A common aspect, to all these scientific approaches, is that they are exercised and validated through simulations over LTE configurations, ignoring key RAN aspects, such as multiple frequency sub-carriers, time sub-frames structures and constellation symbols which isolate radio utilization among a volume of UEs [ETSI10]. Beyond that, they do not explore the integration of MEC systems into SDR technologies. The authors in [Wang et al.17] employ an OpenAirInterface (OAI) SDR system for the testbed, by means of a deep integration on the eNodeB operations building a MAC packet scheduler (L2). Furthermore, other *Content-based* systems characterize the content, using Structural Similarity (SSIM) needing a-priori knowledge [Kourtis et al.17].

5.2.3 Hybrid MEC and Client Adaptation

5.2.3.1 Hybrid system architecture

To achieve a fair and efficient utilization of a shared link among mobile users concurrently consuming media streaming services, this work proposes a novel MEC component, called MEC4FAIR, to be deployed in the eNodeB. Going further, MEC4FAIR is highly suitable to be used jointly with any client-side adaptation algorithm such to achieve a hybrid client-side and MEC solution. The overall scenario of the hybrid solution is depicted in Figure 5.1.

HAS media players execute an application layer (L7) adaptation by switching the quality of the representation contained in the MPD. Thus, each player decision is managed by a proper client-side algorithm aiming to prevent playback degradation.

MEC4FAIR extends the video adaptation by considering data link layer (L2) metrics. MEC4FAIR exploits awareness of L2 cell statistics and CQI reports, probed in eNodeB, to

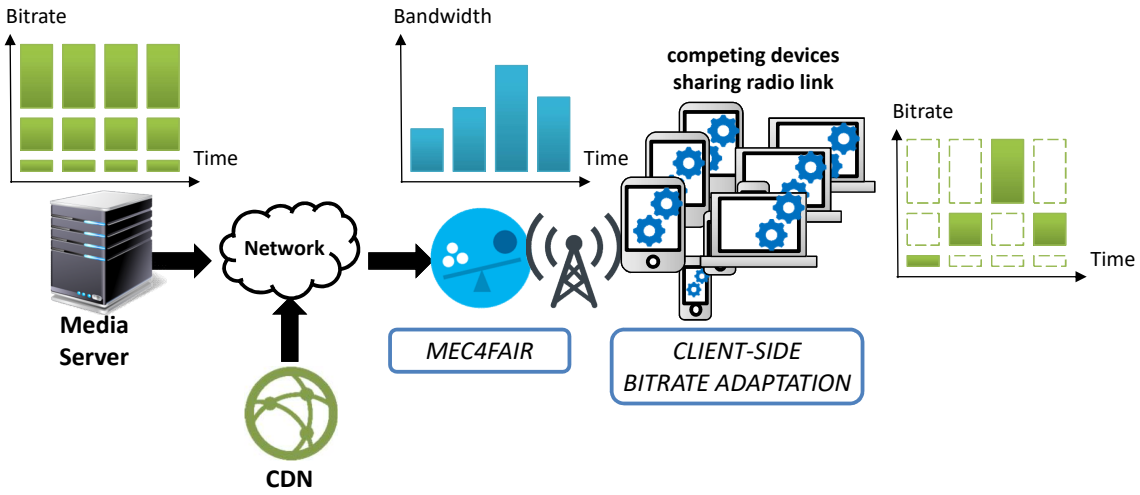


Figure 5.1: General scenario of the proposed solution.

quickly and dynamically control the video representations which are available for delivery. MEC4FAIR service, located at the mobile edge, operates in a transparent manner to service provider and clients. So the HAS principle is maintained, since the quality level selection can still be performed locally and independently by each client, to answer to quick network performance changes. Therefore, MEC4FAIR prevents QoE degradation with a fast convergence to fair and stable radio link utilization.

MEC4FAIR includes the following features:

- Scalable, by means of distributed and capillary nature of the MEC architecture.
- Transparent, without adding out of band signalling, exploiting operational radio reports.
- Encryption-friendly, compliant encrypted videos as MPEG-DASH Common Encryption Scheme (CENC) [ISO16] include encrypted segments, but the MPD is unencrypted. So, the MPD can be parsed and processed.

It is important to highlight that the wired path of the communication is usually more stable and has higher bandwidth than the wireless section. As a consequence its influence is relatively smaller when compared to the performance of wireless network. Therefore, for the representation bitrate adaptation, we focus only on the wireless part since it is the most restrictive and representative for dense client cells.

Although our contribution addresses a solution for LTE networks, this work is extensible to other wireless environments with an equivalent approach regarding the wireless access point.

5.2.3.2 **Hybrid solution workflow**

The sequence diagram with the exchanged messages is depicted in Figure 5.2. First, the MEC proxy server running MEC4FAIR detects HTTP GET request from the UE to download MPD files from the media server. Then, it retrieves the original MPD file from the media server. Once, the MEC proxy has the MPD, it appropriately filters the representations set available in the MPD manifest before it is sent to the UE. To this end, the MEC proxy assesses the effective maximum bitrate from the reports and divides it by the number of concurrent users for a fair utilization of the radio channel. All the representation bitrates exceeding the resulting fair value are dropped from the MPD. Such operations are executed at the stream start and each time that the client asks for a MPD manifest update. Then the UE selects a representation bitrate from the available ones and requests a specific segment file to the CDN, through the MEC proxy.

5.2.3.3 **MEC4FAIR rate adaptation**

A client-side decision algorithm is not sufficient for guaranteeing the best performance since each client is unaware of the presence of others. Client-side adaptation mechanisms take care of their internal state, then their decisions are just maximizing their playback, in particular they measure L3 values and select an L7 throughput accordingly. This client-side decision rule is missing the in-network knowledge. This measurement can be inaccurate compared to L2 measurements especially at the edge of mobile networks in dense client cells. Here, an MEC server located close to the eNodeB can retrieve L2 values for each client gaining an overall knowledge of the network. Hence, the MEC server can produce real-time data for influencing the HAS streams and providing a joint adaptation in a transparent way.

In order to better understand our MEC4FAIR solution, it is useful to overview some LTE fundamentals, such as Modulation and Coding Scheme (MCS) and Resource Block (RB). Table 5.1 shows the symbols that are used in the following discussion. Moreover, Figure 5.3 visualizes the relation among the symbols presented into the table. Notice

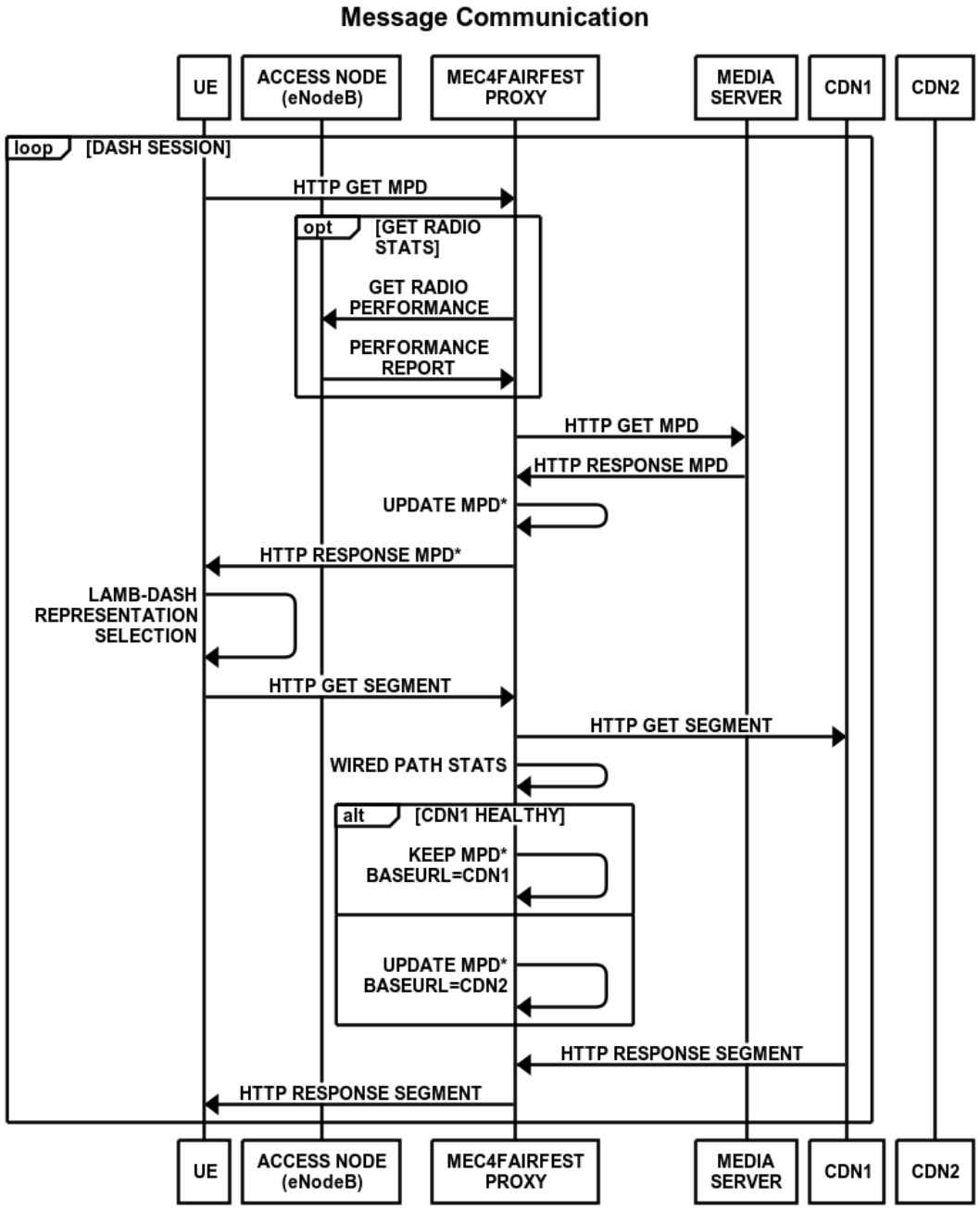


Figure 5.2: Sequence diagram of LAMB-DASH and MEC4FAIR for representation bitrate and CDN decision of a media player at an UE.

that some values are fixed by LTE standard while others depend on the connection state between the eNodeB and the UE.

Table 5.1: List of LTE Symbols used in the paper

Symbol	Description
RB	Resource block
RE	Resource element
N_{RB}	Number of resource blocks
N_{RE}	Number of resource elements
N_{sc}^{RB}	Number of subcarriers per resource block
T_{slot}	Slot time
N_{slot}	Number of slots
N_{symb}^{slot}	Number of symbols per slot
$N_{bit_{tot}}^{symb}$	Total amount of bits per symbol
$N_{bit_{inf}}^{symb}$	Information bits per symbol
CQI	Channel Quality Indicator
MCS	Modulation and Coding Scheme

Here, RB is the minimum temporal/frequency resource that eNodeB can allocate to a specific UE. RE is the minimum resource necessary for transmitting a symbol. Slot is a temporal subsection of an LTE frame, lasting one RB transmission time. CQI is an index ranging 16 levels reported by the UE to the eNodeB to characterise the quality of the communication channel (0 corresponds to unreliable communication, 15 corresponds to highly favourable channel performance). Finally, MCS is the modulation and coding scheme established by the eNodeB as a consequence of the CQI index. CQI mapping into MCS is depicted in Table 5.2. From the table it is also evident that the useful bitrate, i.e. the resulting bitrate by removing redundancy bits due to L2 coding, depends directly on the chosen MCS. There, more complex modulations and higher codes are assigned as CQI index increases.

As a consequence of the above LTE principles, the actual resources allocation mechanism operated by the eNodeB includes the following steps:

1. UE sends CQI to eNodeB
2. eNodeB decides MCS (function of CQI) and RB
3. eNodeB sends MCS and RB setup information to the UE

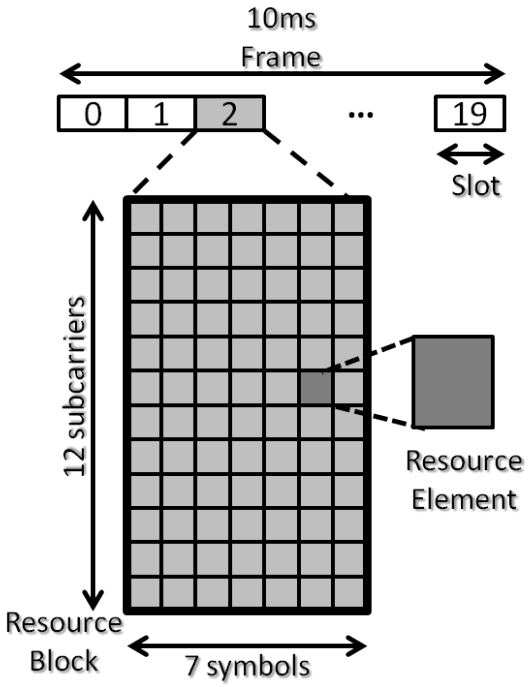


Figure 5.3: LTE resource grid.

Table 5.2: Relation between CQI and MCS [ETSI10, Tab. 7.2.3-1]

CQI index	modulation	code rate x 1024
0	out of range	
1	QPSK	78
2	QPSK	120
3	QPSK	193
4	QPSK	308
5	QPSK	449
6	QPSK	602
7	16QAM	378
8	16QAM	490
9	16QAM	616
10	64QAM	466
11	64QAM	567
12	64QAM	666
13	64QAM	772
14	64QAM	873
15	64QAM	948

The values related to MCS and RB are then processed in order to evaluate the maximum data bitrate, denoted as *PeakDataRate*:

$$PeakDataRate = \frac{N_{RE}(N_{RB}) * N_{bits_{inf}}^{symb}(MCS)}{N_{slot} * T_{slot}} \tag{5.5}$$

In the equation N_{slot} and T_{slot} are constants (2 and 0.5ms respectively) according to the LTE standard. N_{RE} and $N_{bits_{inf}}^{symb}$ depends on the chosen RB allocation and the MCS.

N_{RE} is calculated from the number of RB allocated (N_{RB}) by the following equation:

$$N_{RE} = N_{sc}^{RB} * N_{symb}^{slot} * N_{RB} * N_{slot} \tag{5.6}$$

Except N_{RB}, all the values are defined by the LTE standard. Thus, N_{sc}^{RB}, N_{symb}^{slot} and N_{slot} are 12, 7 and 2.

$N_{bits_{inf}}^{symb}$ is calculated from the MCS by the following equation:

$$N_{bits_{inf}}^{symb} = N_{bits_{tot}}^{symb} * CodeRate = \log_2 M * CodeRate \tag{5.7}$$

where M is the modulation cardinality which is 4 for QPSK, 16 for 16QAM and 64 for 64QAM. CodeRate is given in the Table 5.2.

The decision program of MEC4FAIR (UpdateMPD) is shown in Algorithm 4. UpdateMPD is executed each time a new MPD request is performed by any player. First, GetFairBitrate function assesses the real bandwidth available. Then, CropMPD function removes, for each player, the representation bitrates that could compromise the efficient and fair utilization of the radio link. The inputs of the GetFairBitrate function are the chosen MCS and the number of RB for a specific UE, and the current number of media playing sessions. The output is the highest representation to be used by a specific media player ($R_{max}^{pl_i}$). CropMPD function applies this representation threshold by cropping the MPD served by the media server (MPD_{proxy}).

In this sense, MEC4FAIR is not just a simple pass-through proxy. It has the capability to analyze the traffic, recognize HTTP requests for MPD files, understand their content and adapt it in order to fit with the current network state. Thanks to such MPD filtering, the decision rule of the client could be influenced (the decision rules inside the client are not affected, the HAS principle has to be preserved).

Algorithm 4 Generation of Fair MPD

procedure UPDATEMPD( ) ▷ listen to MPD requests & update MPD

for all MPD request **do** ▷ MPD request from the UEs

MPDrequest() ▷ to the media server

MCS ← eNodeB API

N_{RB} ← eNodeB API

N_{pl} ← eNodeB API ▷ number of media players

$R_{max}^{pl_i}$ = getFairBitrate(MCS, N_{RB}, N_{pl}) ▷ threshold

MPD_{server} ← MPDresponse() ▷ from the media server

MPD_{proxy} = cropMPD(MPD_{server}, $R_{max}^{pl_i}$) ▷ update

MPDresponse(MPD_{proxy}) ▷ MPD response to the UE

function GETFAIRBITRATE(MCS, N_{RB}, N_{pl}) ▷ for each MPD request

Input: MCS ▷ MCS chosen for transmitting to the UE

Input: N_{RB} ▷ number of RBs allocated for the UE

Input: N_{pl} ▷ number of players on the shared radio link

Input: R_j ▷ representations listed inside MPD_{server}

Output: $R_{max}^{pl_i}$ ▷ higher representation allowed for player i

PeakDataRate = $f(MCS, N_{RB})$ ▷ overall, equation (5.5)

PeakDataRate$_{pl_i}$ = $\frac{PeakDataRate}{N_{pl}}$ $\forall$ i=0,1...N_{pl}-1 ▷ player i

$R_{max}^{pl_i}$ = max(R_j | $R_{j\text{-}1} \leq$ PeakDataRate$_{pl_i}$) $\forall$ i=0,1...N_{pl}-1 ▷ higher representation considering eNodeB radio performance & concurrency level

function CROPMPD(MPD_{server}, $R_{max}^{pl_i}$) ▷ for each MPD response

Input: MPD_{server} ▷ biased MPD returned by the server

Input: $R_{max}^{pl_i}$ ▷ higher representation allowed for player i

Output: MPD_{proxy} ▷ unbiased MPD generated by the proxy

MPD_{proxy} = crop(MPD_{server}, $R_{max}^{pl_i}$) ▷ remove representations over $R_{max}^{pl_i}$

On the client-side, such a solution is possible since the MPEG-DASH [Sodagar11] specification guarantees the possibility to update the cached MPD inside the client. In particular it could be done in two ways [Li et al.16]:

- minimumUpdatePeriod field from the MPD, scheduling an MPD update after a number of setup seconds,
- EventStream signalling events in an MPD, mainly designed for advertising purposes [ISO12, ETSI15].

It is worth to note that the first method is only possible when working with a live playlist, i.e. when the content is played as it is generated, and then a playlist update is necessary. Whereas, on-demand playlists are static, so a compliant client will never update its cached MPD, in this case it is only possible to use the second one (EventStream).

5.2.3.4 **CDN performance broker**

Beyond this, dynamically switching in real-time from one CDN to another, can become a reality by using content delivery analytics from the MEC components. Thus, for services delivered over multiple CDN providers, MEC4FAIR approach would be also valid to select in real-time an appropriate CDN for a RAN geo-position according to L3 metrics. To this end, MEC4FAIR would get alternative CDNs to dynamically switch the base URL from other media sessions in the same RAN or from a set of preferred CDN providers from the media service. In case of detected performance degradation, the MEC4FAIR system would replace the base URL field of all the managed sessions to another known CDN endpoint, migrating all of them at once to avoid outages.

5.2.4 **Testbed setup**

To demonstrate the advantages of this hybrid approach in terms of QoE, we exploit NITOS facilities [Makris et al.15]. NITOS provides heterogeneous testbeds in order to execute experiments on a real wireless network. In our tests we use an indoor RF-isolated LTE network deployed at the University of Thessaly's campus building, which is provided with UE, eNodeB and EPC nodes in both commercial and open source implementation. In particular open source setup is based on Universal Software Radio Peripheral (USRP)

devices which can be managed through OAI [Nikaein et al.14] software. The role of OAI is to configure the USRP in order to provide an LTE-compliant network and to run the LTE stack protocol on top of USRP devices. In our case, we are interested in using an open source implementation of the eNodeB and EPC since it allows us to retrieve and manage radio network measurements which are needed by the MEC4FAIR algorithm. The experimental setup comprises:

- UE nodes. 10 Icarus nodes, that feature multiple wireless interfaces (Wi-Fi, WiMAX, LTE), placed in a symmetrical fashion around the isolated environment of NITOS indoor testbed forming a grid topology. The distance between the nodes is fixed at 1.2 meters and the height level is identical for all of them as well. These nodes execute DASH media players running LAMB-DASH, a client-side decision algorithm resulting from a previous work [Martin et al.17], for video rate control.
- eNodeB node. 1 USRP SDR system at the outer edge of the isolated room. This node performs eNodeB stack and retrieves radio performance reports.
- EPC node. 1 wired Icarus node close to the eNodeB. This node executes EPC stack.
- Generic nodes with Ethernet interface. 2 wired Icarus nodes that feature the MEC4FAIR proxy and the Media Server.

In terms of the testbed setup, MEC is a part of the eNodeB run on an external server that can be deployed between the radio base station and the mobile core. Thus, MEC4FAIR is located at the LTE RAN between the UE and the eNodeB. The eNodeB function consists of analyzing the link and continuously adapting the transmission by changing the modulation and coding scheme so that it fits with the current state and guarantees a reliable L2 data transfer. The eNodeB is connected to the EPC which manages mobile related activities at a higher level such as authentication, encryption and provides access to external IP networks. Then, all the traffic which passes through the eNodeB is forced to be processed by the EPC before being transmitted on any other network. Therefore, the eNodeB provides the LTE connection to the UEs which request the content stored on the Media server. This setup is depicted in Figure 5.4.

All the packets to/from the LTE network have to pass through the EPC. The eNodeB only provides L2 support. The main nodes and related function of the EPC are:

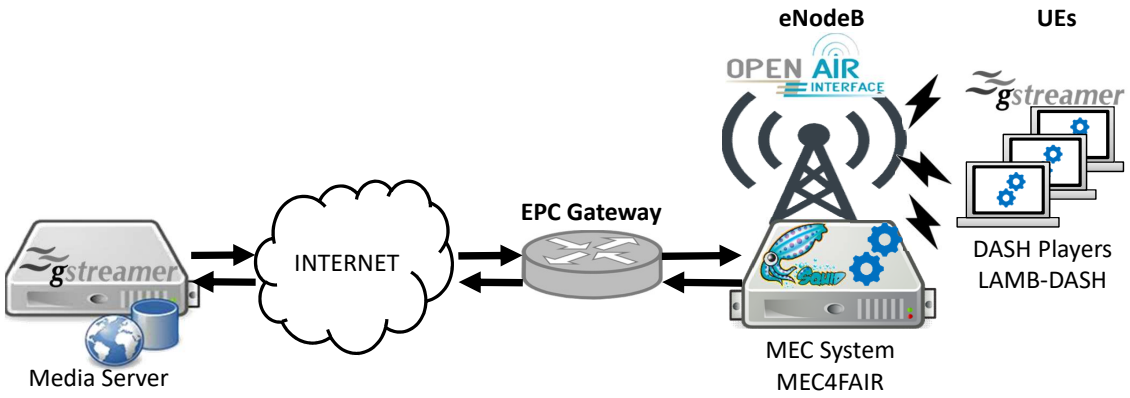


Figure 5.4: Hybrid MEC and client testbed.

