eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Department of Communications Engineering

Network-aware video streaming for future media internet

by:
Roberto Viola

Supervised by:
Dr. Jon Montalbán Sánchez
&
Dr. Ángel Martín Navas

Donostia – San Sebastián, Wednesday 14th July, 2021

eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

DEPARTMENT OF COMMUNICATIONS ENGINEERING

NETWORK-AWARE VIDEO STREAMING FOR FUTURE MEDIA INTERNET

by:
Roberto Viola

Supervised by:
Dr. Jon Montalbán Sánchez
&
Dr. Ángel Martín Navas

Donostia – San Sebastián, Wednesday 14th July, 2021

Dedicated to my family, my friends, my workmates
and every person who supported me and
made me who I am today.

Abstract

The current 5G deployment of 5G networks is bringing new network capabilities, including enhanced mobile broadband, ultra-low latency and massive device connectivity. These enhancements are also pushed by the increasing popularity of wireless and mobile devices. Moreover, softwarization and virtualization technologies, such as Software Defined Network (SDN) and Network Function Virtualization (NFV), are considered as key pillars of 5G, as well as for network generations beyond 5G. NFV paradigm virtualizes all the data centers that are part of the network infrastructure (Core, Edge and Access Networks) to bring cloud technologies into the network operations, while SDN technology centralizes network control and manages the forwarding rules between data centers to adapt networking policies to traffic demands. The combination of them enables to operate and manage network functions, referred as Virtual Network Functions (VNFs), by software running on top of general-purpose hardware. The usage of NFV and SDN can be extended also to the Access Network, where Multi-access Edge Computing (MEC) represents a new architectural paradigm to provide cloud capabilities closer to the clients. MEC allows the deployment of edge services to empower heterogeneous vertical applications.

At the same time, we are witnessing a growth in the usage of video streaming applications, including commonly used services, such as Live Streaming and Video-on-Demand (VOD), and new media applications, such as online gaming and 3D video applications (eXtended Reality, Virtual Reality and Augmented Reality). Moreover, streaming solutions tailored to empower different vertical applications, e.g., Industrial Internet of Things (IIoT), medical imaging and automotive machine-vision, are gaining relevance. These trends in video streaming are shaping network traffic, where

5G networks are expected to cope with the increasing total network traffic, mostly generated by media services.

In this context, technologies included in the 5G ecosystem are considered to be the enablers to overcome the challenges raised by the increasing media content generation and consumption. New network functions should be designed and implemented on top of the 5G infrastructure to support high Quality of Service (QoS) and Quality of Experience (QoE) required by media applications and end users. Moreover, these network functions should exploit metrics coming from the network concerning connectivity performance and player's traffic demand to adapt to changeable network conditions. Enabling dynamic changes during the operation of the streaming system has effects also on Content Provider's business costs, as network functions could be optimized to reduce resource usage to what is actually needed. Then, network resources and their costs can be balanced by defining business rules.

In this Ph.D. thesis, the main objective is to improve video streaming QoS and user's QoE, while reducing CP's business costs, through three different contribution areas. Contributions addressed several stages, such as content encoding and delivery, and network nodes, such as origin server, Content Delivery Network (CDN) and MEC host, involved in video streaming workflow. In any of them, the exploitation of the information, acquired from the analysis of both media content characteristics and network metrics, was decisive for increasing the performance of the streaming system.

First, concerning *Network-aware video encoding*, investigating strategies for encoding and packaging the video content has led to two different implementations that leverage network information when preparing the video content for streaming. In the first one, on top of the Secure Reliable Transport (SRT) protocol, the encoding and packaging configurations are tuned to keep QoS/QoE rates when network capabilities change, to prioritize playback smoothness over video quality. In the second one, the use of Low Latency Common Media Application Format (LL CMAF) has been studied to reduce latency when delivering Dynamic Adaptive Streaming over HTTP

(MPEG-DASH) streams. The trade-off between latency and user's QoE is demonstrated to be an important factor when selecting the encoding and packaging configurations.

Then, regarding *Network performance forecast for video delivery*, the use of Machine Learning (ML) techniques to analyze network metrics has been investigated. A solution that exploits a Long Short-Term Memory (LSTM) model has been implemented to forecast network performance and enhance the selection of a CDN, when multiple CDN are employed, to deliver MPEG-DASH streams. Being able to forecast CDN performance allows the selection of the CDNs according to defined business rules. Then, a trade-off between QoS and business costs for CDN usage is evidenced.

Finally, this thesis has contributed to *MEC-enabled video delivery* with the implementation of two MEC services to be employed on top of the novel 5G MEC architecture. The first solution assesses the user's QoE according to ITU-T P.1203 since QoE knowledge is an important enabling factor for advanced solutions to enforce QoE on the MEC platform. Thus, this solution infers QoE from QoS information collected from MPEG-DASH Media Presentation Description (MPD) and from network monitoring. In the second solution, a MEC service is deployed to enforce and boost MPEG-DASH streams. The solution provides two operations. First, it enables a proactive cache of MPEG-DASH segments at the network edge to reduce CDN usage. Second, it shields from CDN malfunction by switching the download of segments to an alternative CDN to ensure QoE rates. This MEC service supports media playback with steady QoE scores by switching from one CDN to another and enhances it by proactively caching the content.

Acknowledgements

It has been more than five years since I joined Vicomtech for my first work experience, when I was still a master's student, and three years since I started this Ph.D. process. I would like to thank all the great people who have enriched and supported me during this period.

First of all, I would like to thank my Ph.D. supervisors Jon Montalbán and Ángel Martín, my Ph.D. tutor Pablo Angueira, and Mikel Zorrilla, who was my supervisor during the first phase and still supported me throughout the Ph.D. process. Their guidance and support were essential to properly develop the research activity, including the description of the research ideas, the management of projects and the preparation of publications.

I would also like to thank my colleagues at Vicomtech, especially those in the Digital Media and Communications department. Ángel Martín, thank you for your invaluable help throughout the Ph.D process, from reviewing the state of the art to developing the experiments and writing publications; Mikel Zorrilla, thank you for giving me the opportunity to participate in interesting and challenging projects; Felipe Mogollón, thank you for helping me in different projects and for always providing a solution for any problem encountered, even when dealing with unknown or new technologies you always ready to face it and provide your help; Ana Domínguez, thank you for sharing information about Vicomtech and Ph.D. procedures and for lots of conversations during the coffee break; Álvaro Gabilondo, thank you for your positive mood (ya no queda nada..., ya queda menos...) and for many padel matches after work, even if I am still waiting for a bike ride.

I also thank my new colleagues, recently arrived in Vicomtech to enrich the 5G research line, Juncal, Zaloa, Mikel S. and Danilo, and the rest of the department, Iñigo, Stefano, Miguel Sergio and Hector.

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.

Thank you all, eskerrik asko guztioi, gracias, grazie

Roberto Viola

Wednesday 14th July, 2021

Contents

List of Figures

List of Tables

Part I

Introduction

CHAPTER

1

Scope of the research

1.1 Overview

In the recent years, we are witnessing a constant growth in the usage of video streaming applications. Increasing of commonly used Live Streaming and Video-on-Demand (VOD) services, as well as the raising of new media applications involving video streams are gaining relevance and are attracting a wider audience. In this context, online gaming and video conferencing are highly popular, while 3D video formats enable support for eXtended Reality (XR), Virtual Reality (VR) and Augmented Reality (AR). Moreover, tailored streaming solutions are being employed to empower different vertical applications, e.g., Industrial Internet of Things (IIoT), medical imaging and automotive machine-vision, which can benefit from the advances in video streaming. Meanwhile, wireless and mobile devices are becoming the primary User Equipment (UE) for both content generation and consumption [2].

It is evident that video streaming technologies are being used beyond traditional content playback as an alternative to television broadcasting. This means changes at different levels, from players and servers to media formats and delivery protocols, which also affect network traffic. 5G networks are expected to cope with the increasing total network traffic, mostly generated by media services, by means of higher network bandwidth and reduced latency. According to Cisco reports and forecasts, it is estimated that

5G connections will handle nearly three times more traffic than a current LTE connection by 2023 [3]. The estimation is still passive from being reviewed, as a larger usage of media applications than the expectation is being fuelled by the global COVID-19 pandemic. This pandemic is transforming users' habits to access the Internet [4, 5] and media contents [6, 7]. The Broadband Commission for Sustainable Development, a joint initiative of the International Telecommunication Union (ITU) and the United Nations Educational, Scientific and Cultural Organization (UNESCO), is concerned about these trends and is implementing an Agenda for Action to push an emergency response to the pandemic, aiming at Internet access extension and boosting its capacity [8, 9].

All the mentioned factors are driving the evolution and the deployment of 5G networks and services. New network solutions are necessary to support high quality of service (QoS) for media applications, as a best-effort network approach to manage media traffic does not ensure the fulfilment of the requirements requested by the media service in terms of Key Performance Indicators (KPIs). Media services have constraints regarding packet delivery, e.g., on-time delivery or packet loss, which need to be addressed to guarantee a certain level of QoS. Therefore, lower QoS may result in lower user's quality perception, i.e., quality of experience (QoE). A pragmatic example of this QoE degradation are stalls or artifacts during video playback on player devices. Moreover, the heterogeneity of media-related use cases results in having different requirements depending on the specific media service, i.e., real-time communications from a security camera and VOD for entertainment have different requirements in terms of latency and throughput.

An essential system for streaming multimedia content is the Content Delivery Network (CDN). A CDN is the most common solution to increase the performance of online applications, including the video streaming ones. It consists in a geographically distributed hierarchical system to cache and deliver every type of contents, i.e., web objects (HTML web pages) or downloadable files (media files and documents). For video streaming purpose, a CDN provides the infrastructure to deliver Live or on-demand streams by fostering the efficiency and increasing the service coverage. The increasing number of internet users and the proliferation of video and rich media contents over the internet is boosting the demand for CDN solutions. The global CDN market is expected to grow at a rate of 14.1% per year until 2025 [10]. The proliferation of CDNs is making their price to decrease, but the overall cost for the content provider (CP), which makes use of them to deliver its contents, is still increasing, as also the traffic volume from/to CDN is increasing [11].

Beyond CDNs, more advanced solutions based on Software Defined Network (SDN) [12] and Network Function Virtualization (NFV) [13] technologies are being investigated to boost media services. SDN is a network management approach to enable a centralized and programmable way to configure and monitor the network and its performance. It separates the control from the data plane such that the network administrator can configure the network resources in an easy way. The control layer includes a SDN controller provided with an Application Programming Interface (API) that allows to fill and updates the forwarding tables of the network infrastructure (data plane or forwarding plane). The data plane processes and forward the data packets based on the instructions coming from the control plane. NFV is instead a network architecture concept that uses virtualization technologies to deploy and operate an infrastructure totally independent of hardware. In other words, it decouples network functions from the commercial off-the-shelf (COTS) hardware where they are deployed and run. Network functions, called Virtual Network Functions (VNFs), are deployed and connected to create a complete network service (NS) on top of a virtualized infrastructure. Definitively, SDN and NFV represent complementary solutions. NFV virtualizes the network infrastructure (Core, Edge and Access Networks) built on top of data centers, while SDN centralizes network control and manages the forwarding rules between data centers. The combination of them allows to have networks and services that are operated and managed by software systems running on top of COTS hardware. It is important to note that 5G is pushing the employment of SDN and NFV to manage end-to-end connections and to provide them with the demanded network resources. Within the 5G umbrella, Multi-access Edge Computing (MEC) [14] also covers an important role to guarantee end-to-end performance. MEC is a new architectural concept to provide virtualized capabilities at Network Edge. MEC platform consists in an NFV-compliant data center to deploy and run VNFs close to the Radio Access Network (RAN). Additionally, it also provisions a specific API to access Radio Network Information (RNI) [15]. A RNI service (RNIS) oversees the collection of RNI which can be consumed by VNFs running at MEC host.

In video streaming context, VNFs can be employed at any level of the end-to-end communication (Core, Edge and Access Networks) to empower network capabilities when generating, delivering and/or consuming video streaming traffic in an optimized and cost-effective manner [16, 17]. VNFs enable flexible operations with several benefits. Firstly, VNF-based networks monitor objective operational parameters, such as throughput or latency, representative for QoS of the streaming dataflows, which have a direct influence on user's satisfaction. However, QoS metrics do not perfectly map on user experience, as user perceived quality is highly subjective. Additionally, QoE which compiles subjective evaluation elements, including rewards for playback quality and smoothness, and penalties for image freezes and unstable or low quality [18, 19], needs to be considered, too. Secondly, the CP has more control to shape the network traffic and allocate resources since business rules for VNF deployment and life-cycle management could be established. These rules allow to adjust network resources and business costs trade-offs [20], so they are highly relevant.

In general, selecting the right performance metrics to optimize the video streaming is not trivial. There is not a unique metric to evaluate the goodness of a particular media solution, as multiple viewpoints coexist. Network Operator (NO) is interested in providing a QoS according to the Service Level Agreement (SLA) contracted by the CP or the user, while keeping Operational Expenditure (OPEX) under a target threshold. CP has to guarantee the best QoE to the users, while aiming in business costs reduction. Finally, the users are mostly interested in having as higher and steadier QoE as possible. In any case, it is clear that performance assessment and network monitoring are important sources of information that can be exploited for the design of performance-driven VNFs for media services. Thus, information from performance metrics and network monitoring allows to optimize the network service.

1.2 Motivation

As described in the previous section, network services depend on the characteristics and performance provided by the underlying network. Monitoring the network and assessing performance metrics are essential for improving the services. Considering network capabilities enhancement and media service improvement as separate optimization problems is not a good option as they are intrinsically related. 5G ships new parameters and technologies which make the difference in enhancing both the network itself and the services running on it. Media services are deployed as Virtual

Network Functions (VNFs), which run on top of an NFV Infrastructure (NFVI) and are interconnected through SDN rules. Moreover, the benefits gained from deploying a network-aware media service are evident, as the media service can react to variable network conditions. Combining innovative media services with KPIs improvements introduced by 5G, it is possible to achieve more reliable and flexible networks and services, where the exploitation of performance metrics and network information is essential for performance-driven optimization.

HTTP Adaptive Streaming (HAS) [21] is a video streaming technology that was natively designed to exploit measurements performed on the network by the media player. Video content is encoded at different representation bitrates and resolutions and the player is in charge of selecting the one that fits with the network measurements and its internal state (playback buffer size or window of time buffered for the playback). The different representations are also split in segments of fixed duration (between 2 and 10 seconds), enabling switching operations between representations each time a new segment needs to be downloaded. HAS implementations, such as HTTP Live Streaming (HLS) [22] and Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [23], can be furtherly improved, as they present two main limitations. First, uncoordinated operations between players sharing the network assets may cause unfairness, as the resources could not be equally distributed. Second, HAS intrinsically has high latency as a segment cannot be sent until fully generated. The theoretical minimum latency is the segment duration which makes impossible to achieve real-time communications. This thesis is oriented to improve video streaming services by exploiting the information coming from the performance metrics and network traffic monitoring and analysis. Moreover, it aims to provide solutions to overcome the current limitations of HAS.

Figure 1.1 includes all the stakeholders and/or agents coming into play in a common video streaming communication schema. First, the origin server, managed by the CP, generates the content by performing two main operations. It encodes the video content through standard codecs, e.g., H264 [24] or HEVC [25]. Then, it packages it in a streaming format/protocol that is later sent to the CDNs. Here, the encoded and packaged content is stored and distributed to the end users. When considering HAS technologies, an additional manifest file is generated and stored in a media server and it is accessed by the media player to know the available representations and the CDNs where media segments are cached. Optionally, the manifest stored at the media server can be

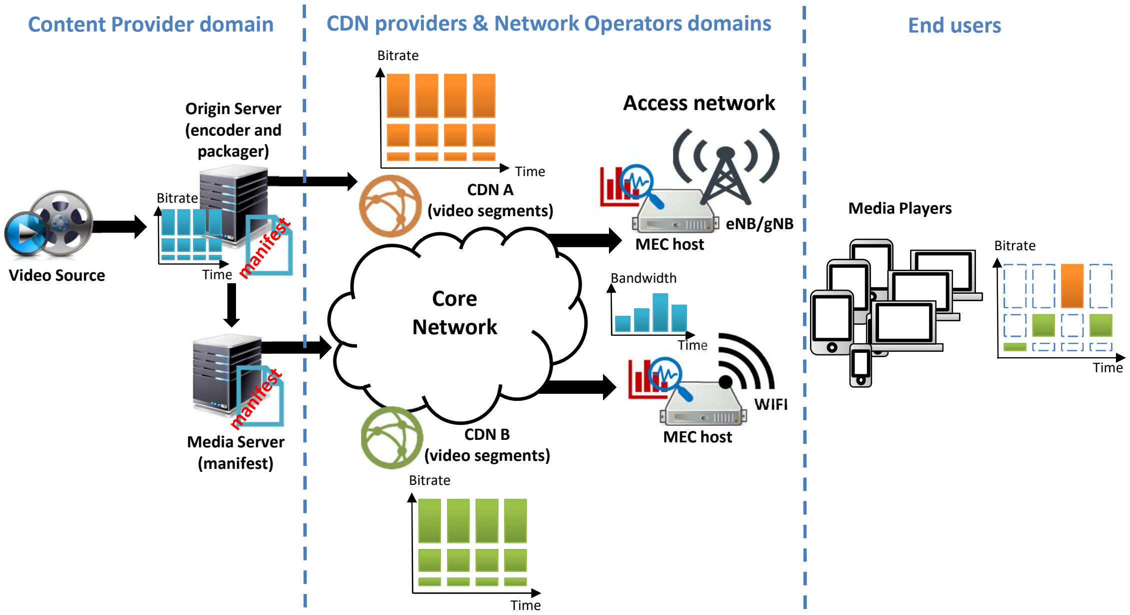


Figure 1.1: Overview of a video streaming communication and involved stakeholders and/or agents.

periodically updated, as declared in the standard, to provide the player with updated information on the available representations and the CDN. Thus, it enables to serve a different manifest to each request and influence the player's behavior in selecting the CDN from which downloading the content, as well as selecting the representation of each segment. Finally, mobility trends are pushing solutions that move from legacy streaming between remote servers (origin server or CDN) and media players to more complex ones that include MEC-based services in the middle of the end-to-end streaming process. HAS enables individual optimization of the QoE performed by each player. It is not an optimal solution, as it may cause unfairness in terms of QoE among end users. The awareness of RNI at the MEC allows to assess network QoS and estimate user's QoE. It enables the possibility to take actions that enforce QoS/QoE and fairness among the players. MEC-enabled solutions include the possibility to locally cache HAS video segments and, again, to modify the manifest in a coordinated way with the remote media server and the CDNs.

This research work is focused on some major changes in the video streaming context. **First**, the knowledge of network information can be exploited to encode the content at the appropriate video bitrate and resolution and select the media container format.

Second, when delivering the content, network metrics can be monitored and analyzed to forecast the behavior of the network and take proactive actions to optimize the delivery. **Third**, the introduction of MEC within the 5G umbrella enables the development and deployment of services close to the RAN, which empower the delivery through network and video analytics. In the following paragraphs, the motivation for each of them is explained.

First, when preparing the video content to be delivered, two main operations are required: compressing the content through standard video codecs, e.g., H.264 or HEVC, and packaging it in media container formats, e.g., MPEG-4 Part 14 (commonly called MP4). Encoding and packaging the content influence user's QoE, which plays a significant role when dealing with media services, as a satisfactory QoE may help to retain the user from leaving the media service. Human Visual System (HVS) has been widely studied for years when developing the actual video codecs in order to increase the compression rate, while keeping the same quality of the image [26], and further improvements are expected by the next generation ones. Moreover, Per-Title Encoding [27] strategy also includes the complexity of the video content itself in the equation. Per-Title Encoding targets to select the encoding bitrate for the chosen codec that best fits with the visual complexity. Thus, it aims to present a paramount view experience.

In any case, video codec development and Per-Title strategy are only focused on the user's QoE and does not consider the possibility to exploit network information to optimize the encoding process. If the network is not able to provide sufficient throughput to cope with encoded bitrate, stalls and video artifacts are experienced at the media player which affects the QoE. HAS aims to reduce the number of stalls by allowing the player to select the video representation, among the available ones, that fits with the experienced network performance. Here, uncoordinated operations and unfairness between players, as well as HAS intrinsic high latency, are still major issues to be addressed.

To overcome such limitations and to provide bitrate adaptation according to network conditions, two different options are envisioned:

- Introducing bitrate adaptation on top of streaming protocols designed for real-time communications.
- Enabling the possibility to send partially generated HAS segment to the client, such that the content is sent to the player with minimum delay after the encoding

operation. This solution is the key of Low Latency Common Media Application Format (LL CMAF) [70], also called Chunked CMAF.

The former solution also enables coordinated delivery as the bitrate is chosen by the origin server which is generating the content, but it lacks scalability as unicast streams are generated independently for each video session. The latter solution keeps the advantages provided by HTTP protocol, such as the possibility to cross Network Address Translation (NAT) systems and firewall devices, and end user device heterogeneity, as every device supports HTTP.

Second, in-network caching is a mechanism to improve the performance when accessing online contents and, in particular, media streaming ones. It aims to prevent negative effects on the QoS/QoE caused by network impairments. In this context, a CDN is the popular solution to cache and deliver video streams. Furthermore, major CPs also moved to multi-CDN strategies to provide a more reliable service while streaming their contents [28, 29]. In a multi-CDN environment, several strategies on how selecting the best performing CDN are applicable by the CP, but they are typically limited to a selection at the startup, keeping the same CDN along all the streaming session [30].

Enabling the capability to switch between alternative CDNs, when the streaming sessions are ongoing, opens to lots of possibilities for optimization. Thus, multi-CDN strategies can be designed to optimize CDNs utilization and reduce the resultant OPEX for the CP. Among these strategies, proactive ones can exploit forecasts to perform actions which cope with a predicted increased demand and/or prevent the effects of predicted network failures.

Third, 5G includes MEC [14] as key pillar to increase the performance of network applications. MEC is a new network architecture concept that enables computing capability close to the RAN to run algorithms and/or services that empower specific applications. RNIS integrated into MEC platform allows the deployed services to access RNI report and exploit it. Thus, it fosters the design and development of use cases and/or vertical sector-specific solutions that exploit the favorable location near the RAN and the context of the network [31, 32].

In any case, the information provided by the RNIS consists in a set of objective metrics, i.e., RNIS assesses only information related to QoS. Designing and developing QoE models for their deployment at the MEC represents a step forward. When considering media streaming applications, a QoE model can infer RNI/QoS metrics to estimate the QoE experienced by the end users. Being able to access both QoS and QoE information at the MEC increases the interest of CPs to design MEC-empowered algorithms and/or services that take actions to improve their media service performance. MEC solutions should address two main objectives. First, a MEC-enabled media solution should maximize both the QoS and QoE, while trying to reduce the network traffic volume. Thus, the objective is to optimize the trade-off between QoS/QoE and business costs for network assets. Second, the assessment of the QoE of each individual player should foster solutions that improve the fair utilization of the network resources. Consequently, fairness in viewing experience can be increased through fair sharing the network assets. This produce similar playback at media players.

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

1. Multimedia consumption is gradually shifting from traditional TV to Over-the-top (OTT) media services where the CP streams media contents via the Internet. Utilization of both mobile devices, i.e., smartphones and tablets, and connected TVs, including flat-panel TVs, set-top boxes and gaming consoles, will grow during the next few years. By 2023, the consumer share of the total devices will be 74%, with business claiming the remaining 26% [3].

2. Video streaming is heavily dominating the traffic over the Internet, as the demand of higher and higher quality video contents is increasing the consumption on network resources. It is fueled by improved cameras with stunning picture quality [33] and the breakthroughs in display technology [34]. By 2023, 66% of connected flat-panel TV sets will be 4K [3] and 47% of all devices and connections will be video capable [35].

3. Increasing video resolutions, such as 4K and 8K, and the rise of new video formats, such XR, VR and AR, is pushing the development of new compression techniques

and codecs. Improved video codecs for 2D images are being released in the recent years [36, 37], while 3D compression is finally being standardized [38].

4. International consortia, such as the European Telecommunications Standards Institute (ETSI) and the International Telecommunication Union (ITU), are driving the digital transformation of the networks. Softwarization and virtualization are changing the economics of the networks and pushing NOs to move from proprietary and specific hardware to virtualized software platforms through the abstraction of the execution environment [20].

5. ETSI proposes the deployment of edge services at MEC infrastructures and includes several use cases related to media streaming [39]. MEC platform can host diverse edge services, which exploit RNI to get a wider view of the local conditions to enhance media streaming service [31, 32]. Moreover, the capillarity of MEC concept allows to run edge services whose nature is distributed.

6. When managing the networks, a Service Level Agreement (SLA) aims to enforce network capabilities for a specific service. Modern network services require different levels of guaranteed bandwidth, latency and priority over other traffic. Thus, avoiding SLA violations is becoming more and more important to guarantee required performance over time [40].

All these mentioned trends are reinforced and aligned with the experience acquired by participating in several national and European projects, such as 5G Euskadi (`https://5g-euskadi.com/`), Open-VERSO (`https://www.openverso.org/en/`) and Fed4Fire+ (`https://www.fed4fire.eu/`). These projects involve first level telecommunication operators, such as Orange (`https://www.orange.es/`) and Euskaltel (`https://www.euskaltel.com/`), technology providers, such as ZTE (`https://www.zte.com.cn/global/`), universities, such as University of Thessaly (`https://nitlab.inf.uth.gr/NITlab/nitos`), and research institutes, such as Vicomtech (`https://www.vicomtech.org/en`), Gradiant (`https://www.gradiant.org/en/`) and i2Cat (`https://i2cat.net/`).

1.3 Hypothesis

The working hypothesis is constructed as a statement of the following expectations:

1. Encoding operations at the origin server are improved if compression schemes and encoding strategies make use of the information provided by both the video content and the network status.
2. Utilization of network assets is optimized by leveraging information achieved by monitoring network and media players' behavior.
3. Distributed nature of MEC concept enables the deployment of services focused on supporting enhanced media session management and performance.

These mentioned expectations involve different stakeholders and/or agents in the media streaming chain shown in Figure 1.1:

- **Content Provider**
  - It manages both origin and media servers, even they do not necessarily belong to it, as they could be deployed in a third-party cloud platform.
  - It aims to reduce business costs for the exploitation of network assets, including cloud and MEC platforms and CDN resources.
- **Content Delivery Network (CDN) vendor**
  - It provides CDN infrastructure to store and deliver media contents.
  - It grants an SLA with CPs that has to respect by avoiding/reducing violations.
- **Network Operator (NO)**
  - It provides network resources (Core, Edge and RAN) to stream the content from the CDNs to the end users.
  - More than one NO's network could be crossed when streaming a content, i.e., multi-domain networks.
  - MEC hosts and services are deployed and managed close to NO's RAN infrastructure.

  - It grants SLAs with both CPs and end users that has to respect by avoiding/reducing violations.

- **End User**
  - It is subscribed to CP's media service.
  - It is connected to the network thanks to its network provider (Network Operator).
  - It is receiving the media stream on its player device and watching it.
  - It aims to have the highest possible QoE.

1.4 Objectives

The main objective of this work is to improve QoS and QoE of media streaming, while reducing CP's business costs. Performance of video streaming services are evolved by means of advanced encoding solutions and new 5G network architecture and paradigms. Furthermore, the main objective is decomposed into more specific objectives:

1. Enabling advanced encoding and packaging strategies with the aim of exploiting the information acquired from the network to improve the encoding operations with focus on media streaming QoS and user's QoE.

2. Empower media delivery by using Machine Learning (ML) algorithms to allow the CP to reduce business costs, while guaranteeing a sufficient level of QoS.

3. Integrate MEC platform and develop services that enforce media streaming sessions and boost the user's QoE.

To achieve these objectives, it is necessary to address and provide solutions to overcome the three main challenges of video streaming (see Figure 1.2):

1. **Video content preparation for streaming:** considering network QoS enhancement and user's QoE improvement as separate optimization problems is not a good option as they are intrinsically related. The origin and media servers should be provided with the knowledge acquired from the network. Network-aware servers can take actions to balance network QoS and user's QoE.

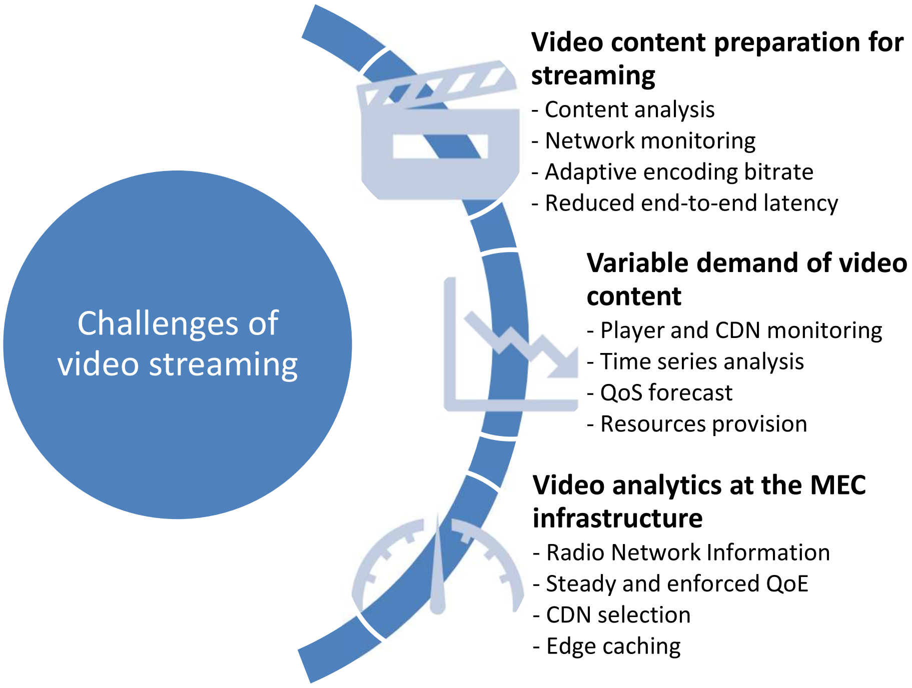

Challenges of video streaming
Video content preparation for streaming
- Content analysis
- Network monitoring
- Adaptive encoding bitrate
- Reduced end-to-end latency
Variable demand of video content
- Player and CDN monitoring
- Time series analysis
- QoS forecast
- Resources provision
Video analytics at the MEC infrastructure
- Radio Network Information
- Steady and enforced QoE
- CDN selection
- Edge caching

Figure 1.2: Main challenges to be addressed in order to achieve the objectives of this Ph.D. thesis.

2. **Variable demand of video contents:** number of users and their demanded contents are varying. It causes complex network dynamics, where the traffic depends on the demanded contents and the representation bitrate chosen to stream them at any moment. Increasing or decreasing the employed network assets depending on the demand means efficiently managing the resources and finding a trade-off that ensures a steady and consistent user's QoE and reduces CP's business costs.

3. **Video analytics at the MEC infrastructure:** the 5G MEC architecture exploits network performance metrics at the RAN to estimate user's QoE. Therefore, the possibility to boost user's media streaming session is envisioned by influencing the caching and CDN selection strategies and minimizing the impact of network faults or issues on the QoE.

1.5 Contributions

The main contribution of this Ph.D. research is founded on the advances in video streaming technologies to provide enhanced performance of media services. Various metrics are defined to consider performance from different point of view, including business costs, network capacity and user's satisfaction. These advances, based on standard protocols and technologies, enable network-aware and performance-driven video streaming solutions which operate on dynamic and flexible networks.

More specifically, the main contribution can be translated into specific outcomes. Figure 1.3 illustrates the three specific contributions of the research in a wider context to address content generation and delivery, and management of the network and its resources.

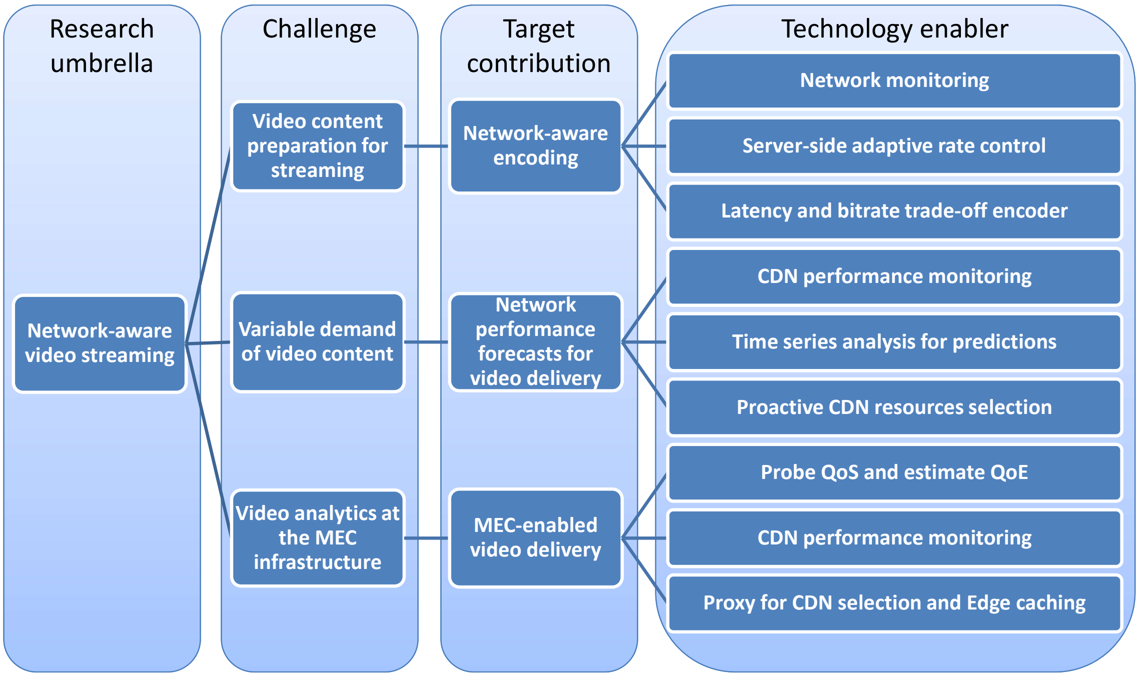

Research umbrella
Challenge
Target contribution
Technology enabler
Network-aware video streaming
Video content preparation for streaming
Variable demand of video content
Video analytics at the MEC infrastructure
Network-aware encoding
Network performance forecasts for video delivery
MEC-enabled video delivery
Network monitoring
Server-side adaptive rate control
Latency and bitrate trade-off encoder
CDN performance monitoring
Time series analysis for predictions
Proactive CDN resources selection
Probe QoS and estimate QoE
CDN performance monitoring
Proxy for CDN selection and Edge caching

Figure 1.3: Diagram of the contributions of the research.

Contribution 1: Network-aware video encoding

Optimizing video encoding and packaging strategy has effects on user's QoE, which plays a significant role when dealing with media services, as a satisfactory QoE may retain the user from leaving the media service. Studies on HVS [26] and Per-Title Encoding [27] are leading to analyse the video image complexity in order to apply custom encoding settings for different classes of video contents (e.g., action movie, sport, security camera, etc.), which optimize the exploitation of video processing resources, while enhancing user's QoE.

Including network information and application context (VOD or real-time communications) represents a further step. Selection of video encoding bitrate and streaming format/protocol should be chosen depending on the application context and network information. In this sense, this thesis has designed and implemented two different solutions that exploit such information.

First, on top of SRT protocol, an Adaptive Rate Control is developed to demonstrate the applicability of network information at the origin server. It enables a coordinated delivery as the encoding bitrate is chosen by the origin server once for all the connected media players. Second, a solution which exploits the wide support of end devices for playing HAS streams, such as MPEG-DASH and HLS, studies LL CMAF to deliver Live Streaming and evaluates the trade-off between latency and QoE.

Publications related to Contribution 1:

- *R. Viola, Á. Martín, J. F. Mogollón, A. Gabilondo, J. Morgade and M. Zorrilla, "Adaptive Rate Control for Live streaming using SRT protocol," 2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1-6, 2020, doi: 10.1109/BMSB49480.2020.9379708*, in Section 3.2.

- *R. Viola, A. Gabilondo, Á. Martín, J. F. Mogollón and M. Zorrilla, "QoE-based enhancements of Chunked CMAF over low latency video streams," 2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1-6, 2019, doi: 10.1109/BMSB47279.2019.8971894*, in Section 3.3.