- Home Subscriber Server (HSS): it is the subscriber database, it stores subscriber's IMSI (International Mobile Subscriber Identity) and provides supports for user authentication and access authorization
- Mobility Management Entity (MME): it deals with the control plane. It allows bearer activation for the communication after that the UE authentication is performed, then it manages intra-cell and inter-cell communication such that it is possible to track UE movements and guarantee continuity during the communication
- Serving Gateway (S-GW): it deals with the user plane, it manages all the traffic that LTE users send or receive in order to adapt it to the wireless environment
- Packet Data Network Gateway (PDN-GW): it is the gateway for inter-network communications, i.e. it allows intercommunication with external IP networks (called Packet Data Network).

The eNodeB configuration of LTE parameters is compiled in Table 5.3, employing the default setup from OAI. This configuration originates from a maximum theoretical bitrate of 23.3 Mbps (CQI index 15 [Ghosh et al.10]). This setup is not able to deliver media streams to 10 media players requesting premium quality (index 6 from Table 6.1 means 3.4Mbps per client). On the client-side LAMB-DASH performs L7 decision rules, without further requirements, then Linux-based devices with commercial LTE dongle are the UE nodes. Finally, we install an HTTP Apache Server on a generic node which

Table 5.3: LTE configuration

N_{RB}	N_{sc}^{RB}	N_{slot}	N_{symb}^{slot}	T_{slot}
25	12	2	7	0.5 ms

acts as Media Server by serving MPEG-DASH manifest and segments files (ISO/IEC 23009-1:2012).

The test sequence employed in our experiments is Big Buck Bunny with a duration of 9 min and 50 s. Its raw version is provided by Xiph.Org Foundation [Xiph.Org17]. Segment files are generated by encoding a test sequence in High Efficiency Video Coding (HEVC) format (ISO/IEC 23008-2:2015) [Sullivan et al.12] and multiplexing in ISO MPEG4 files (ISO/IEC 14496-12 - MPEG-4 Part 12). The chosen duration for each segment is fixed to 5 seconds, granting a balanced live delay and window time for successful segment download trade-off, resulting in 118 segments for each representation. Moreover, the test sequence is encoded into six different representations by considered networks and devices features[1]. Each representation is characterized by a particular video bitrate. The complete characterization of each representation is depicted in Table 6.1. Here, the group of pictures (GOP) size sets the number of frames between key frames.

Table 5.4: Set of MPEG-DASH representations for the tests.

index	profile	bitrate	resolution	GOP size	framerate
1	low	420kbps	288P	72frames	15fps
2	mid-low	1000kbps	360P	90frames	30fps
3	mid	1400kbps	432P	90frames	30fps
4	mid-high	2000kbps	480P	90frames	30fps
5	high	2600kbps	576P	90frames	30fps
6	premium	3400kbps	720P	90frames	30fps

The bitrate adaptation mechanism at the client-side does not target a specific resolution. The most used smartphone screen resolution is 1280x720 [Atlas18] which is the highest bitrate provided by the media server. Thus, all the UEs aim highest available representation bitrate and the tests focus on the dynamics of the network.

[1]Representations employed by Encoding commercial solution: `https://www.encoding.com/http-live-streaming-hls`

Furthermore, this paper also checks the feasibility and performance of an active component of the video delivery chain at the mobile edge. A significant result of the work described in this action is the implementation of the solution to perform the experiments and the tests on a real, rather than simulated, setup. MEC4FAIR is implemented and validated over SDR for network-assisted approach for adaptive HTTP streaming. Hence, this action also provides the evidence of the performance of a theoretical approach to deliver superior video quality while enabling transmission rate savings at the same time, in practice. Therefore, the complexity of integrating these mechanisms into mobile SDR networks is also evaluated.

5.2.4.1 **Candidate strategies**

In the target dense client environment, two different scenarios are presented:

- Synchronous start-up. The clients are synchronized to a common clock joining the live stream at once. This scenario resembles the start of a popular stream (e.g. sports live event). This means clients are concurrently sharing common resources, as they are measuring the same available bandwidth value at once.
- Stochastic start-up. The clients are randomly joining the stream. This scenario resembles the consumption of a popular stream (e.g. TV series). This means clients are measuring different bandwidth values, since they do not download at the same time, then they experience network bandwidth fluctuations.

In the synchronous scenario, the clock employed is based on network time protocol (NTP). The clients employ the ability to become synchronized to a NTP clock in order to synchronize the bootstrapping of the playout. The clock is no longer synchronized to follow the playback time afterwards.

The different candidates control quality switching smoothness and underperformed quality trade-off. We compare the selected bitrate and resulting QoE of the media players in different contexts where LAMB-DASH and MEC4FAIR mechanisms show benefits from a best-effort strategy, as will be shown in the next section. Best-effort strategy means individual players taking instant decisions based on the overall available bandwidth. The selection of the maximum available bitrate is an aggressive approach. It helps to improve the overall perceived quality but could need higher initial delay or

Table 5.5: Tested candidate strategies

Id	Description
sync	10 synch. clients on best-effort regime
async	10 asynch. clients on best-effort regime
sync_c	10 synch. clients with LAMB-DASH
async_c	10 asynch. clients with LAMB-DASH
sync^p	10 synch. clients with MEC4FAIR
async^p	10 asynch. clients with MEC4FAIR
sync_c^p	10 synch. clients with LAMB-DASH & MEC4FAIR
async_c^p	10 asynch. clients with LAMB-DASH & MEC4FAIR

playout freeze, because the needed higher buffering time. With the set of experiments done and compiled in the Table 5.5, this aims to identify the individual and combined contribution of each component and the convenience of the previous scenarios with regard the final result.

Going beyond, it is not possible to compare the real-time measurements of the candidate strategies with the related research, as most of the papers in this application domain employ simulations. The simulations range from the defined model, for the media players dynamics, to the LTE testbed [Li et al.14b, Miller et al.16, Chiariotti et al.16, Li et al.14c, Rubin et al.15, Essaili et al.15, Yan et al.17, Chen and Liu16].

5.2.5 Validation and Results

5.2.5.1 Performance metrics

We carried out a set of tests to validate the capability of the proposed MEC system to manage the efficiency and fairness trade-off. Efficiency assesses the network resources utilization. Thus, a higher average bitrate for all the media players sharing a radio link means a higher efficiency. While fairness refers to a more unbiased and homogeneous QoE across all the media players sharing a radio link. Hence, a lower eMOS deviation enhances the fairness.

As employed in the evaluation of LAMB-DASH work [Martin et al.17], we express our QoE results in terms of eMOS by means of the *QL model*.

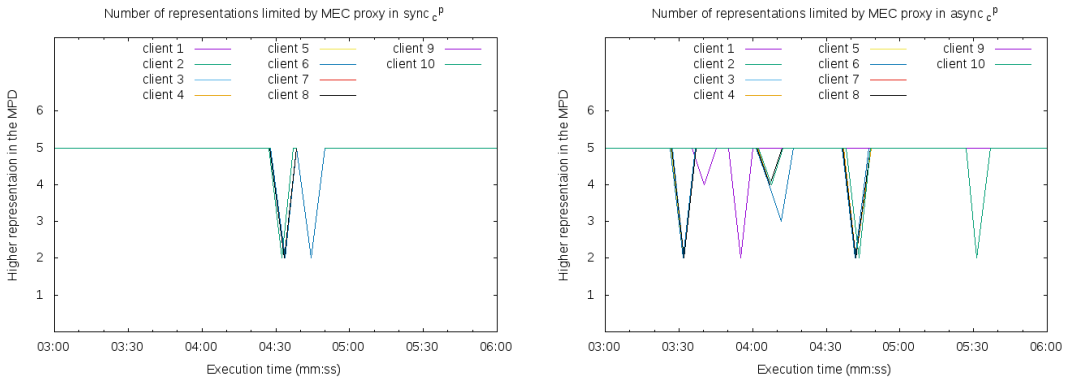
Therefore, we evaluate our hybrid solution per client along the video sequence in terms of:

- number of quality switches (S_{Nb})
- quantity (F_{Nb}) and average duration (F_{avg}) of freezes
- average bitrate (R_{avg})
- eMOS

Work from [Claeys et al.14a] concludes that the operational range of the eMOS is [0; 5.84].

5.2.5.2 **Results**

LAMB-DASH work [Martin et al.17] concluded a more fair radio link utilization on the synchronous scenario, as the estimation of the available bandwidth is more accurate. First of all, we analyze if this conclusion persists once MEC4FAIR comes into play. Figure 5.5 and 5.6 show the behaviour of the proposed hybrid solution executed on 10 competing clients that are sharing an eNodeB network.



(a) **(b)**

Figure 5.5: Ten clients sharing a radio link: scenario for synchronous clients start-up on **a** plot, and scenario for stochastic clients start-up on **b** plot. Plots **a** and **b** show the limitations applied by the MEC4FAIR proxy.

Under the described conditions, the stochastic scenario tends to unfair radio utilization, the greedy behaviour from asynchronous players produces more variable radio performance reports.

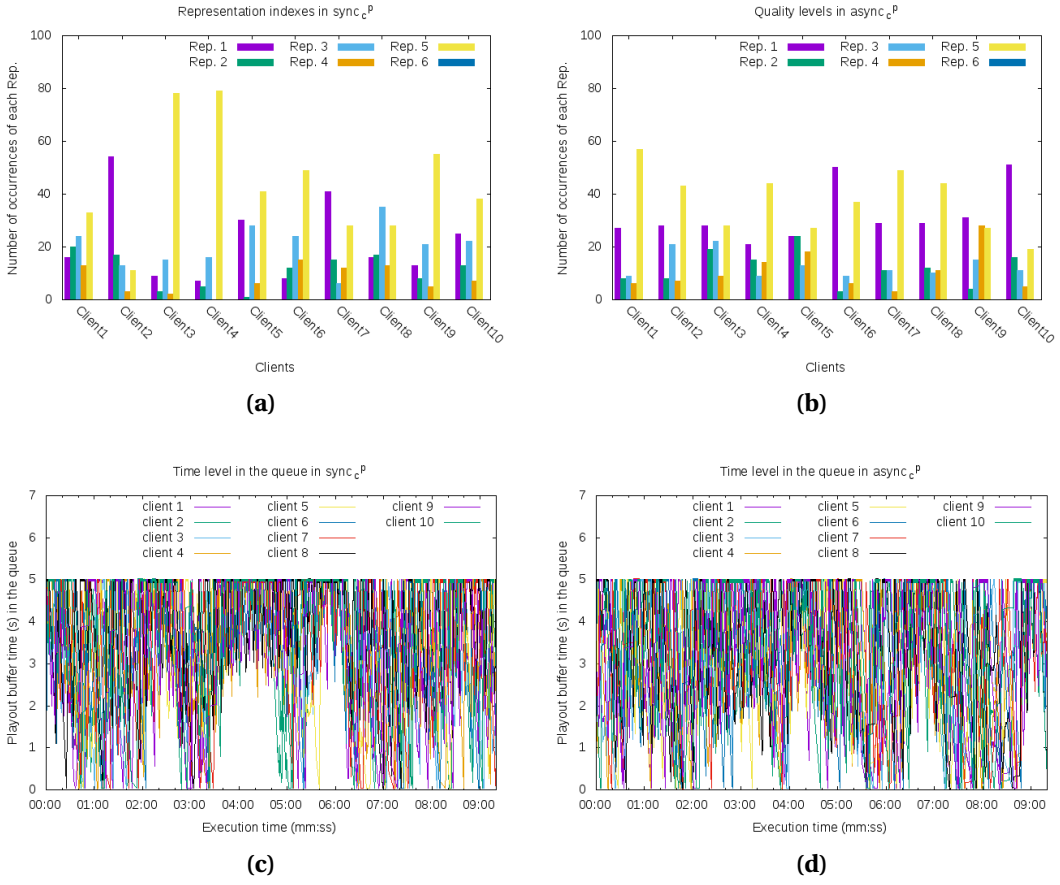


Figure 5.6: Ten clients sharing a radio link: scenario for synchronous clients start-up on **a** and **c** plots, and scenario for stochastic clients start-up on **b** and **d** plots. Plots **a** and **b** display the histogram of the selected representation bitrate for each client. Plots **c** and **d** show the playout buffer lengths.

The LTE setup provides a maximum theoretical bitrate of 23.3 Mbps (for CQI index 15). Hence, the eNodeB is not able to deliver 10 media streams with the highest bitrate (3.4Mbps). Accordingly, in Figures 5.5a and 5.5b, the quality 6 is always dropped from the MPD. In order to limit the intention from asynchronous media players to get a higher bitrate than the effective one the MEC4FAIR proxy has to crop more sharply and frequently the representations available in the MPD (Figure 5.5b). In the synchronous scenario (Figure 5.5a) the MEC4FAIR proxy is not modifying the MPD as frequently as in stochastic scenario, because the simultaneous bandwidth estimation done by LAMB-DASH is more accurate. This means MEC4FAIR will incorporate to stochastic scenarios the environmental concurrence parameters that synchronous mechanisms get independently.

Concerning the histogram of the selected representation bitrates, this is depicted for each client. The dominant utilization of one or two representation bitrates is evident in the synchronous scenario (Figure 5.6a). On the contrary, stochastic scenario (Figure 5.6b) behaves more stochastically with less concentrated representation selections. This means more fluctuation between representations.

Figure 5.6c and 5.6d show the curves of the playout buffer level. In both scenarios the level leans towards 5 seconds, which is the maximum amount of data queued. In our tests the buffer size has been defined to accommodate the duration of the segments. Therefore, everytime the buffer level falls to zero for a time over the duration of one frame, this affects the playback with freezes. Such events occur when clients switch to a representation with a higher bitrate which needs a higher download time causing buffer emptying. As a consequence of buffer emptying, freezes affect the playback. We can see that in the synchronous and the stochastic scenarios the buffer depletion behaviour seems quite similar. Table 5.6 shows that in both scenarios all the clients experience no more than 2 freezes, as the buffers get empty but not for enough time to be perceptible. Going further, the buffer level for synchronous scenario (Figure 5.6c) seems to be more steady, at some moments all the clients tend to have more than 50% of the buffer filled, which hardly ever happens in the stochastic case (Figure 5.6d).

Coincidentally, Table 5.6 also points out the stability of both scenarios in terms of segment quality switches. In the synchronous case the mean values of switches is 25.3 over a total of 118 played segments, while in the stochastic there is a small rise

to 27.1. In the stochastic scenario, MEC4FAIR proxy compensates the higher variable measurement of the available bandwidth exploiting RAN concurrency awareness.

The individual results for each client of the quality evaluation are presented in Table 5.6. As previously explained, the evaluation has been done following the MOS scale, because it gives us a human-like evaluation. The QoE parameter, eMOS, is evaluated for each scenario and client [Claeys et al.14a]. In both scenarios, the range for eMOS spans similar values. Such values correspond to a variation of -3.7%, +25.9% and +9.9% respectively for the minimum, maximum and average value in favour of the synchronous scenario. Therefore this means that a situation where the bandwidth is equally distributed is favourable, since it provides the best overall quality.

Table 5.6: Number of switches (S_{Nb}), number of freezes (F_{Nb}), average freeze duration (F_{avg}), average bitrate (R_{avg}) and eMOS evaluated for each scenario and client.

client	Scenario Synch. $sync_c^p$					Scenario Stoch. $async_c^p$				
	S_{Nb}	F_{Nb}	F_{avg} (ms)	R_{avg} (Mbps)	eMOS	S_{Nb}	F_{Nb}	F_{avg} (ms)	R_{avg} (Mbps)	eMOS
1	22	0	0.0	1.56	3.22	35	1	136.0	1.77	2.47
2	32	1	65.0	0.94	1.04	27	0	0.0	1.64	3.07
3	26	0	0.0	2.09	4.08	26	0	0.0	1.37	2.71
4	22	2	86.5	2.04	2.48	29	0	0.0	1.71	3.24
5	26	1	68.0	1.53	2.33	23	0	0.0	1.44	2.79
6	27	0	0.0	1.82	3.66	25	1	41.0	1.39	1.77
7	23	2	146.0	1.27	1.16	25	1	56.0	1.66	2.22
8	21	0	0.0	1.51	3.05	24	2	49.0	1.62	1.84
9	25	1	124.0	1.81	2.73	29	0	0.0	1.48	2.91
10	29	0	0.0	1.50	2.99	28	0	44.0	1.08	1.28

The overall average and deviation values of Table 5.6 are shown in Table 5.7 ($sync_c^p$ row) and Table 5.8 ($async_c^p$ row). Here, a lower deviation of eMOS ($eMOS_{dev}$) means a more fair QoE across the media players. It is evident that the hybrid solution gets a significant fair result in the synchronous scenario (0.11 for $sync_c^p$) compared to the stochastic one (0.38 for $async_c^p$).

In order to complete the evaluation, in terms of fairness and efficiency, we also compare the result of $sync_c^p$ with the other synchronous candidate strategies detailed in

Table 5.5, compiled in Table 5.7. We can observe from the resulting values how the different components of the hybrid solution contribute to guarantee: efficiency of radio link utilization represented from the the bitrate average (R_{avg}); and fairness among all the clients from the eMOS deviation ($eMOS_{dev}$). The best-effort approach (*sync*) takes as much bandwidth as possible ignoring the balance required to enhance the QoE ($eMOS_{avg}$) and fairness ($eMOS_{dev}$). We can see that the hybrid solution ($sync_c^p$) causes the radio utilization rate to fall to -13.4% while the fairness rate is improved to +85.5% in the best-effort strategy (*sync*).

Table 5.7: Bitrate average and deviation, and eMOS average and deviation, evaluated for all the clients in the synchronous candidate strategies from Table 5.5.

candidate Id	R_{avg}(Mbps)	R_{dev}(Mbps)	eMOS$_{avg}$	eMOS$_{dev}$
sync$_c^p$	1.61	0.03	2.67	0.11
sync$_c$	1.70	0.03	2.21	0.25
sync^p	1.75	0.02	3.34	0.10
sync	1.86	0.27	3.14	0.76

Table 5.8: Average and deviation of bitrate (R_{avg}), and average and deviation of eMOS evaluated for all the clients in the stochastic candidate strategies from Table 5.5.

candidate Id	R_{avg}(Mbps)	R_{dev}(Mbps)	eMOS$_{avg}$	eMOS$_{dev}$
async$_c^p$	1.52	0.14	2.43	0.38
async$_c$	1.65	0.18	2.02	0.67
async^p	1.66	0.05	3.13	0.37
async	1.84	0.09	3.28	0.54

From Table 5.7, the use of LAMB-DASH algorithm ($sync_c$ and $sync_c^p$), in a real setup, contribute to fairness from the best-effort test (*sync*). LAMB-DASH gets a better fair behaviour since the deviation of the eMOS from the mean value is much more steady, the deviation reduction is about +67.1%($sync_c$) and +85.5%($sync_c^p$).

The best results, from Table 5.7, are provided by using MEC4FAIR without LAMB-DASH ($sync^p$) since it increases eMOS by +6.4% compared to best-effort case (*sync*), while the average bitrate is reduced by -5.9%. The advantage in terms of fairness becomes more evident since $sync^p$ case shows lower variability than *sync* (they have an

eMOS deviation of 0.10 and 0.76 respectively). Furthermore, the joint use of LAMB-DASH and MEC4FAIR (sync$_c^p$) is better than just using LAMB-DASH ($sync_c$), as the eMOS average and deviation are enhanced by +20.8% and +56.0%.

Furthermore, hybrid solution ($sync_c^p$) results compared to MEC4FAIR ($sync^p$) are not significantly different in terms of fairness (0.11 and 0.10 respectively). LAMB-DASH still makes the difference to get fairness in those RAN environments where MEC deployment is not feasible (no SDR eNodeB) or affordable (no contracted MEC service on an MNO).

Finally, the synchronous scenario gets more fairness scores (eMOS$_{dev}$) than the stochastic one, by means of comparing Table 5.7 to Table 5.8. Moreover, the application of the hybrid solution to the stochastic scenario ($async_c^p$) also improves the results from the best-effort strategy (*async*).

To sum up, the LAMB-DASH algorithm reacts to buffer emptying by switching to a representation with a lower bitrate. The aim is to get buffer refill and avoid freezes. MEC4FAIR prevents greedy behaviour in a conservative manner by cropping higher bitrate representations for concurrent media players sharing a radio link. LAMB-DASH switches to a higher bitrate in order to improve the quality while MEC4FAIR restores higher representation bitrates when concurrency gets lower. Such adaptability, at buffer and bandwidth level, is brought about by live measurement allowing the algorithms to discover state changes. It means that the hybrid solution can be exploited in heterogeneous environments.

Finally, it should be noted that the MEC4FAIR proxy improved the quality by exploiting effective RAN utilization awareness, granting unsynchronized media players a similar performance to synchronized ones for the initial HTTP requests, when this synchronization is not possible.

5.2.6 Conclusions and Future Work

The objective of the media CPs is to increase audience engagement and retention, where the QoE plays a significant role. Thus, the goal of the network for media services is to deliver a smooth and high quality playback, with low video start times and high bitrates while reducing buffering.

Targeting this goal, we introduce in this paper a hybrid solution for fair and efficient utilization of a radio link in the target scenario, dense client environments. MEC4FAIR, a novel MEC component introduced by the paper, provides RAN awareness in real-time for influencing the HAS streams and providing a joint adaptation in a transparent manner. This entity also includes the reaction to CDN outages or performance degradation by switching to an alternative CDN provider. On the client-side, LAMB-DASH maintains the ability of media players to react to sudden bandwidth fluctuations in the local network.

This approach is ahead of the existing solutions in three key aspects. First, it is transparent to media players and media server compliant with MPEG-DASH and CENC encryption specifications. Second, it exploits MEC architecture by means of a proxy, located at the mobile edge, to operate in a scalable manner, with zero latency and no signal overheads. To this end, MEC4FAIR exploits awareness of L2 cell statistics and CQI reports, probed in eNodeB, to quickly and dynamically control the video representations. Last but not least, the validation is carried out on a real SDR infrastructure including the RAN entities eNodeB and EPC.

The algorithm has been implemented and validated on a real SDR LTE setup where multiple clients share the same path in the network, therefore competing for the available bandwidth. Two different scenarios have been explored. The synchronous scenario runs clients synchronized to a common clock joining the live stream at once. The stochastic scenario arranges clients randomly joining an on-demand stream. Here, they experience stochastic network bandwidth fluctuations.

The results of both scenarios show that the hybrid solution balances the efficiency and fairness trade-off. Here, an enhanced efficiency means high average bitrate while improved fairness means low deviation of eMOS across all the media players sharing a radio link. The hybrid system makes the media players tend to a common and high quality representation bitrate. Moreover, in the stochastic scenario, MEC4FAIR plays a significant role to improve efficiency, in terms of network utilization and quality experienced. Furthermore, the synchronous scenario introduces a more accurate and stable characterization causing the hybrid solution to obtain better scores than the stochastic one. Finally, from the comparison of individual components of the hybrid solution to the best-effort strategy, MEC4FAIR performs better without LAMB-DASH. However,

LAMB-DASH is easily plugged into media players making it convenient in those areas where an MEC service cannot be deployed at the MNO infrastructure.

Future work on this hybrid solution will expand the MEC4FAIR proxy with L3 path performance to decide the CDN base URL, from the ones available for delivery. Therefore, MEC4FAIR prevents QoE degradation and service outages from unhealthy CDNs as well as fair and stable radio link utilization.

Acknowledgment

This work was fully supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action).

CHAPTER

6

Network Resource Allocator

6.1 Context

5G represents the next generation of communication networks and services. Reaching formidable levels of complexity and traffic volume 5G networks brings a new set of challenges for managing the network. Thus, it will be necessary for the network to largely manage itself and deal with organization, configuration, security, and optimization issues.

The evolution of mobile communication has started an overall process towards agile networking with higher performance to meet increasing traffic demands. The change of Internet traffic on this wise makes the capacity of the networks more critical to the user experience. To this end, 5G solutions will address network traffic and resource management challenges. Consequently, novel techniques and strategies are required to address these challenges in a smarter way.

Machine learning ability to learn from historical data, make predictions, dynamically adapt to new situations learning from new data and conduct decisions [Mohri et al.12] can yield insights, detect meaningful events and conditions and enable the management system to respond correctly to them. Machine learning algorithms along with SDN and NFV brings potential to forecast resource demand and to react appropriately. Combining SDN, NFV and machine learning technologies, a centralized network controller

could change the network topology instantiating or removing Virtual Network Functions (VNF) to forward the incoming traffic in an efficient way, removing the unused parts of a network to release these resources [Ismail et al.13].

Accurate service demand prediction and provisioning represents a challenge for virtualised environments. This should allow the network to resize and provision itself, using virtualization, to serve predicted demand according to parameters such as location, time and specific service demand from specific users or user groups. Service demand prediction has the potential to offer an effective solution to such issues, particularly considering the need to preempt and anticipate the amount of network resources that need to be allocated.

A network resource allocator for self-organising networking must take benefit of SDN and VNF technologies in 5G networks. To create such a system, some aspects must be overcome. First, forecast the incoming traffic demands. Then, a mechanism to find the optimal topology for media delivery while assuring a QoE for the incoming traffic demands is needed. Finally, to check that the system can be operated, it is necessary to integrate with representative SDN and VNF frameworks to proactively and dynamically provision the network.

In this line, this section presents a solution for autonomic self-organising network which is capable of achieving or balancing objectives such as high QoS, low energy usage and operational efficiency.

First, Section 6.2 propose a solution where an scalable, real-time and autonomous network management system makes demand prediction to foresee the amount of network resources to be allocated to cope with the traffic demand, and dynamically provisions the network in a proactive way, while keeping network operation inside business ranges. This system is able to scale the network topologies and to address the levels of resource optimization, required for media streaming services.

The implemented experiment in Section 6.2 shows the viability to integrate machine learning methods in a SDN controller to forecast resource demand and to react appropriately, so that this one can learn to instantiate the most efficient network topology in terms of KPIs. The learning can be done based on experience gathered in previous measurements. Thus, the proposed Network Resource Allocator system is a reliable solution that addresses the problems for flexible creation an elastic network in an automated way.

Second, Section A.3 presents a set of use cases and scenarios of 5G in which machine learning can aid in addressing their management challenges. Specifically, 5G challenges such as network resource allocation and network performance degradation would have a big impact on QoE by steering the network performance.

Third, Section A.5 proposes an architecture of an autonomic self-organising network to ensure QoS, improve operational efficiencies and reduce operational expenditure of 5G networks. The state and consumption records on the hardware resources are gathered in real-time from multiple functional blocks constituting the layered architecture. The collected records are processed by the machine learning algorithms in (near) real-time or periodically tailored to identify or forecast specific 5G issues. Based on the output of the machine learning algorithms, the Policy Engine generates actions on network topology that provides high QoS without using excessive resources. It brings a cognitive solution to NFV management.

Finally, Section A.2 describes how the a solution to make self-organising in reconfigurable dynamic networks by using of policy based network management actuation for correction and prevention, and how these policies can be reconfigured based on the updated knowledge from machine learning algorithms.

6.2 Network Resource Allocation system for QoE-aware delivery of media services in 5G Networks

- **Title:** Network Resource Allocation system for QoE-aware delivery of media services in 5G Networks
- **Authors:** Angel Martin, Jon Egaña, Julian Florez, Igor Olaizola, Jon Montalbán, Marco Quartulli, Roberto Viola and Mikel Zorrilla
- **Journal:** Transactions on Broadcasting
- **Publisher:** IEEE
- **Year:** 2018
- **DOI:** `http://dx.doi.org/10.1109/TBC.2018.2828608`

Abstract. The explosion in the variety and volume of video services makes bandwidth and latency performance of networks more critical to the user experience. The media industry's response, HTTP-based Adaptive Streaming (HAS) technology, offers

media players the possibility to dynamically select the most appropriate bitrate according to the connectivity performance. Moving forward, the telecom industry's move is 5G. 5G aims efficiency by dynamic network optimization to make maximum use of the resources to get as high capacity and Quality of Service (QoS) as possible. These networks will be based on Software Defined Networking (SDN) and Network Function Virtualization (NFV) techniques, enabling self-management functions. Here, Machine Learning is a key technology to reach this 5G vision. On top of Machine Learning, SDN and NFV, this paper provides a Network Resource Allocator system as the main contribution which enables autonomous network management aware of Quality of Experience (QoE). This system predicts demand to foresee the amount of network resources to be allocated and the topology setup required to cope with the traffic demand. Furthermore, the system dynamically provisions the network topology in a proactive way, while keeping the network operation within QoS ranges. To this end, the system processes signals from multiple network nodes and end-to-end QoS and QoE metrics. This paper evaluates the system for live and on-demand Dynamic Adaptive Streaming over HTTP (DASH) and High Efficiency Video Coding (HEVC) services. From the experiment results, it is concluded that the system is able to scale the network topology and to address the level of resource efficiency, required by media streaming services.

Keywords: 5G, cognitive network, internet TV, network topology, NFV, QoE, QoS, SDN.

6.2.1 Introduction

Multimedia consumption is gradually shifting from traditional TV to streaming video on connected devices, such as Smart TVs, mobile devices, etc. Furthermore, considering the demographic studies, the trend shows a sharp increase in streamed video viewing, particularly among younger generations [Ericsson15]. Thus, the traffic for videos delivered over the Internet will reach 82% of the total Internet traffic by 2021 with a million minutes of video content crossing the network, according to the report issued by the world IT leader Cisco [Inc17b]. Here Internet video includes web-based video monitoring, short-form Internet video (YouTube), long-form Internet video (Hulu), live Internet video, Internet video to TV (Netflix or Roku) where the Internet video-to-TV traffic means the 26 percent. Meantime, reaching heterogeneous devices gains relevance thanks to

the growth of mobile devices as the entry point to services [Inc17a]. The continuous evolution of the media entertainment industry towards enhanced experiences pushes the Ultra High Definition (UHD) technologies beyond 4K resolutions, High Dynamic Range (HDR), Wider Colour Gamut (WCG) and Higher Frame Rate (HFR). The technical challenges combined with today's consumers viewing habits, have shifted from watching purely linear TV to watching media as part of a multi-screen and multi-tasking activity [Domínguez et al.17, Zorrilla et al.15b]. Thus, the convergence of broadcast technologies and mobile networks is fueled by the change on usage patterns, the regulations staking out and reshuffling traditional broadcasting bands to expand mobile networks, and the proliferation of richer experiences requiring broader bandwidths.

Moreover, quality is a dominant factor that drives demand, customer satisfaction and retention, turning the user experience and the ability to deliver media services to any device key aspects. The user expectations on choice, quality, and convenience will continue to increase for the foreseeable future. In order to meet these future needs the 5G network may have a potential to substantially enhance the user experience and positively impact the audiovisual media value chain, including content production, distribution, and delivery to the user environment. 5G can stimulate new economically-viable services of high societal value like U-HDTV application, acting as a vehicle towards a co-operative use of broadcast and broadband infrastructures and enabling bandwidth intensive and low latency experiences. Some superior 5G features are key to media services: faster access with higher user experience data rate and low latency; reliable and dependable network with zero downtimes; and network agility reducing operational time cycles from hours to minutes to deal with dynamics from speed of mobility and connection density.

In the domain of optimization of media services delivery, SLA (Service Level Agreement) enforcement will take a salient position in the value proposition of 5G [Serrano et al.16]. SLA refers to the level of service guaranteed (often through contract) to a user or service by the network operator. The SLA includes a number of QoS parameters. These metrics include bandwidth, latency, security, geographical coverage qualifications, downtime due to error or faults, and priority that a user or service may expect where contention exists. Users may pay a premium subscription to operators to be guaranteed a higher SLA and certain services (emergency services, government communications) may be required by law to be given a higher SLA than other services.

SLA is transforming the operational features of networking functions from reliability to agility. Traditionally telecoms equipment is expected to provide 99.999% availability [Liu et al.16a]. However, with many modern IT services requiring different levels of guaranteed bandwidth, latency and priority over other traffic, SLAs have become more important and more differentiated depending on the nature of the service. The goal is to provide the best possible QoE according to the SLA, and the appropriate device features to overcome technical limitations in order to get a live, fluent and continuous multimedia experience.

Here, HEVC and MPEG-DASH are key to the media industry. Encoding standards, such as HEVC, relieve the bandwidth usage by minimizing the employed bitrate [Qian et al.17]. Accompanied by MPEG-DASH, multiple bitrate streams are operated by adjusting the play-out rate to stay within the actual network throughput and device capability. Thus, adaptive encoding offers benefits to allow operators to plan the capacity of their delivery networks to match the average, rather than the peak, usage demands. This way, operators save considerable Capital Expenditure (CAPEX) maintaining an uninterrupted user experience by means of client based switching decisions. MPEG-DASH and HEVC technologies catalyse QoE solutions for each connection [Yu et al.17], however, from the point of view of the infrastructure and the network, a global optimization for massive media services must be carried out.

The volume of video affects all parts of the IT infrastructure and the network, posing greater challenges due to the cost and bandwidth constraints. The answer to video overload is simple, reduce the traffic or add more bandwidth. Hence, techniques like bandwidth optimization, QoS, and path selection are vital for the network manager [Xu et al.13]. Therefore, 5G optimization tools must provide elements for control-path selection and managing the prioritisation of different traffic types depending on their importance in a cost-effective manner. There, the paths are directly related with the topology of the network.

5G has to deal with fast, heterogeneous, multi-tier networks, which are also dynamic in nature. NFV [Foundation13] and SDN [Foundation12] are two key enabler technologies of 5G. NFV leads to cost efficiency, improvements in time-to-market and innovation in agile network infrastructure and applications. SDN enables network administrators to manage network services through the abstraction of lower-level functionality. This is achieved by decoupling the system that makes decisions about where traffic is sent

(the control plane) from the underlying systems that forward traffic to the selected destination (the data plane). So, implementation of SDN results in infrastructure savings, operational savings and flexibility [Kim and Feamster13]. Furthermore, on top of SDN technologies, it is possible to develop systems to autonomously improve network agility and flexibility to efficiently support the evolving demands of users.

Machine Learning is a good technology candidate to support the vision of the Self-Organised Network (SON) [Klaine et al.17]. Its ability to learn from historical data, make predictions, dynamically adapt to new situations by learning from new data [Mohri et al.12] and make decisions offers a great potential in the network management area, forecasting resource demand and reacting appropriately.

By combining SDN and NFV concepts, a centralized view of the network can be exploited by Machine Learning aided systems to automatically identify networking issues. Thus, enabling the controller to change the network topology instantiating or removing Virtual Network Functions (VNF) to forward the incoming traffic in an efficient way, thereby removing the unused parts of a network to release these resources [Ismail et al.13].

The core contribution of this work is the design, implementation and deployment of a Network Resource Allocator system. The system provides QoE-aware and autonomous network management which, instead of building a network to meet an estimated maximum demand, dynamically provisions a network topology to accommodate changing demands. This system encourages self-configuration, self-optimization and self-healing, shifting from reactive to proactive by means of Machine Learning, SDN and NFV technologies. It includes the capability to scale the network topology and to address the levels of resource optimization, required for media streaming services, in 5G. To this end, the system processes signals from multiple network nodes and end-to-end QoS and QoE metrics.

The work described in this paper verifies that it is possible to integrate Machine Learning methods in an SDN controller to forecast resource demands and to react appropriately. Hence, the system instantiates the most efficient network topology while satisfying an SLA and operational costs. The learning is done based on experience gathered in previous measurements. This area is known as smart traffic routing.

The rest of the paper is structured as follows. Section 6.2.2 reviews the related work in terms of network solutions for QoS and QoE-sensitive media delivery. Section 6.2.3

describes the main contribution of the article with the definition of a Network Resource Allocator system, autonomously providing an efficient network topology for an existing demand, meeting the SLA performance and operational costs. Section 6.2.4 presents an implementation developed on top of MPEG-DASH and HEVC media services. Section 6.2.5 compiles the results from a set of validation experiments carried out. Finally, we present our conclusions in Section 6.2.6.

6.2.2 Related Work

The increasing rates of video experiences and audience are causing the current Internet architecture to reach saturation point. The Content-Centric Networking (CCN) architecture can be considered to resolve this issue in video transmissions. CCN is a new architecture based on how content is named and stored within the network, rather than where it is located, including the IP addresses of the hosts [Park et al.14]. There are new protocols that can find and retrieve content and make network's performance faster, more resilient, and more secure. With regard to QoS evaluation of video streaming, this work [Rhaiem et al.15] tests routing protocols for CCN-based MANET networks.

However, in order to deal with the imminent mobile broadcasting for digital video, Internet of Things (IoT) and Machine to Machine (M2M) systems, a revolution on the networks is required. Here, 5G promises a leap forward for the network features with ever increasing rates of overall data capacity and user density, requiring low power consumption and low data rates for very large numbers of connected devices and ultra-reliable and low latency communications [ETSI17b].

The main advances of 5G focus in two directions [5GPPP16]. First, the radio access network (RAN), by means of additional spectrum bands and higher spectral efficiency, in order to achieve higher capacity [Chávez-Santiago et al.15]. Second, SDN solutions to empower the core and the edge of the network [Nguyen et al.16].

These two lines respond to the need, of the network manager or telco operator, for tools to improve QoS in a 5G environment, such as the:

- Optimization of traffic when passing across the network (e.g. Radio Access Network (RAN) optimization, specific video optimization tools, management of application traffic, congestion control).

- Selection of the most efficient topology and setup to deliver the best QoS at the best cost (e.g. policy-based or fully dynamic network selection; support for different QoS layers with different cost and service level agreement levels).

For the deployment of the most appropriate network schema, the capacity of the SDN controller to dynamically operate the network is capital. Concerning SDN, the strategy from standardization bodies, such as the European Telecommunications Standards Institute (ETSI), network operators and equipment vendors is to decouple hardware from software and move network functions towards software. The key challenge is to enable direct access and manipulation of the forwarding plane of network devices (e.g. router, switch), by moving the network control out of the networking switches, to logically centralized control software. A logically centralized network intelligence can tune the network control directly without taking into account the underlying infrastructure, which is completely abstract for applications and network services. Thus, networks turn into flexible, programmable platforms with intelligence to dynamically meet performance requirements and react to or prevent degradation symptoms.

The SDN architectures use two interfaces. The Southbound API is employed to communicate the SDN Controller and the network switches and routers. The Northbound API is defined to communicate the SDN Controller and the services and applications running over the network. Hence, the Southbound API facilitates efficient control over the network by enabling the SDN Controller to dynamically apply changes according to real-time demands, while the Northbound API facilitates innovation and allows efficient orchestration and automation of the network.

OpenFlow [Foundation17e], open standard, is the most well-known Southbound interface. It deploys innovative protocols in production networks by means of a communications interface defined between the control and forwarding layers of an SDN architecture. On top of it, a set of representative Open Source projects such as OpenStack [OpenStack17], OpenMano [TID17] or OpNFV [SDxCentral17] deploy NFV and MANO technology stacks. Some examples of VNFs are routers, base stations, core mobile nodes, Evolved Packet Core (EPC), firewalls, intrusion prevention IPS, etc.

These technologies catalyse the transformation of operative switching and forwarding into programmable and configurable functions enabling autonomous network

management. Network management in 5G aims to provide high performance connectivity to an increasing number of users [Sun et al.15]. To this end, intelligent traffic steering systems for packet forwarding while reducing operational expenditure are being explored [Hernandez-Valencia et al.15].

From an ICT operator point of view, the service provisioning, in terms of networking or computing resources, constitutes an important challenge which deeply conditions CAPEX and Operational Expenditure (OPEX) values. This challenge relates to the problem of service demand prediction and resource provisioning which allows the network to resize and provision itself, using virtualization, to serve the predicted demand according to parameters such as location, time and specific service demand from specific users or user groups.

In this context, the simplest and fastest approach is to draw upon over-provisioning which consists of allocating an amount of resources larger than that required. This way the possible demand increase is met avoiding any intervention as long as the upper limit of resources allocated is not exceeded. This approach is operationally effective, but inefficient in terms of resources and energy consumption, therefore it is not cost-effective. It becomes clear how service demand prediction would be an essential solution to this issue. Being able to foresee the amount of network resources to be allocated to cope with the demand fluctuations constitutes a great benefit for a network operator.

In [Caglar and Gokhale14] it is stated that cloud service providers tend to maximize their profit by overbooking their resources. It concludes that an arbitrary overbooking ratio may degenerate into SLA violation and cost penalties especially for online video streaming. To optimize the resource utilization and reduce the risk of SLA violations it introduces an Artificial Neural Network to find correlation in the historical data and predict future resource usage. However, the system is not fully automated. In [Serrano et al.16] a system is proposed where a user can define his/her Service Level Object (SLO) related to a specific QoS metric to detail his/her expectation as well as the penalty for any breach. The policies and configurations will be reconfigured according to the updated SLA and a control program will be responsible for applying the reconfigurations. In [Emeakaroha et al.10] an SLA enforcement strategy is presented, called LoM2HiS (Low Metrics to High Level SLOs), for mapping low level metrics to high level SLA predicates.

Moving forward, this SDN paradigm boosts network adaptability and provides elasticity functions to make the network easily scalable. However, this brings about the need for mechanisms to manage the network due to the increased network complexity. Machine Learning technologies must be considered in order to meet the network resources allocation that dynamically meets changing demands, while achieving the SLA network operation enforcement and keeping networking operation inside business ranges [Buda et al.16].

Most of the works, related to demand prediction, employ neural network-based algorithms [Sandhir and Mitchell08, Edwards et al.97]. Moreover, the use of Machine Learning can be applied as a QoS performance optimizer in a number of ways. Firstly, Machine Learning will be used to assess the current provision of network resources in order to reduce resources when no longer needed. Furthermore, Machine Learning will present suggestions to network operators about the structure of their networks.

In the context of self-configuration, self-optimization and self-healing, some works deal with growth in traffic and provide better QoS and QoE [Klaine et al.17]. From the perspective of self-configuration, in [Wainio and Seppänen16] a system to perform congestion management by means of autonomous deployment of the backhaul's network topology is explored. The goal of this approach is to accommodate traffic demands from a neighbourhood of Base Stations (BSs) of the RAN. The backhaul connects the BSs and the rest of the network. Therefore, the backhaul update process is configured enabling new routing paths while providing better latency, reliability and improves energy saving. Another work on topology management [Farzaneh and Moghaddam08] proposes a backhaul solution to arrange the network topology in response to changes in traffic demand.

An SDN controller could change the network topology instantiating or removing VNF to forward the incoming traffic in an efficient way, removing the unused parts of a network to release these resources [Ismail et al.13]. Today, this vision is not yet realized [Bizanis and Kuipers16]. There is not a reliable solution that addresses the problems for flexible creation by scaling up/down or in/out an elastic network in an automated manner [Szabo et al.15].

In terms of QoS and QoE provisioning in the core and the edge of the network, an automated topology management tool can address flexible QoS schemes, congestion

control mechanisms, load balancing and management features. However, the main limitation in the related work is that QoS and QoE provision are not specifically targeted toward novel use cases of NFV and SDN. Furthermore, most of them are not realistic, simulating operational setups. So, there is a lack of frameworks for SDN and NFV for full automation of SLA management combining all the necessary blocks of cognitive management and proactive provisioning [Bendriss et al.17]. In fact, in [Bendriss et al.17] the network management system only focuses on the cardinality of the size of the network but not on the topology graph itself.

6.2.3 Network Resource Allocator system

The scope of the Network Resource Allocator is to apply an efficient network topology setup for an SLA-enforced media delivery. To this end, the system considers the performance records assessed on past exercised topologies, demand prediction and business constraints, as shown in Figure 6.1. First, Figure 6.1 *a)* depicts the preliminary process to populate the database of the system with real metrics, probed from specific topologies for known media traffic demands, to create a ground truth with performance records. Second, Figure 6.1 *b)* shows the ability to conclude and deploy an efficient topology, already in the database or unseen, to prevent partial network under-performance for a demand forecast in specific paths. The candidate topology results from the records on the database, the predicted volume of demand, the resources availability and the business constraints.