Contribution 2: Network performance forecasts for video delivery

CDN is a common solution to provide caching capabilities with worldwide coverage, as it is a geographically distributed hierarchical system that cache and deliver online contents and, in particular, media streaming ones. Thus, the usage of CDN aims to prevent negative effects on the QoS/QoE caused by network impairments. CPs make extensive use of CDNs, also including strategies that employ several CDNs at the same time [28, 29]. Then, it implies the definition of more complex CDN selection mechanisms. In any case, the typical solution consists in CDN selection at the media player session startup, keeping this selection along all the streaming session [30].

In this context, the ability to switch between different CDNs, when streaming sessions are in progress, represents an interesting approach. It allows to optimize the employed CDN resources by reducing their usage to the effective necessity. As the number of players connected to a CDN increases, migrating some players to another CDN helps maintain QoS and QoE scores. In contrast, when the number of players decreases, migrating all players to a single CDN reduces the CDN resources used.

This thesis designed a multi-CDN strategy that optimizes CDNs utilization and reduce the resultant business costs for it. The solution is empowered with a trained Artificial Neural Network (ANN) model that forecasts network performance to perform proactive actions which cope with a predicted increased demand and/or prevent the effects of predicted network failures.

Publications related to Contribution 2:

- *R. Viola, Á. Martín, J. Morgade, S. Masneri, M. Zorrilla, P. Angueira and J. Montalbán, "Predictive CDN selection for video delivery based on LSTM network performance forecasts and cost-effective trade-offs," IEEE Transactions on Broadcasting, vol. 67, no.1, pp. 145-158, 2020, doi: 10.1109/TBC.2020.3031724,* in Section 4.2.

Contribution 3: MEC-enabled video delivery

MEC architecture [14] has been designed to increase the performance of 5G networks by enabling computing capability closer to the users, in the RAN segment of the network. It allows to run algorithms and/or services that empower specific applications. To further improve MEC services, a RNI Service (RNIS) deployed at the MEC provides a specific API to access RAN information or RNI [15]. The information provided by the RNIS consists in a set of objective metrics that is possible to infer in order to understand the end device and player's behavior. Knowledge of the behavior allows to that take actions to improve the media session performance.

Such information fosters the design and development of use case-specific solutions, including media steaming ones [31, 32], as specified by ETSI [39]. The favorable location near the RAN and the provided RNI represent important assets to be exploited for the design of the MEC service. In this sense, this thesis has designed and implemented two different solutions that infer and exploit information acquired at the MEC platform.

First, a MEC service is designed to run on top of a Wi-Fi access point [41]. It allows to collect player's objective metrics, such as bitrate and resolution of the video representation, switching operations between representations, number of stalls and their duration, and infer them to estimate the user's QoE with high accuracy. Second, a MEC proxy is developed to exploit information at the RAN by enabling local caching and CDN selection at the edge network.

Publications related to Contribution 3:

- *R. Viola, M. Zorrilla, P. Angueira and J. Montalbán, "Multi-access Edge Computing video analytics of ITU-T P.1203 Quality of Experience for streaming monitoring in dense client cells," submitted to Multimedia Tools and Applications (March 25, 2021),* in Section 5.2.

- *R. Viola, Á. Martín, M. Zorrilla and J. Montalbán, "MEC Proxy for efficient cache and reliable multi-CDN video distribution," 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1-7, 2018, doi: 10.1109/BMSB.2018.8436904,* in Section 5.3.

1.5.1 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 hypothesis, the objectives and the contributions of the Ph.D. work.

Part II overviews literature related to network functions for media streaming applications, including media encoding services and delivery solutions.

In Part III, the research results are described in three chapters:

- Chapter 3 describes the contributions to create video encoding and packaging solutions that exploit information acquired from the network and the application context (*Contribution 1*). The goal is to tune the video processing operations in order to find the appropriate trade-off between latency and QoE when considering Live streaming applications.

- Chapter 4 describes the contributions to create video delivery strategies that exploit network performance forecasts in a multi-CDN context (*Contribution 2*).

The goal is to guarantee the required QoS for the media streaming application while reducing the business costs.

- Chapter 5 describes the contributions to create MEC services that estimate QoE scores and take actions to improve the streaming sessions (*Contribution 3*). The goal is to infer RAN information to increase user's QoE by enabling local caching and CDN selection at the edge network.

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

Part V provides an appendix, including other publications of the author, the resume, and a list of the acronyms employed throughout the document.

Finally, Part VI contains the bibliography.

Part II

State of the Art

CHAPTER

2

Related Work

2.1 Context

As explained in Chapter 1, new technologies and paradigms, such as SDN and NFV, are included in the 5G ecosystem and are also considered as key pillars of network generations beyond 5G. The objective of NFV is to virtualize all the building blocks that constitute the network infrastructure (Core, Edge and Access Networks) over the resources available at the data centers. The virtualization technologies, widely proven in cloud platforms, allow to easily scale or migrate a service from a location to another depending on the demand of the service and network status at any moment. SDN enables a centralized network control and the management of forwarding rules between network functions running over data centers. To achieve it, the separation between control and data planes allows the control plane to instruct the data plane to process and forward data packets according to specified forwarding rules. Therefore, it creates an abstraction layer for the network administrator who no longer needs to manually configure each node. The combination of SDN and NFV enables to operate and manage VNFs by software running on top of general-purpose hardware. VNFs instances run on top of NFV Infrastructure (NFVI), where the connection is provided through SDN equipment and forwarding rules. NFV and SDN are also applied at the Access Network, where MEC consists in an NFV-compliant data center. MEC represents a new architectural paradigm to provide cloud capabilities closer to the clients, as to allow the deployment of edge services to empower heterogeneous vertical applications. In addition to other cloud infrastructure, it provisions a specific API to access RNI that can be exploited by VNFs instances running at MEC host.

In a video streaming context, VNFs are designed and employed to deploy an end-to-end media system to empower the generation, the delivery and the consumption of video content in an optimized and cost-effective manner. VNFs implementations include functions that can operate on different nodes on the network. VNFs can be used to encode and package the media content on the origin server or to serve manifests when employing MPEG-DASH or HLS on the media server. Additionally, virtual CDNs and MEC services can be deployed as VNFs to enhance the delivery of the media content.

The use of VNFs enables flexible operations whose benefits are threefold. First, VNF-based networks monitor objective operational parameters, such as throughput or latency, representative for QoS of the streaming dataflows, which have a direct influence on user satisfaction. However, QoS metrics do not perfectly map on user experience, as user perceived quality is highly subjective. Additionally, QoE needs to be considered to compile subjective evaluation elements, including rewards for playback quality and smoothness, and penalties for image freezes and unstable or low quality. Secondly, the CP can monitor network traffic and allocate resources according to its business rules. Thus, it can adjust the balance between network resources and business costs. Last, as the volume, complexity and real-time nature of streaming traffic has an evident impact on energy consumption of the network and devices managing the content, an optimized streaming delivery through VNFs should also consider the energy efficiency.

Section 2.2 includes a survey focused on the application of VNFs for media streaming services. The survey provides the state-of-the-art of the involved technologies and solutions, as well as providing an outlook on pending challenges future research directions.

2.2 A Survey on Virtual Network Functions for Media Streaming: Solutions and Future Challenges

- **Title:** A Survey on Virtual Network Functions for Media Streaming: Solutions and Future Challenges
- **Authors:** Roberto Viola, Ángel Martín, Mikel Zorrilla, Jon Montalbán, Pablo Angueira and Gabriel-Miro Muntean
- **Journal:** IEEE Communications Surveys and Tutorials
- **Publisher:** IEEE
- **Year:** (Submitted May 5, 2021)

Abstract: Media streaming services rely heavily on good and predictable network performance when delivered to large numbers of people. Media services need to ensure enhanced user perceived quality levels during content playback to attract and retain audiences, especially while the streams are distributed remotely via networks. Furthermore, as the quality of media content gets higher, the network performance demands are also increasing and it is challenging to meet them. To this end, Content Delivery Networks (CDN) employ diverse solutions to empower media streaming, including state-of-the-art streaming technologies, such as HTTP Adaptive Streaming (HAS). Network functions should help to further enhance media streaming services and cope with the high dynamics of network performance and user mobility. Furthermore, new networking paradigms and architectures under the 5G networks umbrella are bringing new possibilities to deploy smart network functions, which monitor the media streaming services through live and objective metrics and boost them in real time. This survey overviews the state-of-the-art technologies and solutions proposed to apply new network functions for enhancing quality of service (QoS), quality of experience (QoE) and other business and energy metrics in the context of media streaming.

Keywords: Media streaming, network functions, quality of service, network virtualization, network traffic, network forecast.

2.2.1 Introduction

In the recent years, media streaming traffic is constantly growing. Wireless and mobile devices are becoming main sources for both rich media content generation and consumption. 5G networks must cope with this new traffic demand supporting higher bandwidth and reduced latency. It is estimated that 5G connections will handle nearly three times more traffic than a current LTE connection by 2023 [3]. New applications involving video streams are gaining relevance and are attracting an increased audience, including in areas in which there was little or no rich media presence. Examples of professional applications and application areas which can benefit from advanced media streaming include Industrial Internet of Things (IIoT), medical equipment and connected and autonomous vehicles. Moreover, 3D video formats enable support for new services, such as eXtended Reality (XR), Virtual Reality (VR) and Augmented Reality (AR). Finally, online gaming and video conferencing are also highly popular, especially in the last period. These services have increasing demands in terms of network support. However, although the networks have growing capabilities, there is a large increase in rich media streaming traffic, mostly fueled by the global COVID-19 pandemic. This pandemic is transforming users' habits to access the Internet [4, 5] and media content consumption [6, 7]. The Broadband Commission for Sustainable Development, a joint initiative of the International Telecommunication Union (ITU) and the United Nations Educational, Scientific and Cultural Organization (UNESCO), is also concerned about these user habit changes and it is implementing an Agenda for Action to push an emergency response to the pandemic, aiming at Internet access extension and boosting its capacity [8, 9].

All the above-mentioned factors are inevitably influencing the evolution of all services, and especially affect the rich media ones. It is therefore evident that there is a need for new network-related solutions to support high quality of service (QoS) for these applications. The current network traffic crosses networks working on a best-effort basis where no details regarding packet delivery (e.g., time) is guaranteed. Therefore, best-effort networked-transmitted media traffic may result in lower user quality of experience (QoE). A paradigmatic example of this QoE degradation are stalls or artifacts during media playback on player devices. Employing content delivery networks (CDNs) is the most common solution to prevent negative quality effects and make video delivery more efficient. CDNs are geographically distributed hierarchical systems that cache and store video streams to foster efficiency and increase the service coverage. CDN price is decreasing, but the overall cost for the content provider is increasing, as the traffic from/to CDN is increasing [11].

Beyond CDNs, more advanced solutions based on Network Function Virtualization (NFV) technologies [13] are being investigated to support media streaming services. NFV allows the deployment of Virtual Network Functions (VNF) devoted to empower network abilities when delivering media streaming traffic in an optimized and cost-effective manner [16, 17]. VNFs enable flexible operations whose benefits are threefold. First, VNF-based networks monitor objective operational parameters, such as throughput or latency, representative for QoS of the media streaming dataflows, which have a direct influence on user satisfaction. However, QoS metrics do not perfectly map on user experience, as user perceived quality is highly subjective. Additionally, QoE which compiles subjective evaluation elements, including rewards for playback quality and smoothness, and penalties for image freezes and unstable or low quality [18, 19], needs to be considered, too. Secondly, the Content Provider (CP) has more control to shape the network traffic and allocate resources since business rules for VNF deployment and life-cycle management could be established. These rules allow balancing network resources and business costs trade-offs [20], so they are highly relevant. Last, as the volume, complexity and real-time nature of the media streaming traffic has an evident impact on energy consumption of the network and devices managing the content, an optimized streaming delivery through VNFs should also consider the energy efficiency.

In this context, the main contributions of this survey are:

- The survey discusses widely employed performance assessment solutions and metrics related to media streaming. Metrics are classified into four subgroups: QoS, QoE and fairness, business metrics and energy efficiency;

- The survey provides an extensive overview of the literature involving media traffic monitoring and analysis, including traffic characterization and analysis to enable forecasts. Most common tools for network performance monitoring and simulation are also presented;

- The survey analyzes and classifies the state-of-the-art performance-driven network functions for media streaming;

- The survey presents technologies considered by the telecommunications industry as key enablers for the next generation networks, and discusses remaining challenges.

Table 2.1: List of Acronyms used in the paper.

3GPP	3rd Generation Partnership Project
5G	Fifth Generation
6G	Sixth Generation
AES	Advanced Encryption Standard
AES-CBC	AES block cipher mode
AES-CTR	AES counter mode
ANN	Artificial Neural Network
AR	Augmented Reality
C-RAN	Cloud-RAN
CAPEX	Capital Expenditure
CDN	Content Delivery Network
CSI	Channel State Information
CMAF	Common Media Application Format
CN	Core Network
COTS	Commercial off-the-shelf
CP	Content Provider
CRM	Customer Relationship Management
DASH	Dynamic Adaptive Streaming over HTTP
DNS	Domain Name System
DTN	Delay-tolerant Networking
ESN	Echo State Network
ETSI	European Telecommunications Standards Institute
FeMBMS	Further enhanced MBMS
GUI	Graphical User Interface
HAS	HTTP Adaptive Streaming
HLS	HTTP Live Streaming
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IBN	Intent-Based Network
IoT	Internet of Things
IIoT	Industrial Internet of Things
IM	Instant Messaging

IP	Internet Protocol
ISP	Internet Service Provider
ITU	International Telecommunication Union
KPI	Key Performance Indicator
L1	Physical layer
L2	Data link layer
L3	Network layer
L4	Transport layer
L7	Application layer
LL CMAF	Low Latency CMAF
LL-DASH	Low Latency DASH
LL-HLS	Low Latency HLS
LSTM	Long short-term memory
LTE	Long-Term Evolution
M3U8	HLS playlist
MANO	Management and Orchestration
MBMS	Multimedia Broadcast/Multicast Service
MEC	Multi-access Edge Computing
MLP	Multi-layer Perceptron
MOS	Mean Opinion Score
MPD	Media Presentation Description
MPTCP	Multipath TCP
MMS	Multimedia Messaging Service
Multi-RAT	Multiple Radio Access Technology
NAT	Network Address Translation
NFV	Network function virtualization
NFV-RA	NFV resource allocation
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NS	Network Service
O-RAN	Open RAN
ONAP	Open Network Automation Platform
OPEX	Operational Expenditure
OSI	Open Systems Interconnection
OSM	Open Source MANO
OTT	Over-the-top
P2P	Peer-to-peer
PoP	Point of presence
QoE	Quality of Experience
QoS	Quality of Service

RAN	Radio Access Network
RNI	Radio Network Information
RNIS	RNI service
RNN	Recurrent Neural Network
RTCP	Real-time Transport Control Protocol
RTMP	Real-time Messaging Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SCTP	Stream Control Transmission Protocol
SDN	Software-defined network
SDR	Software-defined radio
SLA	Service Level Agreement
SON	Self-Organizing Network
SRT	Secure Reliable Transport
STUN	Session Traversal Utilities for NAT
SVA	Streaming Video Alliance
SVM	Support Vector Machine
SVR	Support Vector Regression
TCP	Transmission Control Protocol
TURN	Traversal Using Relays around NAT
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UE	User Equipment
UNESCO	United Nations Educational, Scientific and Cultural Organization
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
VNF-CC	VNF Chain Composition
VNF-FG	VNF Forwarding Graph
VNF-FGE	VNF Forwarding Graph Embedding
VNF-SCH	VNF Scheduling
VNFI	VNF Instance
VNFM	VNF Manager
VOD	Video-on-Demand
VR	Virtual Reality
vRAN	Virtual RAN
WebRTC	Web Real-Time Communication
WSN	Wireless Sensor Network

A list of acronyms used throughout the paper is presented in Table 2.1. The rest of the paper is structured as follows. First, section 2.2.2 presents the objective of this work in the context of related surveys. Section 2.2.3 contains an overview of media streaming technologies and protocols, while section 2.2.4 describes the taxonomy of VNFs for media streaming. Section 2.2.5 covers methods for the assessment of performance metrics related to media streaming. In section 2.2.6, we provide an overview of the state-of-art on media streaming network traffic monitoring and analysis. Section 2.2.7 describes the network functions employed to date to enhance the performance of media streaming services, while section 2.2.8 presents the current challenges in the virtualization process of network functions, inside 5G networks and beyond, to assess the open issues and scientific research directions. Finally, we highlight some very valuable international initiatives in section 2.2.9 and assert our conclusions in section 2.2.10.

2.2.2 **Paper Objectives in the Context of Related Surveys**

The objective of this survey is to perform an extensive literature review on the proposed solutions in the realm of VNFs applied to the field of media streaming. The paper also addresses future challenges in this research area. To better understand how VNF solutions fit with media streaming, performance metrics and network traffic monitoring and analysis are necessary aspects to consider. Network analysis allows to design an effective network function by enabling the management and orchestration operations according to network status at any moment. Performance metrics are instead useful to test the effectiveness of the deployed network function to empower the media streaming service. Then, before presenting and comparing state-of-the-art VNF solutions, the paper also covers two more related topics: performance assessment and network traffic monitoring and analysis.

In a media streaming context, performance assessment focuses on the evaluation of the system employed to stream the content. The performance assessment is done by employing metrics and collecting measurements and/or estimations of such metrics. These will involve quantifiable values to track and monitor a streaming session, i.e., video resolution and bitrate, or a related factor which can influence it, i.e., network bandwidth and latency. Metrics are usually collected alongside the streaming session for two main reasons:

Table 2.2: Summary of Previous Surveys on Virtual Network Functions and Media Streaming.

Survey	Scope and topics	Performance assessment	Network traffic	Network virtualization	Year
Adas et al. [42]	Traffic models in broadband networks	-	Modelling	-	1997
Chalmers et al. [43]	QoS in mobile environment	QoS	-	-	1999
Jin et al. [44]	QoS specification for media applications	QoS	-	-	2004
Feng et al. [45]	Network traffic predictors	-	Analysis	-	2005
Chandrasekaran et al. [46]	Network traffic models	-	Modelling	-	2009
Mohammed et al. [47]	Network traffic models	-	Modelling	-	2011
Hoque et al. [48]	Energy efficient media streaming, wireless networks	Energy	Modelling, analysis	-	2012
Alreshoodi et al. [19]	QoS and QoE correlation models	QoS, QoE	-	-	2013
Baraković et al. [49]	QoE assessment over wireless networks, QoE optimization	QoE	-	-	2013
Seufert et al. [21]	QoE assessment for HAS, HAS adaptation strategies	QoE	-	-	2014
Juluri et al. [50]	QoE assessment in VOD services	QoE	-	-	2015
Su et al. [51]	Wireless and mobile networks, video coding, QoE assessment for mobile media streaming	QoE	-	-	2016
Zhao et al. [52]	QoE assessment and management in video streaming	QoE	-	-	2016
Akhtar et al. [53]	QoS and QoE assessment for audio-visual content	QoS, QoE	-	-	2017
Petrangeli et al. [54]	QoE assessment for HAS, QoE-centric management of HAS	QoE	-	-	2018
Skorin-Kapov et al. [55]	QoE assessment for HAS, QoE-centric management of HAS	QoE	-	SDN/NFV, MEC	2018
Barakabitze et al. [56]	QoE assessment, QoE management in SDN/NFV	QoE	-	SDN/NFV, MEC, Cloud/Fog	2019
Barman et al. [57]	QoE assessment and modelling for HAS	QoE	-	-	2019
Zhang et al. [58]	VNF design considerations	-	-	VNF, Cloud/Edge	2019
Navarro-Ortiz et al. [59]	5G Use cases and Traffic Models	-	Modelling	-	2020

- They can be exploited as source of information for the network function, i.e., bandwidth measurement;
- They can be used as a measure of the goodness of a proposed network function. For example, it is possible to collect the video representation level in order to evaluate the effects of the network function on a streaming session.

Network traffic monitoring and analysis is the process of recording and monitoring traffic to gain the required knowledge to back decisions for increasing the network and service delivery performance as well as for executing network operation and management. This is a relevant field of research since the raise of digital telecommunications networks. Network traffic monitoring and analysis has different subareas: traffic characterization or modelling and traffic analysis to allow forecasts.

Traffic characterization or modelling consists in statistical analysis of the traffic in order to create a model which approximately describes the behavior of the network. There is not a universal model which perfectly describes the network, but each one brings its own limitations and is more accurate under specific conditions (network traffic profiles, employed protocols, transmission medium, etc.). Such models are the basis for generating realistic network traffic. Traffic generation attempts to exploit traffic models and provide tools to simulate specific network conditions. The purpose of synthetically replicating real conditions is to test network applications in a known and controlled environment before they come into play in a real deployment.

Finally, traffic analysis allows to predict network events. Here, it is important to study the temporal variability of the network and construct time series models which estimate future traffic patterns based on past observations of the network. The advantage to forecast events is clear, as it facilitates the implementation of proactive actions that prevent from network malfunctions.

Over the years, several surveys focused on performance assessment, network traffic monitoring and analysis or VNF-based solutions, but their scope was limited and did not discuss on the relation between these topics. Table 2.2 shows a summary of related survey papers.

Surveys related to performance assessment usually focus on a specific point of view, e.g., user's QoE or network QoS, meaning that they do not cover all the possible performance metrics. Chalmers et al. [43] provide a literature review of QoS assessment for mobile environment. QoS metrics are grouped into two categories: Technology-Based QoS and User-Based QoS. Jin et al. [44] focus their review on media applications and prefers to group QoS metrics depending on the layers of the end-to-end architecture they belong: Resource layer, Application layer and User layer QoS metrics. Akhtar et al. [53] provide the same classification, but also include QoE performance assessment. Even though QoE assessment should be based on subjective evaluation, a review of objective methods to assess QoE is presented. Objective methods infer QoS performance metrics to estimate subjective scores and are widely employed since they allow to reduce time and resources costs for the execution of the subjective evaluation. QoS and QoE are highly correlated each other, as it is presented by Alreshoodi et al. [19].

Surveys which review QoE assessment methods are wider presented in the literature compared to surveys dealing with QoS. Baraković et al. [49] present the state-of-the-art QoE assessment while using wireless networks, but they do not limit to media streaming services. Su et al. [51] also focus on wireless networks, but they limit to media streaming services. The authors also include reviews of wireless network technologies and video encoding as related topics. Seufert et al. [21], Petrangeli et al. [54] and Barman et al. [57] propose comprehensive surveys on QoE assessment while employing HTTP Adaptive Streaming (HAS) technologies to stream the media content. Performance metrics specific for HAS, referred as influence factors by the former, can directly influence the QoE evaluation. They also provide a review of server and client-side solutions to improve the QoE scores by optimizing the adaptation strategy. QoE assessment by Juluri et al. [50] includes instead a review of both real-time streaming and HAS metrics, but then focuses only on describing methods to assess QoE for Video on Demand (VOD) applications. Zhao et al. [52] include a similar review on state-of-the-art QoE assessment. Both Juluri et al. [50] and Zhao et al. [52] presents and classify objective QoE influence factors. Skorin-Kapov et al. [55] and Barakabitze et al. [56] are the most recent surveys on QoE assessment and describe also SDN/NFV-based approaches to enhance the streaming services.

Concerning energy efficiency performance, Hoque et al. [48] compare energy-related metrics and approaches to improve the efficiency of the streaming service. Solutions for energy efficiency are classified depending on the layer of the Open Systems Interconnection (OSI) model they belong. The authors also provide some considerations on traffic modelling and analysis, as they are enablers for reducing energy consumption.

Differently from the above-mentioned works, our survey covers performance metrics applied to media streaming from all the three different domains separately discussed in previous works: QoS, QoE and energy efficiency. Moreover, we add a review of performance metrics related to business aspect of media streaming services.

Surveys on network traffic do not cover media streaming use cases, but they remain generic, as they do not consider an application-specific traffic. Adas et al. [42] was the first survey on network traffic monitoring, but it was based on research studies on traffic generated before the 2000s. Some of the models are still considered valid, but some new models have been proposed, as described by more recent surveys, such as Chandrasekaran et al. [46] and Mohammed et al. [47]. Navarro-Ortiz et al. [59] is the most recent one and also considers the effects of specific 5G use cases/applications on traffic modelling. Feng et al. [45] lists different approaches for network traffic analysis. Our survey addresses only media streaming use case. Then, we include a review on media streaming-related traffic modelling and analysis.

Several surveys on network virtualization have been published in the last few years, in line with the increased interest in virtualization. Zhang et al. [58] provides generic considerations when designing VNFs. Limited to media streaming scope, as already mentioned before, Skorin-Kapov et al. [55] and Barakabitze et al. [56] discuss the use of virtualized solutions to improve the QoE assessment. Our survey wants to provide a similar review on VNFs, but we want to add other performance metrics and emphasize on media streaming traffic.

Therefore, our survey addresses a more specific scope. From one side, we want to widely present performance assessment, including less discussed metrics in literature, such as energy and business-related metrics. On the other side, we provide a review of network characterization due to media streaming traffic and present network solutions for its optimization. This survey discusses the relation between the VNFs and media streaming, also considering performance assessment and network traffic monitoring and analysis.

2.2.3 Media Streaming Overview

Media streaming refers to the delivery of media content (e.g., live television, video clip, etc.) from a streaming server to a streaming client over a certain network infrastruc-

Table 2.3: Features of streaming technologies.

Tech.	Transport	Manifest file	Common issues	Latency	Available bitrate	Bitrate adaptation	CDN compatible	Encryption
RTP	UDP	no	packets lost & artifacts	very low (≤1sec)	RTCP	encoder	no	no
RTSP	UDP	SDP	packets lost & artifacts	very low (≤1sec)	RTCP	encoder	no	no
RTMP	TCP	no	packets lost & artifacts	low (1-3secs)	RTMP control messages	encoder	no	AES-128 CBC
SRT	UDP	no	packets lost & artifacts	very low (≤1sec)	SRT control messages	encoder	no	AES-128 / 265 CTR
WebRTC	UDP, QUIC-ready	SDP	packets lost & artifacts	very low (≤1sec)	RTCP	encoder	no	AES-128 CTR
HLS	HTTP 1.X / 2.0 over TCP	M3U8	segment buffering & quality switch	high (5-30secs)	representation	player	yes	AES-128 CBC
DASH	HTTP 1.X / 2.0 over TCP, QUIC-ready	MPD	segment buffering & quality switch	high (5-30secs)	representation	player	yes	AES-128 CBC / CTR
LL-HLS	HTTP 2.0 over TCP	M3U8	chunks buffering & quality switch	low (1-3secs)	representation	player	yes	AES-128 CBC
LL-DASH	HTTP 1.1 Chunked over TCP	MPD	chunks buffering & quality switch	low (1-3secs)	representation	player	yes	AES-128 CBC / CTR

ture. The media source can be either live or pre-recorded. In some cases, the Content Provider (CP) is also the owner of the infrastructure employed to stream the content, but recently diverse providers and operators have entered the market successfully with different roles in the media streaming process, e.g., Akamai, Netflix, etc. Some of them have their own proprietary media streaming solutions. However, first solutions were based on the Real-time Transport Protocol (RTP) [60] on top of the User Datagram Protocol (UDP) [61], where the Real-time Transport Control Protocol (RTCP) [60] was employed to monitor network metrics and update the rate control. The choice of UDP was based on its lower latency when compared to the Transmission Control Protocol (TCP) [62], even if it does not guarantee reliability when delivering packets, i.e., lost packets are not re-transmitted when employing UDP. The later explosion of Over-the-top (OTT) services, e.g., Netflix and Hulu, pushed the search for new solutions to deliver Video-on-Demand (VOD) contents, where latency was not a concern, but scalability to cover the increasing user demand for content. In OTT services, the CP streams its content over a public network and an Internet service provider (ISP) is in charge of the actual content delivery. HTTP Adaptive Streaming (HAS) [21] technologies were introduced to deliver OTT content and the use of TCP and HTTP made them attractive since these protocols are ubiquitous. Additionally, almost every device or User Equipment (UE) can establish HTTP-based communications. The HAS-based design has the following advantages over RTP/UDP-based solutions:

- Traverse networks: HAS communications are performed on top of HTTP/TCP stack and uses port 80 and pull-based streaming protocols. These cross current network infrastructure components, such as Network Address Translation (NAT) and firewall devices [63];
- Reuse and scalability: HAS-based media services can reuse existing CDN systems and caching infrastructures without modifications reaching wide audiences;
- User mobility and device heterogeneity: The dynamic content adaptation-enabled player mechanism is accommodated by all latest heterogeneous UEs, i.e., smartphones, tablets, which support user mobility.

Figure 2.1 illustrates the HAS-based adaptive streaming principle. HAS works in pull mode which means that the client pulls the data from a standard HTTP server, which simply hosts the media content. To reduce the effect of network fluctuations on the playback, HAS employs a dynamic content adaptation to provide a seamless streaming experience. The original media content is encoded at multiple representations, which differ from each other in terms of bitrate and/or resolution and are split into segments of fixed time duration (i.e., a segment is usually between 2 and 10 seconds). A manifest file is also generated and stored at the server, which contains information of the available representations including HTTP URLs indicating where to download the segments of each representation. During a typical HAS session, the client constantly measures certain parameters, such as available network bandwidth and playback buffer level. When it requests content, the client first receives the manifest file which is examined. Then, following an internal adaptation algorithm that processes the monitored performance parameters' values and takes decisions according to the desired adaptation policy, the client requests to download from the server the segment of an appropriate representation.

Apart from transport-layer protocols such as RTP, there are also application-layer protocols that are employed in media streaming. Table 2.3 summarizes different aspects

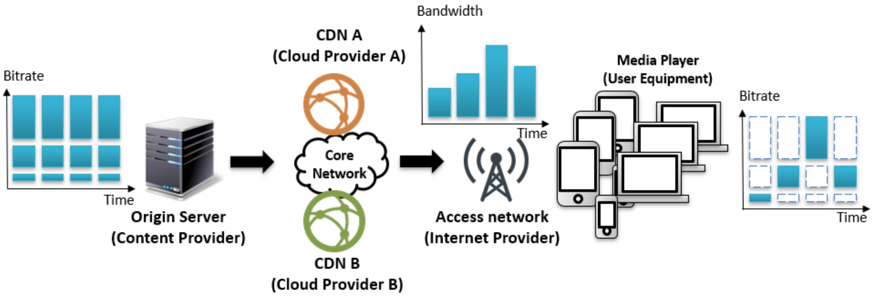


Figure 2.1: HAS-based media streaming principle

of interest when identifying the best protocol candidates to be used when streaming videos. RTP and Real Time Streaming Protocol (RTSP) [64] perform low latency communications compatible with multicast media streaming. TCP-based Real-time Messaging Protocol (RTMP) [65] enables higher reliability compared to RTSP, but at the cost of having higher latency. Secure Reliable Transport (SRT) [66] simplifies the delivery by enabling both push and pull modes of operation. Web Real-Time Communication (WebRTC) [67] enables media streaming through a web browser by exploiting Session Traversal Utilities for NAT (STUN) [68] and Traversal Using Relays around NAT (TURN) [69] protocols provided by third party servers. Both SRT and WebRTC increase the security by including mandatory encryption support, while this is not always required for RTMP. HTTP Live Streaming (HLS) [22] and Dynamic Adaptive Streaming over HTTP (DASH) [23] increase latency due to an internal buffering to overcome network dynamics. In any case, violations on delivery timing could cause stalls and image freezes during the playback if the internal buffer gets empty. To minimise such issues, HAS allows dynamic adaptation mechanisms to track the variability of the network and select appropriate bitrate. Thus, sudden networking problems are prevented by an alternative bitrate selection from the manifest. Common Media Application Format (CMAF) [70] was a proposal to merge major streaming formats around HLS and DASH. Moreover, its Low Latency mode (LL CMAF) aims to reduce the latency by enabling HTTP chunked/push mode. Thus, the latency can be reduced and get closer to UDP-based streaming technologies. In practice, CMAF did not achieve to integration of HLS and DASH streaming formats since the implementations of Low Latency HLS (LL-HLS) [71] and Low Latency DASH (LL-DASH) [72] still present some differences. Thus, LL-HLS and LL-DASH employ different approaches for HTTP transport and encryption schemes.

For instance, a common feature to most HTTP-based solutions is the security by design where different encryption standards protect communications, such as Advanced Encryption Standard (AES) [73] with Cipher Block Chaining (AES-128 CBC) or Counter mode (AES-128 CTR).

Finally, even if most existing media streaming solutions employ UDP and/or TCP, some of them, such as DASH [74] and WebRTC [75], are already evolving and/or being tested with QUIC, a new transport protocol which is expected to substitute TCP when HTTP/3 will replace the current HTTP/2. QUIC lays on top of UDP to provide reduced latency, but with a connection control mechanism to guarantee the same reliability as TCP [76]. There are also proposals to use HAS-based media streaming with protocols such as Stream Control Transmission Protocol (SCTP) [77] and Multipath TCP (MPTCP) [78], which support multihoming, very important in recent heterogeneous network environments. Noteworthy is that MPTCP is backward compatible with the vanilla TCP, which is very useful for service deployment. Finally, efforts are already being made to develop a multipath QUIC [79] protocol to combine the benefits of these approaches, but so far no HAS-based media delivery solution has used it.

2.2.4 Taxonomy of Virtual Network Functions for Media Streaming

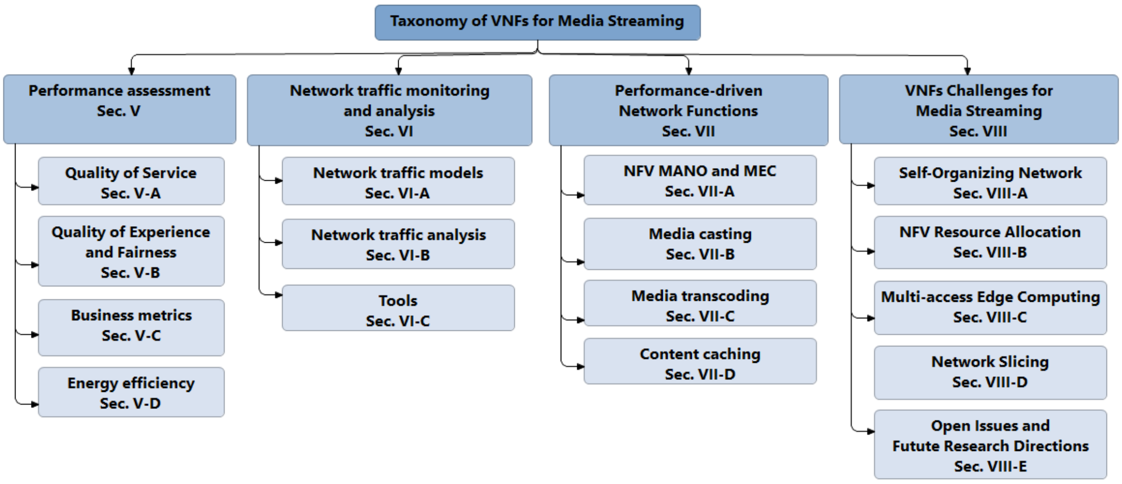

Taxonomy of VNFs for Media Streaming
Performance assessment Sec. V
Quality of Service Sec. V-A
Quality of Experience and Fairness Sec. V-B
Business metrics Sec. V-C
Energy efficiency Sec. V-D
Network traffic monitoring and analysis Sec. VI
Network traffic models Sec. VI-A
Network traffic analysis Sec. VI-B
Tools Sec. VI-C
Performance-driven Network Functions Sec. VII
NFV MANO and MEC Sec. VII-A
Media casting Sec. VII-B
Media transcoding Sec. VII-C
Content caching Sec. VII-D
VNFs Challenges for Media Streaming Sec. VIII
Self-Organizing Network Sec. VIII-A
NFV Resource Allocation Sec. VIII-B
Multi-access Edge Computing Sec. VIII-C
Network Slicing Sec. VIII-D
Open Issues and Futute Research Directions Sec. VIII-E

Figure 2.2: Taxonomy of VNFs for media streaming.

Following the discussion of media streaming solutions, this section reviews the motivations for the applicability of VNFs to improve the media streaming process. The taxonomy of VNFs applied to media streaming is shown in Figure 2.2.

Media streaming can leverage VNFs to enable higher network capacity and stability, media traffic optimization and other performance-related advantages. The final aim is to increase the performances of media streaming, including efficient use of network resources and end device capabilities [80] during their involvement in the streaming service. Media streaming performance indicators include Quality of Service (QoS), Quality of Experience (QoE) and Fairness, Business metrics and Energy Efficiency and illustrated in Figure 2.3. We will discuss media streaming performance assessment from these different perspectives in section 2.2.5.