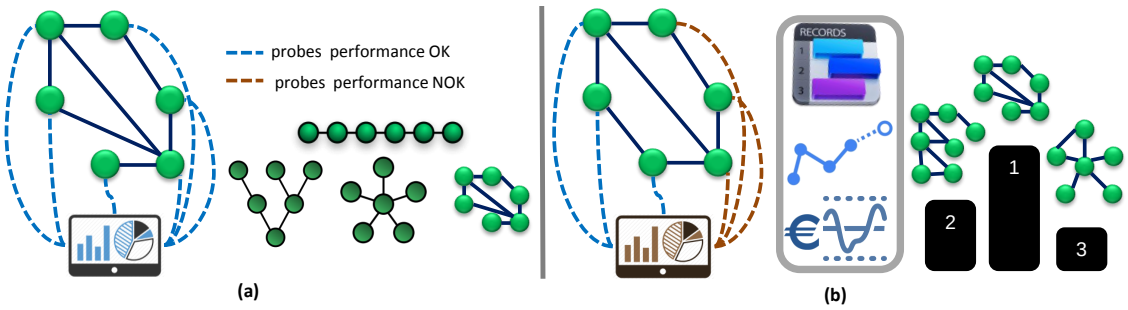


Figure 6.1: a) Assessment of QoS for different exercised topologies; b) Decision making on efficient topology setup to fix potential SLA breaches on specific forwarding nodes and media clients, the resources availability.

To describe the approach of this work in more detail, subsection 6.2.3.1 introduces the system, the individual logic blocks and the dataflow. Afterwards, subsection 6.2.3.2 explains the QoS and QoE metrics, that must be considered for media services, and subsection 6.2.3.3 covers the Key Performance Indicators (KPIs) which come into play. From this list, this work takes the ones more directly related to the SLAs, to forecast potential violations and prevent them.

6.2.3.1 **System components**

The Network Resource Allocator provided in this manuscript processes data from multiple network nodes and enables autonomic infrastructure management. This system demonstrates the capability to scale the network topologies and to address the levels of resource optimization required for 5G.

The overall Network Resource Allocator diagram is depicted in Figure 6.2 where the QoS and QoE metrics from network nodes, media servers and players are probed in real-time and stored in the metrics database to be processed by the machine learning components. In the Algorithm 5 the implemented system workflow is defined. First, the classifier processes the metrics to identify SLA breaches and notifies to the optimizer any network issue raised. Then, the optimizer internally queries to the regressor the performance of a set of topology candidates close to the current one (τ). To provide a result, the regressor takes metrics from the database and returns the network performance scores from records if present or make a prediction. Finally, the optimizer ranks all candidates and suggests to mutate network topology to respond a new volume of traffic demand. As a result, the network resource allocator provisions new resources or frees unused ones and sets up the new topology (τ_{new}) by means of the network controller.

Here, the involved sub-modules are listed:

- Network: set of network elements such as servers, routers or switches that form the network infrastructure and operated through an SDN controller and end-to-end media player metrics. The network includes probes to capture the performance metrics in real-time and agents to send the values to the metrics database.

- Metrics DataBase: monitoring element which measures and stores some key parameters related to the performance of the network.

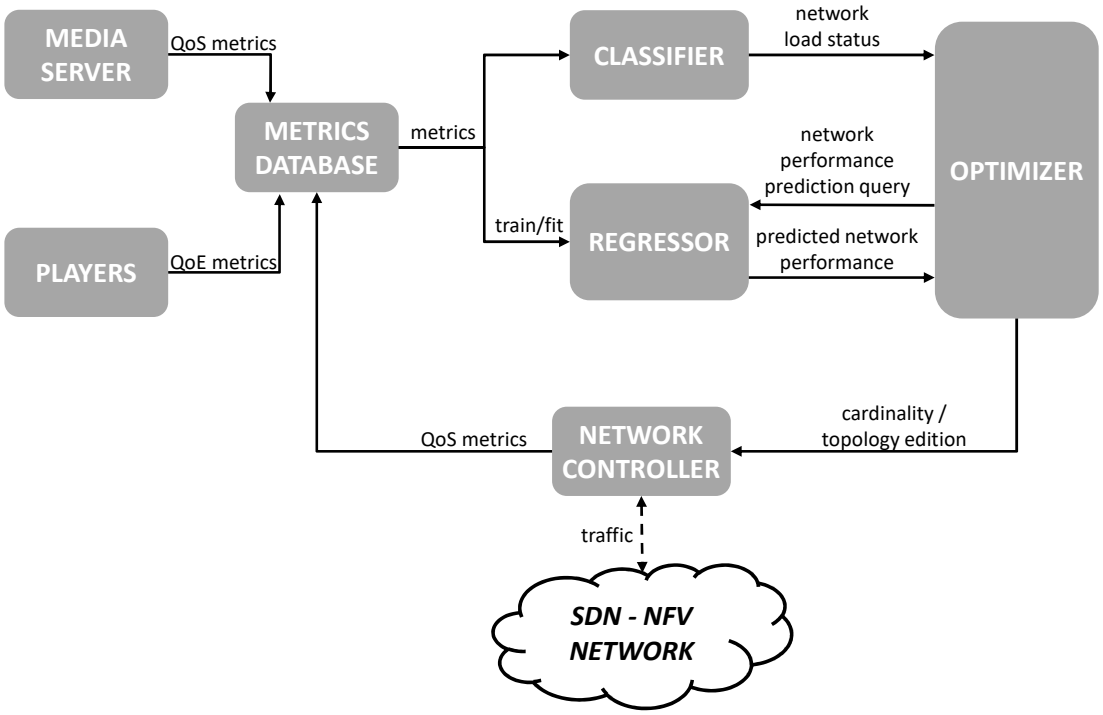


Figure 6.2: Network Resource Allocator workflow.

Algorithm 5 Network Resource Allocator

Input: τ ▷ employed topology
Input: p_{SLA} ▷ target performance
Input: $cost_{max}$ ▷ max expendable cost
  while True **do**
    [bw, l, j] ← ReadLastMetrics() ▷ from the database
    p ← (bw,l,j) ▷ current network performance
    if (Classifier(p,p_{SLA}) == SLA breached) **then** ▷ SLA breach
      τ_{new} ← Optimizer(τ,p,$cost_{max}$) ▷ cost & real-time driven
      $\tau \leftarrow \tau_{new}$ ▷ provision and deploy new topology
    else
      continue

- Supervised Classifier – the *Clustering* module: a module to continuously detect the status of a network which is being used for delivering massive media data flows under conditions of varying traffic. The classifier acts as a network profile classifier checking if the bandwidth, latency and jitter performance metrics (p = *(bw,l,j)*) of the network are inside the operational range defined by the network manager for a specific service. The network must ensure a operational bandwidth (bw) over a nominal value for a high quality video (bw_{min}) and a contained latency (l) and jitter (j) under smooth parameters (l_{max} and j_{max}). In order to avoid instable transitions, when the network underperforms persistently, for a long time (d) with any violation from thresholds (bw-$bw_{min} > 0$; l_{max}-l > 0; j_{max}-j > 0), the classifier triggers the actuation of the Network Resource Allocator to find a capable topology that ensures the SLAs. This process is described in Algorithm 6. The classifier just clusters new data to identify valid or violated performance level. To do so, the classifier ingests data from the metrics database, where the network metrics for each temporal segment have been stored in real-time, and constantly updates the classification model.

- Regressor – the *Regression* module: a module to forecast KPIs of a massive multimedia delivery service over a previously unseen network topology. The regressor is queried by the optimizer each time the classifier notifies to the optimizer an SLA breach situation. The employed modified linear regression is presented in:

$$f_{\tau}^{(m)}(n) = \beta_0 + \beta_{\tau,m} \cdot a_{\tau,m}^{n} + \varepsilon \tag{6.1}$$

  for each node represented by n and the evaluated network performance metrics (bw, l, j) represented by m. Thus, the equation 6.1 is used to predict the KPIs for the unmeasured network configurations. Using least squares:

$$Y = X\beta + \mu \tag{6.2}$$

  where $\mu \sim N(0, \sigma^2)$, the regression function is trained using measurements from past employed topologies. In this training, we get the values of β_0 and $\beta_{t,m}$ which best fits to the training measurements. To generate a forecast the regressor finds

the coefficients of the linear function (β_0 and $\beta_{t,m}$) that allow the Network Resource Allocator predict a topology performance before it comes into play. The regressor takes the metrics of the system and predicts some key features of the model to be fed to the optimizer.

- Optimizer – the *Optimization* module: a module to help human operators to enforce SLAs while keeping operating costs under control. It considers the output of the machine-learning modules and assesses the performance of all the possible and permitted network configurations. As synthesized in Algorithm 7, in order to avoid disruptive changes on the topology, to avoid oscillations and to achieve real-time performance, the optimizer generates a list of candidate topologies that could be applied scaling up or down the current number of nodes of the network topology [McKay and Wormald90]. To this end, the explored space of topology graphs is governed by it_{max}. Each candidate from the list is then queried to the regressor in order to get a performance forecast. With all the results, the optimizer ranks them, to get the better topology to mutate. It then passes the optimum arrangement for the minimum cost, service time and other performance measures in order to comply with the SLA the operator has to comply with. To meet this challenge the algorithm used is Simulated Annealing. This is based upon [Kirkpatrick et al.83] who proposed that it form the basis of an optimization technique for combinatorial problems. As previously stated, within the 5G context, we are focusing on automatic network management. This requires the optimization of a mathematical model representing the network's performance. To solve this issue, the present task focuses on developing optimization algorithms capable of improving the performance of the networks.

The Network Resource Allocator dynamically deploys the concluded topology using the SDN controller.

6.2.3.2 **QoS and QoE metrics**

The essential QoS metrics which come into play for live or on-demand experiences are low latency and high bandwidth. These have a direct impact on the most relevant aspects involved in the QoE when playing HAS media [Chen et al.15]:

Algorithm 6 Classifier

Input: p ▷ current network performance
Input: p_{SLA} ▷ network performance for target SLA
Input: δd ▷ elapsed time between samples
Output: NetworkState ▷ network ensures or breaches target SLA
  d_{max} ▷ max time breaching target SLA
  if (p < p_{SLA}) **then** ▷ network underperforms SLA
    d ← d+δd ▷ accumulated violation time
  else
    d ← 0 ▷ normal or transitory violation
  if (d > d_{max}) **then** ▷ persistent violation
    return SLA breached ▷ fire network management actuation
  else
    return SLA ensured ▷ network status normal

Algorithm 7 Optimizer

Input: τ ▷ employed topology
Input: p ▷ current network performance
Input: $cost_{max}$ ▷ max expendable cost
Output: τ_{best} ▷ best network topology
  it_{max} ▷ number of processed topologies
  $\tau_{best} \leftarrow \tau$ ▷ current one is best costs-driven option
  $p_{best} \leftarrow$ p ▷ current one performance
  for i=1 → it_{max} **do**
    $\tau_i \leftarrow$ Candidate(τ) ▷ generate topology close to current one
    $p_i \leftarrow$ Regressor(τ_i) ▷ query topology performance
    if ($p_i \geq p_{best}$ and Cost(τ_i) ≤ Cost(τ_{best})) **then**
      $\tau_{best} \leftarrow \tau_i$ ▷ update best topology
      $p_{best} \leftarrow p_i$ ▷ update best performance
  return τ_{best} ▷ return best topology

- Initial Delay: the delay between the first client request and the start of the playback.
- Stalling Time: the sum of all playback interruptions.
- Number of quality switches: the total number of quality switches during the playback.
- Inter switching times: the time between quality switches should be imperceptible.

These QoE metrics are intrinsically related to bandwidth and latency [Yu et al.17, Orosz et al.14, Vega et al.18]. In fact, some works [Huang et al.18] simplify the QoE assessment to the monitoring of the buffer level directly related to the bandwidth and latency. The Network Resource Allocator for SLA enforcement computes them to forecast optimal topology. To this end, it compares different SDN configurations and legacy routing protocols in a guaranteed QoS video streaming scenario, using performance metrics such as bandwidth and packet delay to output a configuration for an optimal setup.

6.2.3.3 **Network Management KPIs**

The essential KPIs which come into play for live or on-demand experiences are the same, low latency and high bandwidth. In addition to these, other KPIs to be considered by the Network Resource Allocator are:

- Service scale: minimum size of the network to deliver the traffic volume with the required latency and jitter for the media streaming service.
- Forwarding efficiency: the average throughput and the ratio compared to the theoretical maximum.
- Latency: minimizing end to end delivery time.
- Packet jitter: maximum deviation of packet delivery from the average inter-arrival time.

The Network Resource Allocator scales up or tears down assets to dynamically adapt to the network traffic rate. Each time the network traffic load is modified (increasing or decreasing), the Network Resource Allocator block should be able to spot the traffic trend and choose a suitable model to be applied in the virtual network infrastructure to ensure the most efficient setup.

6.2.4 Implementation

This section describes the implementation details of the proposed Network Resource Allocator. It covers all the aforementioned core processing components, as well as previously mentioned common QoS and KPI metrics, in order to enforce the SLA performance of the network when delivering media services. First, the core components of the processing system are addressed. They deal with the traffic demands and apply a topology in order to satisfy the SLA and make an efficient utilization of the resources under the constraints of available network nodes. Second, the head-end setup is depicted, where the processing system is being exercised. Third, the media services are addressed. They will inject dynamic traffic challenges on the network management. Last but not least, the implementation requirements, key to creating agile networks for a management system, are listed.

6.2.4.1 Network Resource Allocator

The different technologies executing the Network Resource Allocator modules are briefly listed below:

- Supervised Classifier – the "Clustering" module: the metrics flow feeds the Online Classifier implementing a K-Means classifier in addition to an Apache Spark (MLlib).

- Regressor – the "Regression" module: it performs online regression with Python and Spark Streaming API.

- Optimizer – the "Optimization" module: the Optimizer module is a Python library for the global optimization of functions with or without constraints. The

optimization engine is based on the Simulated Annealing metaheuristic algorithm [Du and Swamy16]. Simulated Annealing is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space. It is often used when the search space is discrete. For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, simulated annealing may be preferable to alternatives such as gradient descent. In particular, the discrete capabilities of the Optimizer will help us in the optimization of the topology of the network.

- Network: Mininet [Mininet17] deploys the network infrastructure as explained in the subsection 6.2.4.2.

- Metrics DataBase: the data required to be persistent is stored in a MongoDB database [MongoDB17].

6.2.4.2 **Testbed**

The key activity foreseen in relation to the Network Resource Allocator functionality is the setup of a real experiment. Here, the configuration of an SDN and its managed NFVs are automatically optimized based on the results obtained from the forecasting of relevant metrics. This prediction is in turn based on a description of a current situation. To this end, a set of components is needed:

- Media service: a next-gen standard compliant platform to provide multimedia contents for massive consumption. A GStreamer [GStreamer17] server produces streaming traffic and sends it to the network (e.g. Mininet). Here, an Iperf [ESnet and Laboratory17] server is run on the same machine where the GStreamer media server is located. The Iperf server is employed in order to generate uniformly distributed packets in the network, thus the injected packets are employed to probe the network performance metrics.

- Service probing: a client-side data collection system involves capturing and sharing QoS and QoE metrics and benchmarks. It is described in this previous paper [Martin et al.17]. This can significantly impact in the volume and velocity (less

data to transfer means less time) of data transfers. A pool of GStreamer clients consumes the streaming created by the GStreamer server in a steady and fair manner across media players sharing the connection path. They stream video quality metrics in the QoS metric collector.

- Network monitoring: a system for collecting data from network nodes. It performs data pre-processing to boost data classification by identifying the most significant and irregular data. The messages generated can significantly impact the volume and velocity (less data to transfer means less time) of data transfers. Iperf clients are instantiated in order to receive the traffic sent by Iperf server and collect the network performance metrics into the QoS metric collector.

- QoS metrics collector: it collects metrics received by the GStreamer and Iperf server/client and publishes them through a Kafka [Foundation17b] data stream processing platform to enable Machine Learning processing in streaming mode, which is required for a real-time actuation.

- Network Functions Virtualization: enabled by SDN, it plays an important role to automatically reallocate resources. vSwitch [Foundation17d] on the Mininet testbed routes incoming packets by analysing the flow table configured by the controller.

- Application of Machine Learning algorithms: this virtualized system for predicting service demand and network provisioning allows the network to resize and resource itself. Here, the Network Resource Allocator core comes into play by processing the metrics from a Kafka queue from the QoS metric collector.

- Smart network control and management: a controller provides infrastructures with efficient and flexible provisioning in order to significantly improve end-to-end operations and network efficiencies. An SDN Controller is deployed with OpenDaylight [Foundation17c] which configures the Mininet network to forward the data flows with the configuration provided by the Network Resource Allocator.

The testing setup and the technologies involved are depicted in Figure 6.3, implementing the architecture described in this previous paper [Xu et al.16] and showing

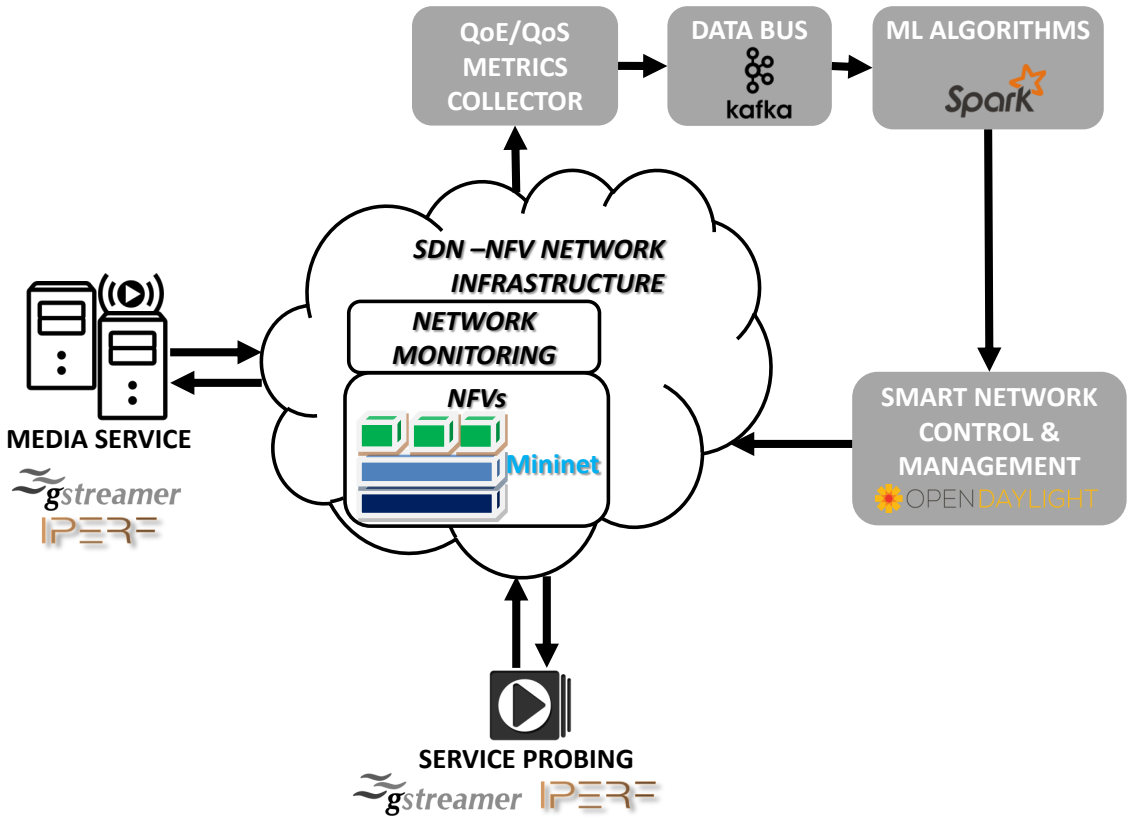


Figure 6.3: Testbed including technologies of logic components.

media services and probes in order to exercise the NFV-based network with the programmed traffic demands. The network nodes benchmarks and the media playout metrics are monitored and captured. They are sent in real-time through the collector, based on a Kafka data bus, to feed the Machine Learning algorithms. The Machine Learning algorithms eventually provide an outperforming topology to be applied. The smart network control and management takes this setup and deploys it by means of OpenDayLight network binding.

6.2.4.3 **Media services**

The test sequence employed in our experiments is Bug Buck Bunny with a duration of 9 min and 50 s. Its raw version is provided by Xiph.Org Foundation [Xiph.Org17]. Segment files are generated by encoding a test sequence in HEVC format (ISO/IEC 23008-2:2015) and multiplexing in ISO MPEG4 files (ISO/IEC 14496-12 - MPEG-4 Part 12). The chosen

Table 6.1: Set of MPEG-DASH representations for the tests.

index	profile	bitrate	resolution	GOP size	framerate
1	low	420kbps	288P	72frames	15fps
2	mid-low	1000kbps	360P	90frames	30fps
3	mid	1400kbps	432P	90frames	30fps
4	mid-high	2000kbps	480P	90frames	30fps
5	high	2600kbps	576P	90frames	30fps
6	premium	3400kbps	720P	90frames	30fps

duration for each segment is fixed to 5 seconds, granting a balanced live delay and window time for successful segment download trade-off. Moreover, the test sequence is encoded into six different representations to allow adaptation to the network dynamics at the client-side. Each representation is characterized by a particular video bitrate. The complete characterization of each representation is depicted in Table 6.1. Here, the group of pictures (GOP) size sets the number of frames between key frames.

In order to generate representative results three video based services generate singular traffic patterns streaming the same content over the network. Thus, all the clients access to one specific service in each run, downloading and/or uploading the content. The considered standard compliant streaming services are:

- Downstream. On-demand video contents are requested and downloaded from the server by the pool of connected clients (Youtube or Netflix like service).
- Upstream. Live video contents are uploaded to the server by the clients (UStream like service).
- Full-Duplex. Live video contents are transmitted in both direction between clients passing through the server (Skype like service).

6.2.4.4 **Implementation requirements**

The developed system offers some essential features when building a network management system to dynamically provision a network to meet changing traffic demands ready as a tool for network management in 5G. These features are:

- Scalability. First, it has been encapsulated on a Docker [Docker17] container in order to deploy the complete execution environment. Second, the Docker machine has been written in an Ansible [Hat17] script in order to facilitate the automatic generation and deployment of different instances in different setups.

- High performance. The system utilizes technologies widely employed in Big Data systems such as Spark and MongoDB to process data.

- Real-time processing. This is possible by means of Apache Kafka providing a channel with data streams coming to the Network Resource Allocator.

- Autonomous actuation. This is achieved thanks to the integration of the system with OpenDayLight APIs to deploy another topology setup.

6.2.5 Validation and Results

6.2.5.1 Validation

Network Resource Allocator takes measures from GStreamer players and network probes, and predicts network KPIs such as path bandwidth, latency and jitter directly related to QoE. Network Resource Allocator meets the network costs and the needs of forwarded services (SLAs) for the three defined video streaming traffic patterns stated in subsection 6.2.4.3. The business limits are considered by the Network Resource Allocator as a range of network size cardinality.

To perform the experiment, we created different networks by varying the cardinality of nodes (2, 4, 8, 16 and 32) with 3 different types of topologies (linear, star and tree). This way, initially, 15 different configurations were generated. However, unseen cardinalities and topologies could be concluded by the Machine Learning algorithm to efficiently cope with the new traffic demand under the SLA constraints. For each of these networks, data were collected to characterize the performance of the GStreamer service in each configuration created. Moreover, to simulate background network traffic, Iperf was launched from the server to the last node on the network. In addition, Iperf was used to continuously measure the free bandwidth (not used by GStreamer). With this information, the network congestion was continuously monitored. Once the

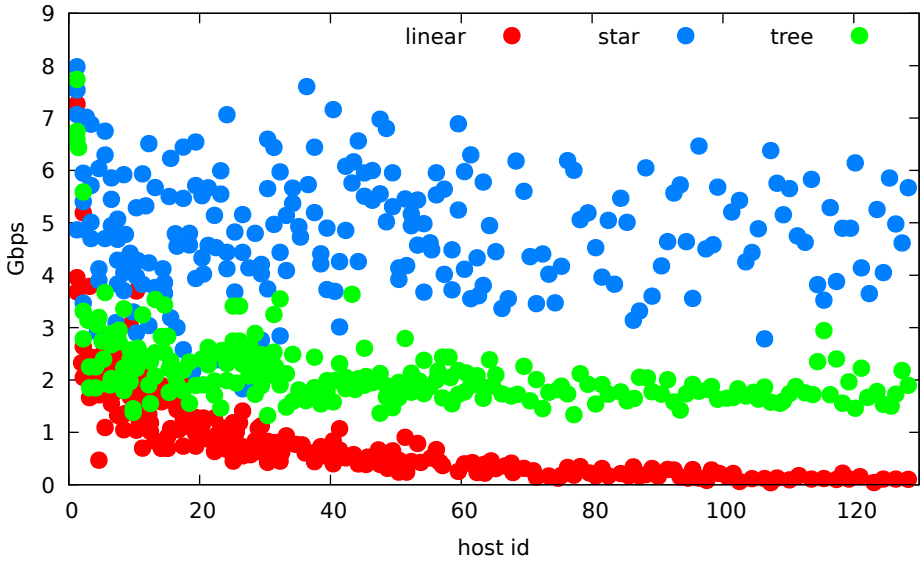


Figure 6.4: Assessed bandwidth for networks counting 32 nodes and 130 clients in different topologies.

network was created and Iperf launched to create background traffic, the GStreamer service was started. The GStreamer server was launched on the first network node serving media to 130 clients in different topologies. The regime of incoming sessions was linear with a new client connecting every 15 seconds. The service did not end until 15 seconds after the last client connected. Therefore, when the last client was connected to the server together with all the rest, they started to leave. Each GStreamer client collected metrics on the latency, bit rate (bandwidth) and jitter of all packets received. The clients were equally distributed across the three considered services, downstream, upstream and balanced.

In these networks we measured the path bandwidth and latency (using Iperf3 and media players' probes) from media players to the server. Figure 6.4 plots the assessed results for bandwidth representing a scatter matrix where each point is a measurement of a different path as described in Figure 6.1 *a)*.

As expected, the concurrency of the paths for linear topologies is similar with higher performance as the clients are closer to the server. For a tree topology, less clients share the full path, increasing the performance. And, for a star topology, the concurrency of packets for different clients is lower minimizing the impact on the communication path.

This way in the star scheme, depending on the position of the clients in respect to the server, the performance is different.

In addition to the described experiment and these primary measures, the goal for the Network Resource Allocator system is multiple. First, to be able to dynamically forecast the network load status for each connection (hop) according to predicted increasing and decreasing traffic demands with enough fidelity in advance (demands changing every 15 seconds). This way the system identifies congestions, bottlenecks or paths not satisfying the bandwidth and latency parameters for all the delivered streams. Second, to find an efficient topology configuration that satisfies the bandwidth and latency needs for all the paths. Last but not least, to mutate the network with the new topology to properly satisfy the incoming media streaming demands.

6.2.5.2 **Results**

The primary goal of the presented method is to be able to forecast the performance of the bandwidth and latency, establishing the thresholds for the target SLA, in order to apply the most efficient topology configuration. Thus, the results of the Network Resource Allocator focus on the accuracy to find an efficient topology for the incoming demand with uniform density distribution of media players.

When the data depicted in Figure 6.4 have been fully collected, we transform and use them to build a Machine Learning regression model which operates by estimating a power law curve for each configuration (cardinality and topology type). Here, only the power law curves for topologies with 128 nodes are depicted learned from the bandwidth for networks counting 2, 4, 8, 16 and 32 nodes in the different topologies. The data and the fitted regression models for the bandwidth, in different topologies for a set of cardinalities, are represented in the plots in Figure 6.5. The predicted values from the regression in Figure 6.5 accurately represents the assessed values shown in Figure 6.4. Furthermore, the fitted regression curves are also accurate, able to reliably represent the data samples. Using these curves, the trained model can predict the characteristics of a given unobserved path (e.g. in terms of bandwidth) in a specific network. The model can also be used in a numeric optimization procedure to determine the characteristics of the best network configuration with respect to the performance of a specific path or set of paths.

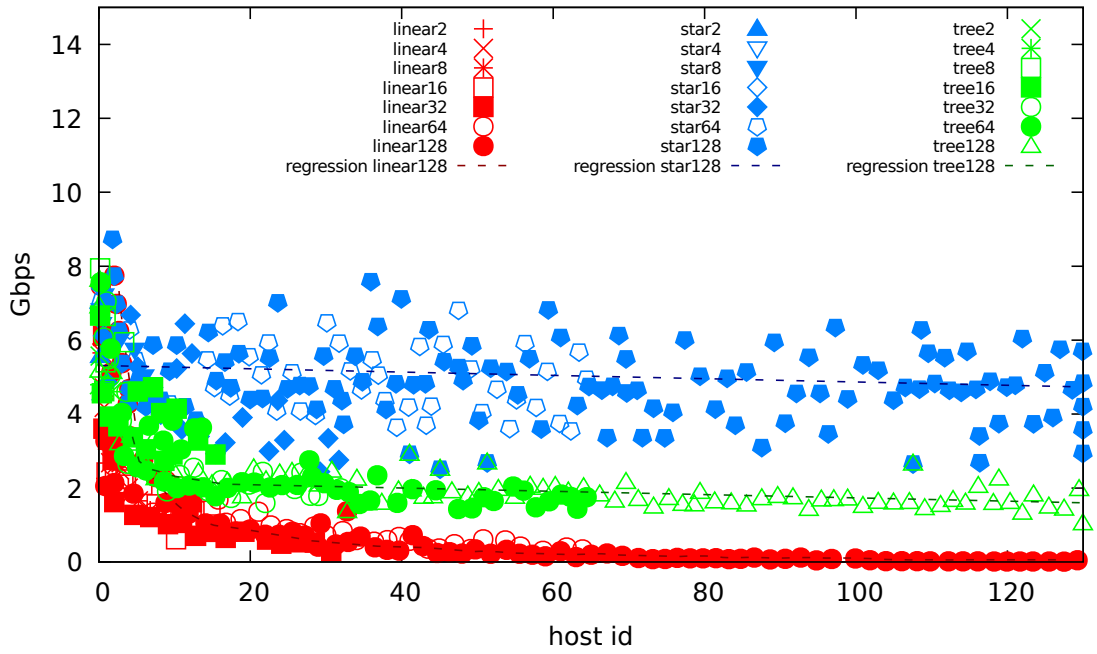


Figure 6.5: Prediction models for the bandwidth performance in different topologies (star, linear and tree topologies) and cardinalities.

In Figure 6.6, the data for the bandwidth, in different topologies for a set of cardinalities, are transformed to the logarithmic domain. Here, only the power law curves for topologies with 4, 64 and 128 nodes are depicted learned from the bandwidth for networks counting 2, 4, 8, 16 and 32 nodes in the different topologies. The prediction, plotted in logarithmic scale, shows that the deviation from the model is visible, particularly in the case of low cardinality values. This means that as the more complex the network, the better the scores this approach gets. The fidelity of the prediction models to provide representations for the achieved performance for the different paths in different topologies and cardinalities is shown.

For the selection of the topology and cardinality configuration to meet the predicted traffic demands, we have to introduce the network topology type, the cardinality and the source and destination host IDs as inputs to a forecasting engine that operates based on the learned model.

The analysis of the performance of the trained models is carried out by splitting the experimental dataset into a training (50%) and a test (50%) set, and by computing error

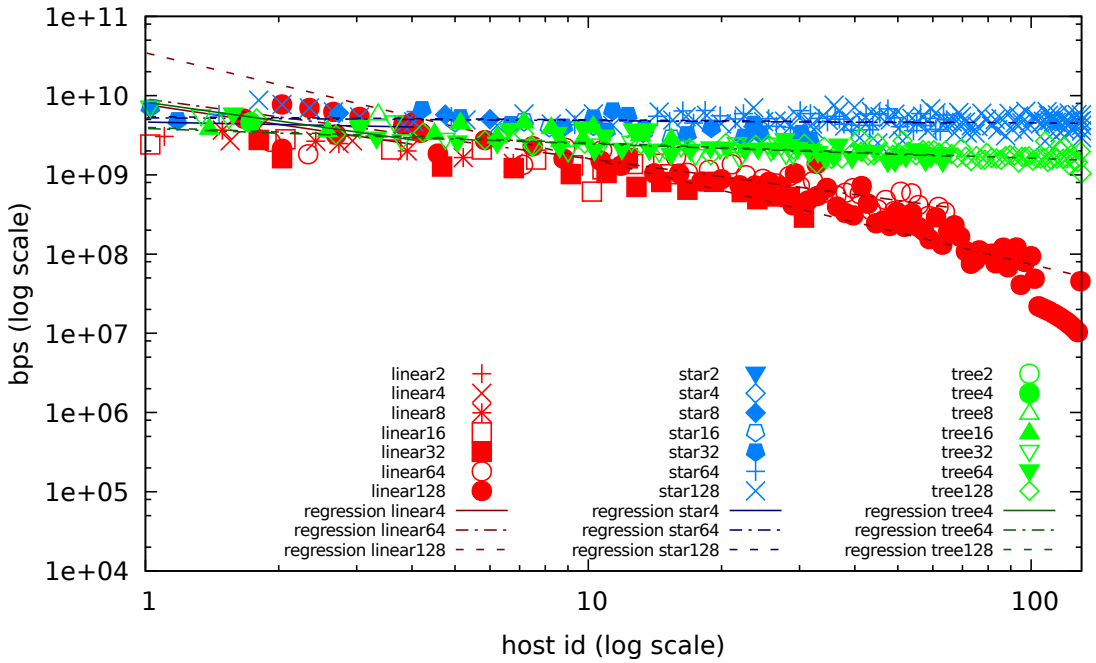


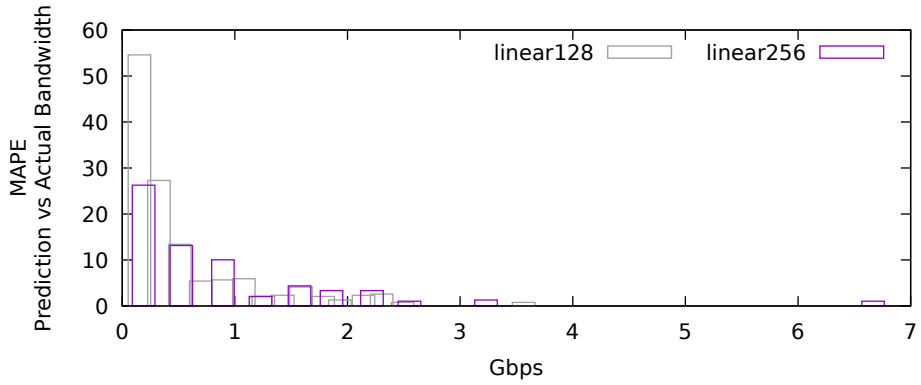
Figure 6.6: Prediction results, in logarithmic scales, for the bandwidth performance in different topologies (star, linear and tree topologies) and cardinalities.

measures on the test samples with respect to the model forecasts.

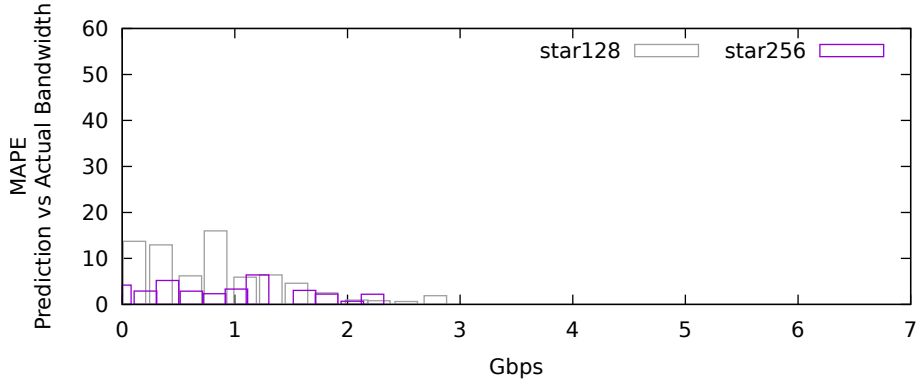
In Figure 6.7 we display histograms of prediction errors for the three topology types and for cardinalities 128 and 256. The mean absolute percentage error (MAPE) measures of prediction accuracy of the bandwidth forecasting. The error distribution depends on the type of the concluded topology and the target network performance. The structure of the star histogram seems to be approximating a uniform probability density function. This might depend on the fact that in such a topology all paths are essentially equivalent. Errors tend to be limited in size. The histograms show forecast results more accurate for high bandwidth and coarser for low bandwidth availability. This way, for higher bandwidth requirements the system results are more accurate.

The results in terms of latency are equivalent to the figures and graphs depicted for bandwidth. Furthermore, the results for download, upload and peer media services are similar without any singular or remarkable feature.

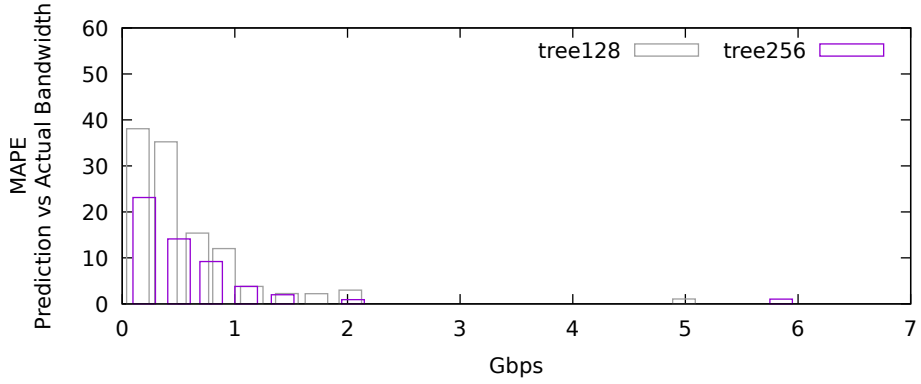
This predicted configuration is then used by an automatic controller. It is applied by a self-management procedure capable of adapting in an agile manner to changing con-



(a)



(b)



(c)

Figure 6.7: Prediction Error Histogram for cardinalities 128 (in gray) and 256 (in violet) in different topologies (**(a)** linear, **(b)** star and **(c)** tree topologies).

ditions, such as the appearance of bottlenecks or the need for more network switches and nodes. So, once we have the network architecture, the OpenDayLight SDN controller changes the actual Mininet network to this network. To this end, the controller instantiates or removes NVF-based network functions and links.

To sum up, the proposed Network Resource Allocator approximates the most efficient network topology for a predicted demand assuring bandwidth and latency performance KPIs which satisfy QoE and consequent SLA constraints. The accuracy of the results is better as the cardinality of the network is bigger and the demands in bandwidth are higher, while the fidelity drops for tiny setups and audiences. So, the more complex the infrastructure and the wider the media service demand, the more confident this approach gets. Furthermore, the Network Resource Allocator is able to conclude unseen configurations (cardinality and topology type), and deploy them to exercise the new network.

Furthermore, the results presented in this paper meet essential 5G requirements of the network core and backhaul. While some significant 5G features are intrinsically related to the new radio specification, the Network Resource Allocator is a cognitive tool to empower network management based on SDN and NFV technologies, boosting autonomous network agility in the context of media delivery services. The integration of the Network Resource Allocator with representative network management technologies and open source stacks to capture QoS metrics, to process data in real-time and provision it with the SDN controller reduces operational time cycles from hours to minutes. The elapsed time from a new prediction demand coming to the system to the conclusion of a better topology takes a time under 1 second, including the data processing by the machine learning algorithms and the generation of visual graphs for network management monitoring and debugging. Then, the actuation in the network by the SDN controller is under 1 minute but it is highly related to the number of changes to apply. This feature is key when meeting environments where dynamics from user mobility and connection densities move fast.

Moreover, the ability of the system to conclude a new network topology, according to target SLA for stable throughput and latency performance, infrastructure costs and soft transition policies, enforces dependability by preventing network from downtimes. This manner, in order to ensure network reliability the network manager only needs to tune the cost range for an expected demand.

6.2.6 Conclusions and Future Work

The increasing rates of video experiences and audience are causing the current Internet architecture reaching the saturation point. Furthermore, in order to deal with incoming mobile terminal broadcasting, IoT and M2M systems, a revolution on the networks is required. One of the goals of 5G is to provide the best possible QoS according to the SLA, and the appropriate device features to overcome technical limitations in order to get a live, fluent and continuous multimedia experience. The quality of the network experience is an important element in customer satisfaction and retention.

5G networks will be highly based on software, enabling self-management functions. Here, Machine Learning is a key technology to reach the vision of a 5G self-organising network. We show that Machine Learning algorithms in addition to SDN technologies can be used to predict path characteristics that directly determine network KPIs. At the moment we have these predictions, we can change the network to obtain the best KPIs, QoE, QoS and identify unused parts of a network for the service we want to forward in our network.

This paper introduces an automated Network Resource Allocator system. The overall solution comprises a network operated by means of an SDN controller which is autonomously and dynamically set up by the Network Resource Allocator. The Network Resource Allocator is engined by Machine Learning algorithms to predict traffic demands, translate them into specific operational thresholds, identify a topology to deliver incoming traffic according to an SLA and operational costs and, eventually, to deploy it through the SDN controller. To this end, the system processes signals from multiple network nodes and end-to-end QoS and QoE metrics. The Network Resource Allocator takes measurements from GStreamer players and network probes and predicts network KPIs such as path bandwidth and latency directly related to QoE.

The experiment results of the Network Resource Allocator system conclude that the more complex the infrastructure and the wider the media service demand, the more confident this approach is. Furthermore, the experiment setup demonstrates that it is possible to integrate machine learning methods in an SDN controller to forecast resource demand and to react appropriately, so that this one can learn to instantiate the better network topology in terms of KPIs. The learning can be done based on experience gathered in previous measurements. The proposed Network Resource Allocator

system is a reliable solution that addresses the problems for the flexible creation of an elastic network in an automated manner. Thus, it enables the controller to change the network topology instantiating or removing NFVs to forward the incoming traffic in an efficient way, removing the unused parts of a network to release these resources.

The Network Resource Allocator has been tested and validated to enforce the network with the needs of forwarded services for three defined video streaming traffic patterns, Netflix, UStream and Skype like services.

In the future we plan to add HAS specific QoE metrics such as initial delay, stalling time, number of quality switches and inter switching times by means of the eMOS scores [Claeys et al.14a, Mok et al.11] coming from media players [Martin et al.17], as well as to go deeper into the topology mutation considering the current topology. Depending on the performance improvement and efficiency rates for the candidate topology, it could be more convenient to keep the current one rather than apply a disruptive topology. This policy would help to minimize the impact on the transition period from the current topology to the one obtaining better performance and more efficient resource utilization.

Acknowledgment

This work was fully supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action).

Part IV

Conclusions

CHAPTER

7

Conclusions

This research work has presented four solutions to improve the QoE when accessing to media services on top of novel delivery solutions and architectures for 5G networks. Furthermore, the feasibility of all the contributions has been demonstrated implementing and deploying them in operational and realistic setups, not simulating theoretical performance. To this end, different corners have been considered to provide an enhanced media service experience, from the media servers, the delivery network and the media players. Consequently, four main challenges have been addressed to improve the QoE on media services: **massive client connections**, **dense client cells**, **edge video analytics** and **self-organising networks**.

In respect of **massive client connections**, a system to exploit distributed tagging resources is introduced, called **SaW**, which aims to complement a Web-based social media service with an elastic cloud of spontaneous connected resources to run delay-tolerant tasks under a Mobile as an Infrastructure Provider (MaaIP) model.

With the aim of achieving enhanced and automatic media tagging over social media datasets, the SaW solution enables the following aspects:

- Foster background dispatching delay-tolerant background tasks of media analysis over connected clients. SaW deploys a pure Web platform for video analysis, adding a delivering computing layer to the stack of the HTML5-based main service.

- Provide high elasticity and address the availability of resources related to the spontaneous presence of users. By means of exploiting high user availability density, the elasticity takes advantage of delay-tolerant target scenarios, with a heterogeneous community of client devices characterised by the assorted availability of resources.

- Cope with the execution of hardware-accelerated image processing tasks in the background, according to the capabilities of each device. The computing tasks are embedded in the foreground social content without draining the bandwidth or affecting the perceived QoE.

- Extend a previous performance model that was focused on CPU resources, by aggregating GPU capabilities to determine suitable scenarios for this MaaIP model. The model illustrates a comparison of distributed computing setups with a local server solution. The maximum benefit is obtained for higher delay-tolerant computational loads with independent tasks capable of being distributed to idle devices, thus compensating the task scheduling management and consolidation overload of the server.

A proof-of-concept implementation of the proposed architecture has also been presented, to evaluate the proposed approach, including existing WebGL and WebCL technologies. The results of the experiments confirm the benefits of the MaaIP approach when the number of devices is high, and the tasks are independent and can be queued.

Regarding **dense client cells**, different mechanisms have been proposed as a solution to fair and efficient radio link utilization in dense client environments, such as a bitrate adaptation algorithm, named LAMB-DASH. LAMB-DASH aims to maximize the QoE through a client-driven selection. LAMB-DASH considers the network conditions during the bitrate adaptation process, while still maintaining the ability to react to sudden bandwidth fluctuations in the local network.

The LAMB-DASH bitrate adaptation mechanism:

- Performs a live assessment instead of preliminary processing for network featuring.