The employment of VNFs in media streaming is growing in the last few years, as the attention increases on media distribution over the newly deployed 5G networks [81]. VNFs are intrinsically designed to follow the principles of modularity, interoperability, scalability and flexibility. However, to use VNFs more effectively in media streaming, knowledge of the network is essential. Characterizing and modelling network behavior, as well as monitoring and analyzing its traffic provide useful information to be exploited while designing, deploying a VNF [16, 17] and managing its life-cycle [82]. Study of networks and traffic can be tackled from different points of view, as shown in Figure 2.4. There is a wide agreement that real world knowledge allows to design a more mature VNF [83]. This knowledge is collected from network monitoring and data analysis. Considerations on network traffic monitoring and analysis are included in section 2.2.6.

Based on the achievements in performance assessment and knowledge acquired in network and traffic characterization, several network solutions to enable a performance-driven management of the resources are already being employed and/or investigated, as shown in Figure 2.5. Section 2.2.7 deals with performance-driven network functions, including a review of the solutions provided in literature.

Finally, in the current deployment of 5G networks the VNFs have a significant role, as 5G aims to having a fully virtualized network deployment. However, there are still several open issues and challenges that need to be address in the future, as shown in Figure 2.10. Section 2.2.8 discusses the future of VNFs in order to enable an improved media streaming process and enhanced user experience.

2.2.5 Performance Assessment

This section presents an overview of performance assessment avenues in the context of VNF-based media streaming. It involves performance aspects from multiple viewpoints, including QoS, QoE and fairness, business metrics and energy efficiency, as illustrated in Figure 2.3.

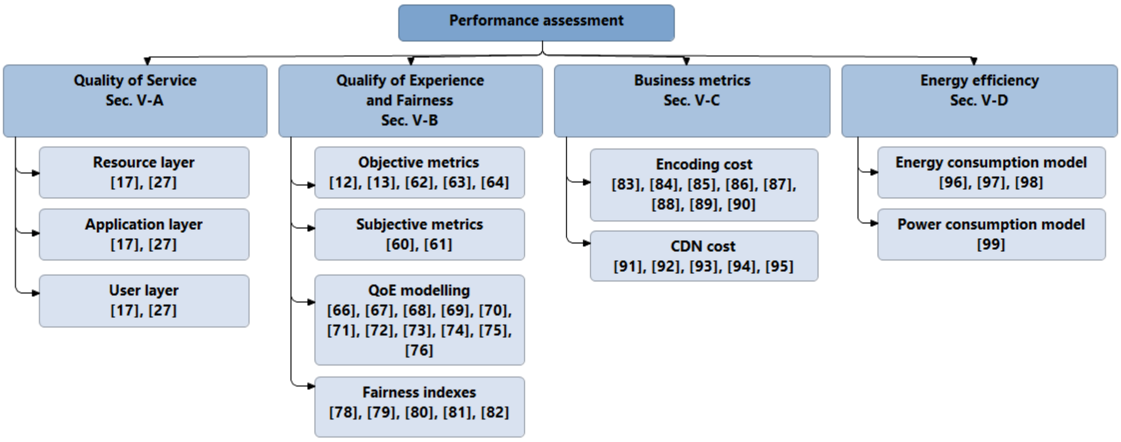

Performance assessment
Quality of Service Sec. V-A
Resource layer [17], [27]
Application layer [17], [27]
User layer [17], [27]
Qualify of Experience and Fairness Sec. V-B
Objective metrics [12], [13], [62], [63], [64]
Subjective metrics [60], [61]
QoE modelling [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76]
Fairness indexes [78], [79], [80], [81], [82]
Business metrics Sec. V-C
Encoding cost [83], [84], [85], [86], [87], [88], [89], [90]
CDN cost [91], [92], [93], [94], [95]
Energy efficiency Sec. V-D
Energy consumption model [96], [97], [98]
Power consumption model [99]

Figure 2.3: Multi-dimensional performance assessment.

2.2.5.1 Quality of Service (QoS)

QoS is related to features which describe the status of network communications and/or the service supported by the network.

QoS properties should be physical and measurable. They are objective performance factors not affected by user's perception of the application, but they will definitely influence this perception. Due to the heterogeneity of networks and/or applications, there is not a unique set of widely accepted QoS properties. Different textbooks and publications introduce differently QoS, but in most cases they focus on a specific network and/or application context. When dealing with a network, QoS assessment consists of measuring network performance, e.g., *network bandwidth*, *packet loss* and *latency*. When considering a particular network application, QoS is linked to a wide range of properties including performance, responsiveness, availability, reliability, and application-related aspects. Each QoS property will be associated to a performance metric. Each metric will facilitate monitoring and characterization of the application property in order to understand the application behavior from that perspective. Furthermore, having a

Table 2.4: QoS performance metrics.

QoS Layer	Property / Category	Parameter / Metric	Description
Resource layer	Timeliness	Packet delay	Time taken to deliver a packet
		Packet jitter	Delay inconsistency between each packet
	Capacity	Channel bandwidth	Occupied frequency range
		SNR	Signal-to-Noise Ratio
		PSNR	Peak Signal-to-Noise Ratio
		MCS	Modulation and Coding Scheme
		Network bandwidth	Maximum (theoretical) data transfer rate
		Throughput	Effective data transfer rate
	Reliability	BER	Bit Error Rate
		PLR	Packet Loss Rate
		Outage probability	Probability that data transfer rate is less than the required threshold
Application layer	Timeliness	Startup delay	Time to receive and display the first video frame
		End-to-End Delay	Time elapsed from content production to its consumption
		Queuing Delay	Time the video frame waits in the playback queue before being displayed
		Audio&Video synchronization	Audio and video are synchronized (no lip sink error)
	Capacity	Audio bandwidth	Audio frequency range
		Audio sampling rate	Audio samples recorded every second
		Video resolution	Pixels in each dimension that can be displayed
		Video frame rate	Video frames recorded every second
		Audio&Video codecs	Codecs employed for audio and video encoding
		Audio&Video encoding bitrates	Bitrates employed for audio and video encoding with the given codecs
		Audio&Video representations	The representation levels presented in HAS
	Reliability	Video Frame Loss	Video frames lost while displaying
		Representation switches	Switches between audio and/or video representation levels
		Stalling ratio	Probability of stalling events
		Stalling duration	Duration of stalling events
User layer	Technical	Device type	Smartphone, tablet, TV, etc.
		Screen/window size	Size of output screen/window
		Content type	Video conferencing (real-time), Live Streaming, Video on Demand, etc.
	Economic	Pricing model	Flat-Rate or Pay-per-Use pricing
		Range of price	High, medium or low price

characterization of the application based on metrics associated with different QoS properties allows to put in place actions to optimize the operations of the application at run-time.

Focusing on media streaming and based on the proposals made in [44] and [53], QoS properties are classified depending on the level of abstraction from the underlying network and hardware/software capabilities. Three main groups of QoS properties (i.e. introducing a QoS layer structure) are defined in relation to different concerns: resources, applications and users. Typical QoS properties and parameters/metrics to measure each property for each these three groups are summarized in Table 2.4 and are discussed next.

The performance metrics at resource QoS layer quantify physical resource properties and are highly dependent on the hardware and platform employed. At this QoS layer, the most interesting ones for media streaming are *timeliness*, *capacity* and *reliability*. Properties at resource layer should fit the requirements needed by the streaming service exploiting those resources. Metrics are not dependent on any particular application and/or user, rather than on the service requirements e.g., video conferencing has tight packet delay and jitter requirements, while video on demand (VOD) requires high throughput. Consequently, the performance metrics are generic for a range of applications and are measured on different OSI network model layers ranging from layer 1 (physical layer) to layer 4 (transport layer). Performance metrics at different OSI layers can characterize the same QoS property, but the abstraction from the physical resource becomes higher as the layer level increases.

At application QoS layer, similar properties to resource layer can be identified, but the metrics are now completely independent from the hardware and platform, as they are application-specific and can be mapped on the network application layer (OSI layer 7). For instance the performance metrics for media streaming at application layer are highly dependent on the video and audio encoding/decoding, streaming technology for the delivery and any other application-level media processing. These metrics are completely abstracted from the network protocols, meaning that they could remain valid even if the content is a file stored locally, as the application layer is agnostic about the origin of the content (local repository or remote server). The content production itself already provides QoS parameters, via audio and video codecs and their encoding bitrates. These parameters are fixed in the encoding and/or decoding process, and they do not depend on the underlying layers. These QoS attributes refer to the characteristics of the encoding at media server and multimedia capabilities at the player device. These content and device characteristics have fixed values during the streaming session and are easily known. In some situations, content characteristics are not enough to describe the QoS at application layer. Additional characteristics dependent on measurements at player side while presenting the content are used. Video impairments provide an objective measurement of the QoS level of the streaming.

Although application layer QoS properties provide high level objective metrics, they are not enough to describe user's point of view. User can be influenced by both objective and subjective factors. User layer QoS aims to identify objective metrics which describe the streaming service from user's point of view. The fact that they remain objective means that they can be measured. Here, these objective properties are completely different since they are neither based on physical resources, nor technological assets, while they deal with external features. We consider that there are two main categories to classify user layer performance metrics: technical metrics and economic ones. Technical metrics describe user device and streaming content. Economic metrics includes consideration on streaming service pricing.

Finally, user subjective metrics are instead not uniquely quantifiable as the user QoS layer ones. They are also referred to as QoE since they are focused on how the user perceives the media service. A more detailed explanation on QoE assessment is described in the next section.

2.2.5.2 **Quality of Experience and Fairness**

Quality of Experience (QoE): QoS performance metrics do not express well users' perceived quality and satisfaction with services. A major reason relies on the fact that human evaluation is influenced by subjective factors that cannot be easily defined by quantifiable parameters and then measured. Therefore, the term QoE is employed to define and describe how a user perceives the media streaming service. Having good values for QoS metrics is not enough to guarantee a certain level of QoE, as it does necessarily imply that the perceived quality is also good. QoS performance metrics can be considered as QoE objective metrics, but additional QoE subjective metrics are necessary. Table 2.5 shows widely used QoE subjective metrics.

Table 2.5: QoE subjective metrics.

Category	Examples	Description
Contextual factors	Location	Home, office, car, etc.
	Environmental characteristics	Noisy or quite, crowded or uncrowded, etc.
	Motion	Sitting or moving, speed, etc.
	Time	Time of the day
Human factors	Age	User's age
	Mood	Emotional state at any time
	Attention level	Attention level at any time
	Goal	User's aim
	Motivation	Level of motivation

Table 2.6: Mean Opinion Score levels.

MOS	Quality	Impairments
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

The International Telecommunication Union (ITU) defines the Mean Opinion Score (MOS) [84] as the measure for the QoE evaluation. It is a widely consolidated way to evaluate the QoE and consists in five quality increasing levels: 1-Bad, 2-Poor, 3-Fair, 4-Good, 5-Excellent. MOS levels are shown in Table 2.6. MOS level achieved by a particular streaming service is assessed by arithmetic mean over all the individual ratings by subjects which take part in the evaluation test. Nevertheless, due to the unpredictability of the subjective factors, a considerable number of scenarios could be possible while assessing the QoE. Then, ITU addresses this issue by attempting to standardize the scenario and environmental variables where the QoE ratings are collected. ITU describes the procedures to assess MOS in the correct way [85]. The procedure intrinsically entails a long time since it requires to select a diverse group of people to represent a good approximation of a typical human audience for a given content. Then, the content should be shown to all the subjects of the chosen set and rated by them.

To simplify QoE assessment, the correlation between QoS and QoE is widely investigated in literature [18, 19] to profile the subjective human perception of the quality. Consequently, quality assessment based on Peak Signal-to-Noise Ratio (PSNR) when considering comparatively the viewed video frames and the original ones has been replaced by more accurate metrics, such as Structural similarity (SSIM) [86], SSIMplus [87], and Netflix' Video Multi-Method Assessment Fusion (VMAF) [88]. While PSNR and SSIM are limited to spatial analysis of video frames, SSIMplus and VMAF include both spatial and temporal analysis. VMAF also moves from employing statistical analysis methods to machine-learning algorithms. VMAF evaluates several elementary metrics which measure content characteristics, type of artifacts and degree of distortion and uses inference to deliver a more accurate final score. Furthermore, Netflix introduced the concept of Per-Title Encoding [27], the same metrics employed to evaluate the user's QoE can be exploited at the server-side while encoding the content. Per-Title encoding allows to select the encoding bitrate which maximizes the user's QoE depending on the type of the media content (i.e. news, sport, action movie, etc.).

Table 2.7: QoE models for HAS.

Model	Description	MOS scale	Year
De Vriendt et al. [89]	Bitrate model, PSNR/SSIM model, chunk-MOS model and Quality level model	yes	2013
Yin et al. [90]	Normalized QoE	no	2014
Xue et al. [91]	Instantaneous and cumulative QoE with exponential decay	no	2014
DASH-UE (Liu et al.) [92]	DASH User Experience model	no	2015
Bentaleb et al. [93]	SSIMplus-based QoE	yes	2016
SQI (Duanmu et al.) [94]	Streaming QoE Index	yes	2016
U-vMOS (Huawei) [95]	User/Unified/Ubiquitous video Mean Opinion Score	yes	2016
ITU-T P.1203 [1]	Parametric bitstream-based quality assessment for HAS services	yes	2017
KSQI (Duanmu et al.) [96]	Knowledge-driven streaming quality index	yes	2019
De Fez et al. [97]	Modified Yin[90]-model, PSNR-based model, VMAF-based model	yes/no	2020
ITU-T P.1204 [98]	Bitstream-based/pixel-based/hybrid models for resolutions up to 4K	yes	2020

Focusing on HAS, diverse metrics have been considered to create QoE models based on its characteristics. QoE models span from pixel-level comparison between received frames and original ones (e.g. PSNR, SSIM, SSIMplus and VMAF) to content-agnostic models with sophisticated equations which consider a wide range of parameters, including available representation bitrates, frequency of bitrate changes and buffering duration. By focusing on objective metrics only, there is an inevitable loss of accuracy, but it has several practical advantages. The absence of human feedback on the QoE reduces the test time and result processing can be automated to be carried out online.

Common QoE models are presented in Table 2.7. Even ITU defines MOS as the standard metric, MOS scale is not employed by all the QoE models as a measure of achieved QoE ratings.

ITU proposes several models, including ITU-T P.1203 [1] and ITU-T P.1204 [98]. ITU-T P.1203 [1] is a parametric bitstream-based model for HAS services which expresses the result in terms of MOS. The model considers both audio and video features, the impact of buffering on perceived quality and also takes into account information on the employed display device. Due to the performed bitstream analysis, a real implementation [99] of this model is computationally intensive as content is analyzed on a per media segment and per video frame basis. The model introduces four modes of operation, from 0 to 3, to tune the trade-off between accuracy and complexity. Lower modes are less accurate to reduce complexity, while higher mode increase the complexity to gain accuracy. Finally, modes 3 and 4 also raise security issues, as the bitstream must be unencrypted/decrypted to access the required input information. ITU-T P.1204 [98] the newest standard from ITU and it is not actually a unique model, but it groups models of different type: bitstream-based, pixel-based and hybrid models. ITU-T P.1204 is meant to be an extension to ITU-T P.1203, as it is focused on higher resolutions (up to 4K). Unfortunately, both ITU models show intrinsic computational complexity.

As an alternative to the complex ITU models, other QoE models are proposed in the literature. In equation (2.1) De Vriendt et al. [89] formulate a general expression for QoE models to predict the results of HAS services on MOS scale.

$$M_{pred} = \alpha * \mu - \beta * \sigma - \gamma * \phi + \delta \tag{2.1}$$

where α, β, γ δ are tunable coefficients. μ and σ are average of the quality of the displayed HAS representations and its standard deviation, respectively. Finally, ϕ takes into account both average duration and frequency of freeze events. From the equation, it is clear that the quality estimation is influenced by some major factors of HAS services: quality associated to each representation, switches between representations and stalling/buffering events. The coefficients are tuned by minimizing the Mean Square Error (MSE) between the predicted MOS values (M$_{pred}$) and the real ones assessed by rating a set of different video clips on different devices, as shown is equation (2.2).

$$\frac{\sum_{n=1}^{N}(M_{pred,n} - MOS_n)^2}{N} \tag{2.2}$$

The selection of values for μ and σ leads to different types of models, as several ways to define the quality of a representation are possible. De Vriendt et al. [89] state that there are at least 4 ways to select the quality:

- Bitrate model: the quality is defined by the bitrate of the representation.
- PSNR or SSIM model: the quality is defined by the average PSNR or SSIM over all the frames of a segment.
- Chunk-MOS model: the quality is not calculated from the representation, but it is part of the same MSE minimization process.
- Quality level model: the quality levels are equally spaced between a minimum and a maximum value.

The authors conclude that the chunk-MOS model has the best performance, with more flexibility for optimization since two more parameters (μ and σ) are varied to improve the model.

Yin et al. [90] suggest a similar approach that considers the same variables to assess a normalized QoE. Later models start from a similar optimization problem expressed by the equation (2.1) and aim to expand by including further variables or defining differently the quality associated to each representation. Liu et al.'s DASH-UE [92], SQI (Duanmu et al.) [94] and Huawei's User/Unified/Ubiquitous video MOS (UvMOS) [95] also include the startup (initial) delay in the equation. They assert that startup delay has negative effects on the user's QoE. Xue et al. [91] perform instantaneous QoE score estimations and introduce an exponential decay to emulate the forgetting curve of human perception when evaluating the cumulative QoE score.

Bentaleb et al. [93] employ SSIMplus [87] to assess the quality related to each representation instead of the four approaches proposed by De Vriendt et al. [89]. Duanmu et al.'s Knowledge-driven Streaming Quality Index (KSQI) [96] considers the same variables, aims to include a human visual system (HVS) analysis result to improve QoE modelling.

The authors derive a system of linear inequalities from QoE subjective studies which allows to improve the optimization problem of QoE modelling.

Finally, De Fez et al. [97] propose three different models. The first one is a modified Yin et al. [90] model which improves accuracy by using the actual video segment bitrate instead of the average value of the segment representation. The second and the third ones are a PSNR-based model and a VMAF[88]-based model, respectively. These models employ PSNR and VMAF metrics to evaluate the quality of each segment instead of the actual video bitrate.

Fairness: While the number and diversity of approaches to assess QoE is large, there are very few metrics to measure fairness. Jain's fairness index [100] is one of the most widely used such metric and was originally introduced to express the fairness of throughput distribution across multiple flows that share a common distribution infrastructure. However, its applicability can be extended to any set of values x_i, which are measured on a scale, where $i = \overline{1,N}$. Note that the minimum Jain's fairness value is $\frac{1}{N}$ and the maximum is 1.

$$J(x_1, x_2, ...x_N) = \frac{(\sum_{i=1}^{N} x_i)^2}{\sum_{i=1}^{N} {x_i}^2} \tag{2.3}$$

Unfortunately, even though many networking parameters can be measured on ratio scales, there are some (i.e., QoE is among them), which are expressed on interval scales, such as the 5-point MOS scale, for instance. For one of these situations, Hoßfeld et al. [101] have proposed a QoE Fairness index based on the lowest L and highest H bounds of the rating scale. In (2.4) σ is the standard deviation and measures the degree of dispersion of the values. Hoßfeld's fairness index has values in the interval $[0,1]$, where 0 is associated with total unfairness and 1 with perfect fairness.

$$F = 1 - \frac{2\sigma}{H - L} \tag{2.4}$$

There are some situations when classic fairness metrics do not reflect well the actual distribution of values. A more generic product-based fairness metric, presented in equation (2.5), was discussed in [102] along with other fairness metrics. In equation (2.5) f is a transformation function which can be defined according to the desired effect, allowing for very high flexibility in the fairness assessment.

Table 2.8: Business costs for media streaming.

Category	Examples	Description
Capital expenditure	Buildings and furniture	Buy buildings, server racks, etc.
	Equipment	Servers, laptops, monitors, etc.
	Intangible assets	Purchased licenses or patents
	Software	Commercial proprietary software
Operational expenditure	Utilities	Electricity, water, etc.
	Employees	Salaries and benefits
	Research and development	Develop/improve media service
	Encoding	Costs due to content preparation
	CDN usage	Costs due to content delivery

$$\mathscr{P}(x) = \prod_{i=0}^{N} f\left(\frac{x_i}{\max(x)}\right) \tag{2.5}$$

The simplest product-based fairness index which uses a linear function $f(x) = x$ is represented in equation (2.6).

$$LP(x) = \frac{\prod_{i=0}^{N} x_i}{\max(x)^N} \tag{2.6}$$

Two other product-based fairness indexes, G's and Bossaer's, are defined using $f(x) = \sin(x\pi/2)^{\frac{1}{k}}$ and $f(x) = x^{\frac{1}{k}}$, respectively. While the first emphasizes the values closer to $\max(x)$, the later inflates the values closer to 0.

Other approaches include the general fairness model proposed by Lan et al. [103] and min-max and max-min-based fairness indexes introduced by Radunovic et al. [104].

2.2.5.3 **Business Metrics**

Achieving higher QoS and QoE values comes at a cost for the Content Provider (CP), because there are generally increased expenses in terms of network/services resources. Nevertheless, CP's strategies should focus on minimizing the business costs, while guaranteeing the same or even higher QoS/QoE. Table 2.8 provides a list of common business costs for a typical CP-based media streaming.

Capital Expenditure (CAPEX) refers to expenses incurred by the CP for the acquisition or improvement of fixed assets that are necessary for the business. CAPEX includes intangible assets, specific software as well as expenses related to licenses and patents payments. In this sense, the use of video codecs is the most evident example. Moving

Picture Experts Group (MPEG) codecs require payments of licenses (royalties) for commercial use. H264 remains widely used, and it is still supported by most end devices. The royalties for using HEVC have increased [105] and this fact prevents some CPs from using HEVC (and maybe its successor VVC) and drives their interest towards royalty-free alternatives [106]. VP8 and VP9 were developed and released by Google, which later joined the Alliance for Open Media (AOM) with other mayor tech companies to work on the AV1 video codec. Nevertheless, as the MPEG-Google/AOM codecs struggle is still on-going, other factors may influence CP decisions on the employed codec, including limitations from device manufacturers (hardware encoding and decoding capabilities) and/or browser capabilities [107].

Operational Expenditure (OPEX) refers to on-going costs for running the business and inherent to the operation of the assets. Except from expenses common with every business, encoding and CDN usage are the most relevant and specific to media streaming services. Once the codec has been chosen, the encoding operations may have other operational costs that vary depending on the encoder choice (i.e., open-source or commercial) and where the encoder runs (i.e., cloud or on-premise encoding). Cloud encoding prices are established by cloud providers [108], while on-premise coding depends on the hardware selection and maintenance. On the other side, on-premise encoding allows to have more control on the processed data and content compared to cloud encoding [109, 110]. The total encoding cost is expressed in equation (2.7).

$$Enc_{cost} = codec_{royalties} + encoder_{price} + server_{cost} + processing_{cost} \tag{2.7}$$

In equation (2.7), codec$_{royalties}$ + encoder$_{price}$ expenses belong to CAPEX, while processing$_{cost}$ is an OPEX. server$_{cost}$ depends on the strategy, a cloud encoder generates an OPEX, while an on-premise encoder needs an equipment investment which is a CAPEX. Focusing on processing$_{cost}$, employing Per-Title encoding and CMAF can reduce the OPEX. Per-Title encoding enables optimization of the encoder adjustment [111], and provides a more effective bitrate and resolution choice to optimize the tradeoff between QoE and processing resources. CMAF guarantees compatibility between different HAS technologies, meaning that the encoded content can be shared between them. Thus, a single encoding operation is necessary for both DASH and HLS [112].

The cost of CDN resources depends on their ongoing utilization [113]. Not all providers publish their own pricing plans since in most of the cases they offer personalized plans to each customer. Nevertheless, [114], [115] and [116] reveal common factors that influence the OPEX for CDN resources, such as the outbound network traffic, storage occupancy and usage time. Thus, CDN OPEX can be expressed as (2.8) [117]:

$$CDN_{cost} = \sum_{i=1}^{N} (\alpha_{loc_i} * Tr_i + \beta_{loc_i} * K_{req_i} + \gamma_{loc_i} * T_i + \delta_{loc_i} * St_i + \epsilon_{loc_i}) \tag{2.8}$$

In equation (2.8), Tr_i and K_{req_i} represent the traffic volume and the number of HTTP requests producing this traffic. T_i and St_i are the utilization time for a CDN (active sessions from video players) and the employed storage at CDN, respectively. Finally, α_{loc_i}, β_{loc_i}, γ_{loc_i}, δ_{loc_i}, and ϵ_{loc_i} are multiplicative coefficients established by the CDN provider and are dependent on the location of CDN resources (the cost of a cloud server depends on the geographical location). CPs usually employ simultaneously more than one CDN to increase coverage and in consequence the addition means a sum over the N available CDNs. The values of the coefficients depend on the business model and the pricing plan of each CDN provider. On the contrary, the variables which depend on the CDN usage (Tr_i, K_{req_i}, T_i and St_i) can be exploited by the CP to optimize the CDN resource selection and achieve a trade-off between QoS/QoE and cost.

2.2.5.4 **Energy Efficiency**

Efficient usage of energy has become a worldwide critical challenge. There is a very strong motivation for researchers to propose and develop energy efficient techniques in order to manage the power consumption in both current and future network environments. The range of green networking solutions covers a wide area. There are centralized network-centric approaches, where operators would deploy and positively influence large scale systems. A different strategy is based on individual solutions, which can be deployed considering a user-centric paradigm. There are energy preservation solutions which target equipment functionality and others which influence data exchange protocols, mechanisms which involve single components and others which target communication and cooperation between units, solutions deployed at a single network layer or across multiple layers, schemes which are made public and approaches which are proprietary, etc.

In this complex energy-aware research, there is a natural interest on solutions for energy efficient delivery of multimedia content with focus on end-user terminal devices. The latest wireless smart mobile devices are deployed with limited battery-based power resources, while computational and content presentation-related complexity has increased exponentially. Kennedy et al. [118] studied mobile device components' energy consumption. These authors have noted that screen, CPU, audio and network units scored the highest in terms of energy consumption, with a large gap between minimum and maximum values for the presentation components (i.e., screen and speakers). Lately, by using hardware optimization solutions for content presentation, the consumption associated to screen and audio interfaces has been reduced at the cost of increased processing complexity as well as increased data transfer. It is therefore fundamental to achieve energy efficiency for rich media content exchange between smart device and other sources or in-between such devices in order to extend operational activity of the devices and support high user QoE.

For a device, energy consumption E is the sum of the energy consumed in data transmitting mode (Tx), data receiving mode (Rx), sleeping mode (Sl) and during state transition (Sw).

$$E = E_{Tx} + E_{Rx} + E_{Sl} + E_{Sw} \tag{2.9}$$

Energy efficiency is generally defined as information bits per unit of transmission energy. A typical function of energy efficiency calculation for an additive white Gaussian noise channel is shown in equation (2.10) [119]:

$$\eta = \frac{2R}{N_0(2^{2R} - 1)} \tag{2.10}$$

where the channel capacity R is defined as in equation (2.11):

$$R = \frac{1}{2}\log(1 + \frac{P}{N_0 B}) \tag{2.11}$$

and P represents the transmit power, N_0 represents the noise power spectral density and B represents the system bandwidth.

However, most solutions require dynamic computation of energy consumption and the components with the largest contribution to the overall energy budget are E_{Tx} and E_{Rx}. Even though transmission energy consumption is expected to exceed the ammount required by reception functions, the literature associates the energy consumption with network interfacing activity in general, and not with any specific communication.

There are two models oftenly used in the literature. In the context of a mobile device, equation (2.12), proposed by Trestian et al. in [120], calculates the energy consumption as follows:

$$E = t(r_t + Th * r_d) \tag{2.12}$$

where E is the estimated energy consumption (Joule) for a RAN, t represents the transaction time (seconds), r_t is the mobile device's energy consumption per time unit (Watt), Th is the throughput (Kbps) and r_d is the energy consumption rate for the data stream (Joule/Kbyte). The parameters r_d and r_t are device specific and differ for various network interfaces present at the device side.

A second power consumption model for a sensor node was introduced by Zou et al. in [121] and is described by equation (2.13). According to this model, the theoretical power consumption p_r of a wireless interface r is proportional to the throughput Th_r, as indicated in equation (2.13).

$$p_r(Th_r) = \alpha_r * Th_r + \beta_r + \gamma_r \tag{2.13}$$

In equation (2.13), p_r is the power expressed in Watts, α_r is the energy consumption rate for data in mJ/Kb for the interface r, Th_r denotes the data rate in $Kbps$ on interface r, β_r is the energy consumption per unit time in $mWatt$ for the interface r and γ_r is a constant which is a tunable value associated with the background energy consumption for interface r. If the node is equipped with a total of R interfaces, the total power consumption is calculated as in equation (2.14):

$$P = \sum_{r \in R} p_r(Th_r) \tag{2.14}$$

2.2.6 **Network Traffic Monitoring and Analysis**

This section overviews major research activities related to network and traffic monitoring. Studies are performed from different points of view, as shown in Figure 2.3. Research activities include investigations on statistical models which can approximate network traffic behavior and analysis of the network traffic to acquire valuable information to be used to improve the network performance. In this section we also include a review of tools employed for performance monitoring and network simulation.

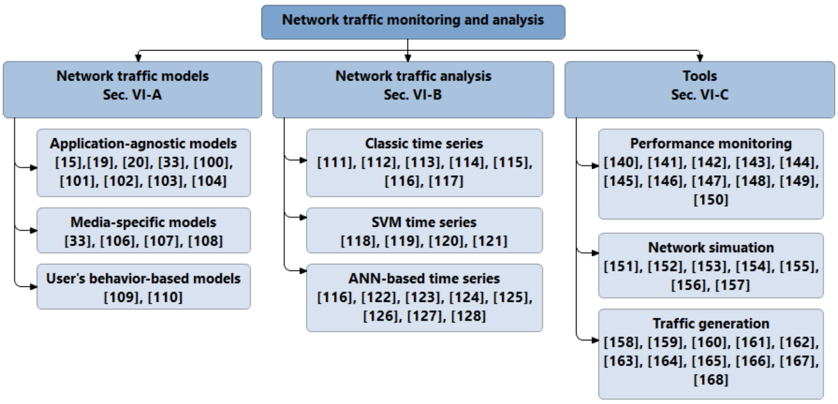

Network traffic monitoring and analysis
Network traffic models Sec. VI-A
Network traffic analysis Sec. VI-B
Tools Sec. VI-C
Application-agnostic models [15],[19], [20], [33], [100], [101], [102], [103], [104]
Media-specific models [33], [106], [107], [108]
User's behavior-based models [109], [110]
Classic time series [111], [112], [113], [114], [115], [116], [117]
SVM time series [118], [119], [120], [121]
ANN-based time series [116], [122], [123], [124], [125], [126], [127], [128]
Performance monitoring [140], [141], [142], [143], [144], [145], [146], [147], [148], [149], [150]
Network simuation [151], [152], [153], [154], [155], [156], [157]
Traffic generation [158], [159], [160], [161], [162], [163], [164], [165], [166], [167], [168]

Figure 2.4: Network traffic monitoring and analysis.

2.2.6.1 **Network Traffic Models**

Table 2.9: Application-agnostic network traffic models.

Model	**Description**
Poisson [122]	Memoryless distribution of the arrivals from independent sources (Poisson sources)
Non-stationary Poisson [123]	Non-stationary Poisson behavior at multi-second time scales
Log-normal [124]	Inter-arrival times from aggregated sources modelling
Pareto [124, 125]	Inter-arrival times from aggregated sources modelling, End-to-end delay modelling
Weibull [126]	Inter-arrival processes (packets, flows and sessions) modelling
Markov [42]	Model of activities of a traffic source with exponentially distributed time between state transitions
Embedded Markov [42]	Model of activities of a traffic source with arbitrary probability distributed time between state transitions

Over the years, network traffic models have been thoroughly studied within the communication networks domain to describe the behaviour of discrete entities, namely, packets, connections, etc. In statistics and probability theory, this kind of traffic is described as a Point Process [127]. Many models have been proposed, with advantages and disadvantages, appropriate or not to different types of networks (i.e. Ethernet, Wi-Fi, LTE/5G, etc.) and with support for diverse scenarios. The choice of traffic model to employ depends on the particular network under study and the demand characteristics. The models are useful to perform any optimization and to produce a robust and reliable network infrastructure design. Moreover, they are essential to design and experiment network services since they can be employed to recreate a realistic traffic scenario in a controlled environment (laboratory). Thus, network services are tested and validated before being deployed in production. Obviously, each model has some assumptions that limits its usage. In other words, the models are not perfect, but their approximation is good enough for experimentation purposes.

In [46] and [47], the most common application-agnostic network traffic models are presented, i.e., these models focus on characterizing generic network packet arrivals. The Poisson distribution model is one of the oldest models, but it is still widely employed across the literature to model packet arrivals from independent sources [122]. The authors of [123] carried out a deep analysis of network traffic to study the limitations of the Poisson model. The authors propose a non-stationary Poisson model as the Poisson model accurately characterizes traffic only at sub-second time scales. At multi-second time scales the traffic seems to have a non-stationary behavior. The Log-normal and Pareto distributions are employed to model inter-arrival times from aggregated sources [124]. Moreover, the Pareto distribution also models end-to-end network delay [125]. The Weibull distribution describes inter-arrival processes at different levels, meaning that it fits with packets, flows and sessions arrivals by tuning its parameters [126]. Markov and Embedded Markov models are used for network sources with a finite number of states, e.g., voice telephony has idle, busy and transmit states [42], and they differ in describing the time between state transitions. Table 2.9 presents the most employed application-agnostic models.

Other studies have focused on media specific applications instead on traffic-agnostic ones. In [59], the authors employ some models proposed in literature to describe the traffic generated by specific 5G use cases/applications. In [128], a traffic analysis of an IPTV CDN network is presented. The authors find that the bitrate of multicast flow is relatively stable and depends on the number of live broadcast channels, while the bitrate of a unicast flow varies along the day and presents differences between weekdays and weekend. In [129], the traffic characteristics of Netflix and YouTube were analyzed, and the findings reveal that data is transferred through ON-OFF cycles, whose duration is dependent on the user's device and browser. In [130], a study of YouTube traffic reveals that the traffic is highly dependent on the hour of the day. Moreover, the inter-arrival between two consecutive video requests depends on the popularity of the video.

Finally, [131] and [132] present two studies of user behavior while accessing streaming services. In [131], the authors focus on VOD streaming and they note that the user inter-arrival rate can be modelled by a modified Poisson distribution. Once streaming was accessed, the session length varied depending on the video duration. Furthermore, they find that more than half of the overall sessions end within ten minutes, while more than one third ended within 5 min. The user behavior has also some variations depending on the day of the week, as during the weekend, the video requests increase. In [132], the authors also consider live streaming. They find that a Poisson distribution is less accurate when modelling user inter-arrival for live streaming services than for VOD ones.

2.2.6.2 **Network Traffic Analysis**

The ability to model and generate realistic network scenarios offers the possibility to design and deploy network functions that adjust to the network traffic at any moment. Analyzing network traffic and applying time series analysis means a further step since it allows to forecast future network traffic. Network functions could move from reactive to proactive approach by exploiting predicted future conditions of the network. Actions are proactively taken when performances are going to not be satisfied. Thus, network under-performance and outages are prevented.

There are many methods proposed in literature for time series analysis. Among them, we distinguish classic time series, Support Vector Machine (SVM) time series and Artificial Neural Network (ANN)-based time series [151]. The choice of a predictor based on one of the different time series approaches depends on the characteristics of the network and different approaches are suitable for traces from different sources [152]. Moreover, in the same scenario different approaches could be combined to predict both long-term traffic demand and short-term network metrics [153]. Classic time

Table 2.10: Methods for time series analysis applied to networks.