- Responds appropriately to different content types and changeable networking conditions, meaning that the algorithm does not require *a priori* knowledge.
- Requires a reduced background computation, when compared to heavier and less flexible alternative computing and optimizing models.
- Exercises a low-complexity heuristic model, based on measurements and estimations from a current stream state.
- Balances QoE in scenarios where several clients compete for the available network resources.
- Obtains a steady and unbiased radio link utilization across the devices sharing a radio link.

The algorithm has been implemented and validated on top of a GStreamer client and tested in a setup where multiple clients share the same path in the network, therefore competing for the available bandwidth. Two different scenarios have been explored. Scenario 1 runs clients synchronized to a common clock joining the live stream at the same time. Scenario 2 arranges clients joining the live stream randomly. Here, they experience stochastic network bandwidth fluctuations.

The results of both scenarios show that the algorithm achieves fairness, since the clients tend to have the same representation bitrate.

In terms of **edge video analytics**, this research proposes a reliable CDN and fair radio link utilization in network edges, by means of exploiting visual analytics on a MEC system, using MEC4FAIR. MEC4FAIR is a novel solution on top of a mobile SDR network.

The MEC4FAIR system:

- Performs real-time updates in the manifest with the available qualities and CDN endpoints. This vision empowers the role of MEC from ETSI for transparent QoE improvement and dynamic CDN selection to shield from service degradation and outages.
- Exploits L2 (link), L3 (network) and L7 (application) metrics to support switching decisions on HAS quality and CDN provider.

- Integrates into a real mobile SDR network and performs validation on a real setup, including an eNodeB and an EPC, checking the performance of an active component of the video delivery chain at the mobile edge.

The algorithm has been implemented and validated on a real SDR LTE setup where multiple clients share the same path in the network, therefore competing for the available bandwidth. Two different scenarios have been explored. The synchronous scenario runs clients synchronized to a common clock joining the live stream at the same time. The stochastic scenario arranges clients joining randomly to an on-demand stream. Here, they experience stochastic network bandwidth fluctuations.

The results of both scenarios show that MEC4FAIR achieves fairness and efficiency, since the clients tend to have a common and high-quality representation bitrate. Moreover, in the stochastic scenario, MEC4FAIR plays a significant role in improving efficiency, in terms of network utilization and quality experienced. Furthermore, the synchronous scenario introduces a more accurate and stable characterization causing the hybrid solution to obtain better scores than the stochastic one.

Finally, in terms of **self-organising networks**, this research proposes an automated setup of network topology for a forecast demand as a solution to automate network management in a scalable and real-time manner. A Network Resource Allocator system is introduced to dynamically provision the network in a proactive way. It predicts demand to foresee the amount of network resources to be allocated to cope with the demand, while keeping the network operation within business ranges. To this end, the system captures and processes performance signals from multiple network nodes and end-to-end QoS metrics from all the media players.

Aimed at achieving an optimal network topology to grant a sufficient KPI level for media service traffic, the Network Resource Allocator system:

- Integrates Machine Learning methods in a SDN controller to forecast resource demand and to react appropriately, so that this one can learn to instantiate the most efficient network topology in terms of KPIs.

- Learns KPI performance based on experience gathered in previous measurements. Moreover, unseen cardinalities and topologies must be concluded by the machine learning algorithm to efficiently cope with the new traffic demand under the SLA constraints.

- Works in an automated manner, thus enabling the network controller to change the network topology instantiating or removing VNFs to forward the incoming traffic in an efficient way and removing the unused parts of a network to release these resources.

The Network Resource Allocator has been tested and validated to optimize the network with the needs of forwarded services for three defined video streaming traffic patterns, Netflix and ADAS-like services, UStream and Enhanced Navigation-like services, and Skype and Car2Car-like services. Network Resource Allocator takes measures from GStreamer players and network probes and predicts network KPIs such as path bandwidth and latency directly related to QoS.

The results from the experiments carried out on the proposed Network Topology Allocator conclude that it approximates the most efficient network topology for a predicted demand assuring bandwidth and latency performance KPIs which satisfy QoS and consequent SLA constraints. Moreover, the accuracy of the results is better as the cardinality of the network is bigger and the demands in bandwidth are higher, while the fidelity drops for tiny setups and audiences. So, the more complex the infrastructure and the wider the media service demand, the more confident the approach becomes. Going beyond, the Network Topology Allocator is able to conclude unseen configurations (cardinality and topology type) and deploy them to exercise the new network.

In a nutshell, this research work provides progress beyond the state-of-the-art for QoE-driven media delivery in 5G networks. Proposed architectures, techniques and systems compiles presenting four main contributions on different aspects of the four identified challenges. First, a platform to complement the media server with a solution to create an elastic cloud of tagging resources populated by massive client devices spontaneously connected to a social media service. The goal of the media service is to engage users with an enhanced browsing and search of the contents catalogue. Second, a client-side bitrate adaptation mechanism brings fair and efficient radio link utilization in dense client environments to have a steady, consistent and unbiased QoE across all the media players sharing a common delivery path. Third, exploiting zero-latency and geo-based video analytics granted by novel 5G MEC architecture systems, a MEC system working on a real SDR setup is presented which achieves a more coordinated

and fair delivery of media services in dense client environments while shielding the media players from performance degradation and outages of the employed CDN. Finally, a network resource allocator provisions an efficient network topology and cardinality to shield the QoE of a traffic demand forecast for media services. To this end, the system is integrated on top of 5G SDN and VNF technologies to achieve agile, dynamic, preventive and automated network operation.

7.1 Future Work

During the research activities, the literature review, the design of solutions, the implementation of operational ranges and the analysis of testing results, different candidate aspects to complement or extend the research presented in this thesis were identified. The main directions are compiled as follows:

- In the research line related to media delivery on social services the blockchain and other distributed ledger technologies (DLTs) must be explored which enable parties, who are geographically distant or have no trust in each other, to interact and exchange value and information on a peer-to-peer basis with fewer to non-existent central intermediaries.

- In dense client environments, we plan to carry out future work on providing dynamic solutions while downloading a segment, in case of detection of sudden changes to network conditions, featuring a multi-pass reactive approach. Furthermore, we will analyse how the proposed solution will work with Common Media Application Format (CMAF) format [1]. CMAF will consolidate all the industry existing formats for HAS into one. The CMAF specification defines the usage of a subset of commonly used standardized media technologies and profiles. These include ISOBMFF, MPEG-4 AVC, HEVC, AAC, VTT, and Common Encryption (CENC). This specification has been proposed in the Moving Picture Experts Group (MPEG), where it is reviewed and updated by a wide range of representatives from the industry, on track to become an international standard. However,

[1] MPEG website for CMAF: `https://mpeg.chiariglione.org/standards/mpeg-a/common-media-application-format`

the feature that is more interesting to study is the performance for low latency media delivery over HAS protocols. CMAF enables chunks including just one frame, which will put completely different dynamics into play in the pull mode basis.

- In the area of MEC for video analytics we plan to perform experiments with representative CDN vendors and expand the MEC system with prediction technologies, including data fusion and aggregation in a distributed manner to forecast degradation and outages issues with CDN providers, while fitting to a well-balanced trade-off between the target QoE and the operational costs.

- Finally, regarding the system for network resource allocation, further work will be done on two main aspects. On the one hand, the penalties to avoid the system to transform the running topology into another completely different. These penalties would establish constraints to apply new setups which do not impact negatively on the network performance along the transition. On the other hand, to explore network slices for different groups of users and applications with heterogeneous SLAs. Moreover, we plan to add HAS specific QoS metrics such as initial delay, stalling time, number of quality switches and inter switching times.

Part V

Appendix

APPENDIX

A

Other Publications

List of other publications:

A.1 Broadcast delivery system for broadband media content

Title: Broadcast delivery system for broadband media content

Authors: Josu Gorostegui, Angel Martin, Mikel Zorrilla, Iñaki Alvaro and Jon Montalban

Proceedings: Broadband Multimedia Systems and Broadcasting (BMSB2017)

Pages: 1-9

Publisher: IEEE

Year: 2017

DOI: `http://dx.doi.org/10.1109/BMSB.2017.7986179`

Abstract: *The pace of technology adoption in the broadcast industry is moving forward slower than in broadband media services because of different aspects. While the broadband penetration rate is growing sharply, the required investment to embrace the broadband content catalogue into the actual broadcast solutions is a major challenge. Nowadays, broadband media services offers more content by means of Internet as the main distribution system for media exchange. Due to the success of Over-the-top (OTT)*

services, there is no doubt that a global transition is about to come. Nonetheless, the broadcast transition period and required investments are considerably higher than in the broadband market. The principal aim of this paper is to assess the key points to take into account to assure the compatibility of OTT content in a broadcast environment. This paper dives into the implementation considerations to make broadband purpose video processing frameworks ready for broadcast pipelines. Market solutions usually perform transcoding for legacy compatibility needing a big processing capacity while losing fidelity during recompression. This approach will generate on the fly live content usable in broadcast contexts and technical environments while saving storage, mantaining the original encoded signal when possible. The approach is a reliable and cost-effective media delivery method optimized for live HTTP-based Adaptive Streaming media and real time broadcast media delivery with muxing correction. In order to show completeness and validate the presented aspects this paper describes the performed implementation. For this purpose, professional tools of broadcast validation and reference broadband sequences have been used.

A.2 Dynamic Policy Based Actuation for Autonomic Management of Telecoms Networks

Title: Dynamic Policy Based Actuation for Autonomic Management of Telecoms Networks

Authors: Martin Tolan, Joe Tynan, Angel Martin, Felipe Mogollon

Proceedings: IEEE European Conference on Networks and Communications (EuCNC)

Publisher: IEEE

Year: 2017

Abstract: *With the proliferation of IoT in society there is a demand for a suitable network that is capable of supporting potentially trillions of wireless connected devices. Current 4G technologies is approaching the limits of what is possible with this generation of radio technology and a suitable replacement is required in order to support the seamless introduction of these devices and their support services. To address this it is planned that the design of the 5G network should be able to accommodate the connection requirements of*

these devices. Some of the key requirements of 5G is its ability to create a network that is highly optimized so as to make maximum use of the available spectrum and data transmission rates to give as high capacity and QoS as possible, and because of the sheer size of the network and number of devices connected, it will be necessary for the network to largely manage itself and deal with organisation, configuration, security, and optimisation issues. This paper will highlight one approach that is being taken by the CogNet H2020 project to provide a type of autonomic management based on the output of Machine Learning that can not only identify current defects but also predict future failures. This paper will also describe how the CogNet project makes self-healing in reconfigurable dynamic networks by using of policy based network management actuation for correction and prevention, and how these policies can be reconfigured based on the updated knowledge of the Machine Learning.

A.3 Can machine learning aid in delivering new use cases and scenarios in 5G?

Title: Can machine learning aid in delivering new use cases and scenarios in 5G?
Authors: Teodora Sandra Buda, Haytham Assem, Lei Xu, Angel Martin et al.
Proceedings: Network Operations and Management Symposium (NOMS)
Pages: 1279-1284
Publisher: IEEE
Year: 2016
DOI: `http://dx.doi.org/10.1109/NOMS.2016.7503003`
Abstract: *5G represents the next generation of communication networks and services, and will bring a new set of use cases and scenarios. These in turn will address a new set of challenges from the network and service management perspective, such as network traffic and resource management, big data management and energy efficiency. Consequently, novel techniques and strategies are required to address these challenges in a smarter way. In this paper, we present the limitations of the current network and service management and describe in detail the challenges that 5G is expected to face from a management perspective. The main contribution of this paper is presenting a set of use cases and scenarios*

of 5G in which machine learning can aid in addressing their management challenges. It is expected that machine learning can provide a higher and more intelligent level of monitoring and management of networks and applications, improve operational efficiencies and facilitate the requirements of the future 5G network.

A.4 CogNet: A network management architecture featuring cognitive capabilities

Title: CogNet: A network management architecture featuring cognitive capabilities
Authors: Lei Xu, Haytham Assem, Imen Grida Ben Yahia, Teodora Sandra Buda, Angel Martin et al.
Proceedings: European Conference on Networks and Communications (EuCNC)
Pages: 325-329
Publisher: IEEE
Year: 2016
DOI: http://dx.doi.org/10.1109/EuCNC.2016.7561056
Abstract: *It is expected that the fifth generation mobile networks (5G) will support both human-to-human and machine-to-machine communications, connecting up to trillions of devices and reaching formidable levels of complexity and traffic volume. This brings a new set of challenges for managing the network due to the diversity and the sheer size of the network. It will be necessary for the network to largely manage itself and deal with organisation, configuration, security, and optimisation issues. This paper proposes an architecture of an autonomic self-managing network based on Network Function Virtualization, which is capable of achieving or balancing objectives such as high QoS, low energy usage and operational efficiency. The main novelty of the architecture is the Cognitive Smart Engine introduced to enable Machine Learning, particularly (near) real-time learning, in order to dynamically adapt resources to the immediate requirements of the virtual network functions, while minimizing performance degradations to fulfill SLA requirements. This architecture is built within the CogNet European Horizon 2020 project, which refers to Cognitive Networks.*

A.5 Machine Learning for Autonomic Network Management in a Connected Cars Scenario

Title: Machine Learning for Autonomic Network Management in a Connected Cars Scenario

Authors: Gorka Velez, Marco Quartulli, Angel Martin, Oihana Otaegui, Haytham Assem

Proceedings: 10th International Workshop on Communication Technologies for Vehicles

Pages: 111-120

Publisher: Springer

Year: 2016

DOI: http://dx.doi.org/10.1007/978-3-319-38921-9_12

Abstract: *Current 4G networks are approaching the limits of what is possible with this generation of radio technology. Future 5G networks will be highly based on software, with the ultimate goal of being self-managed. Machine Learning is a key technology to reach the vision of a 5G self-managing network. This new paradigm will significantly impact on connected vehicles, fostering a new wave of possibilities. This paper presents a preliminary approach towards Autonomic Network Management on a connected cars scenario. The focus is on the machine learning part, which will allow forecasting resource demand requirements, detecting errors, attacks and outlier events, and responding and taking corrective actions.*

A.6 Live HDR Video Broadcast Production

Title: Live HDR Video Broadcast Production

Authors: Igor Olaizola, Angel Martin and Josu Gorostegui

Book: High Dynamic Range Video: Concepts, Technologies and Applications

Pages: 155-170

Publisher: Academic Press

Year: 2016

DOI: http://dx.doi.org/10.1016/B978-0-12-809477-8.00008-X

Abstract: *Among the multiple uses of HDR video, live events can get a big benefit from*

HDR video, especially when it is recorded outdoors with uncontrolled light conditions. HDR technology can provide a better adaptation of cameras to rapidly changing light conditions such as scenes that combine bright sunny areas with dark shadows (football), balls that fly with a very bright sky in the background (golf, football), extremely rapid changes in light conditions (subjective cameras in Formula 1, concerts with flashing lights), etc. All of these cases introduce two main technological challenges: the real-time factor that does not allow any manual intervention nor a computationally demanding image data processing step, and the fact that the end-to-end production pipeline has to preserve all the dynamic range information.

A.7 User interface adaptation for multi-device Web-based media applications

Title: User interface adaptation for multi-device Web-based media applications
Authors: Mikel Zorrilla, Iñigo Tamayo, Angel Martin and Ana Dominguez
Proceedings: International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)
Pages: 1-7
Publisher: IEEE
Year: 2015
DOI: `http://dx.doi.org/10.1109/BMSB.2015.7177251`
Abstract: *The quest to transform the television viewing experience into a digital media service is happening thanks to the addition of companion screens to the TV. Multi-device experiences become more intuitive and easier to use federating cooperative devices. They also bring new creative opportunities to schedule and distribute interactive content synchronised with the TV programme through any connected screen. The rise of HTML5 to develop responsive applications across multiple devices adds a significant amount of improvement enabling universal delivery. A key challenge to harness the power of navigation engaged with the story on the TV is the responsive design of a unique application spanning all the available screens. This paper presents user tests in order to explore the*

relevant parameters to create responsive User Interfaces for Web-based multi-device applications driven by media content.

A.8 Reaching devices around an HbbTV television

Title: Reaching devices around an HbbTV television

Authors: Mikel Zorrilla, Angel Martin, Iñigo Tamayo, Sean O'Halpin and Dominique Hazael-Massieux

Proceedings: International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)

Pages: 1-7

Publisher: IEEE

Year: 2014

DOI: http://dx.doi.org/10.1109/BMSB.2014.6873499

Abstract: *HbbTV takes advantage of the opportunity to expand the broadcast experience exploiting the common media content. However, the time to reach the audience in a different way has come. Aware of the privileged position of the TV in the living room, manufacturers and marketplace app developers have fostered their own bunch of solutions to integrate the big TV display with the mobile ones, to consume broadband media but ignoring the broadcast traction potential. One major challenge of all these approaches is the resource discovery and association step, where different strategies have been employed. A TV content-centric approach opens new possibilities. First, the possibility to enhance the offer services scheduled on a time basis according to the broadcast signalling. Second, the awareness of a common media been played at the same temporal and spatial environment can support the discovery and association of surrounding handheld devices. This paper analyses the capacity of common visual and acoustic environmental patterns to build enhanced discovery and association protocols, concluding a multi-step combined solution as a suitable approach for broadcast-related second screen services.*

A.9 Cloud session maintenance to synchronise HbbTV applications and home network devices

Title: Cloud session maintenance to synchronise HbbTV applications and home network devices

Authors: Mikel Zorrilla, Angel Martin, Iñigo Tamayo and Igor Olaizola

Proceedings: International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)

Pages: 1-6

Publisher: IEEE

Year: 2013

DOI: http://dx.doi.org/10.1109/BMSB.2013.6621754

Abstract: *Second screen services encourage TV audience to enjoy new forms of interaction engaging users around TV content as a main thread. This paper describes a standard-based solution for second screen services synchronised with the broadcast content. The user perceives an enhanced broadcast experience enriched with multimedia, textual and social Internet content through multiple devices. The presented end to end solution delegates to a server the cloud session maintenance in order to pair and synchronise HbbTV applications and HTML5- based second screen ones overcoming existing heterogeneous network interfaces barriers of current technological alternatives. The server decides dynamically the behaviour of the different applications regarding the user context, according to his preferences, device features and number of simultaneous views. It also manages the user interaction providing a full synchronised experience thanks to an event-driven mechanism on top of Websockets and AJAX. The paper analyses the performance of the proposed system evaluating the user interaction latency, the concurrency volume of the server and the interdependence, concluding this solution as a suitable approach for broadcast-related second screen services.*

A.10 Reference Model for Hybrid Broadcast Web3D TV

Title: Reference Model for Hybrid Broadcast Web3D TV

Authors: Igor García Olaizola, Josu Pérez, Mikel Zorrilla, Angel Martin, Maider Laka

Proceedings: 3D Web Technology (Web3D)
Pages: 177-180
Publisher: ACM
Year: 2013
DOI: http://dx.doi.org/10.1145/2466533.2466560
Abstract: *3DTV can be considered as the biggest technical revolution in TV content creation since the black and white to color transition. However, the big commercial success of current TV market has been produced around the Smart TV concept. Smart TVs connect the TV set to the web and introduce the main home multimedia display in the app world, social networks and content related interactive services. Now, this digital convergence can become the driver for boosting the success of 3DTV industry. In fact, the integration of stereoscopic TV production and Web3D seems to be the next natural step of the hybrid broadband-broadcast services. We propose in this paper a general reference model to allow the convergence of 3DTV and 3D Web by defining a general architecture and some extensions of current Smart TV concepts as well as the related standards.*

A.11 HTML5-based System for Interoperable 3D Digital Home Applications

Title: HTML5-based System for Interoperable 3D Digital Home Applications
Authors: Mikel Zorrilla, Angel Martin, Jairo R. Sanchez, Iñigo Tamayo, Igor G. Olaizola
Proceedings: Digital Home (ICDH)
Pages: 206-214
Publisher: IEEE
Year: 2012
DOI: http://dx.doi.org/10.1109/ICDH.2012.21
Abstract: *Digital home application market shifts just about every month. This means risk for developers struggling to adapt their applications to several platforms and marketplaces while changing how people experience and use their TVs, smartphones and tablets. New ubiquitous and context-aware experiences through interactive 3D applications on*

these devices engage users to interact with complex 3D scenes in virtual applications. Interactive 3D applications are boosted by emerging standards such as HTML5 and WebGL removing limitations, and transforming the Web into a horizontal application framework to tackle interoperability over the heterogeneous digital home platforms. Developers can apply their knowledge of web-based solutions to design digital home applications, removing learning curve barriers related to platform-specific APIs. However, constraints to render complex 3D environments are still present especially in home media devices. This paper provides a state-of-the-art survey of current capabilities and limitations of the digital home devices and describes a latency-driven system design based on hybrid remote and local rendering architecture, enhancing the interactive experience of 3D graphics on these thin devices. It supports interactive navigation of sophisticated 3D scenes while provides an interoperable solution that can be deployed over the wide digital home device landscape.

A.12 End to end solution for interactive on demand 3d media on home network devices

Title: End to end solution for interactive on demand 3d media on home network devices

Authors: Mikel Zorrilla, Angel Martin, Felipe Mogollon, Julen García, Igor G. Olaizola

Proceedings: Broadband Multimedia Systems and Broadcasting (BMSB)

Pages: 1-6

Publisher: IEEE

Year: 2012

DOI: `http://dx.doi.org/10.1109/BMSB.2012.6264228`

Abstract: *Smart devices have deeply modified the user consumption expectations getting used to rich interactive experiences around new media services. In this emerging landscape, TV rises as the central media device integrating the home network ecosystem. In the race to create more dynamic and customizable content, computer generated 3D graphics get a prominent position combined with video and audio to provide immersive*

and realistic environments in advanced applications where the user interaction is crucial. However, current home devices lack the required specific hardware to perform it. The proposed 3DMaaS System faces this scenario by performing 3D cloud rendering through streaming sessions with each client device, taking benefit of the Internet connectivity and video streaming management capabilities that most of thin devices have. In order to deal with the wide spectrum of device features, 3DMaaS provides a complete set of streaming formats, including RTSP, HLS and MPEG-DASH, that also fits new trends in media consumption brought by HTML5 and HbbTV. This paper presents latency performance profiling over the different streaming protocols which have a direct influence on the user interaction experience.

APPENDIX

B

Curriculum Vitae

Angel Martin is with the Department of Digital Media, Vicomtech. He received his engineering degree (2003) from University Carlos III. He collaborated with Prodys developing a standard MPEG-4 AVC/H.264 codec for DSPs (2003-2005). He started to work on Telefonica going deeper into image processing area in terms of 3D video and multiview coding (2005-2008). He worked in Innovalia as R&D Project consultant related with smart environments and ubiquitous and pervasive computing (2008-2010). Currently he is on Vicomtech managing and developing R&D projects around multimedia content services.

APPENDIX

C

Glossary

Acronyms

ACID Atomic, Consistent, Isolated, Durable

AJAX Asynchronous JavaScript and XML

API Application Programming Interface

AVC Advanced Video Coding

AWS Amazon Web Services

BAT Basic Attention Token

BS Base Station

CAPEX Capital Expenditure

CDF Cumulative Distribution Function

CDN Content Delivery Network

CENC Common Encryption Scheme

CMAF Common Media Application Format

cMOS chunk Mean Opinion Score

CP Content Providers

CQI Channel Quality Indicators

CRAN Cloud Radio Access Network

CSRF Cross-Site Request Forgery

DASH Dynamic Adaptive Streaming over HTTP

DLT Distributed Ledger Technologies

DoS Denial of Service

ECDF Empirical Cumulative Distribution Function

eMOS estimated Mean Opinion Score

EPC Evolved Packet Core

ETSI European Telecommunications Standards Institute

GOP Group Of Pictures

HAS HTTP-based Adaptive Streaming

HDR High Dynamic Range

HEVC High Efficiency Video Coding

HFR Higher Frame Rate

HSS Home Subscriber Server

HTTP Hypertext Transfer Protocol

IaaS Infrastructure as a Service

IMSI International Mobile Subscriber Identity

IoT Internet of Things

IP Internet Protocol

IT Information Technology

ITU International Telecoms Union

KPI Key Performance Indicator

LTE Long Term Evolution

M2M Machine to Machine

MaaIP Mobile as an Infrastructure Provider

MaaSB Mobile as a Service Broker

MaaSC Mobile as a Service Consumer

MaaSP Mobile as a Service Provider

MCC Mobile Cloud Computing

MCS Modulation and Coding Scheme

MEC Multi-access Edge Computing

MGC Mobile Grid Computing

MME Mobility Management Entity

MNO Mobile Network Operator

MOS Mean Opinion Score

MPD Media Presentation Description

MPEG Moving Picture Experts Group

NAT Network Address Translation

NFV Network Function Virtualization

NoSQL Not only Structured Query Language

NTP Network Time Protocol

OAI OpenAirInterface

OPEX Operational Expenses

OTT Over The Top

P2P Peer-to-Peer

PaaS Platform as a Service

PDN-GW Packet Data Network Gateway

PPS Picture Parameter Set

QL Quality Level

QoE Quality of Experience

QoS Quality of Service

RAN Radio Access Network

RB Resource Block

RDBMS Relational Database Management Systems

RTT Round Trip Time

S-GW Serving Gateway

SaaS Service as a Service

SaW Social at Work

SDN Software Defined Network

SDR Software Defined Radio

SLA Service Level Agreement

SLO Service Level Object

SON Self-Organising Network

SPS Sequence Parameter Set

SSCS SaW Scalable Cloud Server

SSIM Structural Similarity

SVC Scalable Video Coding

TTL Time To Live

UE User Equipment

UHD Ultra High Definition

USRP Universal Software Radio Peripheral

vMOS video Mean Opinion Score

VNF Virtual Network Function

WCG Wider Colour Gamut

Part VI

Bibliography

Bibliography

[5GPPP16] 5GPPP. 5G PPP Architecture Working Group: View on 5G Architecture, (2016). https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-July-2016.pdf, 2016. [Online; accessed 03-November-2017]. 36, 148

[Adhikari et al.12] V. K. Adhikari, Yang Guo, Fang Hao, M. Varvello, V. Hilt, M. Steiner, and Z. L. Zhang. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *2012 Proceedings IEEE INFOCOM*, pages 1620–1628, March 2012. 40

[Aho et al.12] Eero Aho, Kimmo Kuusilinna, Tomi Aarnio, Janne Pietiainen, and Jari Nikara. Towards real-time applications in mobile web browsers. In *Embedded Systems for Real-time Multimedia (ESTIMedia), 2012 IEEE 10th Symposium on*, pages 57–66. IEEE, 2012. 32, 40, 56

[Ahuja and Myers06] Sanjay P Ahuja and Jack R Myers. A survey on wireless grid computing. *The Journal of Supercomputing*, 37(1):3–21, 2006. 40, 46, 50

[Akhshabi et al.12] Saamer Akhshabi, Lakshmi Anantakrishnan, Ali C. Begen, and Constantine 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*, NOSSDAV '12, pages 9–14, New York, NY, USA, 2012. ACM. 11, 110

[Amazon17] Amazon. Amazon Web Services (AWS). http://aws.amazon.com, 2017. [Online; accessed 30-November-2017]. 52

[Amdahl67] Gene M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, AFIPS '67 (Spring), pages 483–485, New York, NY, USA, 1967. ACM. 68

[Analytics17] Strategy Analytics. Strategy Analytics. `https://www.strategyanalytics.com/`, 2017. [Online; accessed 30-November-2017]. 69

[Anttonen et al.11] Matti Anttonen, Arto Salminen, Tommi Mikkonen, and Antero Taivalsaari. Transforming the web into a real application platform: new technologies, emerging trends and missing pieces. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 800–807. ACM, 2011. 31, 40, 47, 53

[Armbrust et al.09] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009. 39

[Armbrust et al.10] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010. 40, 50

[Atlas18] Device Atlas. Device Atlas: Most popular smartphone screen sizes 2017. `https://deviceatlas.com/blog/most-used-smartphone-screen-resolutions-in-2017`, 2018. [Online; accessed 30-January-2018]. 129

[Awiphan et al.13] S. Awiphan, T. Muto, Y. Wang, Z. Su, and J. Katto. Video streaming over content centric networking: Experimental studies on planetlab. In *2013 Computing, Communications and IT Applications Conference (ComComAp)*, pages 19–24, April 2013. 40

[Barth et al.08] Adam Barth, Collin Jackson, and John C Mitchell. Robust defenses for cross-site request forgery. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 75–88. ACM, 2008. 60

[BAT17] BAT. Basic Attention Token website, (2017). https://basicattentiontoken.org/, 2017. [Online; accessed 17-April-2018]. 32

[Begen et al.11] A. Begen, T. Akgul, and M. Baugher. Watching video over the web: Part 1: Streaming protocols. *IEEE Internet Computing*, 15(2):54–63, March 2011. 5, 82, 110

[Bendriss et al.17] J. Bendriss, I. G. Ben Yahia, and D. Zeghlache. Forecasting and anticipating slo breaches in programmable networks. In *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pages 127–134, March 2017. 40, 152

[Bizanis and Kuipers16] Nikos Bizanis and Fernando A Kuipers. Sdn and virtualization solutions for the internet of things: A survey. *IEEE Access*, 4:5591–5606, 2016. 7, 40, 151

[Buda et al.16] T. S. Buda, H. Assem, L. Xu, D. Raz, U. Margolin, E. Rosensweig, D. R. Lopez, M. I. Corici, M. Smirnov, R. Mullins, O. Uryupina, A. Mozo, B. Ordozgoiti, A. Martin, A. Alloush, P. O'Sullivan, and I. G. Ben Yahia. Can machine learning aid in delivering new use cases and scenarios in 5g? In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pages 1279–1284, April 2016. 38, 151

[Caglar and Gokhale14] F. Caglar and A. Gokhale. ioverbook: Intelligent resource-overbooking to support soft real-time applications in the cloud. In *2014 IEEE 7th International Conference on Cloud Computing*, pages 538–545, June 2014. 38, 40, 150

[CanIUse17] CanIUse. CanIUse.com Support of WebGL in different Web Browsers. http://caniuse.com/, 2017. [Online; accessed 30-November-2017]. 67

[Catak and Balaban13] F Ozgur Catak and M Erdal Balaban. Cloudsvm: training an svm classifier in cloud computing systems. In *Pervasive Computing and the Networked World*, pages 57–68. Springer, 2013. 40, 53

[Cedexis17] Cedexis. Cedexis website. https://www.cedexis.com/, 2017. [Online; accessed 30-November-2017]. 35, 40, 116

[Chandra et al.13] Aniruddha Chandra, Jon Weissman, and Benjamin Heintz. Decentralized edge clouds. *Internet Computing, IEEE*, 17(5):70–73, 2013. 40, 53, 54

[Chang et al.15] H. B. Chang, I. Rubin, S. Colonnese, F. Cuomo, and O. Hadar. Joint adaptive rate and scheduling for video streaming in multi-cell cellular wireless networks. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2015. 35, 40, 117

[Chávez-Santiago et al.15] Raúl Chávez-Santiago, Michał Szydełko, Adrian Kliks, Fotis Foukalas, Yoram Haddad, Keith E. Nolan, Mark Y. Kelly, Moshe T. Masonta, and Ilangko Balasingham. 5g: The convergence of wireless communications. *Wireless Personal Communications*, 83(3):1617–1642, Aug 2015. 37, 40, 148

[Checko et al.15] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann. Cloud ran for mobile networks - a technology overview. *IEEE Communications Surveys Tutorials*, 17(1):405–426, Firstquarter 2015. 116

[Chen and Liu16] Y. Chen and G. Liu. Playout continuity driven framework for http adaptive streaming over lte networks. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–1, 2016. 36, 40, 117, 131

[Chen et al.13] Xiang Chen, Yiran Chen, Zhan Ma, and Felix C. A. Fernandes. How is energy consumed in smartphone display applications? In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, HotMobile '13, pages 3:1–3:6, New York, NY, USA, 2013. ACM. 22, 40, 58

[Chen et al.15] Y. Chen, K. Wu, and Q. Zhang. From qos to qoe: A tutorial on video quality assessment. *IEEE Communications Surveys Tutorials*, 17(2):1126–1165, Secondquarter 2015. 40, 156

[Chen et al.16a] Dongliang Chen, Jonathon Edstrom, Xiaowei Chen, Wei Jin, Jinhui Wang, and Na Gong. Data-driven low-cost on-chip memory with adaptive power-quality trade-off for mobile video streaming. In *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ISLPED '16, pages 188–193, New York, NY, USA, 2016. ACM. 34, 114

[Chen et al.16b] Junyang Chen, Mostafa Ammar, Marwan Fayed, and Rodrigo Fonseca. Client-driven network-level qoe fairness for encrypted 'dash-s'. In *Proceedings of the 2016 Workshop on QoE-based Analysis and Management of Data Communication Networks*, Internet-QoE '16, pages 55–60, New York, NY, USA, 2016. ACM. 33, 86, 88, 113

[Chiang and Zhang16] M. Chiang and T. Zhang. Fog and iot: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, Dec 2016. 116

[Chiariotti et al.16] Federico Chiariotti, Stefano D'Aronco, Laura Toni, and Pascal Frossard. Online learning adaptation strategy for dash clients. In *Proceedings of the 7th International Conference on Multimedia Systems*, MMSys '16, pages 8:1–8:12, New York, NY, USA, 2016. ACM. 34, 40, 85, 86, 87, 88, 114, 131

[Claeys et al.14a] M. Claeys, S. Latre, J. Famaey, and F. De Turck. Design and evaluation of a self-learning http adaptive video streaming client. *IEEE Communications Letters*, 18(4):716–719, April 2014. 34, 40, 87, 97, 98, 114, 115, 132, 135, 172

[Claeys et al.14b] Maxim Claeys, Steven Latré, Jeroen Famaey, Tingyao Wu, Werner Van Leekwijck, and Filip De Turck. Design and optimisation of a faq-learning-based http adaptive streaming client. *Connect. Sci*, 26(1):25–43, January 2014. 33, 40, 86, 114

[Conti and Kumar10] Marco Conti and Mohan Kumar. Opportunities in opportunistic computing. *Computer*, 43(1):42–50, 2010. 31, 40, 49

[Conviva17] Conviva. Conviva website. `http://www.conviva.com/`, 2017. [Online; accessed 30-November-2017]. 35, 116

[Cushing et al.13] Reginald Cushing, Ganeshwara Herawan Hananda Putra, Spiros Koulouzis, Adam Belloum, Marian Bubak, and Cees De Laat. Distributed computing on an ensemble of browsers. *Internet Computing, IEEE*, 17(5):54–61, 2013. 40, 54

[De Francisci Morales et al.11] Gianmarco De Francisci Morales, Aristides Gionis, and Mauro Sozio. Social content matching in mapreduce. *Proceedings of the VLDB Endowment*, 4(7):460–469, 2011. 40, 53

[DMTF17] DMTF. Cloud Infrastructure Management Interface (CIMI). http://dmtf.org/standards/cloud, 2017. [Online; accessed 30-November-2017]. 52

[Docker17] Docker. Docker website, (2017). https://www.docker.com/, 2017. [Online; accessed 03-November-2017]. 164

[Domínguez et al.17] A. Domínguez, M. Agirre, J. Flórez, A. Lafuente, I. Tamayo, and M. Zorrilla. Deployment of a hybrid broadcast-internet multi-device service for a live tv programme. *IEEE Transactions on Broadcasting*, PP(99):1–11, 2017. 145

[Du and Swamy16] Ke-Lin Du and MNS Swamy. Simulated annealing. In *Search and Optimization by Metaheuristics*, pages 29–36. Springer, 2016. 160

[Eaton17] John W. Eaton. GNU Octave Website. https://www.gnu.org/software/octave/, 2017. [Online; accessed 16-February-2017]. 74

[EC15] EC. H2020 CogNet 5GPPP Phase 1 project. http://www.cognet.5g-ppp.eu/, 2015. 10

[Edwards et al.97] T Edwards, D Tansley, R Frank, and N Davey. Traffic trends analysis using neural networks. In *Procs of the Int Workshop on Applications of Neural Networks to Telecommunications*, 1997. 40, 151

[Emeakaroha et al.10] V. C. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar. Low level metrics to high level slas - lom2his framework: Bridging the gap between monitored metrics and sla parameters in cloud environments. In *2010 International Conference on High Performance Computing Simulation*, pages 48–54, June 2010. 40, 150

[Ericsson15] Ericsson. Ericsson: Ericsson Mobility Report: Growing up streaming, (2015). https://www.ericsson.com/sectionspage/growing-up-streaming_498116887_c, 2015. [Online; accessed 03-November-2017]. 10, 144

[ESnet and Laboratory17] ESnet and Lawrence Berkeley National Laboratory. Iperf website, (2017). https://iperf.fr/, 2017. [Online; accessed 03-November-2017]. 160

[Essaili et al.15] 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*, 25(6):988–1001, June 2015. 35, 40, 117, 131

[Ethereum17] Ethereum. Ethereum website, (2017). `https://www.ethereum.org/`, 2017. [Online; accessed 17-April-2018]. 32

[ETSI10] ETSI. European Telecommunications Standards Institute (ETSI), Technical Specification LTE: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (3GPP TS 36.213 version 10.1.0 Release 10) (2010). `http://www.etsi.org/deliver/etsi_ts/136200_136299/136213/10.01.00_60/ts_136213v100100p.pdf`, 2010. [Online; accessed 03-November-2017]. xxv, 118, 123

[ETSI15] ETSI. ETSI TS 103 285 V1.1.1 (2015-05). MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks, 2015. 126

[ETSI17a] ETSI. ETSI. Mobile Edge Computing. `http://www.etsi.org/images/files/ETSITechnologyLeaflets/MobileEdgeComputing.pdf`, 2017. [Online; accessed 03-November-2017]. 110, 111

[ETSI17b] ETSI. ETSI website of 5G, (2017). `http://www.etsi.org/technologies-clusters/technologies/5g`, 2017. [Online; accessed 03-November-2017]. 148

[ETSI18] ETSI. ETSI website of Multi-access Edge Computing, (2018). `http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing`, 2018. [Online; accessed 17-April-2018]. 6

[f. Lai et al.15] C. f. Lai, R. h. Hwang, H. c. Chao, M. M. Hassan, and A. Alamri. A buffer-aware http live streaming approach for sdn-enabled 5g wireless networks. *IEEE Network*, 29(1):49–55, Jan 2015. 35, 40, 117

[Fajardo et al.15] J. O. Fajardo, I. Taboada, and F. Liberal. Improving content delivery efficiency through multi-layer mobile edge adaptation. *IEEE Network*, 29(6):40–46, Nov 2015. 36, 40, 117

[Farzaneh and Moghaddam08] N. Farzaneh and M. H. Y. Moghaddam. Virtual topology reconfiguration of wdm optical networks using fuzzy logic control. In *2008 International Symposium on Telecommunications*, pages 504–509, Aug 2008. 40, 151

[Fette and Melnikov11] Ian Fette and Alexey Melnikov. The websocket protocol. 2011. 40, 47, 64

[Fortune and Wyllie78] Steven Fortune and James Wyllie. Parallelism in random access machines. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 114–118. ACM, 1978. 69

[Foster et al.01] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International journal of high performance computing applications*, 15(3):200–222, 2001. 40, 49

[Foundation05] Apache Software Foundation. Apache Hadoop framework. `http://hadoop.apache.org/`, 2005. [Online; accessed 30-November-2017]. 40, 53

[Foundation12] Open Networking Foundation. Open Networking Foundation White Paper (2012) Software-Defined Networking: The New Norm for Network., 2012. 7, 37, 146

[Foundation13] Open Networking Foundation. Open Networking Foundation (2013) OpenFlow switch specification, version 1.4.0., 2013. 7, 37, 146

[Foundation17a] Apache Software Foundation. Apache Cordova Website. `https://cordova.apache.org/`, 2017. [Online; accessed 16-February-2017]. 40, 47

[Foundation17b] Apache Software Foundation. Kafka website, (2017). `https://kafka.apache.org/`, 2017. [Online; accessed 03-November-2017]. 161

[Foundation17c] Linux Foundation. OpenDaylight website, (2017). `https://www.opendaylight.org/`, 2017. [Online; accessed 03-November-2017]. 15, 37, 161

[Foundation17d] Linux Foundation. vSwitch website, (2017). `http://openvswitch.org/`, 2017. [Online; accessed 03-November-2017]. 37, 161

[Foundation17e] Open Networking Foundation. OpenFlow website, (2017). https://www.opennetworking.org/sdn-resources/openflow, 2017. [Online; accessed 03-November-2017]. 37, 149

[Garrett et al.05] Jesse James Garrett et al. Ajax: A new approach to web applications. 2005. 40, 47, 65

[Ghosh et al.10] Arunabha Ghosh, Jun Zhang, Jeffrey G Andrews, and Rias Muhamed. *Fundamentals of LTE*. Pearson Education, 2010. 128

[GStreamer17] GStreamer. GStreamer website, (2017). https://gstreamer.freedesktop.org/, 2017. [Online; accessed 03-November-2017]. 160

[Gum12] Gum Gum. Gum Gum in-image advertising platform. http://gumgum.com/, 2012. [Online; accessed 30-November-2017]. 40, 54

[Hagos16] Desta Haileselassie Hagos. The performance of network-controlled mobile data offloading from lte to wifi networks. *Telecommun. Syst.*, 61(4):675–694, April 2016. 11

[Han et al.11] Jing Han, E Haihong, Guan Le, and Jian Du. Survey on nosql database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, pages 363–366. IEEE, 2011. 40, 56

[Hat17] Red Hat. Ansible website, (2017). https://www.ansible.com/, 2017. [Online; accessed 03-November-2017]. 164

[Hernandez-Valencia et al.15] E. Hernandez-Valencia, S. Izzo, and B. Polonsky. How will nfv/sdn transform service provider opex? *IEEE Network*, 29(3):60–67, May 2015. 40, 150

[Huang et al.13] Dijiang Huang, Tianyi Xing, and Huijun Wu. Mobile cloud computing service models: a user-centric approach. *Network, IEEE*, 27(5):6–11, 2013. 31, 40, 46, 51

[Huang et al.18] 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(99):1–12, 2018. 158

[Ietf10] Ietf. RFC 6020: YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF), October 2010. 15

[Inc17a] Cisco Inc. Cisco Inc: Global Mobile Data Traffic Forecast Update, 2016-2021. (2017). https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, 2017. [Online; accessed 15-September-2017]. 10, 81, 110, 145

[Inc17b] Cisco Inc. Cisco Inc: Visual Networking Index: Forecast and Methodology, 2016-2021. (2017). https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html, 2017. [Online; accessed 15-September-2017]. 10, 81, 110, 144

[Ismail et al.13] M. Ismail, A. Abdrabou, and W. Zhuang. Cooperative decentralized resource allocation in heterogeneous wireless access medium. *IEEE Transactions on Wireless Communications*, 12(2):714–724, February 2013. 15, 38, 40, 142, 147, 151

[ISO12] ISO. ISO/IEC 23009-1 (2012), Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats, Draft Amd.1 (available as w13497), 2012. 126

[ISO16] ISO. ISO/IEC 23001-7 (2016). Information technology — MPEG systems technologies — Part 7: Common encryption in ISO base media file format files, 2016. 119

[ITU] ITU. ITU-T Recommendation P.800: Mean opinion score (MOS) terminology. (2016), year = 2016. 34, 86, 114

[Jarp et al.12] Sverre Jarp, Alfio Lazzaro, and Andrzej Nowak. The future of commodity computing and many-core versus the interests of hep software. In *Journal of Physics: Conference Series*, volume 396, page 052058. IOP Publishing, 2012. 40, 55

[Jeon et al.12] Won Jeon, Tasneem Brutch, and Simon Gibbs. Webcl for hardware-accelerated web applications. In *TIZEN Developer Conference May*, pages 7–9, 2012. 31, 40, 47, 55

[Jiang et al.14] 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*, 22(1):326–340, Feb 2014. 33, 40, 85, 114

[Kalooga12] Kalooga. Kalooga discovery service for image galleries. `http://www.kalooga.com/`, 2012. [Online; accessed 30-November-2017]. 40, 54

[Kathirgamanathan et al.15] P. Kathirgamanathan, L. M. Bushby, M. Kumaraverl, S. Ravichandran, and S. Surendrakumar. Electroluminescent organic and quantum dot leds: The state of the art. *Journal of Display Technology*, 11(5):480–493, May 2015. 10, 81

[Khronos12] Khronos. OpenCL Specification. `http://www.khronos.org/registry/cl/`, 2012. [Online; accessed 30-November-2017]. 40, 52

[Khronos17] Khronos. WebGL Security Website. `https://www.khronos.org/webgl/security/`, 2017. [Online; accessed 16-February-2017]. 61

[Kim and Feamster13] H. Kim and N. Feamster. Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2):114–119, February 2013. 7, 37, 40, 147

[Kirkpatrick et al.83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. 156

[Klaine et al.17] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza. A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Communications Surveys Tutorials*, 19(4):2392–2431, Fourthquarter 2017. 37, 38, 40, 147, 151