Method	References	Approach	Number of variables	Configuration / parameters	Description
ARIMA	[133, 134, 135, 136, 137]	classic	univariate	regression, integration and moving average parameters	Autoregressive integrated moving average
SETARMA	[135]	classic	univariate	regression, moving average and threshold delay parameters	Self-exciting threshold autoregressive moving average
GARCH	[135]	classic	univariate	regression and lag length parameters	Generalized autoregressive conditional heteroskedastic
Holt's linear trend	[134]	classic	univariate	smoothing factor	Secondary or double exponential smoothing time series
Holt-Winters' seasonal	[138, 139]	classic	univariate	smoothing factor	Cubic or triple exponential smoothing time series
SVR	[140, 141]	SVM	multivariate	weight vector and offset	Support Vector Regression
H-SVM	[142]	SVM	multivariate	weight vector and offset	Hierarchical Support Vector Machine
Multi-class SVM	[143]	SVM	multivariate	weight vector and offset	Multi-class Support Vector Machine
Feed-forward NN	[144]	ANN	multivariate	weight and bias	Feed-forward neural network
MLP	[138, 145]	ANN	multivariate	input vector, weight vector and bias	Multi-layer Perceptron
FNN	[145]	ANN	multivariate	input vector, weight vector and bias	Fuzzy Neural Network
RNN	[146]	ANN	multivariate	input, output and forget factors	Recurrent neural network
LSTM	[147, 148, 149]	ANN	multivariate	input, output and forget factors	Long short-term memory
ESN	[150]	ANN	multivariate	input, reservoir and output weights	Echo State Network

series approaches are well known, as they were defined prior to the raise of telecommunication networks. On the contrary, SVM and ANN-based solutions can be seen as new contenders to classic ones, as Machine Learning (ML) application for time series prediction [154] is relatively new (SVM and ANN are two different supervised learning approaches). Time series methods employed for network forecasting are shown in Table 2.10. The table also presents the main differences between them, such as the number of input and output variables and the selection of internal parameters. ML (SVM and ANN) models take the advantage from the knowledge of several variables as input (multivariate), while classic ones are limited to one (univariate). The same is valid for output variables, ML models can output more than one. The outcomes of [155] and [156] reveal that a higher number of input variables improves the traffic predictions of a ML model (the authors employ a Long short-term memory model) when compared to a classic one (the authors employ an autoregressive integrated moving average model).

The autoregressive integrated moving average (ARIMA) [157] is one of the oldest time series method and widely employed in literature as reference method for evaluating any other time series approach. In [133], ARIMA is employed to predict the workload of cloud services. Historical observed requests are exploited to predict the volume of requests during the next time interval. The authors find limitations to track traffic peaks accurately. In [137], ARIMA is instead employed to predict the request number and the amount of data traffic. Other limitations to ARIMA are found in [135] and [136] when modelling QoS attributes which have non-linear behaviors, i.e., time between QoS violations. Thus, they do not fit the linear assumption of ARIMA. Self-exciting threshold autoregressive moving average (SETARMA) [135] and generalized autoregressive conditional heteroskedastic (GARCH) [136] are integrated with ARIMA in hybrid linear and non-linear models to overcome ARIMA limitations.

Exponential smoothing [158] is a subset of classic time series method. Holt's linear trend method (secondary or double exponential smoothing) is employed in [134]. The authors find it complementary to ARIMA when predicting throughput in an LTE network. ARIMA outperforms the exponential smoothing on weekdays, while the exponential smoothing prediction are more accurate on weekends. Holt-Winters' seasonal method (cubic or triple exponential smoothing) is instead employed in [138] and [139]. In [138], it is employed to implement an anomaly detection, while, in [139], its aim is to predict cloud resource provisioning.

Among SVM time series [159], in [140], a Support Vector Regression (SVR) model is employed to predict TCP throughput. A similar approach with SVR is presented in [141], but it aims to predict network links load and not limited to TCP traffic. In [143], the authors use Channel State Information (CSI) and handover history to determine a user's mobility pattern by means of a Multi-class SVM. The next cell can be predicted based on the previous crossed cells, user's trajectory, and CSI. The problem of estimating the location of mobile nodes is investigated also in [142], but limited to an indoor wireless network, and employing a hierarchical SVM model composed of four different levels. The same method is also employed to estimate channel noise.

Concerning ANN-based approaches, in [144] a Feed-forward Neural Network (Feed-forward NN) for predicting the execution time of services while varying the number of requesters is presented. In [146], a Recurrent Neural Network (RNN) is instead employed to forecast the end-to-end delay from RTT metrics. In [147], a Long short-term memory (LSTM) model, a particular type of RNN, is proposed to process downlink control information (DCI) messages, such as resource blocks, transport block size, and scheduling information. LSTM is also employed in [148] to solve a problem of traffic matrix prediction and in [149] to forecast stalling events during a video streaming session. An Echo State Network (ESN), also a kind of RNN, is employed in [150] to predict traffic volume in a city for various network applications, such as Multimedia Messaging Service (MMS), Web, media streaming, Instant Messaging (IM) and Peer-to-peer (P2P) communication. In [138], a Multi-layer Perceptron (MLP) model is employed to detect anomalies in network traffic. MLP is used jointly with a Fuzzy Neural Network (FNN) in [145] to forecast one-step ahead value of the MPEG and JPEG video, Ethernet, and Internet traffic data. The combined results of the two ANNs outperforms the results achieved by employing only one method.

Being able to forecast network traffic and performances is definitely interesting to provide proactive actions in response to future network issues. In any case, there is not an optimal method, as the better performing method depends on the considered metrics and scenarios. As a result, some hybrid solutions are also being investigated to exploit both the advantages of classic methods and ML (SVM or ANN) ones [160, 161].

2.2.6.3 **Tools**

Performance monitoring:

Table 2.11: Tools for performance collection and visualization.

Domain	Processing model	Automation mode	Forecast Skills	Name	Description
General	Real-time inputs	API and manual GUI	Not Applicable	Prometheus [162]	General-purpose monitoring system and time series database
General	Real-time inputs	API and manual GUI	Not Applicable	InfluxDB [163]	General-purpose monitoring system and time series database
General	Real-time inputs	API and manual GUI	Not Applicable	Grafana [164]	General-purpose platform for decision-making with focus on customizable data charts
General	Real-time inputs	API and manual GUI	Predictions	Elastic Stack [165]	General-purpose platform for monitoring and decision-making with focus on customizable alerts and data charts
Business	Batch and real-time inputs	No API, manual GUI	Predictions and Simulations	Board [166]	Business-purpose platform for decision-making with focus on customizable CRM data charts
Business	Batch, scheduled and real-time inputs	No API, manual GUI	Not applicable	Tableau [167]	Business-purpose platform for decision-making with focus on customizable CRM data charts
Web sessions	Real-time inputs	No API, manual GUI	Not Applicable	Citrix Analytics [168]	General-purpose web activity including user session performance and application usage
Web sessions	Real-time inputs	API and manual GUI	Not Applicable	Google Analytics [169]	General-purpose web activity
Media	Batch, scheduled and real-time inputs	API and manual GUI	Not applicable	Akamai Media Analytics [170]	Media streaming service-specific Analytics solution
Media	Real-time inputs	API and manual GUI	Not Applicable	Conviva Streaming Analytics [171]	Media streaming service-specific Analytics solution
Media and Data	Real-time inputs	API and manual GUI	Not Applicable	Amazon Kinesis [172]	Media streaming service-specific Analytics solution

Performance monitoring, including metrics collection and visualization, can be done though several visual analytics tools, as shown in Table 2.11. Tools are classified depending on the application domain.

Prometheus [162], InfluxDB [163], Grafana [164] and Elastic Stack [165] are open-source and general-purpose solutions. Prometheus [162] and InfluxDB [163] are time series database to collect, monitor and visualize real-time information. Anyway, their visualization capabilities are limited, and thus, they are usually employed jointly with external tools to create and visualize interactive data charts. Grafana [164] is the most common tool for these interactive data charts. It can connect to both Prometheus and InfluxDB or any other database to access data and manage them to create interactive web visualization. Several data charts can also be visualized at the same time by generating a unique dashboard to simplify decision-making operations. Elastic Stack [165] is an alternative to Grafana, but it comes with its own database, called Elasticsearch, and data visualization component, called Kibana, to generate data charts and dashboards. It has a modular architecture to allow adding optional add-ons to increase its capabilities. Among these add-ons, a Machine Leaning (ML) one can enable algorithms to analyze the data.

Board [166] and Tableau [167] are commercial software intended for business analytics. They focus on Customer Relationship Management (CRM) and data charts creation for enabling decision-making. Citrix Analytics [168] and Google Analytics [169] aim to track web activities (browser video players). While Citrix Analytics is a commercial solution, Google Analytics has both a commercial and a free version. This free version is usually enough for research activities. Akamai Media Analytics [170] and Conviva Streaming Analytics [171] are meant for media-specific application, as they manage metrics related to online streaming. Finally, Amazon Kinesis [172] is focused on both media and generic data streaming, as it allows to collect real-time data from heterogeneous sources, such as video and audio, application logs and IoT telemetry.

Network simulation and traffic generation:

Achievements in network traffic modeling and analysis are widely exploited to develop utility software which simulates real networks and/or generates realistic traffic for experimentation. Table 2.12 shows several tools enabling research activity and experimentation with network traffic.

Network simulators allow to simulate networks without having to deploy a real one. A single node running a simulator is employed to generate a network whose capabilities and performance are configurable. Network simulators replicate the physical layer (L1), wireless (Wi-Fi, LTE, 5G) or wired (Ethernet) [173, 174, 175], and configure network typologies to be employed during the experiments [176, 177]. Moreover, almost all simulators are designed to enable the exchange of packets belonging to different L2/3/4 protocols (Ethernet, IP, UDP/TCP). In some cases, they also allow to reproduce more specific network technologies or environments (IoT, WSN, DTN) [178, 179]. Definitely, they are useful when testing through a real network is not feasible due to several reasons, such as equipment costs or physical space for assets.

Table 2.12: Tools for network simulation and traffic generation.

Category	Name	OSI layers	Description
Network simulator	OMNeT++ [173]	L1/2/3/4	Simulation of communication networks, multiprocessors and distributed or parallel systems
Network simulator	NS-2 [174] / NS-3 [175]	L1/2/3/4	The Network Simulator (NS) -2 / -3, Simulation of TCP, routing, and multicast protocols over wired and wireless networks
Network simulator	OPNET [176]	L1/2/3/4	Optimized Network Engineering Tool (OPNET), Simulation of network typologies, nodes and flows
Network simulator	Mininet [177]	L1/2/3/4	Instant Virtual Network to develop and experiment with SDN
Network simulator	NetSim [178]	L1/2/3/4	Simulation of heterogeneous networks and protocols (5G NR, IoT, WSN, Cognitive Radio, TCP)
Network simulator	The ONE [179]	L1/2/3/4	The Opportunistic Networking Environment (ONE) simulator, Evaluation of DTN routing and application protocols (sparse mobile ad-hoc networks)
Traffic generator	iPerf [180]	L3/4	Tool for active network performance measurement
Traffic generator	packETH [181]	L3/4	Packet generator tool for Ethernet
Traffic generator	pktgen [182]	L3/4	Testing tool included in the Linux kernel
Traffic generator	Moongen [183]	L3/4	Flexible high-speed packet generator
Traffic generator	Brute [184]	L3/4	Brawny and RobUstT Traffic Engine (Brute), Generation of traffic workloads having common traffic profiles
Traffic generator	Harpoon [185]	L3/4	Application-independent tool for generating representative packet traffic at the IP flow level
Traffic generator	Ostinato [186]	L3/4/7	Generation of specific traffic flows with various protocols
Traffic generator	TRex [187]	L3/4/7	TRex - Realistic Traffic Generator, Emulation of L3-7 traffic
Traffic generator	D-ITG [188, 189]	L3/4/7	Distributed Internet Traffic Generator, Synthetic network workload generator to emulate various applications (DNS, Telnet, VoIP and network games)
Traffic generator	Seagull [190]	L3/4/7	Multi-protocol traffic generator test tool

On the contrary, if a real network is available for experimentation, it is necessary to ensure that the traffic crossing the network has similarities with a real one. In this sense, the use of traffic generators become prominent to guarantee that the network exhibits a realistic behavior. There is a huge number of network simulators having a wide range of capabilities. Basic tools are already provided by Linux kernel-based OS distributions [180, 181] or provide a more user-friendly access to Linux kernel modules to generate traffic [182], but their capabilities are usually limited when needing to generate a specific packet distribution profile. More sophisticated solutions allow to select a specific traffic patterns generated at different OSI layers. The simplest ones are limited to model L3/4 packets [183, 184, 185], while others enable also L7 [186, 187, 188, 189, 190]. While L3 generation aims is to characterize IP flows and L4 generation is mostly limited to choose between employing UDP or TCP-based packets, at L7 there is a wide range of applications. Then, each traffic generator that works at such layer has to specify which applications can be simulated. Different solutions allow to simulate Web traffic, e.g., HTTP/HTTPS [186, 187] or VoIP [188, 189], and also 3rd Generation Partnership Project (3GPP) protocols [190].

2.2.7 **Performance-driven Network Functions**

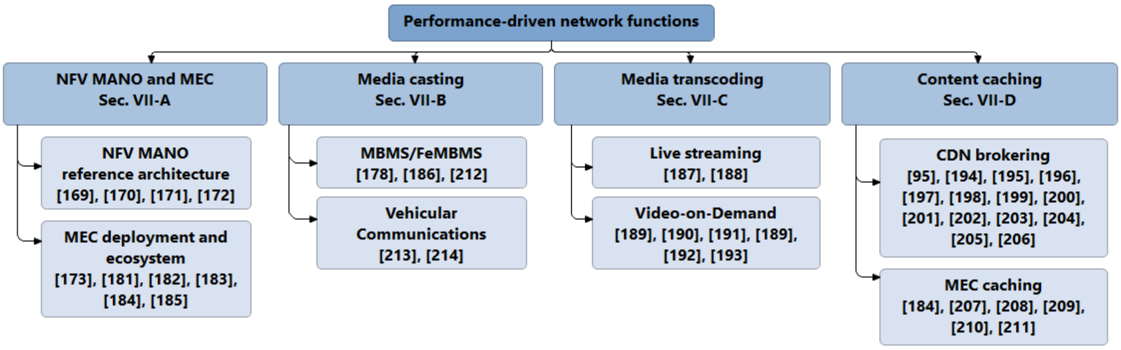


Figure 2.5: Performance-driven Network Functions.

This section presents an overview of VNF-based solutions designed to improve the performance of media streaming. These solutions employ knowledge that comes from network studies and data acquired from live monitoring of network traffic. Figure 2.5 illustrates the major avenues that performance-driven VNF solutions take. First we introduce NFV Management and Orchestration and Multi-access Edge Computing, as

VNFs rely on these paradigms introduced by European Telecommunications Standards Institute (ETSI) and embraced by 5G networks. Then, we discuss the state-of-the-art of most relevant media-related functions such as media casting, media transcoding and content caching.

2.2.7.1 **NFV Management and Orchestration and Multi-access Edge Computing**

Apart from the performance leaps on Key Performance Indicators (KPI) in terms of speed, capacity, mobility, and reliability, brought by 5G radio technologies, the network core is also fully engaged in a revolution, involving its own digital transformation. The concept that one network fits all is over. It is time to adapt the network according to applicable resources efficiency and delivery performance trade-offs. The goal is to allow network management systems to coordinate the systems comprising an agile, programmable and efficient network. This vision is being fueled by the transformation of network functions into dynamically controllable and configurable software components, which are virtualized exploiting cloud technologies and their scalable mechanisms, where orchestration of distributed network functions is done on top of the dynamic configuration of software systems. Going beyond, catalyzed by the network slices concept, the network would also connect groups of virtualized functions devoted to specific data flows or groups of users of specific services, handling independently Service Level Agreements (SLAs) of multiple points of presence (PoPs) over a common bare-metal infrastructure.

To achieve it, 5G network embraces NFV and VNF [191] concepts and comes with a NFV Management and Orchestration (MANO) architecture [192], standardized by ETSI.

NFV brings the primary virtualization step, providing computing, memory, storage and network resources from a bare-metal infrastructure (NFV Infrastructure or NFVI). The utilization of NFV contributes to the deployment of a network providing hardware and software decoupling. Thus, commercial off-the-shelf (COTS) hardware can be used to run every network function having a software implementation (VNF). This architecture is mainly employed by cloud vendors in order to provide Infrastructure as a Service (IaaS) solutions. Thus, hosting for systems on top of hardware and connectivity setup is performed on demand.

VNFs goes a step further in virtualization deploying specific network functions on top of NFVI. VNFs can be deployed, configured, started or stopped in a programmable manner. Thus, VNFs are intended to enable modularity, interoperability, scalability and flexibility when a media streaming service is managed, and the generated traffic is delivered.

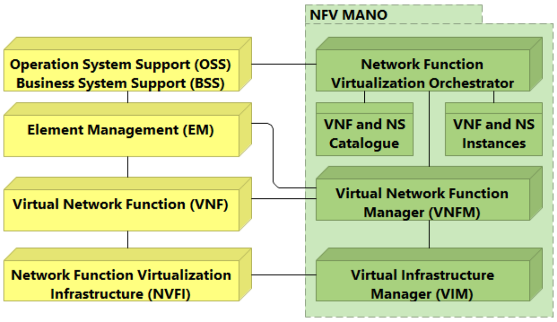

NFV MANO
Operation System Support (OSS)
Business System Support (BSS)
Element Management (EM)
Virtual Network Function (VNF)
Network Function Virtualization Infrastructure (NVFI)
Network Function Virtualization Orchestrator
VNF and NS Catalogue
VNF and NS Instances
Virtual Network Function Manager (VNFM)
Virtual Infrastructure Manager (VIM)

Figure 2.6: ETSI NFV MANO architecture.

NFVI and VNFs are managed and orchestrated by NFV MANO, whose reference architecture is shown in Figure 2.6. Its functional blocks are:

- Virtual Infrastructure Manager (VIM): It manages and controls physical and virtual resources (compute, storage and networking resources). Once a VNF is instantiated (VNF Instance or VNFI), it provides the VNFI with the resources it requires.

- VNF Manager (VNFM): It is responsible for the management of the life cycle of VNFI through the resources provided by the VIM.

- NFV Orchestrator (NFVO): It combines more than one VNF to create end-to-end services. Several VNFs could share VIM resources and be meant to be used for the deployment of a unique Network Service (NS), e.g., one VNF deploys the back-end and another one the front-end, the combination of the two VNFs constitute the NS.

Since 5G architecture allows for both public and private network deployment, existing NFV MANO-compliant solutions encompass both commercial and open-source alternatives for each of the three components. Some examples are Open Source MANO

(OSM) [193], whose development is promoted by ETSI, and Open Network Automation Platform (ONAP) [194], supported by Linux Foundation.

All the described technologies that turn network functions into virtualized software systems facilitate a high level of automation and orchestration by network management systems. This trend is being deeply explored and investigated in the current generation of mobile networks (5G) and it will be key pillar for next ones (beyond 5G) and Multi-access Edge Computing (MEC) infrastructures [14]. MEC architectures enable context-aware applications. It opens computing infrastructures co-located with the base stations to host services closed to the mobile users exploiting the capillary distribution of cloud computing infrastructures at the edge of the cellular Radio Access Network (RAN).

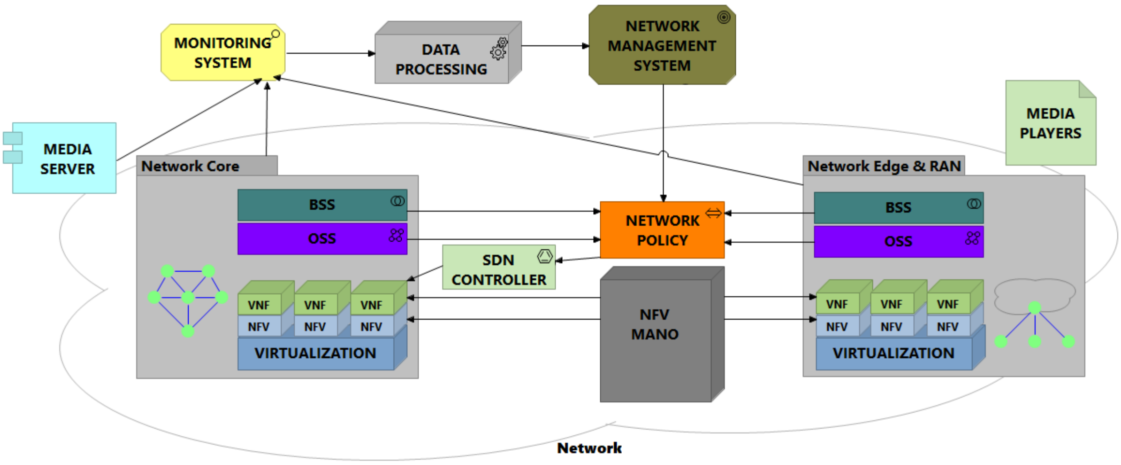


Figure 2.7: ETSI NFV architecture applied to Media streaming services.

The application of NFV and VNF technologies at the edge and the evolution of the RAN towards software components boosted by open-source software, such as OpenAirInterface [195] or srsLTE [196], eases the integration of MEC services with RAN systems. These solutions implement the Mobile Packet Core (Evolved Packet Core for LTE, 5G Core for 5G) and the RAN on top of open-source hardware enabling the deployment, management and orchestration through NFV MANO of both the mobile packet core [197, 198] and RAN [199]. A RAN deployment through NFV and VNF is usually referred as virtual RAN (vRAN). vRAN is also evolving towards the concept of Open RAN (O-RAN) [200], having open interfaces and network intelligence as key enablers to manage and tailor the network based on vendors and operators' requirements. O-RAN enables multi-vendor vRAN deployments, resulting in a more competitive and richer ecosystem [201]. In this context, MEC is a NFV MANO-compliant platform that comes also with a specific API to access Radio Network Information (RNI) [15].

Figure 2.7 shows how the virtualized (NFV) and softwarized (VNF) systems at the network core and edge are monitored and orchestrated according to business and technical policies which ask for changes in the NFV MANO system or SDN controller. Thus, any dynamic changes of the network can be applied over a widely-employed technology stack.

While focusing on the edge architecture, Figure 2.8 illustrates the MEC components and their interactions with the rest of the building blocks of RAN and Core Network (CN). The MEC host manages the User-plane, while the Data-plane communication is managed by the CN (LTE Evolved Packet Core or 5G Core). Depending on whether the deployment is within an LTE or 5G network, MEC host is equipped with User-plane Serving and Packet Gateways (SGW-U and PGW-U) or User Plane Function (UPF), respectively. These components are connected directly to the base station (eNB for LTE or gNB for 5G) and provide access to Internet. Inside the MEC Host, the RNI service (RNIS) oversees collecting RAN information which is later consumed by the application VNFs. Specifically VNFs can be designed to exploit such information to increase the overall system performance.

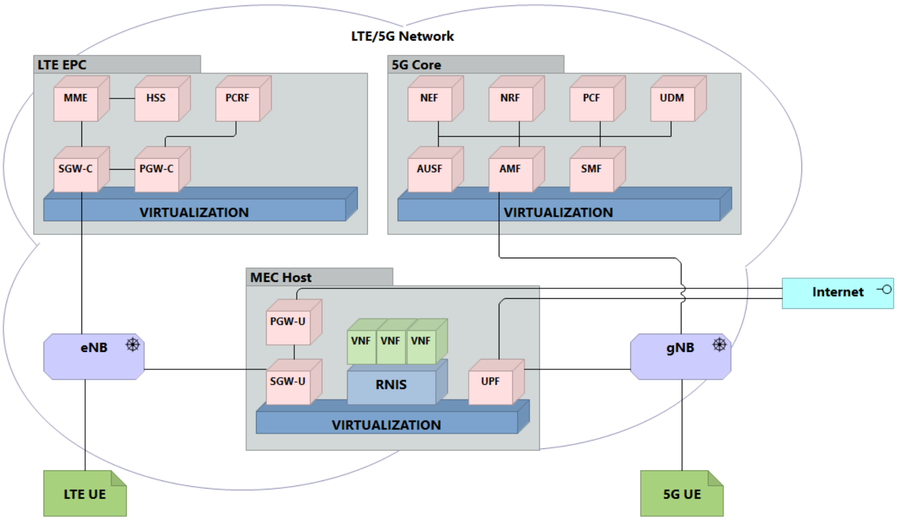


Figure 2.8: MEC architecture and connection with RAN and CN.

Beyond this design, VNF is also applicable for media-specific network functions beyond the 5G core and RAN, involving:

- media casting, in order to perform massive delivery of live data flows,
- media transcoding, such as streaming rate matches network available bandwidth, resulting in higher quality at destination and
- content caching, including storing popular data to help improved high traffic conditions, and managing alternative endpoints to balance the data requests.

All these network functions perform specialized functions of the media applications in order to improve network efficiency, saving bandwidth overheads and favoring the allocation of idle resources to other network flows, and to enhance quality of experience with enforced KPIs according to SLAs.

ETSI includes several use cases related to media streaming to be considered for MEC deployment [39] empowering traditional media streaming applications, which are based on interaction between remote server (origin server or CDN) and client, as shown in Figure 2.9. MEC platform can host diverse VNFs, which exploit RNI to get a wider view of the local conditions to enhance media streaming service. In this line, some solutions, such as [32, 202, 203], exploit standard RAN interfaces and data reports to conclude better decisions for media applications.

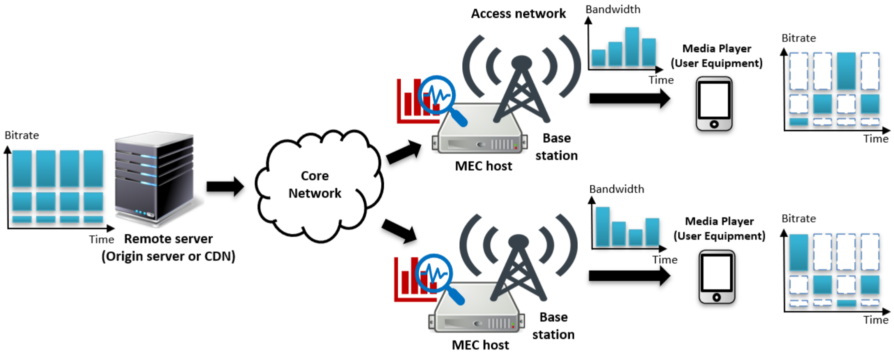


Figure 2.9: MEC-powered media streaming.

The following sections analyze how the described core technologies of 5G are applied to expand the network functions with core components for improved delivery of media streams, resulting with benefits in terms of enhanced quality and efficient resources utilization. Accordingly, Table 2.13 compiles and classifies all the research activities exploiting 5G to support performance-aware networking. The classification highlights the main features implemented, as well as secondary aspects, as sometimes the same approach is applicable to more than one solution. Some proposals are limited to architecture design and do not achieve a real implementation and experimentation. The implemented ones differ in terms of activation and processing approach, as they could operate in reactive or proactive manner and, in same cases, embed a processing algorithm (classic or ANN-based). All proposed solutions aim to have direct impact on the performance of the media streaming systems, ranging from QoS and QoE enhancement to more effective business costs and energy saving. However, most of them do not provide specific validation tests, especially in terms of HAS-centric QoE metrics, or insights on applicable cost models which include business aspects or evidence on energy footprint.

2.2.7.2 **Media casting**

For massive delivery of common data at once, synchronously, broadcast is still much more efficient that unicast communications widely employed by cellular networks. That is why 3GPP introduced Multimedia Broadcast/Multicast Service (MBMS) specification in Long-Term Evolution (LTE) release 9, which has been evolved towards further enhanced MBMS (FeMBMS) in release 14 to enable higher per cell bandwidth for MBMS services and simultaneous reception of both unicast and multicast services [204]. Furthermore, release 16 includes feedback for increased reliability [226].

In fact, as this technology is tied to the RAN system, it has sense in some use cases as firmware/software updates, clock synchronization, alarms and massive media contents to be turned in the network edge from unicast communications to broadcast signals. This would need the support from MEC systems which will turn popular streams into broadcast flows to expand the capacity of a cell. This is feasible as manifests of HAS technologies, such as HLS or DASH, even for encrypted contents keep the manifests unencrypted allowing a simple processing to parsed them by intermediaries, such as CDNs or MEC systems, for efficient and smart media delivery.

This architecture brings three major benefits by means of attracting all the ongoing live sessions to consume the broadcast dataflow, instead of establishing concurrent

Table 2.13: Performance driven networking for media streams using 5G technologies.

Main Feature	Secondary Feature	Activation	Processing approach	References	Network features	Description
Casting	-	Not Applicable	Not Applicable	[204]	FeMBMS	Design of 3GPP architecture for media multicast
Casting	-	Reactive	Not Applicable	[199]	FeMBMS, VNF, SDR	Virtualization of FeMBMS with SDR setup
Transcoding	-	Not Applicable	Not Applicable	[205]	NFV, VNF, 5G Core	Design of centralized virtual transcoder solution at 5G Core
Transcoding	-	Reactive	ANN	[206]	VNF, MEC	On-the-fly transcoder at the network edge
Transcoding	Caching	Proactive	Classic	[207]	L1 MC-NOMA, MEC	Solution empowered by mulitcarrier non-orthogonal multiple access
Transcoding	Caching	Reactive	Classic	[208]	MEC, VNF	Transcoding and cache location in virtualized edge infrastructures
Transcoding	Caching	Reactive / Proactive	Classic	[209]	MEC, VNF	Transcoding and cache location when content popularity is known (proactive) or not (reactive)
Transcoding	Caching	Proactive	Classic	[210, 211]	MEC, VNF	Transcoding and cache location based on known content popularity
CDN Brokering	-	Reactive	Not Applicable	[28, 29, 30]	L7	Proprietary solution for selection of CDN vendor at startup
CDN Brokering	-	Reactive	Classic	[212, 213, 214]	L3 DNS	Performance-driven solution based on DNS resolution
CDN Brokering	-	Not Applicable	Not Applicable	[215, 216, 217, 218]	L3 DNS	Design of CDN-ISP collaborative solutions
CDN Brokering	-	Proactive	ANN	[117]	L7, L3	Solution for proactive CDN selection employing ANN algorithm to forecast network metrics
CDN Brokering	-	Reactive	Not Applicable	[219, 220, 221]	L7	Cloud solution for cost-effective CDN switching
Caching	CDN Brokering	Reactive / Proactive	Classic	[222]	L7, L3, MEC	Statistical solution for CDN selection (reactive) and content caching (proactive)
Caching	-	Not Applicable	Not Applicable	[223]	VNF, Orchestration	Design of virtual CDNs for media distribution
Caching	-	Proactive	Classic	[31]	MEC, SDR	Solution at edge exploiting radio network information
Caching	Fair QoE	Reactive	Classic	[32]	MEC, SDR	Solution at edge exploiting radio network information
Caching	Fair QoE	Reactive	Classic	[224]	MEC, SDR	Solution at edge exploiting radio network information and content popularity
Caching	-	Proactive	ANN	[225]	MEC	Solution for proactive caching employing ANN technologies to predict popularity

unicast sessions:

1. Efficiency at the radio link, as the broadcast stream reduces radio link usage. Data traffic is independent of volume of users since everyone is consuming the same broadcast signal.

2. Optimal fidelity, as the network is able to deliver to all the audience the maximum resolution (bitrate representation).

3. Enhanced QoE, as the media players sharing the radio-link do not have to struggle with independent adaptive mechanisms executed in each player competing for the available bandwidth. This means no bitrate or resolution changes to track time-varying network conditions and no freezes to refill the buffer.

This approach is possible thanks to the application of virtualization and softwarization paradigms to RAN technologies, making vRAN and the containerization of some RAN network functions such as FeMBMS feasible [199].

Specifically, broadcast communications are gaining relevance in the vehicular communications field as they allow synchronous provisioning of common awareness to vehicles, pedestrians and Road-Side Units (RSU) in a surrounding area. Common awareness can be essential for Cooperative, Connected and Automated Mobility (CCAM) applications related to safety of autonomous driving [227]. In these applications media flows are important as the vehicles gets fitted with more camera-like sensors capturing the environment and exchanging the raw/compressed data or processed insights/summaries from on-board computer vision systems [228].

2.2.7.3 **Media transcoding**

Media services have become a fundamental service in 5G networks. There, as summarized in Table 2.3, HAS technologies, such as DASH or HLS, are widely employed and need the provision of several representations meaning different resolutions and bitrates [208]. Thus, VNF-based transcoders are being developed under international funding initiatives aiming to empower different use cases, e.g., live 3D media streaming [205] or automotive [229]. Here, the generation of representations at edge servers is gaining relevance to get higher efficiency by distributing the higher fidelity through the core and generating variants at the edge. This would reduce overheads in the core to send all the possible media variants. To this end, the media transcoding at the edge is essential [209], stressing the fronthaul capacity and requiring Cloud-RANs (C-RANs) or MEC systems in order to minimize the network delivery cost. Furthermore, the capillarity of the MEC systems brings a better adaptation to the local needs when transcoding to produce variants.

However, transcoding is a heavy process which needs a smart mechanism to gain efficiency. Transcoding at resource-constrained MEC server means a challenge for delay-sensitive services. Here, different works deal with the optimal position of transcoding systems in different edge hosts to respond to a distributed demand more efficiently and quickly, where players use a specific base station as a gateway linked to host and an edge server. To overcome this challenge, a mechanism for optimal request forwarding which respects the resources limitations and minimize serving latency is required [207]. In [210], different short/long-term decisions are concluded to deal with the time-varying conditions in terms of demand and network dynamics.

Beyond the planning of such transcoding process, other approaches consider different algorithms for reactive or proactive planning [209]. In this case, the dynamics have a big impact on the reaction time and forecast range. These aspects are minimized using a segment duration in the HAS stream with favor steady short-term conditions as changes comes in a segment duration-basis.

These works focus on enhancing QoS metrics while managing capacity of each processing asset. However, they do not consider heterogeneous SLAs and cost penalties to apply trade-off policies. As the GPU assets are required for HW-accelerated transcoding to ensure parallelization of transcoding threads and they have a big impact on infrastructure costs, this aspect should be a primary feature to evaluate.

It is important to underline that these solutions are often linked to caching strategies as both can be executed at the edge to better match the local conditions, patterns and demand features. Therefore, they design a joint strategy for transcoding processing and caching [207, 208, 209, 210, 211]. In [206], the authors only transcode the content on-the-fly if the content is not cached.

2.2.7.4 Content caching

CDN brokering: Caching is the most employed network function to improve the performance when accessing online contents and, in particular, media streaming ones. In this context, a CDN is the most popular network solution aiming to provide caching capabilities. It consists of a geographically distributed network of proxy servers and data centers to provide high availability of the contents. Caching mechanisms are key inside a CDN, as CDN proxy servers work by selectively storing the content such that the users can quickly access it from nearby locations. The employment of CDN service by the CPs increased in the last years, as the number of CDN vendors increased. Furthermore, major CPs also moved to multi-CDN strategies to provide a more reliable service while streaming their contents. Thus, an improved service also generates more satisfaction among the customers. Nevertheless, how the different CDNs are employed can differ from a CP to another. Static selection of the CDN when a streaming session starts is the easiest and widely employed solution among the CPs. In 2012, this strategy was used by Netflix [28] and Hulu [29], with big similarities [30]. In both cases, they were using three different CDN vendors. They used to map the player device to a CDN depending on to its location or the subscriber when the streaming session starts. Moreover, the CDN is never changed during the streaming session, even when the performances decrease. Other solutions include client-side CDN selection [212] or Domain Name System (DNS)-based solutions [213]. Client has a privileged position to measure end-to-end QoS metrics (network bandwidth and latency) when choosing the CDN, but it has the advantage to produce an uncoordinated decision as each client selects the CDN independently from the others. A DNS-based solution means resolving a fixed hostname owned by the CP into different IP addresses referring to several CDNs. Depending on the DNS resolution, the client receives the content from the appropriate CDN. In any case, a sub-optimal CDN server selection could lead to performance decreasing [214], affecting the user's satisfaction.

In the last years, other network caching solutions are also raising to empower the delivery. The same Netflix changed its streaming strategies. It developed and deployed an in-house CDN, called Open Connect [215], to reduce the dependency from CDN vendors and streaming costs. Moreover, Open Connect is meant to be run also inside the ISP infrastructure, i.e., closer to the user, to guarantee better performances in terms of network bandwidth and latency [230]. The use of Open Connect also helps Netflix and other CPs having in-house solutions to better control the resources enabled for the streaming session and to reduce the costs. Anyway, it requires a large investment to have such a solution and it could not be affordable by small CPs.

The Streaming Video Alliance (SVA) is a joint initiative which works on different aspects of media streaming and aim to standardize the employed protocols and technologies. Its membership includes some of the major world-wide agents in content production and streaming. Among its activities, the SVA Open Caching Working Group [216] oversees identifying the critical components of a non-proprietary caching system and establishing the basic guidelines for its implementation inside the ISP infrastructure. Thus, it wants to promote an architecture similar to Netflix' Open Connect, but with the advantage to be standardized.

Other collaborations between CDN and ISP are proposed in literature. In [217], ISP provides the CDN provider with information concerning geographical user distribution and allows the CDN provider the possibility to allocate server resources inside the ISP network. The authors of [218] use a redirection center instance inside the ISP network which intercepts the client requests and selects the appropriate CDN server. The process is transparent to the client as the redirection center employs a CDN surrogate to store the content and instructs an OpenFlow controller to migrate the traffic to the CDN surrogate. Beyond the employment of multi-CDN solutions, there are still possibilities of improvements. CDN Brokering [231] is proposed to make more effective CDN utilization in a multi-CDN environment. It redirects clients dynamically between two or more CDNs.

CDN brokers work as switching services that dynamically and seamlessly select the optimal CDN to use at any time. To achieve this, CDN brokers collect and analyze in real time the performance of the available CDNs to select the best one. Thus, network analytics have a prominent role in CDN selection, in contrast with traditional multi-CDN strategies where the same CDN is kept during the streaming session. The approach from [117] applies ANN technologies to forecast dynamic demand and changeable performance to make decisions including cost-performance trade-offs. In this context, a representative example is Eurovision Flow [219], proposed by the European Broadcasting Union (EBU). Similar solutions are also provided by Citrix [220] and Haivision [221].

Edge caching: In [222], a MEC proxy retrieves media streaming metrics of video players at the access point and CDNs performance metrics to enhance DASH media streaming. The MEC proxy evaluates the performance of different CDNs and switches players' sessions when a CDN is under-performing and cannot support the demanded traffic. Moreover, it features a local edge caching to reduce network traffic. Recurrent content is downloaded and cached once for every player. In [223], a similar MEC cache is proposed for empowering the delivery.

With a deeper integration with RAN interfaces, in [31] and [32] the MEC cache is improved by exploiting RNI. The media segments and representations are selectively cached depending on the network state. In [224], both RNI and knowledge of segment popularity are employed to decide the segments to cache. Moving from a reactive to a proactive approach, the authors of [225] empower the edge cache with neural collaborative filtering to predict content popularity. The predictions are exploited to proactively cache the content at the MEC, as more content popularity means higher probability to be requested by the users.

2.2.8 Challenges of Virtual Network Functions for Media Streaming

VNF solutions play a significant role in the successful deployment of 5G networks. This is backed by evidence, especially for supporting rich media applications such as multimedia streaming, as described in section 2.2.7. However, VNF applications still require some challenges and open issues to be addressed, as shown in Figure 2.10. This section discusses and classifies these challenges around some key features studied in relation to 5G networks and presents the open issues in the context of the 6G networks' roadmap.

2.2.8.1 Self-Organizing Networks

Agile deployment and life-cycle management of VNFs exploiting a NFV MANO architecture are essential features to satisfy the expectations of smart 5G networks, but further research is still ongoing to increase network automation. In this context, the Self-Organizing Network (SON) paradigm [232] represents a next step to achieve a fully virtualized and automated network, as it empowers the network with specialized decision-making algorithms which monitor network resources and traffic patterns, and

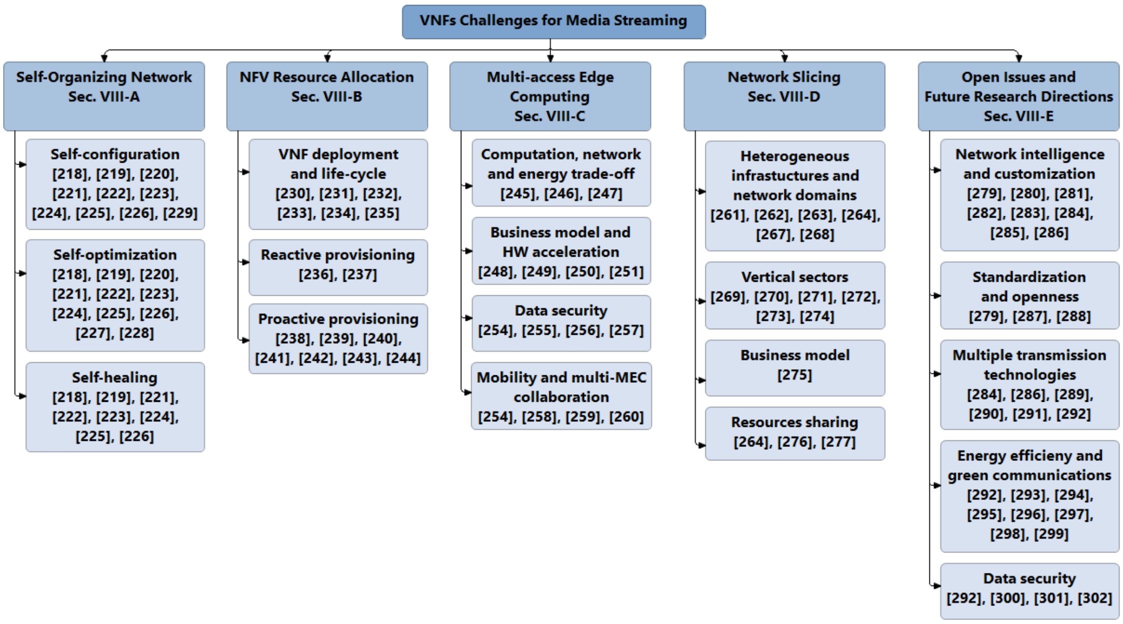

VNFs Challenges for Media Streaming
Self-Organizing Network Sec. VIII-A
Self-configuration [218], [219], [220], [221], [222], [223], [224], [225], [226], [229]
Self-optimization [218], [219], [220], [221], [222], [223], [224], [225], [226], [227], [228]
Self-healing [218], [219], [221], [222], [223], [224], [225], [226]
NFV Resource Allocation Sec. VIII-B
VNF deployment and life-cycle [230], [231], [232], [233], [234], [235]
Reactive provisioning [236], [237]
Proactive provisioning [238], [239], [240], [241], [242], [243], [244]
Multi-access Edge Computing Sec. VIII-C
Computation, network and energy trade-off [245], [246], [247]
Business model and HW acceleration [248], [249], [250], [251]
Data security [254], [255], [256], [257]
Mobility and multi-MEC collaboration [254], [258], [259], [260]
Network Slicing Sec. VIII-D
Heterogeneous infrastructures and network domains [261], [262], [263], [264], [267], [268]
Vertical sectors [269], [270], [271], [272], [273], [274]
Business model [275]
Resources sharing [264], [276], [277]
Open Issues and Future Research Directions Sec. VIII-E
Network intelligence and customization [279], [280], [281], [282], [283], [284], [285], [286]
Standardization and openness [279], [287], [288]
Multiple transmission technologies [284], [286], [289], [290], [291], [292]
Energy efficieny and green communications [292], [293], [294], [295], [296], [297], [298], [299]
Data security [292], [300], [301], [302]

Figure 2.10: Virtual Network Functions Challenges for Media Streaming.

autonomously take actions to enforce or optimize network operations [233]. SON capabilities were initially meant to be included as add on features of LTE, as 3GPP Release 8 started defining LTE and already set the basis for SON concepts and requirements [234]. However, SON is expected to enhance 5G network management providing automation to cope with increasing network complexity [235].

Specifically, in the media streaming context, SON should provide the required network resources and guarantee target QoS or QoE scores when delivering media streams. More generally, SON turns static networks into dynamic ones by configuring network parameters, optimizing the allocated resources and fixing or preventing issues in real time.

A SON-enabled system can accomplish tasks belonging to three categories: self-configuration, self-optimization and self-healing [233]. Self-configuration techniques adjust network operational parameters to change network behavior and rules, according to specific business policies and node neighborhood context. Self-optimization strategies are dynamically applied to ensure that the network performance is near optimal. They include real-time network monitoring and performance metrics processing to proactively apply enhancement operational parameters. Self-optimization techniques can be applied in many areas: load balancing, resource selection, caching infrastruc-

Table 2.14: SON categories and use cases.

Self-configuration	**Self-optimization**	**Self-healing**
• IP address & connectivity • neighbour & context discovery • radio access parameters • policy management	• load balancing • resource selection • caching infrastructure • coverage & capacity • radio interference management • mobility & handover	• fault detection • fault classification • countermeasures operations

ture, coverage and capacity, radio interference management, mobility and handover. Last, self-healing is necessary to generate a prompt reaction when faults, failures or any operational range violations in the network occur. The objective is to continuously monitor the system and ensure a fast and seamless recovery, whatever reason causes the failure. In case of a failure event, self-healing functions detect (fault detection) and diagnose (fault classification) it. Then, according to applicable policies and current setup, the appropriate countermeasure are applied to reestablish the desired network performance.

All these SON flavours need actionable data to process decision making algorithms. It is therefore very important to collect and exploit network data. Current networks are ready to probe and provide a huge amount of data. However, it is clear that specialized intelligence needs to be deployed within the network to infer valuable and useful information from the collected data [236]. Such information helps taking automatic actions to reach, recover or even improve the network performance. In the context of media streaming, it means that the SON paradigm has the potential to increase the QoS/QoE, while decreasing the business costs and energy consumption to maintain the network. In this context, the use of ML techniques will become prominent, even if the selection of the right algorithm is not trivial and depends on the considered use case [235, 237]. Table 2.14 shows the most common use cases belonging to the three SON categories, as seen from the network operator's perspective. Some SON applications are already provided by network vendors included in their commercial hardware equipment. Some examples are HCL's SON [238], Nokia's EdenNet [239] and Ericsson's SON Optimization Manager [240].

In any case, SON systems need to have a wider view of the delivered traffic beyond the metrics from the network functions and including service domain. It means that operated SON policies are usually steered by network statistics rather than application characteristics. Nevertheless, the communication dynamics of applications delivered on top of the network have an impact on network performance. Thus, the authors of [241] propose to design an application-driven SON in order to widen the view with both network performance and user's QoE metrics. When considering media streaming applications, data are available from network functions in the path and from playback devices. Thus, data exploitation inside a SON-enabled system needs further investigation, as the multi-domain data exploitation is still underexplored. Few solutions are available in the literature that apply a SON paradigm to media streaming scenarios. The authors of [242] propose a SON-enabled media transcoder to be deployed within the network. In [243] the authors introduce a self-organizing Unmanned Aerial Vehicle (UAV)-based communication framework for media streaming.

2.2.8.2 **NFV Resource Allocation**

The deployment of a VNF over a distributed platform requires the allocation of network and computing assets to be provisioned to host the VNF. Network and computing resource allocation is a challenging feature whose interest is raising and focusing on VNFs deployment and life-cycle management [244]. In this context, the NVFO is in charge of selecting the appropriate resources, among the available ones at the NVFI, when deploying a VNF, which is usually referred to as the NFV resource allocation (NFV-RA) problem. The NFV-RA includes three stages [244]: VNF Chain Composition (VNF-CC), VNF Forwarding Graph embedding (VNF-FGE) and VNF Scheduling (VNF-SCH).

VNF-CC deals with the composition of several VNFs to be deployed jointly by the NFVO. How the traffic flows between VNFs is also described trough the definition of VNF Forwarding Graphs (VNF-FG). Thus, any network service can be considered as composed of a set of VNFs and VNF-FGs. Each VNF executes a small function of the entire application or service [245]. VNF-FGE focuses on how to embed the VNFs and VNF-FGs into the infrastructure. It aims to find suitable resources and locations where to allocate the VNFs in NFVI. At this stage, resource selection and optimization must be accomplished with regard to the specific constraints defined by SLA [246]. Finally, VNF-SCH determines how to schedule the processing operations of the deployed VNFs [247].

When the VNFs are already deployed and running, the required resources vary during their life-cycle, as they depend on user demand of the running function provided by the VNFs. Allocated resources could be optimized to fit with the variable demand by the user. Increasing or decreasing the allocated resources means the VNFs also need to dynamically scale up/down. Then, an efficient orchestration and automation of the VNFs requires supporting this dynamic allocation of resources. This assumption was already envisioned when designing the NFV MANO architecture [248], where mechanisms to scale are essential to enable a flexible management of the running services.

However, the easiest and fastest approach consists of employing an over-provisioning strategy, where the amount of allocated resources for each VNF is larger than what is required. In case of experiencing an increasing demand, the VNF can manage overheads without any intervention as long as the allocated resources are not exceeded. This approach is operationally effective, but inefficient in terms of OPEX and energy consumption generated by the allocated resources which are not actually employed. This means that this approach is not cost-effective, as it is clear that adjusting the resources allocated for the VNF to the actual demand would avoid over-provisioning and reduce costs. Employment of dynamic provisioning strategies results in OPEX reductions for network operators and/or service providers [20].

Enabling dynamic resource allocation for VNFs allows scaling up and down and therefore coping with network traffic fluctuations and changeable demands from connected users. Dynamism, scalability and automation are important features for resource management [249]. Changes in resource allocation should be applied according to real-time network traffic and service demands. Dynamic resource allocation can be performed in reactive or proactive manners. Simple solutions involve a reactive provisioning approach which means changing the allocated resources to react when traffic and/or demand change. In [250] the authors design an online algorithm for VNF scaling in cloud data centers. The authors of [251] aim to minimize the OPEX by considering the trade-off between bandwidth and host resource consumption under diverse workload variations. All these reactive solutions have the advantage of a simple design, as there is no need for any complex algorithms for provisioning. On the other side, such an approach does not prevent any network issues or service faults from happening, affecting the services.

A more sophisticated approach consists of proactive provisioning where the future traffic and/or demand is predicted. Being able to foresee the amount of resources to be allocated constitutes a great benefit, as it enables to avoid network issues or service faults by proactively resizing the employed resources and scaling the deployed VNFs. In such a context, the problem of service demand prediction constitutes a mayor challenge. Most of the literature on demand prediction employs ANN algorithms [252, 253]. However, the application of such algorithms in practical solutions is limited, being mostly theoretical.

Among the most innovative solutions proposed, [254] describes a novel FTRL online algorithm for VNF provisioning which handles workload fluctuations. The solution in [255] employs an ANN algorithm to predict future resource requirements for each VNF contributing to a network service. The authors of [256] propose the POLAR algorithm, which combines online learning and online optimization of proactive provision resources with VNFs provisioning, while the VNFs chaining in a network service is ignored. In [257] a proactive failure recovery is proposed when considering VNF deployed at distributed edge computing nodes. In [17] a proactive VNF chaining aims to find the optimal number of VNFs and their location inside a CDN in order to minimize costs. Finally, the authors of [258] propose a multi-layer resource allocation solution, which aims to proactively provide resources to the VNFs deployed in several VIMs and network resources between VIMs.

2.2.8.3 **Multi-access Edge Computing (MEC)**

MEC represents a novel technological solution integrated in 5G networks to bring computation closer to the user. MEC infrastructures create new potential revenue flows to network operators opening their edge infrastructures to host specialized services at network edge. There are many aspects which require investigation to achieve a complete integration of MEC into the current network architecture and services. However some avenues are already seen as highly beneficial for MEC deployment and use. For instance, media streaming is a key application of MEC solutions, as ETSI considers it as one of MEC core use cases [39]. MEC platforms can host edge services to empower media streaming applications, which traditionally were based on server-client communications. As explained in the previous section, MEC and VNFs enable the deployment of innovative media-related services such as media casting, media transcoding and content caching.

More specifically, MEC resources are exploited by both the server and clients to offload computation tasks [259, 260]. Offloading server tasks targets reducing network traffic and latency, as the processing is performed close to UEs. MEC resources are shared between different service providers, but how the resources are distributed among different service providers is still undefined. The authors of [259] propose to allocate MEC resources proportionally to the demanded resources and payment of each service provider. If an UE offloads tasks to the MEC host, it reduces not only the device computation load, but also its power consumption, as computing-intensive tasks heavily impact on the battery duration. In [260], a video telephony application employs MEC to encode the content. It reduces processing operations at the UE, but increases network traffic since uncompressed raw content is sent to the base station. The authors focus on power consumption, but they do not consider operational costs generated by using the MEC platform. In general, how to balance network traffic, power consumption and operational costs trade-off needs to be studied. In [261], optimization of the allocation of both computing and network resources is discussed, while taking into account the energy efficiency. Even in this case, operational costs are not considered in the optimization problem. In general, business aspects raise complex discussions due to the lack of a clear business model [262]. MEC needs a business model equivalent to the one applicable in cloud computing infrastructures. However, unlike cloud computing, the decentralized location and utilization of shared resources between services makes the cost model more complex. Resource accounting and monitoring have to also be determined in order to create a complete business model. The debate on the business model is even more intricate if we consider hardware-acceleration assets, such as GPUs, required to accomplish critical tasks where general-purpose hardware (CPU) has limitations [263]. Some works suggest to employ Field-Programmable Gate Array (FPGA) approaches instead of GPU solutions due to their reduced price and power consumption [264, 265], but this possibility is again underexplored.

Regarding to accessible information at MEC, the API to communicate with RNIS [15] has been recently standardized and its development is on going [266, 267]. It means that services running at the MEC host cannot be further optimized. When RNIS implementations will be available, edge services could embed more complex and precise algorithms (classic or ML models), aiming to exploit RNI in order to improve their operations and the performance of the overall system. However, improved capabilities due to RNI exploitation raise some security concerns on how to manage information at MEC hosts, an aspect that needs further investigation [268, 269]. In order to exploit a MEC decentralized approach, the deployment of location-aware services is necessary. Thus, mechanisms for user privacy protection and anonymity are needed. Moreover, modification of the networks to introduce MEC capabilities opens the door for potential attacks, including DDoS attacks, malware injection, authentication and authorization attacks [270, 271].

Mobility remains another major concern and is becoming critical, as the explosion in availability and type of mobile devices (e.g., smartphone and tablets) involves an increasing number of UEs to be served. The same way the connectivity is guaranteed when moving from a cell to another in a cellular network, migration support for MEC services is also required. Consequently, the investigation on a multi-MEC cooperation should be addressed in order to guarantee seamless migration of sessions across MEC servers [268, 272].

From the perspective of media services, user QoE plays an important role and a wide MEC deployment definitely should target it, especially as transcoding and caching capabilities would be provided closer to UEs. How to balance the cost of MEC-based caching and transcoding and provision of high user QoE is an important direction for future research [268]. Moreover, it becomes relevant the ability to find suitable locations where MEC instances should be deployed, as it may affect the fulfilment of the demanded requirements. It is especially true for low latency multimedia services, where the distance between the MEC host and UE affects the overall delay [273]. Finally, content caching mechanisms in the network have been studied both at the core and at the edge, but a convergent solution has not identified yet. Caching solutions that integrate both core and edge caching could result in better network performance in terms of the energy consumption, network throughput, latency, and user QoE [274].

2.2.8.4 **Network Slicing**

While NFV-RA is limited to provide NFVI resources deployed for a specific section of the network (CN or RAN/MEC), network slice has a wider scope, as it is able to provide network and computing resources even across different networks. Network slicing [275, 276] is introduced in 5G networks as a solution involving several virtual/logical networks (slices) on top of a common physical network, where each virtual/logical network delivers the traffic generated by a specific service [277, 278]. It can be considered that a network slice is associated with a set of network resources and VNFs, which can be provided by that slice. In this context, NFV MANO and SDN play an important role, especially in the deployment and management of network slices [279, 280]. NFV MANO enables life cycle management and orchestration of the VNFs, while SDN allows for the configuration and control of the routing and forwarding planes of the underlying network infrastructure, providing communication between the deployed VNFs. This results in a logical network of resources and VNFs built over a common underlying physical infrastructure, separated into diverse network slices. Each network slice provides the service as an end-to-end connectivity, meaning that network slicing provisioning refers to three different aspects: at the air interface, in the RAN and in the CN [281, 282].

Network slicing at the air interface refers to partitioning physical radio resources (physical layer or L1) into subsets of several physical resources, each one for a different network slice, then mapping into logical resources to be provided to the Medium Access Control (MAC) sublayer at the datalink layer (or L2) and higher layers.

In the RAN, network slicing changes RAN operations, including MEC-operated ones, such as device association and access control, from a cell-specific perspective to a slice-specific one. Thus, the RAN operations are service-oriented instead of physical cell-oriented. Configuration of control and user planes is tailored and/or tuned considering the requirements of each slice individually. Then, factors such as QoS requirements, traffic load or type of service/traffic are prominent when operating the RAN.

Finally, network slicing in the CN enables the definition of vertical networks, where each one aims to support a service belonging to a specific vertical industry. NFV MANO and SDN have a higher impact in this aspect of the network, where each vertical industry should be able to run its VNF-specific solutions. CN needs flexible management to enable resource scalability and migration when required by the network traffic associated with a service.

A videoconferencing system is deployed in [283] through the deployment of two different slices to split audio and video transmissions, as they have different requirements in terms of network throughput. In [284], the authors focus on the eHealth vertical, where services are typically media-rich and mission-critical and are high QoS demanding. Then, a MEC-based application, empowered with end-to-end network slicing, is designed and developed to enable in-ambulance applications. The application is accessed by paramedics in the ambulance and sends audiovisual data to the hospital/-doctor. The same vertical is addressed by [285] to enable a real-time communication between hospital staff and patients. In [286] and [287], applications of network slicing for Vehicle-to-Everything (V2X) services are investigated. Different use cases are considered in a vehicle, including related to safety and traffic efficiency, autonomous or tele-operated driving, media & entertainment and remote diagnostics. Each use case means different requirements in terms of latency, throughput and communication reliability. Consequently, different network slices with different configurations are required on top of the same physical network of resources. The authors of [288] present several use cases belonging to different verticals, such as protection and smart metering in the smart grid sector, car and passenger data exchange in an intelligent transportation system and best-effort data delivery in a multimedia system. Each use case and vertical sector requires different capabilities in terms of latency and throughput. The different types of traffic are prioritized by splitting them into specialized network slices.

Network slicing-related research has increased importance in the current 5G network context. Ongoing challenges include solutions to allow wide employment and operation of slices for different industry verticals. Most of slicing operations relate to the exploitation of resources provided by the network operator, but the effects of changes in network operator's business models for operating network slicing are unknown [289]. The increase in the number of devices belonging to different verticals and their mobility management in the presence of different technologies (LTE, 5G, Wi-Fi) also need further investigation [256]. An end-to-end network slice implies that slice segments potentially stretch across different administrative domains. There are two requirements in order to achieve a unified control of the network slice. First, an exchange point that performs the resource negotiation between different administrative domains is necessary to enable multi-domain slices. Then, standardized APIs should make transparent the underlying domains and simplify the negotiations to provide the control on the slice [276]. Finally, network slicing leverages algorithms to accommodate applications with widely diverse requirements over the same physical network. Thus, complex algorithms are necessary for deciding how to efficiently allocate, manage, and control the physical resources to be shared across diverse slices [290]. Concerning these algorithms, the application of ML in network systems is capturing increased research attention lately and this trend is expected to continue in the future [291].

2.2.8.5 **Open Issues and Future Research Directions**

The benefits of virtualization for media streaming communications will increasingly evident in the next few years, as the 5G coverage will be extended. Complementary technologies such as MEC, SON and network slicing are still not fully integrated. Further efforts in integrating all these new paradigms and/or architectures are envisioned to provide a more efficient and intelligent network [292].

ML-powered network intelligence to manage NFV and VNFs is only partially achieved in 5G networks, but it will be also a key factor for the future 6G networks [293]. The concept of Intent-Based Networks (IBN) [294] means employing ML solutions to transform business intents into network configuration, operation, and maintenance strategies. In order to meet the massive service demands and overcome limitations due to time-varying network traffic, the network can continuously learn and adapt to the time-varying network environment based on the massive collected network data in real-time. An intelligent-native network exploits ML algorithms to improve its capabilities and reduce the business costs for service deployment and management [295, 296]. The advantages of an intelligent-native network are two-fold. First, the network can analyze user's behavior in real-time and autonomously learn its needs to predict its future behavior. Then, user's information can be employed for network customization to achieve a user-centric network [297]. Second, the network can met changing requirements of a network service during its life-cycle by autonomously matching the requirements to the corresponding network communication, computing and caching assets. This is also valid for new emerging services. Holographic (AR and VR) and haptic communications are meant to be wider available thanks to the future 6G network

[298]. Moreover, the global COVID-19 pandemic is accelerating the digital transformation of multiple and heterogeneous verticals, such as development of new services for smart cities and innovation in the eHealth including telemedicine, medical and thermal imaging, and robotics for medicine practice [299, 300].

Openness is also an important aspect to achieve flexible network and services [293]. Having open network platform and interfaces (O-RAN, NFV MANO, SDN, etc.) allows interconnection and interoperability of different vendors, which is essential for sharing a physical infrastructure. Thus, agents of diverse vertical industries may deploy their private physical infrastructure and manage it though NFV MANO solutions and SDN controllers independent from public networks operated by mobile network operators [301]. Standardization process will continue in the next years to fulfil the remaining gaps and guarantee interoperability of heterogeneous implementations of open network solutions [302].

The cooperation of different physical networks will also attract attention. Multiple Radio Access Technology (multi-RAT) aims to employ different access network to improve the overall connectivity [303]. Its application to improve media streaming is already being investigated [304, 305], but new transmission solutions based on space, UAV-based and underwater communications will be integrated with terrestrial ones [298, 300]. Flexibility to operate the network at any level (spectrum/band, physical and MAC, etc.), despite the different involved technologies, will be imperative [306].

Energy efficiency and green communications [307] are envisioned to enable more sustainable networking [308]. Energy efficiency concerns are also relevant for media streaming services [309, 310]. Here, low-power wireless devices could harvest energy from the available high-power radio waves [306]. Thus, battery-free implementations will be an interesting topic to be further explored in different use cases, e.g., IoT [311] and media streaming communications [312, 313].

Finally, the growth of network and media traffic will have consequences for security. Critical media use cases, e.g., eHealth applications [314] and autonomous driving systems [315], need to be secured with security mechanisms which will complement the conventional cryptography-based ones. Increasing security will be assured with the design of cross-layer algorithms to protect the transferred information [306, 316].

2.2.9 International Initiatives

Table 2.15: Major SDN/NFV related research activities.

Project	Time period	Area of concern	References
CogNet (Building an Intelligent System of Insights and Action for 5G Network Management)	2015-2018	Architecture	[317, 318, 319]
SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks)	2015-2018	Architecture	[320, 321]
SliceNet (End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks)	2017-2020	Architecture	[284, 322, 323]
SoftFIRE (Software Defined Networks and Network Function Virtualization Testbed within FIRE+)	2016-2018	TestBeds	[324, 325]
FLAME (Facility for Large-scale Adaptive Media Experimentation)	2017-2020	TestBeds	[326, 327]
5GTango (5G Development and validation platform for global industry-specific network services and Apps)	2017-2020	TestBeds	[328, 329, 330]
5G-Media (Programmable edge-to-cloud virtualization fabric for the 5G Media industry)	2017-2020	Application Verticals	[82, 331, 332]
5Growth (5G-enabled Growth in Vertical Industries)	2019-2021	Application Verticals	[333, 334]
5GCity (A Distributed Cloud and Radio Platform for 5G Neutral Hosts)	2017-2020	Application Verticals	[335, 336]
OpenAirInterface Software Alliance	2014-	Development Platforms	[195, 337]
Mosaic5G	2016 -	Development Platforms	[338, 339]
O-RAN Alliance	2018-	Development Platforms	[200, 337]

Employing VNFs for media streaming is a research topic that has attracted the attention of international organizations and international funding programs for many years now. Recently, the European Commission has funded numerous research projects aiming at developing and implementing VNFs for different research scenarios and vertical industries. Table 2.15 summarizes the most relevant actions. The project list includes initiatives targeting generic architectural design (i.e., CogNET [317], SELFNET [320] and SliceNet [320]), activities building testbed environments and pilot environments for use case definition and testing (FLAME [326], SoftFIRE [324] and 5GTango) [328], projects targeting specific application verticals and developing required functionalities (5G-Media [331], 5Growth [333], 5GCity [335]) and finally international software communities to provide open-source platforms (OpenAirInterface Software Alliance [337], Mosaic5G [338] and O-RAN Alliance [340]).

Regarding architectural definition, SELFNET H2020 project designed and tested an autonomous network management framework capable of the automatic detection and mitigation of common failures in the network [321]. Among others, it proposed the smart integration of state-of-the-art technologies in NFV. One of the outcomes is presented in [341], where the SELFNET framework preserves the health of the network maximizing the QoE and minimizing the end-to-end energy consumption. SliceNet project addressed both management and control planes of network slicing to leverage QoS for sliced services [342]. The project proposed an integrated network management, control and orchestration framework and applied the concept to a variety of use cases. One of those cases, related to multimedia health services is described in [284], where demanding QoS requirements (i.e., latency) need to be fulfilled. The network intelligence topic is tackled by CogNET, a project that focused on realizing the well-known control loop MAPE (Monitor, Analyze, Plan and Execute) with Machine Learning techniques and policy-based mechanisms for a vision of softwarized 5G networks. COGNET validated its vision in different use cases that include SLA Enforcement and Mobile Quality Predictors [319], [318].

A second group of projects aimed at creating platforms and testbed environments where specific use cases, applications, algorithms, and interoperability solutions could be designed and validated. FLAME stands out in this area as a facility for experimenting large scale experiments in the field of Adaptive Media. Since 2017, FLAME has hosted different proposals [327] to offload proactively video content to the edge of the network on an SDN/NFV environment. FLAME tests include augmented reality applications as well as smart video surveillance for aiding impaired citizens. SoftFIRE is another testbed environment to experiment VNF services and applications in SDN/NFV. SoftFIRE aims at assessing the level of maturity of solutions in programmability, interoperability and security and showing how they can support the full potential of these properties in a real-world case [325]. Finally, 5GTango puts the focus on network flexible programmability [329] by providing software development kits (SDKs) [330]. This project included qualification and verification mechanisms as well as a modular service platform to bridge the gap between business needs and network operational management systems. 5GTango was demonstrated in two vertical through specific pilots: advanced manufacturing and immersive media [329].

The third category encompasses some examples of projects designing the required building blocks that enable the applications for specific vertical sectors. 5GCity was an H2020 project aiming at designing, implementing and demonstrating a distributed cloud and radio platform for municipalities and infrastructures with neutral hosting capabilities. One of the main outcomes of the project was the 5GCity Orchestration Platform, which supported the NFV MANO model. In [336], the authors demonstrate that the virtualized platform was able to address different use cases related to media streaming such as real-time video acquisition and production at the edge, UHD Video Distribution and immersive services or mobile real-time transmission. 5G-MEDIA [331] exploits the principles of NFV and SDN to facilitate the development, deployment, and operation of VNF-based media services on 5G networks. Key in this project is the development of a platform for service virtualization that provides an advanced cognitive management environment for the provisioning of network services and media applications [82]. The use cases include tele-immersive gaming, mobile journalism and UHD content distribution [332]. 5Growth [333] supports diverse industry verticals developing the tools for interfacing those verticals with the 5G end-to-end platforms. The system provides the creation of network slices with closed-loop automation and SLA life-cycle service control. ML-driven solutions are also part of the project targets to optimize access, transport, core and cloud, edge and fog resources, across multiple technologies and domains [334].

Finally, OpenAirInterface Software Alliance [337], Mosaic5G [338], and O-RAN Alliance [340] are mixed academic and industrial communities to create ecosystems of open-source projects for studying, building, and sustaining open flexible and integrated 5G network. OpenAirInterface Software Alliance [337] provides 5G network tools extensively used by researchers from both industry and academia. This initiative gathers developers from around the world, who work together to build wireless cellular RAN and CN technologies [195]. Mosaic5G [338] develops a set of 5G software solutions and has already hosted experiments targeting low latency MEC services, orchestration solutions and programmable RANs [339]. O-RAN Alliance [340] is pushing the standardization and the development of the O-RAN. RAN industry is moving towards open, intelligent, virtualized and fully interoperable RAN [200].

2.2.10 Conclusions

The popularity of media streaming services is constantly growing due to increasing number of users and diversity of rich media applications, e.g., online gaming, VR/AR applications, etc. The latest smart mobile devices also have an important role in the success of media streaming, as their processing and rendering capabilities support streaming content at very high resolutions, e.g., Ultra-High-Definition (UHD) or 4K. Consequently, media streaming traffic accounts not only a very large share of the total Internet traffic, but, more importantly, also an increasing one.

To cope with this increasing media traffic and high dynamics of network performance and user mobility, improved network capabilities are required to maintain high QoS and QoE performance, while also achieving the best trade-off with business costs and energy efficiency. 5G networking is bringing new possibilities to deploy smart network functions, which monitor both the media streaming service through live and objective metrics and boost it in real time. Under the 5G umbrella, NFV and SDN will have a prominent role in the virtualization of network functions and their management and orchestration.

In this context, this work provided a state-of-the-art on VNFs applied to media streaming. To this end, we considered the factors that concur to the design and implementation of a stable VNF. Monitoring and collecting performance metrics enable their exploitation as source of information for the VNF life-cycle deployment and management, as well as to evaluate the effects of the capabilities provided by the VNF on the media streaming session. Moreover, network traffic monitoring and analysis allow to create models to approximate the behavior of the network and predict future network events to take actions in a proactive manner. Thus, any network malfunction or issue that affects the media steaming session can be prevented.

Several VNF solutions to improve media streaming are presented. Solutions including media casting, media transcoding and content caching can be employed at any segment of the network. Thanks to the NFV MANO architecture, the deployment of VNFs is not limited to the Network Core, but they can be also run at MEC infrastructures. Capillarity of the MEC allows computing operations close to the base stations and reduces the latency when dealing with live streaming services.

Finally, research challenges and open issues have been presented in the realm of VNFs applied to media streaming services. The achievement of dynamic resource allocation, complete MEC integration and network slicing are the main venues where the research will focus in the next few years. Long-term research directions will also address a strong employment of ML to foster network capabilities and the utilization of open network solutions and/or new access technologies, also combining them to increase the capacity. Green communications and security will also be major concerns, as the future networks should reduce their impact on the environment and guarantee the security of the processed information. In conclusion, VNFs represent an important enabler to improve the media streaming services, but despite the research done under international initiatives that are pushing 5G and network virtualization, several research challenges still exist and provide opportunities for further research activities.

Acknowledgment

This work was fully supported by Open-VERSO project (Red Cervera programme, Spanish government's Centre for the Development of Industrial Technology) and by the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 870610 for the TRACTION project. The support of the Science Foundation Ireland (SFI) Research Centres Programme Grant Numbers 12/RC/2289_P2 (Insight) and 16/SP/3804 (ENABLE) is also gratefully acknowledged.

Part III

Research Results

CHAPTER

3

Network-aware content encoding

3.1 Context

The original uncompressed video content undergoes two main operations before being delivered to the user: compression through one of the widely known video codecs, e.g., H.264 or HEVC, and packaging in a media container format, e.g., MPEG-4 Part 14 (commonly called MP4). Encoding and packaging the content influence user's QoE, which plays a significant role when dealing with media services. Thus, optimizing video encoding and packaging strategy contributes to increase the user's satisfaction and to retain the user from leaving the media service. Considering network information and application context (VOD or real-time communications) may lead to a better selection of video encoding bitrate and streaming format/protocol. In this sense, this thesis investigated the possibility of considering such information by designing and implementing two different solutions that exploit it.

MPEG-DASH natively allows encoding bitrate selection at the client side which enables to mitigate network performance fluctuations. This format also fits with the Video on Demand (VOD) scenario where the latency between content packaging and playback is not an issue. On the contrary, it is not suitable when latency constraints come into play. Live streaming applications, such as video surveillance and video conference, cannot work with typical operational ranges meaning tens of seconds of delay of MPEG-DASH.

In Section 3.2, an Adaptive Rate Control on top of SRT protocol is developed to demonstrate the applicability of network information at the origin server. Differently from MPEG-DASH, SRT protocol is meant for guaranteeing low latency required for Live streaming, but it does not provide the capability to adapt the encoding bitrate. The implemented Adaptive Rate Control-enabled SRT server periodically changes the video resolution and encoding bitrate to adapt live streams accordingly to the information concerning the network throughput and reported by the connected clients. When network throughput decreases, the resolution and encoding bitrate are decreased to prioritize the playback smoothness over video quality. On the contrary, if the throughput increase, encoding bitrate and resolution are also increased. This solution does not need any additional communication, as SRT protocol already provide feedback mechanisms for reporting network status. Then, this paper proposes a real implementation of an Adaptive Rate Control for SRT streams by including the following relevant contributions:

- A server-side Adaptive Rate Control implementation on top of open-source framework for SRT streaming applications. This Adaptive Rate Control exploits the network reports employed by SRT protocol to enable the adaptation of the resolution and encoding bitrate of the content.

- A coordinated delivery of the stream as the encoding bitrate is chosen by the origin server at once for all the connected media players. It differs from MPEG-DASH, where each client autonomously choses the representation bitrate.

- Evaluation of the effects on user's Quality of Experience (QoE) when compared the proposed solution to a legacy one. In both cases, the player does not need any modification as a legacy SRT client can decode the Adaptive Rate Control-enabled stream.

Compared to a legacy SRT solution, the results show that the Adaptive Rate Control-enabled SRT delivery experiences fewer freeze events by enabling switching operations to lower representation bitrates. It reduces the average representation bitrate to prioritize playback smoothness. Moreover, in terms of initial delay, there is not a noticeable difference with legacy SRT since the Adaptive Rate Control does not introduce any delay while starting the streaming session.