[Kosba et al.16] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 839–858, May 2016. 32, 40

[Kourtis et al.17] M. A. Kourtis, H. Koumaras, G. Xilouris, and F. Liberal. An nfv-based video quality assessment method over 5g small cell networks. *IEEE MultiMedia*, 24(4):68–78, October 2017. 40, 118

[Lab17] Kaspersky Lab. Miners on the Rise, (2017). https://securelist.com/miners-on-the-rise/81706/, 2017. [Online; accessed 17-April-2018]. 32

[Langhans et al.13] Philipp Langhans, Christoph Wieser, and François Bry. Crowdsourcing mapreduce: Jsmapreduce. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 253–256. International World Wide Web Conferences Steering Committee, 2013. 40, 54

[LegitReviews12] LegitReviews. Google Nexus 7 Tablet Review. http://www.legitreviews.com/article/1988/2/, 2012. [Online; accessed 30-November-2017]. 69

[Lentisco et al.17a] 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*, 5:20684–20694, 2017. 40, 115

[Lentisco et al.17b] 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*, 5:20684–20694, 2017. 34

[Li et al.14a] B. Li, H. Li, L. Li, and J. Zhang. λ domain rate control algorithm for high efficiency video coding. *IEEE Transactions on Image Processing*, 23(9):3841–3854, Sept 2014. 83

[Li et al.14b] 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*, 32(4):719–733, April 2014. 33, 40, 85, 114, 131

[Li et al.14c] Zhi Li, Ali C. Begen, Joshua Gahm, Yufeng Shan, Bruce Osler, and David Oran. Streaming video over http with consistent quality. In *Proceedings of the 5th ACM Multimedia Systems Conference*, MMSys '14, pages 248–258, New York, NY, USA, 2014. ACM. 34, 40, 85, 87, 88, 114, 131

[Li et al.16] Y. Li, P. A. Frangoudis, Y. Hadjadj-Aoul, and P. Bertin. A mobile edge computing-based architecture for improved adaptive http video delivery. In *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–6, Oct 2016. 36, 40, 117, 126

[Lin et al.10] Heshan Lin, Xiaosong Ma, Jeremy Archuleta, Wu-chun Feng, Mark Gardner, and Zhe Zhang. Moon: Mapreduce on opportunistic environments. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pages 95–106. ACM, 2010. 40, 53

[Liu and Wei16] Z. Liu and Y. Wei. Hop-by-hop adaptive video streaming in content centric network. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2016. 40

[Liu et al.11] Chenghao Liu, Imed Bouazizi, and Moncef Gabbouj. Rate adaptation for adaptive http streaming. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, MMSys '11, pages 169–174, New York, NY, USA, 2011. ACM. 33, 40, 113

[Liu et al.16a] J. Liu, Z. Jiang, N. Kato, O. Akashi, and A. Takahara. Reliability evaluation for nfv deployment of future mobile broadband networks. *IEEE Wireless Communications*, 23(3):90–96, June 2016. 11, 40, 146

[Liu et al.16b] J. Liu, R. Xie, and F. R. Yu. Resource allocation and user association for http adaptive streaming in heterogeneous cellular networks with small cells. *China Communications*, 13(9):1–11, Sept 2016. 112

[Liu13] Huan Liu. Big data drives cloud adoption in enterprise. *IEEE internet computing*, (4):68–71, 2013. 40, 50

[MacWilliam and Cecka13] Tommy MacWilliam and Cris Cecka. Crowdcl: Web-based volunteer computing with webcl. In *High Performance Extreme Computing Conference (HPEC), 2013 IEEE*, pages 1–6. IEEE, 2013. 40, 56

[Maillé and Schwartz16] P. Maillé and G. Schwartz. Content providers volunteering to pay network providers: Better than neutrality? In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 484–489, April 2016. 82, 110

[Makris et al.15] N. Makris, C. Zarafetas, S. Kechagias, T. Korakis, I. Seskar, and L. Tassiulas. Enabling open access to lte network components; the nitos testbed paradigm.

In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, pages 1–6, April 2015. 126

[Martin et al.17] Angel Martin, Roberto Viola, Josu Gorostegui, Mikel Zorrilla, Julian Florez, and Jon Montalban. Lamb-dash: a dash-hevc adaptive streaming algorithm in a sharing bandwidth environment for heterogeneous contents and dynamic connections in practice. *Journal of Real-Time Image Processing*, Oct 2017. 127, 131, 132, 160, 172

[Mathematics and Science96] ANL Mathematics and Computer Science. MPI: Message Passing Interface. `http://www.mcs.anl.gov/research/projects/mpi/`, 1996. [Online; accessed 30-November-2017]. 40, 55

[McKay and Wormald90] Brendan D. McKay and Nicholas C. Wormald. Uniform generation of random regular graphs of moderate degree. *J. Algorithms*, 11(1):52–67, February 1990. 156

[Miller et al.16] Konstantin Miller, Abdel-Karim Al-Tamimi, and Adam Wolisz. Qoe-based low-delay live streaming using throughput predictions. *ACM Trans. Multimedia Comput. Commun. Appl.*, 13(1):4:1–4:24, October 2016. xxii, 33, 40, 85, 87, 88, 89, 91, 99, 114, 131

[Mininet17] Mininet. Mininet website, (2017). `http://mininet.org/`, 2017. [Online; accessed 03-November-2017]. 160

[MIT17] MIT. Hijacking Computers to Mine Cryptocurrency Is All the Rage, (2017). `https://securelist.com/miners-on-the-rise/81706/`, 2017. [Online; accessed 17-April-2018]. 33

[Mohaisen et al.14] A. Mohaisen, H. Tran, A. Chandra, and Y. Kim. Trustworthy distributed computing on social networks. *IEEE Transactions on Services Computing*, 7(3):333–345, July 2014. 32, 40

[Mohri et al.12] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012. 37, 40, 141, 147

[Mok et al.11] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang. Measuring the quality of experience of http video streaming. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 485–492, May 2011. 34, 40, 97, 114, 172

[MongoDB17] MongoDB. MongoDB website, (2017). `https://www.mongodb.com/`, 2017. [Online; accessed 03-November-2017]. 160

[Moreno-Vozmediano et al.13] Rafael Moreno-Vozmediano, Rubén S Montero, and Ignacio M Llorente. Key challenges in cloud computing: Enabling the future internet of services. *Internet Computing, IEEE*, 17(4):18–25, 2013. 40, 52

[NAPTE14] NAPTE. New NAPTE and CEA research finds show producers and creators see second screen becoming permanent part of viewing experience. `https://www.natpe.com/press/release/130`, 2014. [Online; accessed 30-November-2017]. 57

[Neumann et al.11] Dirk Neumann, Christian Bodenstein, Omer F Rana, and Ruby Krishnaswamy. Stacee: enhancing storage clouds using edge devices. In *Proceedings of the 1st ACM/IEEE workshop on Autonomic computing in economics*, pages 19–26. ACM, 2011. 31, 40, 45, 51

[Nguyen et al.16] Van-Giang Nguyen, Truong-Xuan Do, and Younghan Kim. Sdn and virtualization-based lte mobile network architectures: A comprehensive survey. *Wirel. Pers. Commun.*, 86(3):1401–1438, February 2016. 37, 40, 148

[Nielsen14a] Nielsen. Sports Fans Amplify The Action Across Screens. `http://www.nielsensocial.com/sports-fans-amplify-the-action-across-screens/`, 2014. [Online; accessed 30-November-2017]. 57

[Nielsen14b] Nielsen. Who's tweeting about tv? `http://www.nielsensocial.com/whos-tweeting-about-tv/`, 2014. [Online; accessed 30-November-2017]. 57

[Nikaein et al.14] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. Openairinterface: A flexible platform for 5g research. *SIGCOMM Comput. Commun. Rev.*, 44(5):33–38, October 2014. 127

[NokiaResearch17] NokiaResearch. Nokia WebCL Extension for Firefox. https://github.com/toaarnio/webcl-firefox, 2017. [Online; accessed 30-November-2017]. 67

[Nurmi et al.09] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. The eucalyptus open-source cloud-computing system. In *Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on*, pages 124–131. IEEE, 2009. 50

[Nvidia07] Nvidia. CUDA computing platform and programming model. http://www.nvidia.com/object/cuda_home_new.html, 2007. [Online; accessed 30-November-2017]. 40, 55

[of California99] University of California. SETI@home Project. http://setiathome.berkeley.edu/, 1999. [Online; accessed 30-November-2017]. 31, 40, 50

[of Virginia Computer Graphics Lab12] University of Virginia Computer Graphics Lab. Hadoop Image Processing Interface. http://hipi.cs.virginia.edu/, 2012. [Online; accessed 30-November-2017]. 40, 53

[OGF17] OGF. Open Cloud Computing Interface (OCCI). http://occi-wg.org, 2017. [Online; accessed 30-November-2017]. 52

[Olaizola et al.14] Igor G Olaizola, Marco Quartulli, Julian Florez, and Basilio Sierra. Trace transform based method for color image domain identification. *Multimedia, IEEE Transactions on*, 16(3):679–685, 2014. 68

[OpenMP13] OpenMP. OpenMP Specification. http://openmp.org/wp/openmp-specifications/, 2013. [Online; accessed 30-November-2017]. 40, 55

[OpenStack17] OpenStack. OpenStack website, (2017). http://www.openstack.org/, 2017. [Online; accessed 03-November-2017]. 37, 149

[Orosz et al.14] P. Orosz, T. Skopkó, Z. Nagy, P. Varga, and L. Gyimóthi. A case study on correlating video qos and qoe. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–5, May 2014. 40, 158

[Pang et al.15] Z. Pang, L. Sun, Z. Wang, E. Tian, and S. Yang. A survey of cloudlet based mobile computing. In *2015 International Conference on Cloud Computing and Big Data (CCBD)*, pages 268–275, Nov 2015. 40, 51

[Pardalos93] Panos M Pardalos. *Complexity in numerical optimization*. World Scientific, 1993. 92

[Park et al.14] C. Park, M. Lee, N. A. Nasir, and S. H. Jeong. Video quality measurement in content-centric wireless networks. In *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 532–534, Oct 2014. 40, 148

[Petrangeli et al.15] Stefano Petrangeli, Jeroen Famaey, Maxim Claeys, Steven Latré, and Filip De Turck. Qoe-driven rate adaptation heuristic for fair adaptive video streaming. *ACM Trans. Multimedia Comput. Commun. Appl.*, 12(2):28:1–28:24, October 2015. 33, 35, 40, 86, 113, 117

[PhoneGap17] PhoneGap. PhoneGap Website. `http://phonegap.com/`, 2017. [Online; accessed 16-February-2017]. 40, 47

[Project17] OpenNebula Project. OpenNebula Project. `http://opennebula.org/`, 2017. [Online; accessed 30-November-2017]. 40, 52

[Qian et al.17] L. Qian, Z. Cheng, Z. Fang, L. Ding, F. Yang, and W. Huang. A qoe-driven encoder adaptation scheme for multi-user video streaming in wireless networks. *IEEE Transactions on Broadcasting*, 63(1):20–31, March 2017. 146

[Quinlan et al.15] J. J. Quinlan, A. H. Zahran, K. K. Ramakrishnan, and C. J. Sreenan. Delivery of adaptive bit rate video: balancing fairness, efficiency and quality. In *The 21st IEEE International Workshop on Local and Metropolitan Area Networks*, pages 1–6, April 2015. 79, 110

[Rainer and Timmerer14] Benjamin Rainer and Christian Timmerer. Quality of experience of web-based adaptive http streaming clients in real-world environments using crowdsourcing. In *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, VideoNext '14, pages 19–24, New York, NY, USA, 2014. ACM. 40, 91

[Rhaiem et al.15] O. B. Rhaiem, L. C. Fourati, and A. Masmoudi. Content-centric network-based manet for streaming video transmission. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 1022–1027, Oct 2015. 40, 148

[Rong et al.16] Bo Rong, Songlin Sun, and Michel Kadoch. Traffic prediction for reliable and resilient video communications over multi-location wmns. *Journal of Network and Systems Management*, 24(3):516–533, Jul 2016. 40

[Rubin et al.15] I. Rubin, S. Colonnese, F. Cuomo, F. Calanca, and T. Melodia. Mobile http-based streaming using flexible lte base station control. In *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–9, June 2015. 35, 40, 117, 131

[Saad et al.15] M. A. Saad, M. H. Pinson, D. G. Nicholas, N. V. Kets, G. V. Wallendael, R. D. Silva, R. V. Jaladi, and P. J. Corriveau. Impact of camera pixel count and monitor resolution perceptual image quality. In *2015 Colour and Visual Computing Symposium (CVCS)*, pages 1–6, Aug 2015. 10, 81

[Sandhir and Mitchell08] P. Sandhir and K. Mitchell. A neural network demand prediction scheme for resource allocation in cellular wireless systems. In *2008 IEEE Region 5 Conference*, pages 1–6, April 2008. 40, 151

[SDxCentral17] SDxCentral. OpNFV website, (2017). `https://www.sdxcentral.com/listings/opnfv/`, 2017. [Online; accessed 03-November-2017]. 37, 149

[Serrano et al.16] Damián Serrano, Sara Bouchenak, Yousri Kouki, Frederico Alvares de Oliveira Jr., Thomas Ledoux, Jonathan Lejeune, Julien Sopena, Luciana Arantes, and Pierre Sens. Sla guarantees for cloud services. *Future Gener. Comput. Syst.*, 54(C):233–246, January 2016. 40, 145, 150

[Seufert et al.15] 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*, 17(1):469–492, Firstquarter 2015. 6, 22, 40, 82, 87, 88, 110

[SNIA17] SNIA. Cloud Data Management Interface (CDMI). http://www.snia.org/cdmi, 2017. [Online; accessed 30-November-2017]. 52

[Sodagar11] I. Sodagar. The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia*, 18(4):62–67, April 2011. 10, 82, 83, 126

[Spectrum17a] IEEE Spectrum. Blockchains: How They Work and Why They'll Change the World, (2017). https://spectrum.ieee.org/computing/networks/blockchains-how-they-work-and-why-theyll-change-the-world, 2017. [Online; accessed 17-April-2018]. 32

[Spectrum17b] IEEE Spectrum. Do You Need a Blockchain?, (2017). https://spectrum.ieee.org/computing/networks/do-you-need-a-blockchain, 2017. [Online; accessed 17-April-2018]. 32

[Stone et al.10] John E Stone, David Gohara, and Guochun Shi. Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in science & engineering*, 12(1-3):66–73, 2010. 31, 40, 52

[Sullivan et al.12] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, Dec 2012. 83, 129

[Sun et al.15] S. Sun, M. Kadoch, L. Gong, and B. Rong. Integrating network function virtualization with sdr and sdn for 4g/5g networks. *IEEE Network*, 29(3):54–59, May 2015. 40, 150

[Sweeney et al.11] Chris Sweeney, Liu Liu, Sean Arietta, and Jason Lawrence. Hipi: a hadoop image processing interface for image-based mapreduce tasks. *Chris. University of Virginia*, 2011. 40, 53

[Szabo et al.15] R. Szabo, M. Kind, F. J. Westphal, H. Woesner, D. Jocha, and A. Csaszar. Elastic network functions: opportunities and challenges. *IEEE Network*, 29(3):15–21, May 2015. 7, 40, 151

[Tan et al.13] Wei Tan, M Brian Blake, Iman Saleh, and Schahram Dustdar. Social-network-sourced big data analytics. *IEEE Internet Computing*, (5):62–69, 2013. 40, 56

[TID17] TID. OpenMano website, (2017). `https://github.com/nfvlabs/openmano`, 2017. [Online; accessed 03-November-2017]. 15, 37, 149

[Tilkov and Vinoski10] Stefan Tilkov and Steve Vinoski. Node. js: Using javascript to build high-performance network programs. *IEEE Internet Computing*, (6):80–83, 2010. 40, 54

[TomsHardware11] TomsHardware. AMD Bulldozer Review: FX-8150 Gets Tested. `http://www.tomshardware.com/reviews/fx-8150-zambezi-bulldozer-990fx,3043-14.html`, 2011. [Online; accessed 30-November-2017]. 69

[Toni et al.15] Laura Toni, Ramon Aparicio-Pardo, Karine Pires, Gwendal Simon, Alberto Blanc, and Pascal Frossard. Optimal selection of adaptive streaming representations. *ACM Trans. Multimedia Comput. Commun. Appl.*, 11(2s):43:1–43:26, February 2015. 40, 87, 88

[Valiant90] Leslie G Valiant. A bridging model for parallel computation. *Communications of the ACM*, 33(8):103–111, 1990. 69

[Vega et al.18] M. Torres Vega, C. Perra, and A. Liotta. Resilience of video streaming services to network impairments. *IEEE Transactions on Broadcasting*, PP(99):1–15, 2018. 158

[Vimeo17] Vimeo. Vimeo Web site. `https://vimeo.com/`, 2017. [Online; accessed 30-November-2017]. 21, 58

[Vleeschauwer et al.13] D. De Vleeschauwer, H. Viswanathan, A. Beck, S. Benno, G. Li, and R. Miller. Optimization of http adaptive streaming over mobile cellular networks. In *2013 Proceedings IEEE INFOCOM*, pages 898–997, April 2013. 35, 40, 117

[Vriendt et al.13] J. De Vriendt, D. De Vleeschauwer, and D. Robinson. Model for estimating qoe of video delivered using http adaptive streaming. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 1288–1293, May 2013. 34, 40, 87, 97, 114

[Wainio and Seppänen16] P. Wainio and K. Seppänen. Self-optimizing last-mile backhaul network for 5g small cells. In *2016 IEEE International Conference on Communications Workshops (ICC)*, pages 232–239, May 2016. 40, 151

[Wan et al.11] Shuai Wan, Junhui Hou, and Fuzheng Yang. *Two-Dimensional Rate Model for Video Coding*, pages 155–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 83

[Wang et al.17] C. C. Wang, Z. N. Lin, S. R. Yang, and P. Lin. Mobile edge computing-enabled channel-aware video streaming for 4g lte. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 564–569, June 2017. 36, 40, 118

[Williams et al.13] Dan Williams, Hani Jamjoom, and Hakim Weatherspoon. Plug into the supercloud. *Internet Computing, IEEE*, 17(2):28–34, 2013. 40, 50

[Xamarin17] Xamarin. Xamarin Website. `https://www.xamarin.com/`, 2017. [Online; accessed 16-February-2017]. 40, 47

[Xiao et al.13] Z. Xiao, W. Song, and Q. Chen. Dynamic resource allocation using virtual machines for cloud computing environment. *IEEE Transactions on Parallel and Distributed Systems*, 24(6):1107–1117, June 2013. 39

[Xiph.Org17] Xiph.Org. Xiph.Org Foundation Video test sequence repository, (2017). `http://media.xiph.org/video/derf`, 2017. [Online; accessed 03-November-2017]. 129, 162

[Xu et al.13] C. Xu, T. Liu, J. Guan, H. Zhang, and G. M. Muntean. Cmt-qa: Quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks. *IEEE Transactions on Mobile Computing*, 12(11):2193–2205, Nov 2013. 7, 40, 146

[Xu et al.16] L. Xu, H. Assem, I. G. B. Yahia, T. S. Buda, A. Martin, D. Gallico, M. Biancani, A. Pastor, P. A. Aranda, M. Smirnov, D. Raz, O. Uryupina, A. Mozo, B. Ordozgoiti, M. I. Corici, P. O'Sullivan, and R. Mullins. Cognet: A network management architecture featuring cognitive capabilities. In *2016 European Conference on Networks and Communications (EuCNC)*, pages 325–329, June 2016. 161

[Yan et al.17] Z. Yan, J. Xue, and C. W. Chen. Prius: Hybrid edge cloud and client adaptation for http adaptive streaming in cellular networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(1):209–222, Jan 2017. 36, 40, 117, 131

[Yang et al.11] Chao-Tung Yang, Chih-Lin Huang, and Cheng-Fang Lin. Hybrid cuda, openmp, and mpi parallel programming on multicore gpu clusters. *Computer Physics Communications*, 182(1):266–269, 2011. 40, 55

[Youtube17] Youtube. YouTube Web site. `https://www.youtube.com/`, 2017. [Online; accessed 30-November-2017]. 21, 58

[Yu et al.17] L. Yu, T. Tillo, and J. Xiao. Qoe-driven dynamic adaptive video streaming strategy with future information. *IEEE Transactions on Broadcasting*, 63(3):523–534, Sept 2017. 146, 158

[Zhang et al.16] Jingyu Zhang, Gan Fang, Minyi Guo, and Chunyi Peng. How video streaming consumes power in 4G LTE networks. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, June 2016. 22, 40, 58

[Zhang et al.17] Lei Zhang, Di Fu, Jiangchuan Liu, Edith Cheuk-Han Ngai, and Wenwu Zhu. On energy-efficient offloading in mobile cloud for real-time video applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(1):170–181, 2017. 40, 51, 58

[Zorrilla et al.13] Mikel Zorrilla, Andrew Martin, Inigo Tamayo, Naiara Aginako, and Igor G Olaizola. Web browser-based social distributed computing platform applied to image analysis. In *Cloud and Green Computing (CGC), 2013 Third International Conference on*, pages 389–396. IEEE, 2013. 49, 61, 69, 71, 72

[Zorrilla et al.15a] Mikel Zorrilla, Njål Borch, François Daoust, Alexander Erk, Julián Flórez, and Alberto Lafuente. A web-based distributed architecture for multi-device adaptation in media applications. *Personal and Ubiquitous Computing*, 19(5-6):803–820, 2015. 40, 58

[Zorrilla et al.15b] Mikel Zorrilla, Njål Borch, François Daoust, Alexander Erk, Julián Flórez, and Alberto Lafuente. A web-based distributed architecture for multi-device adaptation in media applications. *Personal and Ubiquitous Computing*, 19(5):803–820, Aug 2015. 145

[Zorrilla et al.17] M. Zorrilla, J. Flórez, A. Lafuente, A. Martin, J. Montalbán, I. G. Olaizola, and I. Tamayo. Saw: Video analysis in social media with web-based mobile grid computing. *IEEE Transactions on Mobile Computing*, PP(99):1–1, 2017. 34, 114

[Zorrilla17] Mikel Zorrilla. Program to experiment with the SaW Cost Model. `https://github.com/mikelzorrilla/SaW`, 2017. [Online; accessed 16-February-2017]. 74

[Zyskind et al.15] G. Zyskind, O. Nathan, and A. '. Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pages 180–184, May 2015. 32, 40