Section 3.3 presents a study of LL CMAF to deliver Live Streaming which is carried out to evaluate the trade-off between latency and QoE. CMAF is a technological solution which has two major benefits. First, it pushes MPEG-4 Part 14, usually referred as MP4, as a common file format for different streaming technologies, such as MPEG-DASH or HLS. This feature makes media storage more efficient as different manifests (MPD for MPEG-DASH and M3U8 for HLS) may index the same media segments. Therefore, even if the players download different manifests depending on their supported streaming technologies, they download and play the same media segments. Thus, the remote server (origin server or CDN) needs lower storage capacity. Secondly, it defines a low latency mode, also called chunked mode, named LL CMAF or Chunked CMAF, which enables latency enhancement of the stream, reducing the time elapsed between media packaging and its playback. A typical MPEG-DASH segment contains a single MP4 fragment. On the contrary, LL CMAF enables a single segment to contain multiple fragments. A MP4 fragment is the minimum amount of data required by the player to start decoding the stream. Therefore, the shorter fragment duration allows a promptly playback start, removing the limitation to fully download the entire segment, which usually lasts some seconds. This paper includes:

- A server-client solution delivering LL CMAF streams on top of open-source framework.
- A comparison with a legacy MPEG-DASH stream having segments of 2 seconds duration to underline the limitations of the setup widely employed for live/low latency video streaming.
- The evaluation of the effects on user's QoE while varying the fragment duration and the resulting latency. The employed fragment durations are 33 ms, 100 ms and 167 ms that correspond to fragments containing a Group of pictures (GOP) with 1, 3 or 5 frames for a video with a nominal framerate of 30 frames per second, respectively.

The results show that media players gain lower latency in any of the LL CMAF configurations with respects to legacy MPEG-DASH setup. However, when using an aggressive configuration with a small GOP size and fragment duration, the playback has lower protection against freezes which reduce the QoE. To balance the latency and QoE trade-off, a more conservative configuration of LL CMAF is suggested.

3.2 Adaptive Rate Control for Live streaming using SRT protocol

- **Title:** Adaptive Rate Control for Live streaming using SRT protocol
- **Authors:** Roberto Viola, Ángel Martín, Juan Felipe Mogollón, Alvaro Gabilondo, Javier Morgade and Mikel Zorrilla
- **Proceedings:** 2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)
- **Publisher:** IEEE
- **Year:** 2020
- **DOI:** 10.1109/BMSB49480.2020.9379708

Abstract: Media delivery represents one of the main challenges for future networks which aim to converge Broadcast and Broadband video traffic into a common telecommunication network architecture. Nowadays, contents streamed over Internet are delivered in two different manners depending on the application: Video on Demand and Live Streaming. For the former, HTTP-based streaming technologies, such as Dynamic Adaptive Streaming over HTTP (MPEG-DASH), are widely employed for unicast and broadcast communications. It also enables Adaptive Rate Control on the client device allowing players to select a representation and bitrate matching the capabilities of the network at any moment. For the latter, MPEG-DASH does not provide low latency for Live streaming when compared to a Broadcast service. Secure Reliable Transport (SRT) is proposed by SRT Alliance to overcome such limitations of unicast and broadcast communications. Nevertheless, it misses the adaptation of the content to the available network resources. In this paper, we show an implementation of Adaptive Rate Control for SRT protocol which exploits periodical network reports in order to adapt the content encoding process. The evaluation includes a real deployment of the solution and a comparison with a legacy SRT stream.

Keywords: Rate Control, Traffic and performance monitoring, Secure Reliable Transport, Video coding and processing.

3.2.1 Introduction

MPEG-DASH [23] and other HTTP-based alternatives are widely employed solutions for media services. It is compatible with existing HTTP-based Internet infrastructure and allow resolution and encoding bitrate selection to mitigate network performance fluctuations in unmanned networks. These solutions perfectly fit for Video on Demand (VOD) scenario where the latency between content packaging and playback is not an issue. On the contrary, they are not suitable when latency constraints come into play. Live streaming applications, such as video surveillance and video conference, cannot work with tens of seconds of delay of HTTP-based solutions.

Real Time Streaming Protocol (RTSP) and Real Time Messaging Protocol (RTMP) are legacy protocols for Real-time streaming which enable lower latency than HTTP-based solutions. Nevertheless, they are designed to work in unicast mode, meaning that the communication is based on a server-client delivery where the server sends the content in push mode. This communication model fails when players scale up to broadcast concurrency rates, since the server should push as many unicast steams as the number of connected players. Moreover, these solutions suffer of network restrictions applied by network functions, such as firewall and NAT, blocking the delivery of those streams.

Secure Reliable Transport (SRT) protocol [343] is the proposal of SRT Alliance to fill this gap. It lets to gain scalability of Broadcast delivery while guaranteeing low latency required for Live streaming. SRT has been designed to work in both push and pull mode, then allows to stream content even when firewalls and NATs network functions are present. The protocol also includes forward error correction (FEC) [344] which enforces resilience from transmission errors. SRT server also employs network reports from the client to adapt packet overhead. Thus, the server upload speed depends on network throughput and packets are not lost when the available throughput is enough to send the content to the client. Lost packets are re-transmitted only if the network throughput can absorb such overhead.

SRT does not interfere with content encoding, then network reports are never exploited to adapt resolution and encoding bitrate of the content as it happens in HTTP-based solutions. Enabling content adaptation on top of SRT allows two main advantages. First, in case of network degradation, it shields from playback stalls by reducing the bitrate of the content to be send, as MPEG-DASH does. Second, it allows to send a representation matching the client display features. This work proposes a real implementation of an Adaptive Rate Control for SRT streams. This solution includes two relevant contributions:

- A server-side Adaptive Rate Control implementation on top of Open Source framework for SRT streaming applications. This Adaptive Rate Control exploits the network reports employed by SRT protocol to enable the adaptation of the resolution and encoding bitrate of the content.
- Evaluation of the effects on user's Quality of Experience (QoE) when compared the proposed solution to a legacy one.

The rest of this paper is organized as follows. First, section 3.2.2 presents the background of Video Streaming solutions and performance metrics. Then, section 3.2.3 shows the implementation of our Adaptive Rate Control on top of SRT streams. In section 3.2.4 we describe the experiments and present the results. Finally, in section 3.2.5 we expose the conclusions and future work.

3.2.2 **Related Work**

3.2.2.1 **Overview of Video Streaming**

MPEG-DASH [23] was developed by MPEG and standardized by ISO/IEC. MPEG-DASH is a pull-based streaming technology over HTTP, where the client requests the content from a conventional HTTP server which stores it split into segments and encoded at many representation levels. A segment consists in a unique ISO Base Media File Format fragment, usually called MP4 fragment, which is the minimum playable data. First, the client fetches a manifest file, referred as Media Presentation Description (MPD), and parses it to be aware of the different representations of the content. Then, the client downloads the segments corresponding to the representation that matches the device capabilities and user preferences in terms of resolution, language, codec and bitrate. Each time the client requests the next segment, it can switch to a different representation depending on network performance to avoid playback degradation and maximize user's QoE. Thus, the Adaptive Rate Control is fully managed by the client during the streaming session. However, the delay of MPEG-DASH streams is high since, by design, it is not possible to go behind the segment duration, which usually ranges from 2 to 30 seconds. Thus, MPEG-DASH is not suitable for real-time applications. Apple HTTP Live Streaming (HLS) and Microsoft Smooth Streaming are other solutions working with a similar workflow to MPEG-DASH, where the format of the manifest file differs, and experience delays with the same order of magnitude.

The Common Media Application Format (CMAF) [70], proposed by ISO/IEC, tries to overcome such latency limitations of MPEG-DASH by introducing a Low Latency mode, namely Low Latency or Chunked CMAF. Chunked CMAF allows the presence of several MP4 fragments inside one segment. Consequently, the buffering done by the client is shorter as it can start to play the content even if the segment is not completely downloaded, it just needs to have a MP4 fragment. In [345] and [346], two different implementations of Chunked CMAF are presented. The former, [345] still evidences latency in the order of 1 second, the latter, [346] reduces latency behind one second by generating fragments containing just one frame. The use of just one frame introduces a heavy overhead inside the communication since MP4 header must be replicated each time a fragment is sent. Moreover, it comes at cost of reduced QoE since smaller fragments causes a smaller playout buffer which more easily can go empty [347]. Nevertheless, Chunked CMAF is not currently used by the media industry, but it is a promising solution for the future.

In any case, HTTP-based solutions were not designed for real time applications. Thus, achieving similar latency performance as real-time designed protocols, based on Real-time Transport Protocol (RTP) and User Datagram Protocol (UDP) [60], is complicated. RTP Control Protocol (RTCP) [348] was designed to work jointly with RTP. Hence, RTCP does not transfer streaming data, but it provides RTP with an out of band channel to get feedback on the network statistics, enabling RTP to control the transferring rate. However, such adaptation is made at the network interface level. Then, it does not imply changes on the bitrate of the encoder that could make the difference to adapt the throughput to the available network bandwidth, preventing packet losses. RTP-based solution has also some drawbacks. On the one hand, RTCP is designed to work with unicast streams and, on the other, RTCP is only compatible to push communications mode. Thus, it is difficult to scale as the number of clients increases and when firewalls and NATs are present.

Periscope, one of the most common live streaming services, overcomes these issues by using a hybrid RTMP and HLS solution [349]. Here, the streaming protocol is chosen depending on the volume of clients and latency trade-off. For streaming sessions involving few clients, RTMP is preferred to reduce latency. When the number of clients increases, HLS is exploited to reduce overheads at the server.

SRT [343] protocol, proposed by SRT Alliance, is the media industry solution to transfer live broadcast streaming under the constraint of low latency. The protocol also includes a mandatory encryption to enforce the security. SRT allows both push and pull modes which means that, in case of network traversal barriers, pull mode could bridge them. Moreover, the use of a FEC mechanism [344] enforces resilient communication. Network feedback reports are also exploited to tune the packet overhead and provide protection against transmission errors. In case of packet losses, they are re-transmitted or discarded depending on the configured maximum latency and on the network possibilities to support such overhead.

Finally, SRT includes many advantages compared to conventional real time protocols, but it still lacks the capability to adapt the bitrate throughput of the content when the network bandwidth changes. Network reports are exploited to tune the transferring rate, but they are not accessible by the encoding process. Consequently, resolution and encoding bitrate of the content are neither adapted at the server nor at the client as it happens in HTTP-based solutions.

3.2.2.2 **Performance metrics**

All the proposed streaming technologies have a common aspect, they need to focus not only on reducing the latency to allow live streaming, but also on maximising the QoE to retain user when satisfying expectations. The QoE is a key aspect for user satisfaction and retention when rating streaming services. An exhaustive QoE evaluation requires a demographic perception study to get a Mean Opinion Score (MOS) [84].

Nevertheless, there are many studies in literature which demonstrate that the use of objective performance metrics is helpful to provide an estimation of user's QoE [19].

In [350] the authors consider stalling time, number of representation switches and inter-switching time as objective metrics to estimate user's QoE. Recently, the work [351] also includes initial buffering. In both cases, the proposed performance metrics are applicable only for HTTP-based streaming applications involving content adaptation. Since our solution aims to include the same feature on top of SRT protocol, the same performance metrics can be assessed.

3.2.3 **Adaptive Rate Control Implementation**

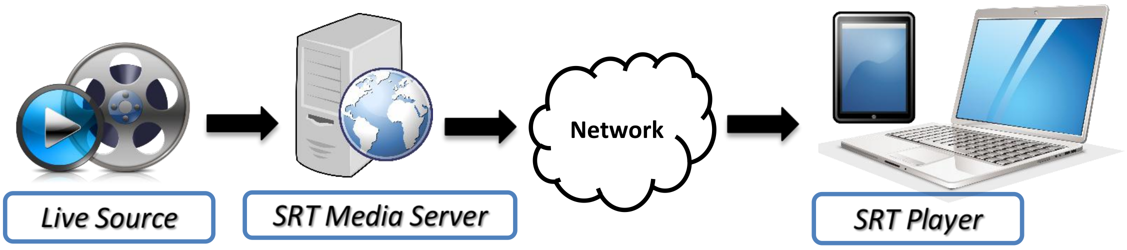


Figure 3.1: System architecture for SRT streaming.

The system architecture for delivering SRT streams is depicted in Figure 3.1. The system is composed by a Live Source, a SRT Media Server and a SRT Player.

The Live Source is the node which provides the content to the processing and delivery pipeline. Here, many different entities can act as a Live source, e.g. a camera or a video software editor.

The SRT Media Server processes the content ingested by the Live Source and delivers it to the SRT Player after encoding and packetizing it into an SRT-compliant stream. Thus, it accomplishes the following tasks:

- Encoding: it encodes the content into a live H.264 bitstream [24]. In a legacy SRT solution, the video frame resolution and encoding bitrate is chosen when launching the encoding process and kept unaltered during all the process. In our approach, both resolution and bitrate can be dynamically changed during the streaming session to provide different representation levels of the same content.

- Muxing: H.264 bitstream is packetized into a MPEG-2 Transport Stream (MPEG-TS) container [352].

- Encryption: MPEG-TS is encrypted though 128/256 bit Advanced Encryption Standard (AES) [73]. This is a mandatory feature included in SRT to enforce end-to-end security.

- Delivery: SRT employs User Datagram Protocol (UDP) to transmit data over the network to the client since it guarantees lower latency than Transmission Control Protocol (TCP), which is commonly used by HTTP-based streaming solutions. However, UDP is not reliable since it does not provide mechanisms to compensate for transmission errors. Then, SRT includes Forward Error Correction (FEC) and re-transmission mechanisms on top of UDP delivery to allow the SRT Player to recover from lost or corrupted packets.

- Monitoring: SRT server receives network reports from the SRT Player which contain information related to network status (bandwidth and delay) and packet transmission (sent, lost or re-transmitted packets number). In a legacy SRT solution, reports are only employed to tune the sending transmission rate and schedule the transmission of new and/or lost packets. In our approach, network reports are also captured and employed to select the appropriate representation level (resolution and bitrate) to be used by the encoding process.

We employ GStreamer [353] in its v1.14 stable release to develop our Adaptive Rate Control-enabled SRT Media server. We select and setup the following plugins to accomplish the above tasks:

- *H.264 encoder*: the setup of the encoder is key to allow the adaptation of the representation (resolution and bitrate) of the content according to the measured network statistics. *Keyframes* (I-frames) do not require any other frames to be decoded, so the player can always start decoding a stream from a *keyframe*. Thus, *keyframes* are essential for live streaming to start playing the content as soon as possible when the player starts receiving the content. Moreover, in our Adaptive Rate Control-enabled stream, when it switches the representation level, it introduces a discontinuity. Thus, it makes new frames, with a different resolution and encoding bitrate, not possible to be decoded based on the previous frames. The player needs a new *keyframe* to decode the stream every time the representation level changes. Then, the Adaptive Rate Control forces the encoder to introduce a *keyframe* every time a representation switch is performed.

- *MPEG-TS muxer*: it packetizes H.264 encoded frames into MPEG-TS chunks. Each chunk cannot contain data at different representation levels. Then, it is mandatory that each MPEG-TS chunk starts with a *keyframe*.

- *SRT server sink*: it receives MPEG-TS chunks from the muxer, it encrypts and encapsulates them into UDP packets before sending them to the player. This plugin gets active, sending packets, only when a client is connected. It also monitors the network by getting network statistics measured during the transmission of the video stream to the client. A legacy SRT server sink uses statistics only to adapt the network overhead of the transmission, to reduce packets lost and to avoid re-transmissions. Additionally, the proposed Adaptive Rate Control-enabled solution exploits this information to dynamically change the setup of the encoder plugin to switch to a suitable representation level.

Finally, SRT player is implemented with GStreamer release (v1.14) [353]. This version provides SRT client and decoding capabilities. Thus, it can play SRT legacy streams. Moreover, the aggregation of the Adaptive Rate Control to the SRT Media server does not cause any relevant change in the protocol. This means that developing a custom client application is not required and the generated SRT stream can be played by any SRT-compliant player.

The communication between the systems in the setup is shown in Figure 3.2. SRT Media Server starts to encode and packetize video frames when the Live Source is connected. Video frames are H.264 encoded and MPEG-TS muxed, then packetized into SRT chunks. Nevertheless, data are not transmitted until an SRT player connects to the SRT Media Server. It means that chunks can be discarded by the server if there are no players. The SRT Media Server only stores the most recent chunks with a buffer size of "maximum allowed latency". This operation is necessary in order to guarantee that only the most recent chunks are sent, then achieve a low latency live streaming. In

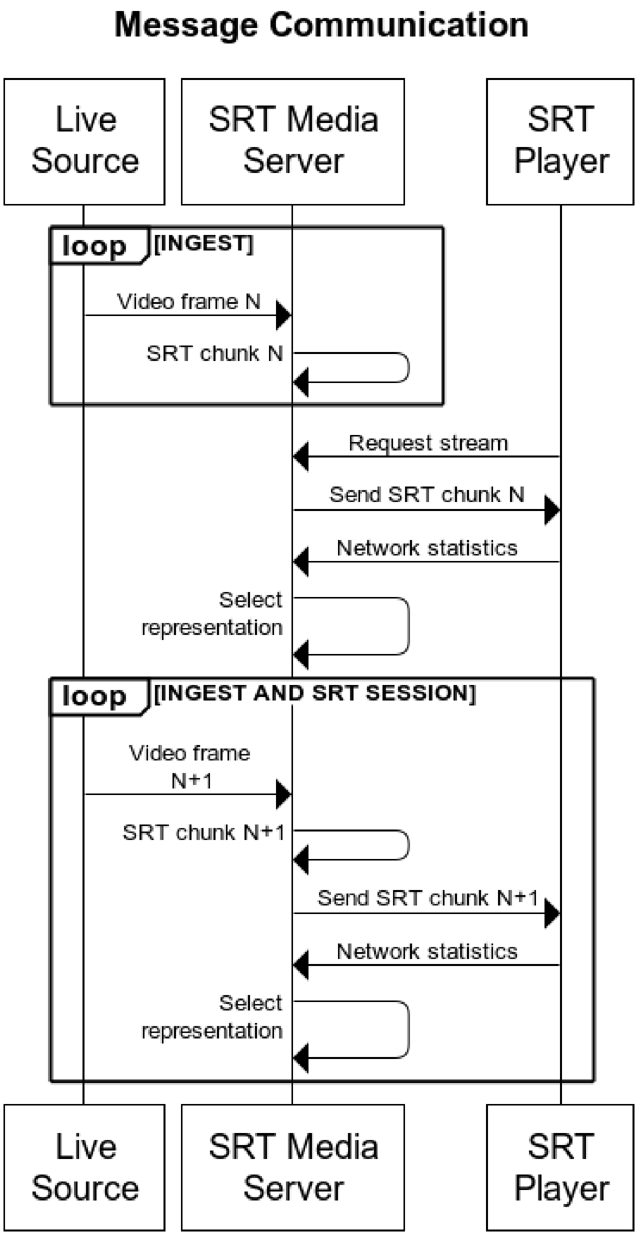


Figure 3.2: Sequence diagram.

GStreamer, maximum allowed latency is configurable, but we decide to keep it to its default value which is 125 ms. Once the SRT player is connected, SRT Media Server starts delivering the content to the player. While receiving the content, the SRT player stores it in the playback buffer before decoding and displaying it. The playback buffer is set to 1 seconds to balance low latency, packet losses reliability and changeable network conditions.

To adapt the representation, the implemented Adaptive Rate Control accesses network statistics from the SRT server sink plugin and exploits the information to tune the configuration of the H.264 encoder. This evaluation is performed by the Adaptive Rate Control once per second. The decision algorithm of the implemented Adaptive Rate Control is shown in Algorithm 1. The algorithm takes the last measured network bandwidth (bw_t), round-trip delay time (rtt_t) and send rate ($rate_t$) from SRT network reports, the current employed representation level (rep_t) and the list of all the available ones ($\{rep_{list}\}$). Bandwidth and round-trip delay time are employed to evaluate the maximum allowed network throughput ($throughput_{max_t}$) through the Equation 3.1.

Algorithm 1 Adaptive Rate Control

```
  function ADAPTIVERATE($\mathrm{bw}_t$, $\mathrm{rtt}_t$, $\mathrm{rate}_t$, $\mathrm{rep}_t$, $\{rep_{list}\}$)
Input: $\mathrm{bw}_t$                                        ▷ measured bandwidth
Input: $\mathrm{rtt}_t$                                       ▷ measured delay
Input: $\mathrm{rate}_t$                                      ▷ measured send rate
Input: $\mathrm{rep}_t$                                       ▷ current representation
Input: $\{rep_{list}\}$                                       ▷ available representations
Output: $\mathrm{rep}_{t+1}$                                  ▷ next representation
    $\mathrm{throughput}_{max_t} \leftarrow \mathrm{bw}_t, \mathrm{rtt}_t$      ▷ network throughput
    for all $\mathrm{rep}^i \in \{rep_{list}\}$ do                ▷ for each representation
        $\mathrm{bitrate}_t^i \leftarrow \mathrm{rep}^i, \mathrm{rate}_t, \mathrm{rep}_t$     ▷ minimum network bitrate
        if ($\mathrm{throughput}_{max_t} > \mathrm{throughput}_t^i$) then
                                                  ▷ network admits the representation
            $\mathrm{rep}_{t+1} \leftarrow \mathrm{rep}^i$                ▷ next representation
```

$$throughput_{max_t} = bw_t * \frac{1}{1 + \frac{rtt_t}{2}} \tag{3.1}$$

Then, for each available representation, the required network throughput to allow its transmission is calculated. It is important to note that each representation means a different throughput configured by the encoding bitrate. Furthermore, the encoding bitrate needs to accommodate a gap to allow that protocols messages and extra information from other levels of the ISO/OSI model, added to the encoded video payloads, still meet network bandwidth thresholds.

Consequently, to establish if it is possible to stream a specific representation to the client, we should compare the maximum allowed network throughput ($throughput_{max_t}$) with the throughput that the representation would generate ($throughput_t^i$). Since we cannot access the employed throughput before streaming the content, we estimate it from the current send rate ($rate_t$) provided by the network reports. Equation 3.2 estimates the necessary network throughput ($throughput_t^i$) to allow the representation (rep^i) bitrate to be streamed. The ratio between current send rate ($rate_t$) and current representation encoding bitrate (rep_t) is the current overhead, then we multiply it per the representation encoding bitrate (rep^i).

$$throughput_t^i = rep^i * \frac{rate_t}{rep_t} \tag{3.2}$$

If the estimated throughput ($throughput_t^i$) is lower than the maximum allowed throughput ($throughput_{max_t}$), it means that the representation can be sent. The output of the algorithm is the selected representation to be employed at the H.264 encoder. The encoder is configured to immediately generate a *keyframe* and switch at the new selected representation resolution and encoding bitrate.

3.2.4 Results

The experimental setup employed for testing the implemented Adaptive Rate Control is presented in Figure 3.3. The overall setup comprises the following nodes:

- STR Media Server and Traffic Control: this is a unique physical node which embeds two logical systems. We employ a Docker containerization [354] to run different functions in separated environments. A Docker container running Ubuntu 19.04 OS includes the Adaptive Rate Control-enabled SRT Media Server developed through GStreamer framework [353]. The container communicates with the host machine running Ubuntu 16.04 OS which forwards data to the physical network interface. On the host machine, we periodically modify bandwidth and latency of the network interface. Traffic Control [355] is the utility to change the uplink capacity. To emulate an LTE network, we use the *European Broadband user experience* dataset collected and publicly provided by the Joint Research Centre (JCR) of the European Commission [356]. The dataset provides both bandwidth and latency that we apply through Traffic Control utility. The interval between two consecutive network changes is set to 100 ms.

- Network switch: this node provides wired network access to both server and player nodes to communicate each other. It forwards all the incoming traffic on both sides.
- SRT Player: this node run a GStreamer application to receive the SRT stream and play it.

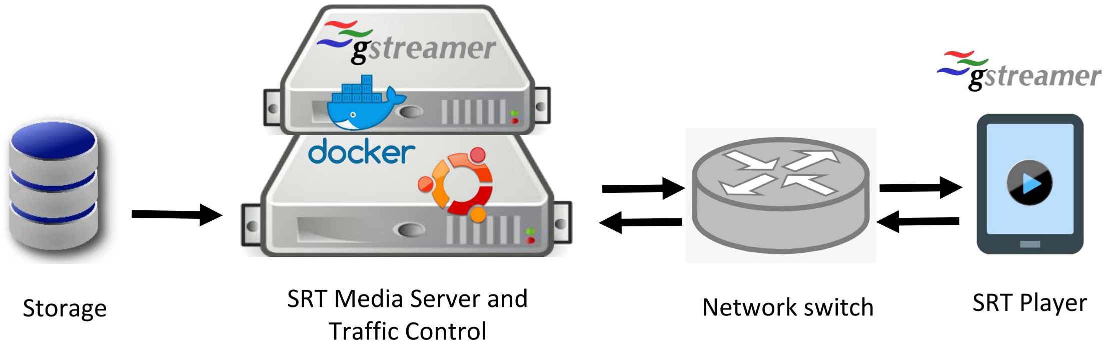


Figure 3.3: Experimental setup.

Table 3.1: Set of representations employed in the experiments.

Index	bitrate (kbps)	resolution	framerate (FPS)
1	1200	640x360	24
2	2250	1280x720	24
3	4500	1920x1080	24

We perform different experiments to compare the Adaptive Rate Control-enable SRT stream with a legacy SRT stream. We use a locally stored Big Buck Bunny test sequence to feed the SRT Media Server. Its raw version is provided by Xiph.Org Foundation [357].

For the Adaptive Rate Control-enabled stream, we established three different representation levels to be employed by the H.264 encoder, while legacy one employs only the higher one. The representations are shown in Table 3.1.

The duration of each SRT streaming session lasts 594 seconds which is the duration of the employed test sequence. The results of the two strategies in terms of representation switches, freezes, initial delay and average representation bitrate are shown in Table 3.2.

Table 3.2: Number of switches (S_{Nb}), number of freezes (F_{Nb}), average freeze duration (F_{avg}), initial delay (D) and average representation bitrate (R_{avg}) for both legacy and Adaptive Rate Control-enabled SRT streams.

SRT server	S_{Nb}	F_{Nb}	F_{avg}(ms)	D(ms)	R_{avg}(kbps)
Legacy	0	122	820	972	4500
Adaptive Rate Control	197	87	727	986	4028

In terms of switches, Adaptive Rate Control solution performs 197 representation switches, while it is not applicable to the legacy SRT stream. Figure 3.4 shows the distribution of the switches across the streaming session. Legacy stream is stable to 4500 kbps encoding bitrate, while seems that Adaptive Rate Control one never uses the lowest encoding bitrate (1200 kbps) but moves between the other two (4500 and 2250 kbps). Thus, it causes that the average representation bitrate is 10% lower when using the Adaptive Rate Control (4028 kbps against 4500 kbps). In terms of initial delay, there is not a noticeable difference since the Adaptive Rate Control does not introduce any delay while starting the streaming session. On the contrary, our Adaptive Rate Control solution outperforms legacy one while considering number of freezes and their duration. Here, the proposed solution scores 29% less freezes events, and their average duration is 11% shorter.

These results show that our solution sacrifices average representation bitrate in order to reduce freezes events. Fewer freezes events lead to a smoother video playback for the end user.

3.2.5 Conclusions and Future Work

This paper proposes an Adaptive Rate Control for SRT protocol to deliver live streaming content, while coping with transmission variability due to network degradation or

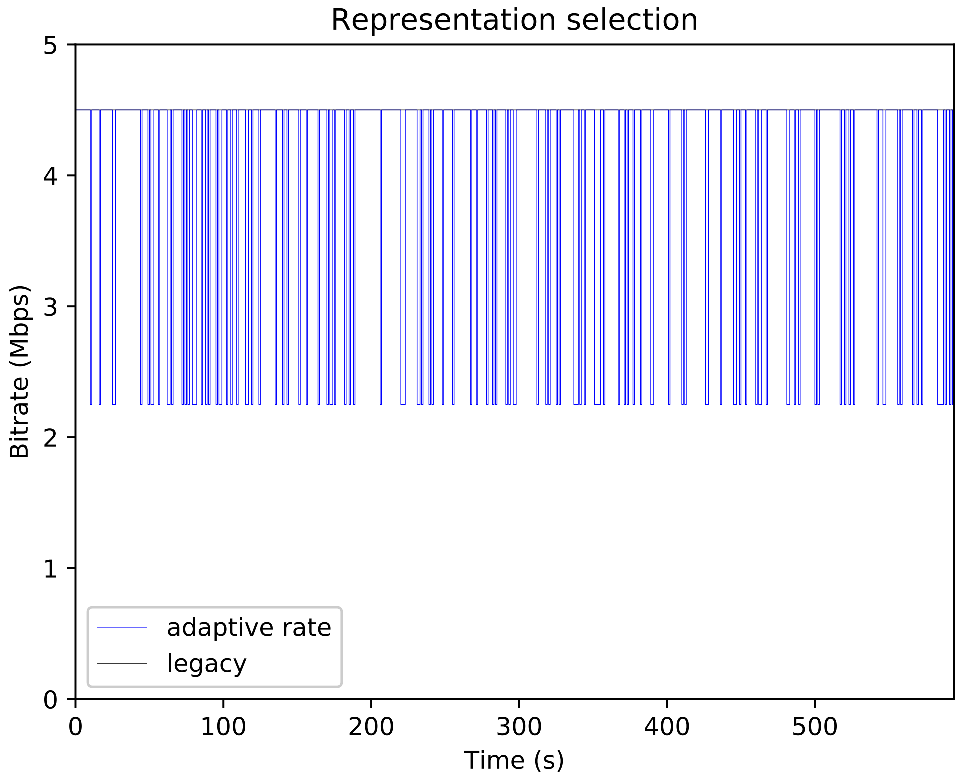


Figure 3.4: Representation bitrate selection.

issues.

The proposed solution is integrated with Open Source GStreamer multimedia framework and tested by altering the network capabilities according to real LTE measurements provided by a publicly available dataset.

Compared to a legacy SRT solution, the results show that Adaptive Rate Control-enabled SRT delivery experiences fewer freeze events by enabling switching operations to lower representation bitrates. Thus, it reduces the average representation bitrate to prioritize playback smoothness.

3.3 QoE-based enhancements of Chunked CMAF over low latency video streams

- **Title:** QoE-based enhancements of Chunked CMAF over low latency video streams
- **Authors:** Roberto Viola, Alvaro Gabilondo, Ángel Martín, Juan Felipe Mogollón and Mikel Zorrilla
- **Proceedings:** 2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)
- **Publisher:** IEEE
- **Year:** 2019
- **DOI:** `10.1109/BMSB47279.2019.8971894`

Abstract: 5G infrastructures are in the roadmap of content delivery services, aiming to forward all broadcast and broadband video traffic using a common telecommunication network architecture. Streaming services will benefit from 5G networks which promise higher capacity, higher bandwidth and lower latency than current infrastructures. However, the widely employed streaming technologies, such as Dynamic Adaptive Streaming over HTTP (MPEG-DASH), require an intrinsic high latency of tens of seconds to enforce the Quality of Experience (QoE). These conditions turn MPEG-DASH unfavourable when compared with a traditional broadcast pipeline for live events in terms of latency. Therefore, improvements on latency of streaming technologies are necessary to deliver live broadcast services over 5G networks. The media industry proposed a Chunked Common Media Application Format (Chunked CMAF) in order to achieve latency under a second. In this paper, we show an implementation of a Chunked CMAF for MPEG-DASH live videos in a real deployment. To further evaluate the benefits of CMAF we evaluate the QoE results when delivering a legacy MPEG-DASH live content compared to a Chunked CMAF-powered one.

Keywords: Chunked CMAF, Future broadcasting services, MPEG-DASH, Quality of Experience, Video coding and processing.

3.3.1 Introduction

Video streaming services represent a large fraction of the Internet traffic. In fact, live video traffic will grow 3-fold from 2017 to 2022, accounting the 17% of all internet video traffic [358]. This trend would be fuelled by the deployment of 5G networks, which will enable the cooperation between broadband and broadcast services [359].

5G networks promise high capacity, high bandwidth and low latency to cope with demanding traffic and services. Nevertheless, MPEG-DASH and other HTTP-based alternatives require to packetize video contents in segments with a duration in the order of seconds to enforce the QoE, leading to tens of seconds of delay between content generation and consumption [360]. This delay is produced by the duration of packaged media segments, designed to match changeable network delivery performance, and the required buffering, done by the media player to achieve a smooth playback for on-demand and live streams. This delay is not significant for on-demand contents, but it is too high to deliver live streams with comparable performance to the broadcast live services. Moreover, when deploying hybrid broadcast broadband services, the common mechanism to synchronize broadcast and broadband signals consists in delaying the broadcast source as broadband stream is usually 30-40 seconds delayed with respect to the broadcast service.

The Common Media Application Format (CMAF) [70], proposed by ISO/IEC, is the solution for delivering live contents from media industry. CMAF includes two major benefits. First, it ensures the use of the ISO Base Media File Format, usually referred as MP4, as a common file format when combined with different streaming technologies such as MPEG-DASH or HTTP Live Streaming (HLS). This feature makes media storage more efficient as different manifests (MPD for MPEG-DASH and M3U8 for HLS) may index the same segments. Each media client can download a different manifest depending on its supported streaming technologies. Then, it plays the content by downloading the same media segments. Thus, the server needs lower storage capacity. Secondly, it defines a chunked mode, named Chunked CMAF or Low Latency CMAF, which enables latency enhancement of the stream, reducing the time elapsed between media packaging and its playback.

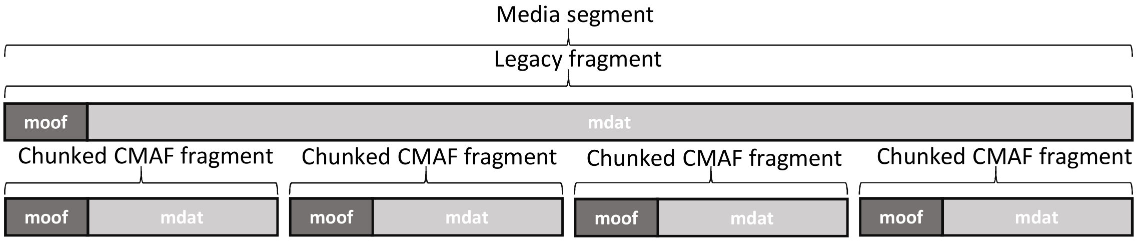


Figure 3.5: Legacy fragment and Chunked CMAF fragment.

Typical MPEG-DASH media segments contain a single MP4 fragment with the fragment duration equal to the segment duration. Here, common values for segment duration are from 2 to 30 seconds. On the contrary, Chunked CMAF enables a single segment to contain multiple fragments as depicted in Figure 3.5. Therefore, Chunked CMAF exploits the use of short MP4 fragments, including the minimum data required by the player to start decoding the stream. Therefore, the shorter fragment duration allows a promptly playback start, removing the limitation to fully download the entire segment.

This work proposes a real implementation of a server-client solution delivering Chunked CMAF streams of live contents. This solution has been achieved by providing two relevant contributions:

- Chunked CMAF has been integrated with an Open Source MPEG-DASH framework.
- The evaluation measures the effects on user's QoE while varying the fragment duration of the Chucked CMAF segments and the resulting latency.

The remainder of this paper is organized as follows. First, section 3.3.2 presents the background of the MPEG-DASH standard and Chunked CMAF for deploying live media services. Then, section 3.3.3 shows the implementation of Chunked CMAF solution. In section 3.3.4 we describe the experiments and present the results. Finally, in section 3.3.5 we expose the conclusions and future work.

3.3.2 Related Work

In this section, it is included an overview of MPEG-DASH for live streaming services and afterwards, a comprehensive review about the State of Art of Chunked CMAF is described.

3.3.2.1 Overview of Live MPEG-DASH

MPEG-DASH was developed by MPEG and standardized by ISO/IEC. In MPEG-DASH, first, the client fetches a Media Presentation Description (MPD) and parses it to be aware of the different representations of the content. Then, the player chooses the representation that fits in the device capabilities in terms of resolution, language, codec and bitrate. Accordingly, the client requests and downloads the corresponding segment from the server. Once a segment has been played, the next one from the MPD is requested. During the playback, the player can switch to a different representation depending on its preferences and network performance in order to minimize any impact on the QoE.

The live playback is possible through the *availabilityStartTime* field in the MPD, which marks the UTC time when the stream is made available. The client continuously compares it with the current time to fetch the last available segments. In the case of a legacy live MPEG-DASH content, the *availabilityStartTime* has to correspond to the time when the first media segment is fully available on server side. Then, the segment duration, which ranges from 2 to 30 seconds, is the minimum delay that a client experiences during a live stream. Encoding and network latency also influence this delay, but their weights in the resulting latency are 10-100ms when aggregated to the segment duration.

3.3.2.2 Chunked CMAF

The latency is a key factor when dealing with live streaming contents. Current MPEG-DASH-based solutions are not able to operate with a similar latency to the current broadcast solutions. This is a major challenge when targeting broadcast levels of QoE.

The use of Chunked CMAF, together with improved network bandwidth and latency of the 5G networks, aims to reduce live streaming latency and keep it behind a second [72].

Contrary to the legacy MPEG-DASH streams, a Chunked CMAF-compliant MPEG-DASH distinguishes between MP4 fragment duration and media segment duration with different values. The reduction of the fragment duration makes the media units of the stream quickly available enabling prompt playback. Consequently, the segment contains multiple fragments and the *availabilityStartTime* attribute contained in the MPD must be set at the time the first fragment is available, even if the segment is not completely written. The smaller the fragment is, the smaller delay is experienced by the player. Theoretically, fragment duration can be reduced to one frame duration. To this end, it is required a proper server, which must be able to split the HTTP response sending fragment units instead of the full segment.

Several implementations serve Chunked CMAF with HTTP 1.1 Chunked Transfer Encoding. The server encapsulates each MP4 fragment in a HTTP chunk and deliver it over time, instead of sending the entire segment at once. In [361], Chunked Transfer Encoding allows a HTTP 1.1 server to split the response in small HTTP chunks. The paper shows that the latency does not depend on the segment duration but depends on the duration chosen for the HTTP chunks. This approach still uses one second duration chunks while splitting the HTTP connection between server and player. The author of [346] also implements a MPEG-DASH delivery involving Chunked CMAF, but it varies the duration of the fragment. Both papers provide performance results in terms of overall latency.

The work shown in [345] uses HTTP 2.0 to exploit the push-mode added in the new HTTP version for reducing the latency. HTTP 2.0 does not have a Chunked Transfer Encoding mode, since it already employs a frame-based delivery, i.e. it splits the response in several frames which contain the Chunked CMAF MP4 fragments. This solution has the advantage of reducing the protocol header overhead since HTTP 2.0 header is simplified when compared to HTTP 1.1. However, push-mode reduces the adaptation possibilities at the client-side to dynamically select an appropriate representation for the network performance conditions. Decisions can be still done by the server, modifying the MPD, but this approach does not scale as the player-side decisions when working in pull-mode.

3.3.2.3 QoE Metrics

The reduction of the latency of the service is the major goal to all these scientific approaches. Nevertheless, when evaluating streaming services, it is essential to focus in user's QoE. The QoE is a key aspect for user satisfaction and retention when rating streaming services. Hence, any solution trying to enhance media delivery needs to consider QoE metrics. No one of the above works consider QoE as a metric in order to evaluate the real benefits to end users.

A commonly used scale to evaluate QoE is the Mean Opinion Score (MOS) which consists of five increasing quality levels (from 1 to 5) [84]. In the literature many models are available to profile the subjective human perception of the quality and estimate the MOS through objective metrics. In [350] the author uses metrics like initial delay, stalling time, number of representation switches and inter-switching times in order to get an estimated Mean Opinion Score (eMOS). This eMOS quantifies the quality of video streaming services based on objective streaming connectivity and buffering measures of players without a demographic perception study of users. In [362] the author proposes to evaluate the MOS depending on the initial delay and the numbers of freezes. It concludes that is preferable a higher initial delay than freezes in order to have a better human perception. Recently, the work [351] investigates a new model for MOS, called Ubiquitous-Mean Opinion Score for Video (U-vMOS), which makes initial buffering more dominant than [350].

3.3.3 Chunked CMAF Implementation

3.3.3.1 System Model

The implemented end-to-end system for delivering Chunked CMAF MPEG-DASH streams is depicted in Figure 3.6. The system is composed by the following nodes: Video ingest, Media Packager, HTTP Server and DASH Player.

Video ingest is the system which injects the content into the processing and delivery chain. Many and different entities can act as a live source for media production, e.g. a live camera or any video software editor.

Media Packager is based on GStreamer [353]. It is in charge of processing the content ingested and generating a standard compliant Chunked CMAF MPEG DASH stream. It encodes the content, packetizes it into MP4 fragments and segments and creates a

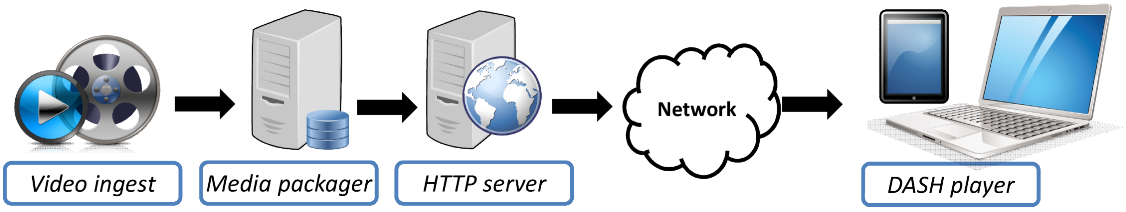


Figure 3.6: End-to-End DASH streaming system.

live DASH Media Presentation Description (MPD) to expose the stream to the clients. The recent stable GStreamer release (v1.14) does not provides capabilities for generating a Chunked CMAF steam. Then, to experiment with Chunked CMAF, we introduced additional features and properly tuned the following plugins based on GStreamer v1.5:

- *H264 encoder*: the performance and setup of the encoder is key to favour a low latency stream, enabling the player to request the last generated MP4 fragment and to start its playback. *Keyframes* (I-frames) do not require any other frames to be decoded, so the player has to start decoding a stream from a *keyframe*. Thus, *keyframes* are essential for live streaming and each generated fragment should start with a *keyframe* to boost the playback start. The Group of Pictures (GOP) is a collection of successive pictures within a coded video stream where the *keyframe* always indicates the beginning of a GOP. In terms of header information, some fields are mandatory for playing the stream, e.g. Sequence Parameter Set (SPS) and Picture Parameter Set (PPS) provide basic parameters like the frame size. Consequently, we encode the content forcing the presence of a *keyframe* and all the header information at the beginning of each MP4 fragment. Moreover, for the remaining frames we avoid to use bidirectional predicted frames (B-frames) since they add additional latency into the encoding process due to required frames reorder. The resulting stream contains only key and predicted frames (I-frames and P-frames).

- *MP4 muxer*: it packetizes the encoded frames into MP4 fragments and segments. Since we established that each fragment contain only one GOP, with a *keyframe* at the beginning, the muxer works by switching from a fragment to another each time it recognizes a *keyframe* generated by the encoder. In case a minimum theoretical latency wants to be exercised, putting bandwidth efficiency aside for unlimited connectivity, the encoder could just use *keyframes*, i.e. no predicted frames are used, meaning the MP4 fragment contains just one frame. According to the specifications, each fragment contains header information (moof) and a payload (mdat) with the encoded data.

- *MPEG-DASH filesink*: it receives the MP4 fragments from the muxer and aggregates them in order to write the segments on the disk. Since the fragments can be decoded independently each other, the fragments are concatenated by appending the new fragment at the end of the previous one. Following this strategy, it is not necessary to receive all the fragments to start writing a segment, which is progressively written on the disk. The filesink also creates the MPD manifest which contains the URL where the player can download the last fragment or, in case of legacy live MPEG-DASH stream, the segment. The filesink also updates the *availabilityStartTime* field of the MPD manifest to allow the player to calculate the last generated fragment (or segment) time with accuracy and to download it. The generated MPD and the segments are directly written by the filesink in the storage of the HTTP Server.

HTTP Server is based on Node.js [363]. It is in charge of serving the content generated by the Media Packager to the player. When the player connects to the HTTP Server, the server loads the content from its storage and serves each fragment promptly to the player. Its functions include:

1. It loads partial segments which are still being generated by Media Packager.
2. It analyzes a segment and recognizes the contained MP4 fragments.
3. It serves the fragments to the client through HTTP 1.1 chunked transfer encoding. Each HTTP chunk contains one fragment.

DASH player is based on the last stable GStreamer release (v1.14) [353]. This version already provides capabilities to parse the manifest and request the last generated segment, then decoding and displaying it. Thus, it is able to play legacy MPEG-DASH streams. Moreover, GStreamer HTTP source plugin is able to receive a HTTP 1.1 chunked response when using a fragmented segment, but it does not pass the downloaded fragments to the decoding pipeline until it receives the whole segment. Thus, this is not valid in case of low latency streaming and the implementation of a Chunked CMAF Media Packager and HTTP Server would not be applicable. On the contrary, the HTTP source can request a section of a file if the exact byte range of the fragment inside a segment is known. The player can request the fragments separately and forward it to the decoding pipeline. Consequently, we modify the capabilities of the Media Packager in order to add *mediaRange* [364] attributes inside the manifest which explicitly provides the DASH Player the byte range of the fragment. The player parses this attribute and requests separately the fragments to the server by adding Range header [365] into the HTTP 1.1 request. The HTTP Server receives the requests, analyzes the segment and send the fragment included in the chosen range. When the player receives the fragment, it forwards the fragment to the decoding pipeline. Furthermore, to reduce latency, it is also important to take into account the internal playout buffer at the player since it is a widely used mechanism for preventing image freezes. However, it adds delay when playing the content. To overcome this limitation, we tuned the buffer size to be equal to one fragment duration. Finally, to synchronize the player and calculate the last fragment time with accuracy, we employ the network time protocol (NTP) to keep the Media Packager at server side and player device at client side synchronized. To sum up, the player does the following tasks:

1. It parses the MPD manifest in order to get the *availabilityStartTime*
2. It compares the *availabiltyStartTime* with NTP clock time in order to know which is the last generated fragment or, in case of legacy MPEG-DASH stream, segment.
3. It requests the last generated fragment (or segment) from the HTTP Server.
4. It decodes and displays the received stream.

The communication between the nodes of the system is shown in Figure 3.7.

Media Packager begins to encode and packetize the content into MP4 fragments when the live source is connected. The Chunked CMAF content is directly stored inside the storage located at the HTTP Server. Meanwhile the content is generated, the clients can connect to the HTTP Server which serves the segments.

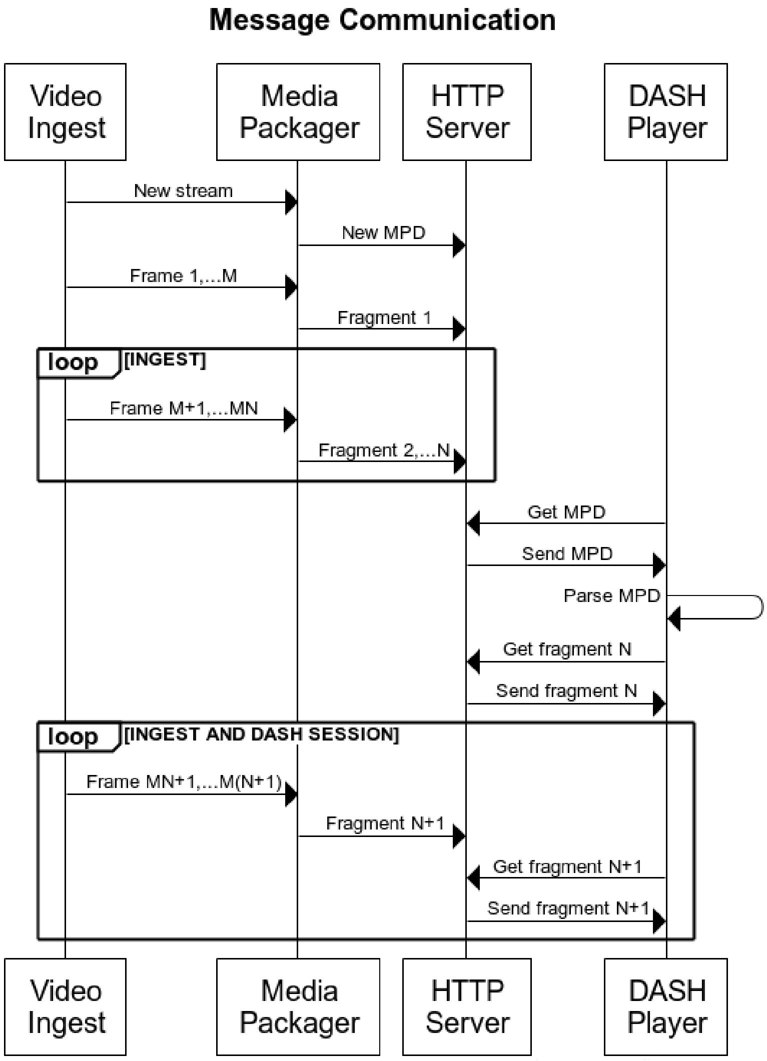


Figure 3.7: Sequence diagram.

3.3.3.2 QoE Metrics

The measurements of the implemented solution aim to identify the effects on QoE. From the work of Claeys et al. [350], the QoE is related to objective metrics such as frequency and duration of freezes that we can measure directly introducing some probes into the player while playing the content. We consider that the playback freezes when the internal buffer of the player goes empty and defines the duration of the freeze the time between the buffer goes empty and it starts to refill.

Moreover, Hossfeld et al. [362] investigates the effects of playout delay on the user and concludes that the user's satisfaction decreases while the playout delay increases. The playout is the elapsed time between the moment the user pushes the play button and the first frame is displayed on the screen. Anyway, in case of low latency, the playout depends also on content generation since the player needs to synchronize with sender. Consequently, in case of low latency steaming, it is more useful to measure the end-to-end latency of the system. From the work of Essaili et al. [346], the latency is the elapsed time between the frame ingest at the Media Packager (T$_{in}$) and the visualization time on the player screen (T$_{out}$), it can be express through the Expression 3.3.

$$Latency = T_{in} - T_{out} = T_{enc} + d_F + T_{fetch} + T_{dec} \tag{3.3}$$

The latency depends on processing time at Media Packager (T$_{enc}$), the fragment duration (d$_F$), the time for fetching the fragment from HTTP Server (T$_{fetch}$) and the decoding time at the player (T$_{dec}$).

We evaluate the latency of the end-to-end system by summing up all the components which appear in the Equation (3.3). Since in case of live streaming the Media Packager should work on real-time, T$_{enc}$ is inversely proportional to the input framerate. Accordingly, T$_{enc}$ is considered a fixed value. In the next section the remaining values of Equation (3.3) are evaluated.

3.3.4 Results

To test the implemented end-to-end Chunked CMAF solution, a live MPEG-DASH dataset using Big Buck Bunny test sequence was employed. Its raw version is provided by Xiph.Org Foundation [357]. The raw video was encoded in H264/AVC format

(ISO/IEC23008-2:2015). Then, it is multiplexed in ISO MPEG4 fragments (ISO / IEC 14496-12 - MPEG-4 Part 12) and split into segments. We experiment with the different representation levels shown in Table 3.3.

Table 3.3: Set of MPEG-DASH representations employed in the experiments.

Index	bitrate	resolution	framerate
1	1200kbps	352x288	30fps
2	1600kbps	640x360	30fps
3	2250kbps	960x540	30fps
4	2000kbps	704x576	30fps
5	4500kbps	1280x720	30fps
6	8000kbps	1920x1080	30fps

Moreover, the tests were carried out using different fragment configurations settings while generating each representation level to compare a legacy MPEG-DASH live stream and a Chunked CMAF enabled one. In Table 3.4 the employed fragment configurations are shown. The chosen duration for the segments along the tests is fixed to 2 seconds. This is a widely used value for legacy MPEG-DASH live streams, while the fragment duration employed is set to 33 ms, 100 ms or 167 ms for the Chunked CMAF live streams. These values correspond to fragments containing a GOP with 1, 3 or 5 frames, respectively.

Table 3.4: Tested fragment configuration

ID	Frames per fragment	Fragment duration (d$_F$) (ms)
F_1	1	33
F_3	3	100
F_5	5	167
S (Legacy)	60	2000

The experimental setup employed for the executed tests is presented in Figure 3.8. The overall setup comprises the following nodes:

- Server: this node is in charge of creating and distributing the content, i.e. it runs both the Media Packager and the HTTP Server. It creates the live stream and serves it to the client when required by the client itself.

- Wireless access point: to provide wireless capabilities, an access point is used, which provides a wireless local area network (WLAN) using 2.4 GHz band. The only role of the access point consists in forwarding all the incoming traffic on both sides (server and player).
- Player: a wireless network node connected to the access point, which is running the DASH Players. It uses probes in order to collect network information and player internal status.

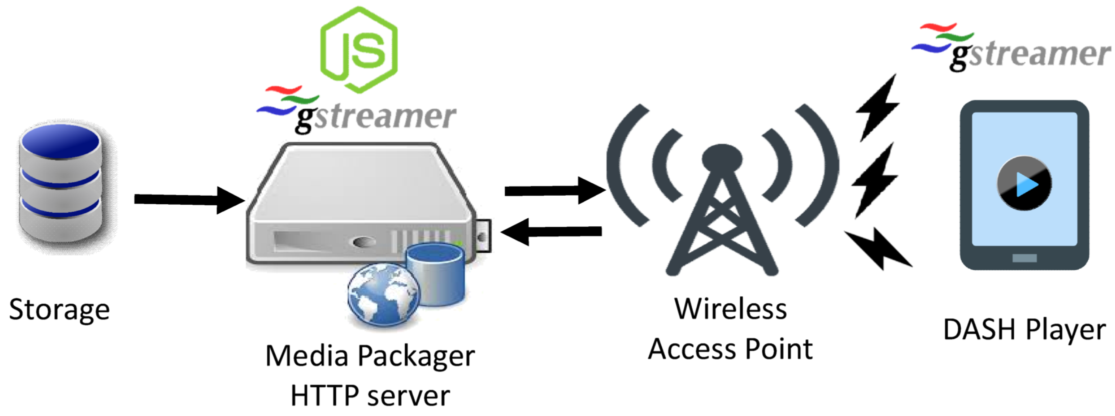


Figure 3.8: Experimental setup.

The Table 3.5 shows the results for the different fragment configurations and the employed representations in terms of the number of freezes and their average duration, and the overall latency according to the Equation (3.3). These parameters are the common factors employed by [350, 362] for the assessment of MOS metrics.

It becomes clear that the reduction of the fragment duration means a reduction in latency, but it also increases the number of freezes as the network performance is not enough to deliver the fragments in time. The duration of each freeze is not related to the fragment duration as the freezes spans a duration between 300 and 400 milliseconds independently of the fragment duration. Thus, it looks like an intrinsic limit of the wireless setup. The configuration S (legacy) is the only one which is not affected by the freezes, except for the highest representation level. In any case, even when higher network resources are needed, the playout buffer is big enough to shield against any network fluctuation. As expected, the configuration F1 (1 frame per fragment) provides the lowest latency. However, the latency score is not exactly proportional to the GOP

Table 3.5: Number of freezes (F_{Nb}), average freeze duration (F_{avg}) for each fragment configuration.

Conf. ID	Rep. index	F_{Nb}	F_{avg} (ms)	Latency (ms)	Conf. ID	Rep. index	F_{Nb}	F_{avg} (ms)	Latency (ms)
F_1	1	34	423	117	F_3	1	9	300	184
F_1	2	35	410	124	F_3	2	3	492	187
F_1	3	38	401	116	F_3	3	4	332	186
F_1	4	30	466	117	F_3	4	9	294	186
F_1	5	31	434	120	F_3	5	6	346	198
F_1	6	13	445	126	F_3	6	11	401	222

Conf. ID	Rep. index	F_{Nb}	F_{avg} (ms)	Latency (ms)	Conf. ID	Rep. index	F_{Nb}	F_{avg} (ms)	Latency (ms)
F_5	1	4	452	249	S	1	0	-	2196
F_5	2	14	312	256	S	2	0	-	2231
F_5	3	5	380	262	S	3	0	-	2288
F_5	4	5	493	259	S	4	0	-	2261
F_5	5	12	432	285	S	5	0	-	2482
F_5	6	13	430	317	S	6	2	463	2196

size since the latency reduction means -38% compared to F3 (3 frames per segment) while theoretically latency should decrease 3 times. This effect is mainly produced by two factors. First, the increasing HTTP overheads when the request/response speed and volume is very high. Second, the higher data rates to be transferred due to the utilization of more keyframes meaning high bitrates and lower compression efficiency as a GOP needs to start with a keyframe to allow instant consumption of a live stream. Moreover, the number of freezes for configuration F1 is three times compared to F3 and F5 (5 frames per fragment), which means that, the maximum technical reduction of latency may significantly damage the user's QoE with higher freezes along streaming sessions.

The results in terms of number of freezes along the video playback are presented in Figure 3.9, comparing the occurrence for the different representations and fragment configurations. It is visually evident that the number of freezes is lower as the fragment duration is higher. Finally, the configuration F3 and F5 present almost the same number of freezes and duration and then F3 is preferable in order to reduce latency.

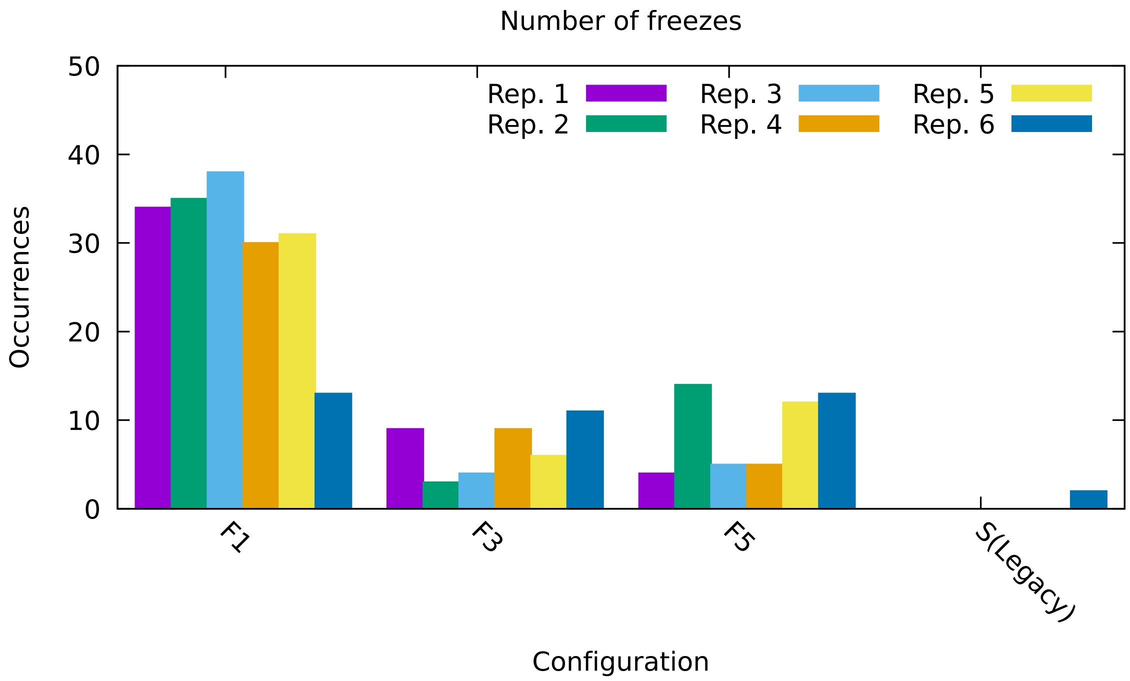


Figure 3.9: Number of freezes.

3.3.5 Conclusions and Future Work

This paper proposes an end-to-end system for delivery contents though a Chunked CMAF enabled MPEG-DASH live stream which aims to reduce latency, while trying to preserve major parameters of user's QoE.

The proposed solution has been integrated with Open Source MPEG-DASH framework and tested by performing experiments though a real testbed. The target of the experiments is the evaluation of the effects on user's QoE while tuning GOP and fragment duration during the Chunked CMAF packetizing in order to vary the latency of the system.

The results show that DASH players gain lower latency in any of the Chunked CMAF configuration with respects to a legacy solution but when using an aggressive configuration with a small GOP size and fragment duration the playback is frequently affected by freezes which reduce the QoE. So, to balance the latency and QoE trade-off a more conservative configuration of Chunked CMAF is suggested.

CHAPTER

4

Network performance forecasts for content delivery

4.1 Context

In the video streaming context, caching is a fundamental mechanism that aims to prevent negative effects on the QoS/QoE caused by network impairments. A CDN is a widely employed solution to cache and deliver video streams. Furthermore, it is becoming usual for CPs to employ alternative CDNs from different vendors or geographic locations to provide a more reliable service. However, the typical multi-CDN strategy is limited to selecting the CDN to be used at the start of the media session, maintaining it throughout the content playback. Then, the selected CDN is kept along all the streaming session. Moving to more dynamic solutions, that enable to switch between different CDNs when the streaming sessions are ongoing, opens lots of possibilities for optimization. Moreover, employing times series analysis to forecast network performance enables to perform proactive CDN selection and to consider the trade-off between performance (QoS) and costs (Operational Expenditure or OPEX).

Section 4.2 proposes to optimize the employed CDN resources by reducing their usage to the effectively necessary moments, when delivering MPEG-DASH streams. The objective is to avoid over-provisioning of CDN resources, as it affects CP's OPEX. The proposed solution, called intelligent network flow (INFLOW), consists in a multi-CDN strategy designed to optimize CDNs utilization and reduce the resultant business costs for it. It exploits periodical MPEG-DASH media presentation description (MPD) updates to apply dynamic switching among the available CDNs at the players in a standard compliant manner. The MPD with the appropriate CDN endpoint is served by the INFLOW Media Server, which works jointly with the INFLOW Forecast Service. The INFLOW Forecast Service provides network metrics predictions based on a Long Short-Term Memory (LSTM) network, a kind of Recurrent Neural Network (RNN), when fed with the historical values of network metrics. The integration of the Forecast Server into the delivery chain allows the Media Server to serve an MPD containing the *BaseURL* of the CDN, which matches target QoS and CP's business requirements. Thus, INFLOW allows for proactive and cost-effective video streaming delivery. This paper comprises the following relevant contributions:

- Exploitation of network performance metrics and MPD information to apply common decisions to ongoing streaming sessions. Captured network metrics are employed to forecast CDN serving capacity (throughput) and then to select a CDN only if it would ensure the viability to serve the content at a representation bitrate from the available ones in the MPD that matches with the target minimum QoS.

- A dynamic approach switching from a CDN server to another depending on the performed predictions at any time. Thus, in contrast to current solutions, a streaming session is not served from a single CDN provider.

- Practical application of a forecast model. The literature proposing a forecast model for QoS network metrics is usually limited to theoretical analysis and simulations where the predictions are not turned into video streaming actions. On the contrary, in this work the predictions are effectively employed to switch the players among the available CDN servers, then proactively acting on the delivery.

- Business constraints are considered for the CDN selection. Metrics for both the OPEX and the QoS have been considered in the algorithm which selects the ideal CDN to be employed. Thus, this sophisticated approach favors the dynamic utilization of a CDN marketplace to deal with cost-effective trade-offs. The efficient utilization results in OPEX reduction, while keeping the QoS, which is a major concern for practical deployments in real-world streaming services.

- The evaluation includes a comparison with other CDN selection strategies in terms of QoS metrics and business cost.

The proposed solution has been implemented and validated in a distributed and heterogeneous testbed employing real network nodes and including both wired and wireless nodes. The wireless nodes were connected through a real Long-Term Evolution (LTE) network deployed with Software Defined Radio (SDR) equipment and OpenAirInterface (OAI) open-source software. The traffic demand on video players was generated according to a probability distribution widely employed in the literature. The results highlight the advantages of INFLOW for reducing the overall usage time of the available CDNs, while guaranteeing a minimum level of network bandwidth to every player.

4.2 **Predictive CDN selection for video delivery based on LSTM network performance forecasts and cost-effective trade-offs**

- **Title:** Predictive CDN selection for video delivery based on LSTM network performance forecasts and cost-effective trade-offs
- **Authors:** Roberto Viola, Ángel Martín, Javier Morgade, Stefano Masneri, Mikel Zorrilla, Pablo Angueira and Jon Montalbán
- **Journal:** IEEE Transactions on Broadcasting
- **Publisher:** IEEE
- **Year:** 2020
- **DOI:** `10.1109/TBC.2020.3031724`

Abstract: Owing to increasing consumption of video streams and demand for higher quality content and more advanced displays, future telecommunication networks are expected to outperform current networks in terms of key performance indicators (KPIs). Currently, content delivery networks (CDNs) are used to enhance media availability and delivery performance across the Internet in a cost-effective manner. The proliferation of CDN vendors and business models allows the content provider (CP) to use multiple CDN providers simultaneously. However, extreme concurrency dynamics can affect CDN capacity, causing performance degradation and outages, while overestimated demand affects costs. 5G standardization communities envision advanced network functions executing video analytics to enhance or boost media services. Network accelerators are required to enforce CDN resilience and efficient utilization of CDN assets. In this regard, this study investigates a cost-effective service to dynamically select the CDN for each session and video segment at the Media Server, without any modification to the video streaming pipeline being required. This service performs time series forecasts by employing a Long Short-Term Memory (LSTM) network to process real time measurements coming from connected video players. This service also ensures reliable and cost-effective content delivery through proactive selection of the CDN that fits with performance and business constraints. To this end, the proposed service predicts the number of players that can be served by each CDN at each time; then, it switches the required players between CDNs to keep the (Quality of Service) QoS rates or to reduce the CP's operational expenditure (OPEX). The proposed solution is evaluated by a real server, CDNs, and players and delivering dynamic adaptive streaming over HTTP (MPEG-DASH), where clients are notified to switch to another CDN through a standard MPEG-DASH media presentation description (MPD) update mechanism.

Keywords: Content Delivery Network, MPEG-DASH, Operational Expenditure, Quality of Service.

4.2.1 Introduction

In the last few years, the demand for video content across the Internet has constantly increased. Video streams from professional applications, such as Industrial Internet of Things (IIoT), medical equipment, connected and autonomous cars, and from domestic services such as gaming, virtual reality, augmented reality, video over IP (VoIP) sports services, and over-the-top (OTT) platforms are flooding networks with real-time data intensive sessions.

This evolution of Internet traffic makes evident the severity of the network's capacity to guarantee a certain quality of service (QoS) for the video applications. To prevent network flooding and to make video delivery more efficient, content delivery networks

(CDNs) employ geographically distributed and cost-effective infrastructures as a service (IaaS) to enhance media availability and delivery performance across the Internet. This hierarchical system that caches and stores video streams fosters efficiency while geographical locations track human daytime life cycles, which have a close relation with local content demands.

Furthermore, the current video traffic crosses networks working on a best-effort basis where the delivery time of network packets is not guaranteed. Thus, it may cause stalls during playback on player devices, damaging the quality of experience (QoE). The popularity of video streaming services over the Internet pushed video industry-Moving Picture Experts Group (MPEG)-and standardization bodies to create new formats which enable adaptive streaming over the already existing Hypertext Transfer Protocol (HTTP) infrastructures. Thus, they allow the player devices to adapt the content representation to the specific device capabilities (resolution, codecs, etc.) and the changeable network connectivity.

Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [23], which was designed to mitigate problems due to fluctuations on best-effort networks, is the solution adopted by the video industry [366]. In fact, MPEG-DASH enables pull-based streaming [367] and allows for scalable distribution as it has a CDN-ready design [368] that enables the exploitation of existing HTTP caching infrastructures without modifications. To this end, the MPEG-DASH pipeline splits the video content into segments of fixed duration, usually between 2 and 10 seconds; then, it encodes them at different representation levels with a nominal resolution and bitrate. Thus, for each segment request, the player can switch from one representation to another depending on the assessed network status.

Nevertheless, the MPEG-DASH client-driven approach presents some drawbacks. First, each player is not aware of the existence of the others, leading to high network dynamics as the content download is not coordinated. Second, each player strives to achieve optimized individual quality, which may lead to unfairness when a congested connection path is shared [369]. Thus, it is challenging for a content provider (CP) to ensure a certain level of quality to end users, who are accessing large volumes of content through the same access point and competing for the available bandwidth independently. Here, some issues, such as initial buffering delay, temporal interruptions, unsteady video resolution, and bitrate changes, may damage the QoE [21].

Currently, CDNs are used to enhance media availability and delivery performance across the Internet. The proliferation of CDN vendors and business models allows the CP to use multiple CDN providers simultaneously [28], [29], [30]. However, extreme concurrency dynamics can affect CDN capacity, causing performance degradation and outages, while overestimated demand affects costs, thereby increasing the operational expenditure (OPEX) of the CP [370].

Upcoming 5G networks will need advanced and intelligent mechanisms to dynamically deliver each data flow according to the required service level agreement (SLA) and considering performance costs trade-offs. This concept is where the approach proposed by this paper takes place, fusing network characteristics and media service options to match user satisfaction and business policies.

4.2.1.1 **Contribution**

This work proposes a novel solution called intelligent network flow (INFLOW) for CDN selection in a multi-CDN delivery environment. It exploits periodical MPEG-DASH media presentation description (MPD) updates to apply dynamic switching among the available CDNs at the video players in a standard compliant manner. The MPD with the appropriate CDN endpoint is served by the INFLOW Media Server, which works jointly with the INFLOW Forecast Service. The INFLOW Forecast Service provides network metrics predictions based on a Long Short-Term Memory (LSTM) network, a kind of Recurrent Neural Network (RNN), when fed with the historical values of network metrics. The integration of the Forecast Server into the delivery chain allows the Media Server to serve an MPD containing the *BaseURL* of the CDN, which fits target QoS and CP's business requirements. Thus, INFLOW allows for proactive and cost-effective video streaming delivery. The proposed solution comprises the following relevant contributions:

- Exploitation of network performance metrics and MPD information to apply common decisions to ongoing streaming sessions. Captured network metrics are employed to forecast CDN server capacity, then select a CDN only if it would guarantee the viability to serve the content at a minimum representation bitrate from the available ones in the MPD.

- Dynamic CDN server switching. We employ a dynamic approach switching from a CDN server to another depending on the performed predictions at any time. Thus, in contrast to current solutions, a streaming session is not served from a single CDN provider.

- Practical application of a forecast model. The literature proposing a forecast model for QoS network metrics is usually limited to theoretical analysis and simulations, and the predictions are not turned into video streaming actions. On the contrary, we exploit the predictions to switch the players among the available CDN servers, then proactively act on the delivery.

- Business constraints are considered for the CDN selection. We include metrics for both the OPEX and the QoS in the algorithm which selects the ideal CDN to be employed. Thus, this sophisticated approach favours the dynamic utilisation of a CDN marketplace to deal with cost-effective trade-offs. OPEX reduction, while keeping the QoS, is a major concern for practical deployments in real-world streaming services.

To achieve the above contributions, we develop a Forecast Service and a Media Server as complementary parts of the proposed INFLOW solution. Forecast Service executes an LSTM network performing real-time predictions of the QoS metrics. Media Server updates the MPD according to the network metrics predictions and CP's business rules and serves it to the clients. The solution was integrated and tested in a real setup employing a multi-modal testbed including both wired and wireless nodes. The wireless nodes were connected through 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). The traffic demand on video players was generated according to a probability distribution widely employed in the literature.

The paper is structured as follows. First, section 4.2.2 reviews related work in the field of video delivery based on CDN performance and network traffic generation and forecast. Then, section 4.2.3 introduces the proposed INFLOW server, a novel media server equipped with a forecast service that tunes the delivery and applies a CDN selection mechanism based on QoS metrics and business rules, as the main focus of the article. Section 4.2.4 describes the implemented setup using a real testbed, while section 4.2.5 presents the results of the validation experiments. Finally, we assert our conclusions and future work in section 4.2.6.

4.2.2 **Related Work**

4.2.2.1 **CDN resource selection**

A CDN is a network function widely employed to improve content delivery by means of cloud service provisioning cache features. Fueled by the CDN vendor proliferation, media platforms exploit multi-CDN strategies to obtain more reliable content delivery that provides a steadier QoS and higher customer satisfaction. Nevertheless, the CDN selection criteria can be different for any CP.

A widely employed solution applies a static selection made by the media server when a new streaming session starts. This is used by Netflix [28] and Hulu [29], with big similarities [30]. They use three different CDN vendors mapping CDNs to the location of client device or to a subscriber. Moreover, they evidence that, the selected CDN is fixed during the streaming session even when the QoS degrades. Thus, providers are more prone to lower the representation bitrate instead of operating alternative CDNs. Hence, the authors conjecture that CDN selection is most likely based on business policies.

However, Netflix has changed its strategy over the years, and nowadays it uses its own CDN, which is called Open Connect [215]. Open Connect can be run inside the ISP infrastructure so that a better QoS can be achieved as the content is closer to the user. Netflix' solution is not that different from the open CDN architecture proposed by [371]. The authors propose collaborative participation of CPs and ISPs. On one hand, cost reductions are realized as the ISP acts as a CDN. On the other hand, the ISP provides better performance to the clients and reduces traffic as the content is already present in its network infrastructure.

The awareness of end-to-end QoS metrics measured by the client can make the difference when the employed CDN is dynamically chosen by the clients. The authors of [212] propose a client-side CDN selection. As a drawback, client-side strategies do not produce a coordinated decision as each client analyses the network performance of each CDN independently, introducing bias and communication overheads. Hence, a client-side CDN selection is not an optimal solution.

An intermediate solution consists of Domain Name System (DNS) resolution. Here, the DNS server can resolve a fixed hostname owned by the CP into different IP addresses of several CDNs. Depending on the DNS resolution, the client is directed to an appropriate CDN. The YouTube DNS-based solution is shown in [213]. YouTube goes further as it allows for the use of a hybrid DNS and application-level CDN selection. First, the DNS redirects the client to a server. Then, the server accepts or reject the client depending on the workload. If the client is refused, the DNS redirects the client to another server.

In [214], the effects of DNS resolution for CDN selection are further studied. The authors conclude that, depending on the DNS service provider, Akamai and Google CDN servers are chosen differently. Consequently, CDN performance highly depends on the load balancing rules of the DNS server. Here, a suboptimal CDN server selection leads to a higher round-trip delay time (RTT). To solve this problem, the authors propose a DNS-proxy running on the client. This proxy forwards the DNS requests to different DNS servers; then, it compares the responses to identify the best performing CDN server. However, these solutions are loosely coupled from media player requests slots applying balancing policies independently of media player timing. Instead, out approach is triggered by the requests with the most recent available information.

CDN Brokering [231] is the ability to redirect clients dynamically among two or more CDNs. CDN brokers collect and analyze the performance metrics of the available CDNs to select the one that performs best. Their work, in contrast to traditional multi-CDN strategies, is not limited to the selection of the initial CDN for each client. This solution also moves clients between CDNs when performance degradation is detected in real time. Thus, the CDN is dynamically and seamlessly changed. As an example, the European Broadcasting Union (EBU) proposed the EBU Flow Multi-CDN [219], which consists of a CDN switching service that selects the optimal CDN at any time. Similar approaches are provided by Citrix [220] and LightFlow [221]. Thus, these solutions are usually provided by intermediaries, federating infrastructures from different vendors. However, our approach keeps the control to the media service manager able to dynamically change the business policy or tune the cost function.

Edge computing systems promise a revolution on smart delivery of media traffic fueled by Multi-access Edge Computing (MEC) [14] architectures from 5G. Thus, new solutions for improving multi-CDN delivery involve investigating MEC services. In [222] a MEC proxy is proposed. The proxy can retrieve video streaming metrics of video players at the access point transparently and CDNs performance metrics from the wired link. Compared to a pure client-decision, a MEC proxy can evaluate the performance of each CDN just once and apply conclusions to other sessions (independently from the number of connected players). Moreover, it empowers the delivery through a local edge cache. This feature guarantees traffic reduction compared to server-side CDN selection as recurrent content can be downloaded and cached once for every client. In [223], a similar solution for a MEC-based cache is proposed. However, till the moment the edge systems realize, current CDN infrastructures makes the difference, and universal solutions that dynamically manage balancing are required.

In [217], a prototype of CDN and ISP collaboration is proposed. The ISP provides the CDN provider with services that allow the CDN provider to retrieve geographical user distribution and allocate server resources inside the ISP's network topology. The authors of [218] propose a similar solution without the binding to allocate resources in the ISP's infrastructure. In this case, a redirection center inside the ISP's network intercepts the client's requests and selects the appropriate CDN. This process is transparent to the client as the redirection center stores the content in a CDN surrogate and instructs an OpenFlow controller to migrate the traffic to a CDN surrogate.

Contrary to those approaches provisioning or balancing serving resources, other works focus on the selection of the appropriate bitrate to avoid congestion for a static number of servers [372] or network assets [202]. To this end, the MPD is parsed and heavier options are cropped. As implemented in these works, our approach employs a compliant MPD update mechanism. But, in our case, it is exploited to dynamically manage the CDN resources at any moment depending on network performance forecasts and CP business rules. The CP can tune the media server, thereby influencing CDN selection when the video player requests an MPD update.

4.2.2.2 **Time series for network traffic forecast**

The goal of applying time series analysis to network traffic data is to forecast future conditions to take actions proactively when actuation performance or cost policies are satisfied. These techniques allow network management systems to prevent network under-performance and outages, thereby addressing network congestion preemptively.

Table 4.1: Forecast models comparison

Model	Approach	Number of variables	Internal parameters
ARIMA	statistical	univariate	a-priori (regression, integration and moving average parameters)
Exponential smoothing	statistical	univariate	a-priori (smoothing factor)
SETARMA	statistical	univariate	a-priori (regression, moving average and threshold delay parameters)
GARCH	statistical	univariate	a-priori (regression and lag length parameters)
Feed-forward NN	neural network	multivariate	trained (weight and bias)
RNN	neural network	multivariate	trained (input, output and forget factors)
LSTM	neural network	multivariate	trained (input, output and forget factors)

The auto-regressive integrated moving average (ARIMA) is employed in [133] to predict the workload of cloud services. It employs historical records of observed requests to predict the volume of requests for the following time interval. The results reveal that the model can obtain the general trend, but it lacks the ability to accurately and timely track traffic peaks. The authors of [134] apply both ARIMA and exponential smoothing models to predict throughput in an LTE network. The two models are complementary, with ARIMA outperforming the exponential smoothing models on weekdays and the exponential smoothing models outperforming ARIMA on weekends.

The authors of [135] and [136] found limitations in ARIMA while modelling QoS attributes. QoS attributes such as bandwidth or latency have nonlinear behaviors that do not fit the linear assumption of the ARIMA model. They overcame this by introducing hybrid linear and non-linear models. The linear model was represented by the ARIMA model. For the non-linear model, [135] used the self-exciting threshold autoregressive moving average (SETARMA) model, while [136] employed a generalized autoregressive conditional heteroscedastic (GARCH) model. In both cases, the proposed solution outperforms a standalone ARIMA model in forecasting the time between QoS violations.

In recent years, machine learning (ML)-based techniques for time series prediction have exhibited satisfactory performances. Specifically, neural networks (NNs) are gaining adoption in the generation of time series models. The authors of [144] propose a feed-forward NN for predicting the execution time of services while varying the number of requesters. In [146], a recurrent NN (RNN) is employed to forecast the end-to-end delay from RTT metrics.

In [147], an LSTM model, a particular type of RNN, was proposed. The authors employed a multivariate time series model where data input was probed from downlink control information (DCI) messages, such as resource blocks, transport block size, and scheduling information.

Table 4.1 shows the main differences between the employed techniques for time series forecasting. Clearly, NN-based approaches have the advantage to employ several variables as input and/or output of the models, while statistical ones are limited to one. Moreover, statistical approaches need a-priori evaluation of internal parameters, while NN-based ones are trained through a dataset of previous collected metrics. Here, the parameters for statistical approaches are intended for the whole model, while for NN-based ones, parameters must be trained for each internal cell.

From the available algorithms to analyze and predict time series, we employ LSTM network model as it satisfies two requirements. First, in terms of accuracy, statistical solutions (ARIMA and its derivatives) are slow when tracking quick fluctuations in time series as they tend to concentrate on the average value of the past observed values, as revealed in [373]. Second, in terms of multivariate time series, statistical solutions only can predict one variable. Thus, ARIMA would require separate models for forecasting both latency and bandwidth. On the contrary, LSTM is ready to process multivariate time series. The authors of [155] and [156] revealed that, the higher the number of input variables, the better the traffic predictions of LSTM when compared to ARIMA.

4.2.3 **INFLOW solution**

4.2.3.1 **System architecture**

To achieve reliable and cost-effective video delivery, we propose the inclusion of our INFLOW solution in the video delivery chain. The overall scenario of the solution is depicted in Figure 4.1. The INFLOW solution is composed of two components:

- INFLOW Forecast Service: it receives the QoS performance metrics from the video players and processes them to predict the QoS values in the future.

- INFLOW Media Server: it exploits the results provided by the Forecast Service by combining them with the CP's business rules to select the appropriate CDN for each client.

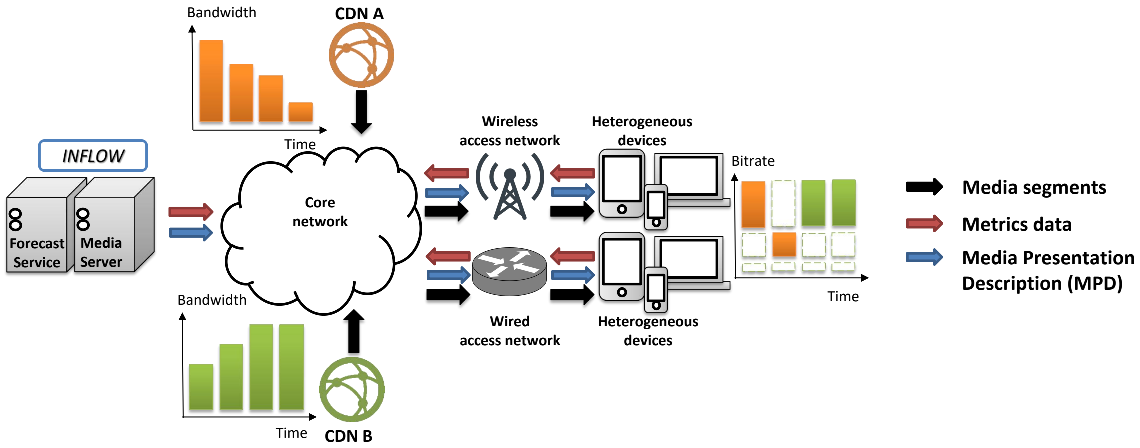


Figure 4.1: General scenario of the proposed solution.

Owing to the utilization of MPEG-DASH, INFLOW includes the following features:

- Scalability. New CDNs can be easily managed by adding them to the initial MPD.
- Real-time migration of video players to CDN providers. Supported by standard-compliant MPEG-DASH MPD update mechanism manages the utilization of a CDN by the video player according to the gathered metrics and business policies.
- MPDs can be parsed and processed even when the content is encrypted with the MPEG-DASH Common Encryption Scheme (CENC) [374]. The CENC format encrypts the media segments indexed in the MPD, but the MPD is not encrypted.

The sequence diagram of the exchanged messages is depicted in Figure 4.2. Media segments are stored at different CDNs, while the MPD is served by the INFLOW Media Server. It is important that the media server uses a dynamic MPD as it forces the player to periodically update, overwriting the Minimum Update Period attribute from the MPEG-DASH standard [360]. On the other side, each video player downloads the initial MPD and starts requesting for segments from the initial CDN. A client-side adaptation mechanism constantly monitors the statistics of the downloaded segments to select a representation level among those available that fits with the experienced network performance. Thus, the video player aims to prevent stalls during playback. Typical monitored metrics are the network bandwidth and latency, which provide a direct measure of the QoS experienced by the client. Moreover, these measurements are sent to the INFLOW forecast service. Thus, video player should support a mechanism for sending feedback to the forecast service, such as Server and Network-assisted DASH (SAND) standard [375]. Finally, INFLOW forecast service stores the measurements and uses them to predict the future values.

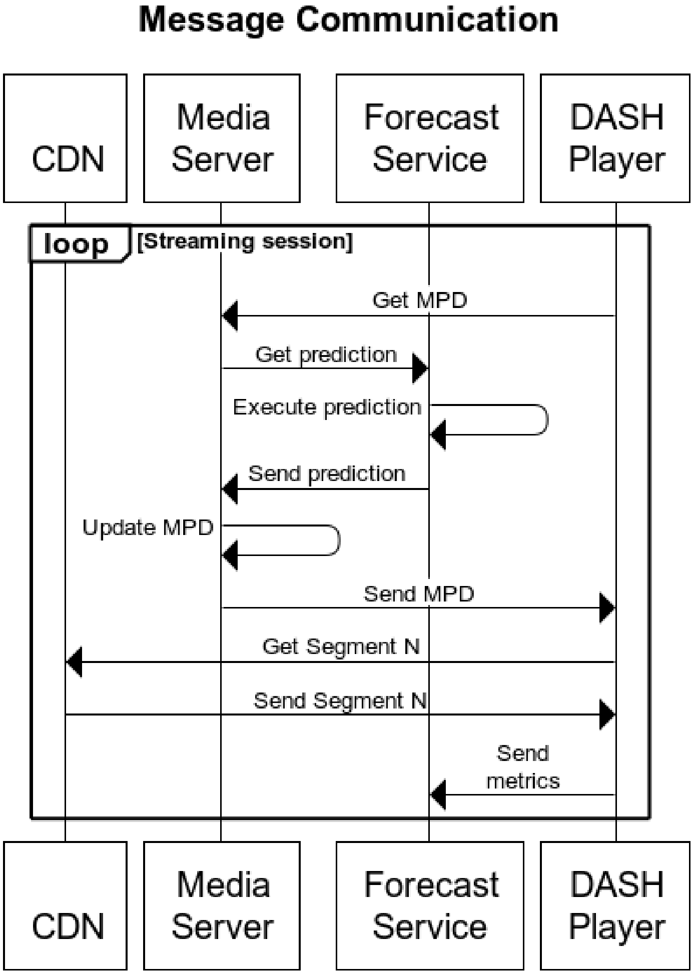


Figure 4.2: Sequence diagram of the INFLOW solution for video delivery.

The MPD served by the media server is fully conditioned by the prediction of the forecast service. Every time a player requests an updated version of the MPD, the media server retrieves a prediction from the forecast service and decides to serve the current MPD or to change the CDN included. Therefore, the forecast service does not apply any QoS or business rules—it simply processes the information provided by the players. The QoS and business-based decisions are made by the media server, and this decision process is executed in real-time as the predictions are rendered out of date after the predicted interval, leading to a new decision. Therefore, the shorter is the segment duration, the more immediate is the forecast validity and the prompter is the MPD update.

The QoS forecasts serve two roles. First, the INFLOW Media Server can select an appropriate CDN to shield from CDN service degradation and outages based on the most recent detected performance. To this end, the server receives alternative CDNs from the initial MPD and replaces the *BaseURL* tag in the MPD with another CDN endpoint to migrate a client. Second, the media server can count the video sessions served by each CDN. On top of this information, the media server can apply cost-effective policies, allocating extra CDN resources to enforce QoS or retiring CDN assets to reduce the number of employed CDN servers. Thus, the media service can manipulate the OPEX ranges to meet the business model.

In the following section, we describe separately the two components of the INFLOW solution.

4.2.3.2 **INFLOW Forecast Service**

The INFLOW forecast service is in charge of collecting network metrics probed and sent by the video players and processing them to predict the values in future slots. The most recent metrics are processed while older ones are discarded using a sliding window mechanism. The decision program of the forecast service is shown in Algorithm 2.

The input are the last N historical values of network bandwidth and latency measured and reported by the players for a specific CDN (CDN^k). The samples comprised in the most recent period are captured during last segment download. The video player requests segments in a regular pace to fill its buffer, according to the media duration of the segment. In total, the algorithm processes two variables taken in N time instants. Every new sample should be taken at a fixed temporal distance from the previous one. Nevertheless, this assumption of equal distance among the samples is not guaranteed as players usually run asynchronously, and then the reports are sent in a random time inside a segment slot. To overcome this problem, the INFLOW forecast server employs a

Algorithm 2 INFLOW Forecast Service

```
function PREDICTMETRICS($\overline{bw^k_{t-1}}$, $\overline{l^k_{t-1}}$, $CDN^k$)        ▷ for each CDN infrastructure
Input: $\overline{bw^k_{t-1}}$        ▷ bandwidth mean for the most recent period @$CDN^k$
Input: $\overline{l^k_{t-1}}$        ▷ latency mean for the most recent period @$CDN^k$
Output: $\widehat{bw^k_t}$        ▷ bandwidth prediction
Output: $\widehat{l^k_t}$        ▷ latency prediction
    $\{bw^k\} = \{\overline{bw^k_{t-1}},...,\overline{bw^k_{t-N}}\}$        ▷ N bandwidth samples
    $\{l^k\} = \{\overline{l^k_{t-1}},...,\overline{l^k_{t-N}}\}$        ▷ N latency samples
    $\widehat{bw^k_t},\widehat{l^k_t} \leftarrow$ LSTM($\{bw^k\}$,$\{l^k\}$)        ▷ update forecast model $CDN^k$
```

mean value of samples within a second as the input of the algorithm ($\{\overline{bw^k_{t-1}},...,\overline{bw^k_{t-N}}\}$ and $\{\overline{l^k_{t-1}},...,\overline{l^k_{t-N}}\}$). The input is processed through the LSTM network to predict the values in the next second ($\widehat{bw^k_t}$ and $\widehat{l^k_t}$). The predicted values are the output of the algorithm.

It is important to underline that the benefits of an LSTM network over statistical approaches are two-fold. First, the LSTM network performs better when time series includes quick fluctuations [373]. Second, it is valid for multivariate time series, such as bandwidth and latency, where statistical methods fail to simultaneously process several components [155], [156].

4.2.3.3 **INFLOW Media Server**

The media server must serve the MPD of the video players to provide awareness on available representations, content formats and metadata, and CDN endpoints. In our case, as we were interested in CDN localization, the served MPD could include one or more *BaseURL* tags containing the URLs of the CDN servers. In cases with only one *BaseURL*, the client is forced to use it.

The INFLOW media server stores an initial MPD containing different *BaseURL* tags and modifies it while excluding CDN alternatives to force a CDN to perform according to the algorithm outcomes, which exploits the predictions provided by the INFLOW forecast service. The decision program of the media server is shown in Algorithm 3.

Algorithm 3 INFLOW Media Server

```
  function UPDATEMPD($urlMPD$, $SLA$)    ▷ for each MPD request
Input: $urlMPD$    ▷ requested MPD
Input: $SLA$    ▷ applicable SLA
Output: $MPD$    ▷ updated MPD
      $MPD \leftarrow$ initial($urlMPD$)    ▷ requested MPD file
      $bw_{min} \leftarrow$ targetQoS($SLA$)    ▷ minimum bandwidth per player
      $d_s \leftarrow MPD$    ▷ segment duration
      $\{CDN_{list}\} \leftarrow MPD$    ▷ set of alternative CDNs
      for all $CDN^k \in \{CDN_{list}\}$ do    ▷ for each CDN
          $\overline{bw^k_{t-1}} \leftarrow$ mean($\{bw^k_{[t-1,t)}\}$)    ▷ average for most recent period @ $CDN^k$
          $\overline{l^k_{t-1}} \leftarrow$ mean($\{l^k_{[t-1,t)}\}$)    ▷ average for most recent period @ $CDN^k$
          $\widehat{bw^k_t}, \widehat{l^k_t} \leftarrow$ predictMetrics($\overline{bw^k_{t-1}}, \overline{l^k_{t-1}}, CDN^k$)
          $n^k \leftarrow$ sessions($CDN^k$)    ▷ total $CDN^k$ sessions
          $\widehat{n^k} \leftarrow$ policy($\widehat{bw^k_t}, \widehat{l^k_t}, n^k, bw_{min}, d_s, CDN^k$)    ▷ $CDN^k$ capacity
          if ($\widehat{n^k} > n^k$) then    ▷ $CDN^k$ admits more sessions
              $BaseURL \leftarrow$ URL($CDN^k$)    ▷ write $CDN^k$ URL
              $MPD \leftarrow$ update($MPD$,$BaseURL$)    ▷ update MPD
```

The algorithm takes an initial configuration of minimum bandwidth to be provided to the clients (bw_{min}) according to the SLA, and the initial MPD. From the MPD, it retrieves the segment duration (d_s) and a list of the CDNs ($\{CDN_{list}\}$). When an MPD request reaches the media server, it selects an appropriate CDN (CDN^k) from the CDN list. This list ($\{CDN_{list}\}$) is ordered in ascending order according to expenses. Thus, the media server first employs the affordable providers, migrating users to cheaper services when possible. The media server retrieves the prediction for each CDN from the forecast service ($\widehat{bw^k_t}$ and $\widehat{l^k_t}$) and stops if the expected capacity ($\widehat{n^k}$) is higher than the current ones (n^k). In other words, it selects the most affordable CDN that has the capacity to serve more players. The number of expected players is evaluated through the predictions and the initial configuration by means of Equation 4.1.

$$\widehat{n^k} = \frac{(d_s - \widehat{l_t^k}) * \widehat{bw_t^k} * n^k}{d_s * bw_{min}} \tag{4.1}$$

To be timely delivered, the theoretical maximum download time of a segment should be lower than the segment duration. Furthermore, a padding time must be considered to take into account the delay introduced by the network during the transmission. Consequently, the predicted latency ($\widehat{l_t^k}$) is used as a penalization factor to estimate the effective download time (d_s-$\widehat{l_t^k}$). Then, this value is multiplied by the predicted average bandwidth per video player ($\widehat{bw_t^k}$) to assess the average volume of data that each player can download. The total data capacity is obtained by multiplying the number of sessions in the CDN (n^k) by the average volume of data that each player can download. Finally, the overall traffic demand is divided by the amount of data that a player should download during a segment duration (d_s) according to the SLA using the minimum bandwidth provided (bw_{min}). The final value is the CDN capacity according to an SLA, which is indicative of the number of video streaming sessions that the CDN can serve ($\widehat{n^k}$).

Once a CDN is assigned to a session, the media server selects the *BaseURL* corresponding to the CDN and generates a new MPD by modifying the initial one. The order of the CDNs in the list is important as they are ranked depending on the cost. Thus, the media server chooses the first CDN that fits with the necessary resources; then, the most affordable ones are quickly booked to reduce CP's OPEX.

4.2.4 **Testbed setup**

To demonstrate the cost-effective advantages of the INFLOW approach in terms of QoS enforcement and CP's OPEX reduction, we deployed a heterogeneous and distributed setup employing both FED4FIRE+ facilities [376] and our facilities at Vicomtech (San Sebastián, Spain). Fed4FIRE+ is a Horizon 2020 project that provides open and accessible testbeds to support research and innovation initiatives in Europe. Among the available facilities, we employed NITOS's network infrastructure [377] at University of Thessaly's campus (Volos, Greece). NITOS provides heterogeneous testbeds to execute experiments on real wired and wireless networks.

We use D-DASH dataset and infrastructure [378], with Dynamic Adaptive Streaming over HTTP (DASH) standard content mirrored over different sites at different locations to perform CDN-based scientific evaluations. The dataset includes the Red Bull Playstreet video sequence, which is owned by Red Bull Media House and licensed for scientific purposes. This sequence is encoded for 17 video representations through advanced video coding (H.264/AVC) and 4 dual channel audio representations through advanced audio coding (AAC). Both audio and video are segmented with different segment lengths of 2, 4, 6, 10, and 15 seconds, and multiplexed in ISO MPEG4 files (ISO/IEC 14496-12 - MPEG-4 Part 12). For our experiments, we employed 2 seconds segments to focus on live video content where dense client cells and congestion of CDNs were likely. We did not modify the video representations; instead, we used the available representations in the dataset. The representations range from a resolution of 320 x 240 and 30 fps at 100 kbps to a resolution of 1920 x 1080 and 30 fps at 6000 kbps. As the client-side bitrate adaptation mechanism works on a best-effort basis and do not take care of the presence of other connected players, each player struggles to achieve the highest representation bitrate.

The final experimental setup comprises the following:

- 4 UE nodes: client nodes located at NITOS and running 100 DASH video players based on GStreamer multimedia framework [353]. They feature both Ethernet and LTE interfaces and are placed in the isolated environment of the NITOS indoor testbed where they form a grid topology.
- 1 eNodeB: USRP provided node performing eNodeB stack located at NITOS. It forwards the packets from the clients to the Access and Core Network.
- 1 EPC node: wired node close to the eNodeB that executes the EPC stack.
- 1 INFLOW Media Server: node at Vicomtech based on a virtual machine with 2 GB RAM and single-core CPU. It is provided with a public IP address to serve the MPD to the video players. It runs a Node.js [363] server application which applies QoS and CP's business rules when sending the MPD to the client.
- 1 INFLOW Forecast Service: node at Vicomtech based on a physical machine having 12 GB RAM and quad-core Intel i5 6500 CPU. To perform predictions, it

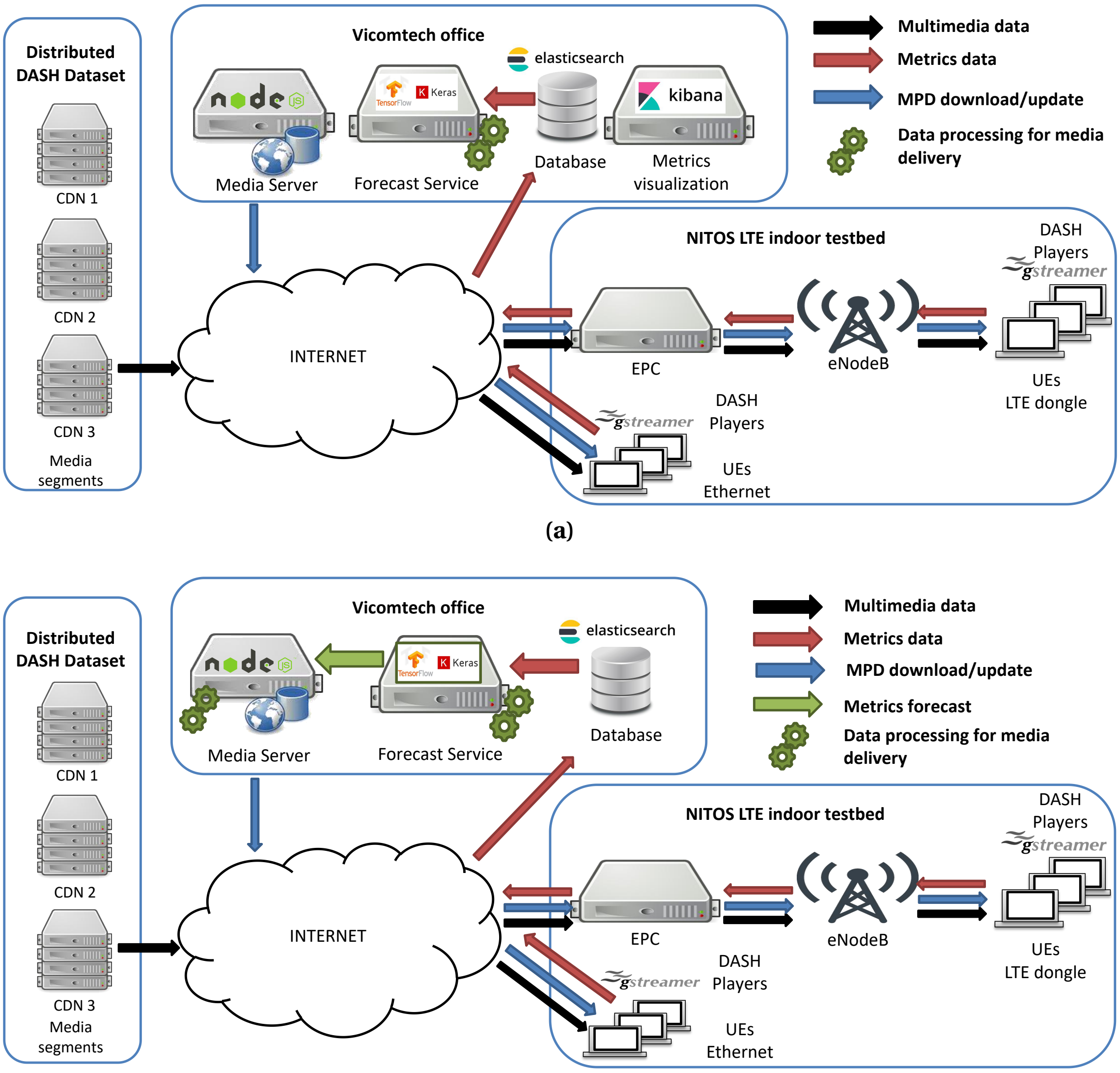
features NVIDIA GTX 1050 TI executing the LSTM model based on TensorFlow [379].

- 3 servers: they belong to the D-DASH dataset [378] providing alternative CDNs storing the media segments to perform CDN-based scientific evaluations. They are located at different sites with different nominal performances in terms of bandwidth and latency.

To distribute the video streaming sessions between the wired and LTE network interfaces, we considered the last Cisco report concluding that mobile traffic covers 9% of the total IP video traffic [358]. Hence, the experiment setup includes nine video players connected through the LTE interface and 91 players employing Ethernet interface. The use of different access networks is helpful for demonstrating its applicability in representative and multi-modal scenarios. Moreover, we modeled player inter-arrival rate and session duration according to [131], which provides an extensive analysis on user behavior while accessing streaming services. Thus, the inter-arrival time distribution is a modified version of the Poisson distribution, while the session duration follows the declared sections of 5 (37.44%), 10 (52.55%), or 25 min (75.25%).

During the experiment, a preliminary step was performed to generate a QoS performance metric dataset for training our LSTM model at the Forecast Service. The setup for dataset creation is depicted in Figure 4.3a. Here, the Media Server serves a static MPD. It does not allow for player migration among the different CDNs so that full characterization of a specific CDN can be achieved, resulting in a time series for a CDN. Then, during streaming sessions, network bandwidth and latency measurements provided by the video players are stored in an Elasticsearch [380] database and employed to train the forecast service predictor. Optionally, Kibana [381] dashboards are available to visualize the collected metrics and guide LSTM training and tuning.

After the training phase is completed, a new setup for testing the proposed INFLOW solution is applied (4.3b). Now, the collected metrics sent with a SAND-alike mechanism are consumed by the forecast service to execute bandwidth and latency predictions for the next period. The predictions are employed by the media server to apply its decision rules for CDN selection. In this case, the media server serves an MPD dynamically updated to force video players to periodically request it. The update period is equal to



(a)

(b)

Figure 4.3: Testbed setup: configurations for dataset creation (**a**) and for INFLOW enabled delivery (**b**).

the segment duration (2 seconds) and it is set through the *minimumUpdatePeriod* tag inside the MPD.

In this setup, we aimed to compare INFLOW with other CDN selection strategies. Then, we compared the results for the following common CDN selection strategies:

- *Single CDN (SC)*: this experiment does not involve multiple CDNs. It uses just the most affordable CDN for all the clients.

- *Equal selection (ES)*: this experiment consists of balancing the occupancy rate of each CDN assigned when the session starts. Therefore, every CDN has the same number of connected clients, and the video players do not migrate between CDNs.

- *Progressive selection (PS)*: this experiment consists of progressive allocation of new CDNs when the used one(s) gets exhausted, i.e., when the theoretical maximum number of connected clients is reached and the bandwidth from the SLA is consumed. The maximum number of clients is set to 33 (100 players / 3 CDNs). The clients do not migrate between CDNs.

- *INFLOW selection (INFLOW)*: this experiment exploits the capabilities of the proposed INFLOW solution to dynamically migrate the clients depending on the predictions and the applicable cost ranges. It aims to minimize the use of CDN providers at any moment.

It is important to note that INFLOW needs to be set with the SLA for the clients to avoid any violation on QoS. Setting a bandwidth threshold lower than the minimum representation bitrate (100 kbps) is useless as the players should always experience at least the minimum representation bitrate to play the content. In the same way, a value higher than the maximum representation bitrate is not valid. Then, we decided to set the minimum bandwidth to 4 Mbps; this was enough to play a smooth 1080p video, which corresponds to two-thirds of the maximum available representation bitrate (6 Mbps).

4.2.5 Validation and Results

4.2.5.1 Predictor validation

The generation of the LSTM model consisted of three steps: training, validation, and testing. Both the training and validation steps employed a training dataset, where 80% of the samples were used for training and the remaining 20% were used for the validation step. The training dataset consisted of a multivariate time series, and bandwidth and latency measurements were taken for three hours. The collection was performed in three different sessions lasting one hour each. Each session was executed on a different day and employed a different CDN to download the content. The testing process employed a testing dataset. The testing dataset consisted of the training dataset with an extra hour of data collected on a different day that was independent of the training dataset.

To guarantee that the LSTM model used equal spaced input measurements, the simple moving average (SMA) was applied to both datasets so that an average bandwidth and latency value could be computed each second. This resulted in 10800 samples for the training dataset and 3600 samples for the testing dataset. A total of 8640 samples of the training dataset (80%) constituted the training set, while the remaining 2160 samples (20%) were used as the validation set.

The training set was employed in the first phase to generate the LSTM model. The autocorrelation plot, depicted in Figure 4.4, shows a clear correlation of the tuple (bandwidth, latency) in the time series. Here, the autocorrelation is lower for samples that are more distant. Consequently, samples which are closer to the one we want to predict are the most valuable. The LSTM model provides next values based on the last N bandwidth and latency measurements. N has been empirically set to 7. A shorter window had a big impact on LSTM accuracy, while a longer one did not result in a significant increase in LSTM forecast fidelity. The accuracy results when $N = 6$ dropped by 0.2% for the bandwidth and by 2.7% for the latency. The accuracy increased when $N = 8$ was under 1% for both time series.

A comparison of the values measured and predicted during the validation is shown in Figure 4.5. The graphs show that the predictor can follow the trend of the time series, but it cannot predict sudden and drastic changes (high or low outliers). We calculated

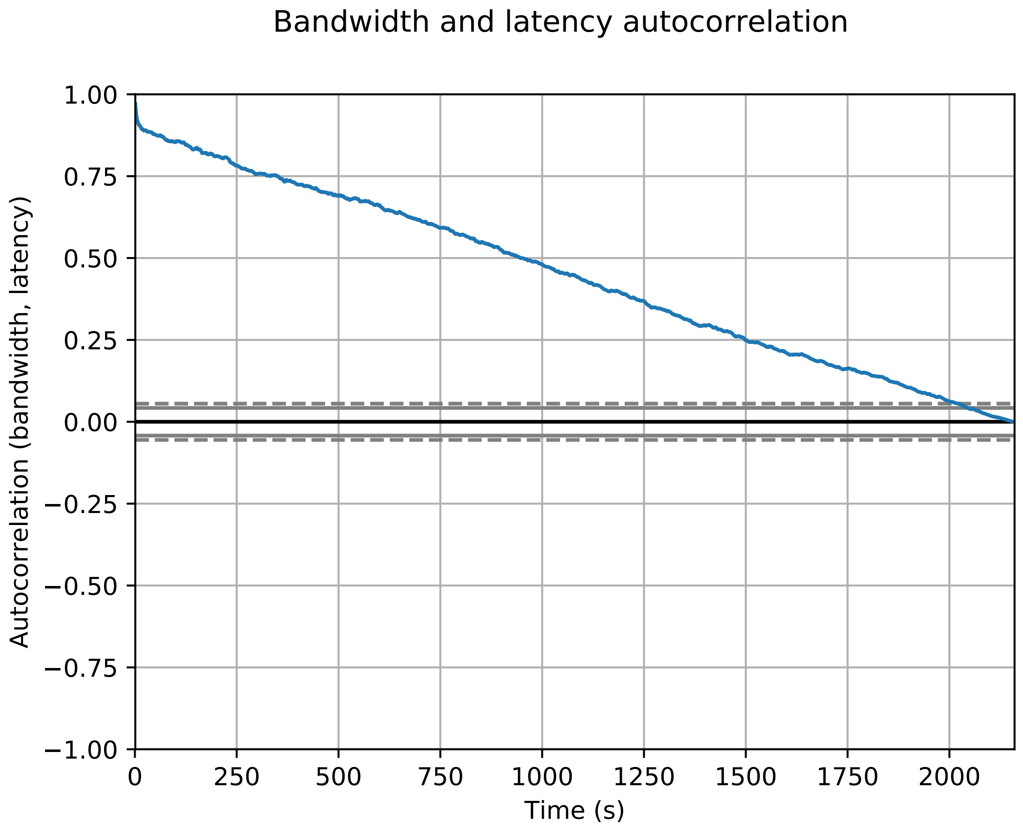


Figure 4.4: Bandwidth and latency autocorrelation.

the mean absolute error (MAE) and the root mean square error (RMSE) for both bandwidth and latency. The MAE values were 0.76 Mbps and 11 ms for bandwidth and latency, respectively, while the RMSE values were 0.99 Mbps and 27 ms, respectively.

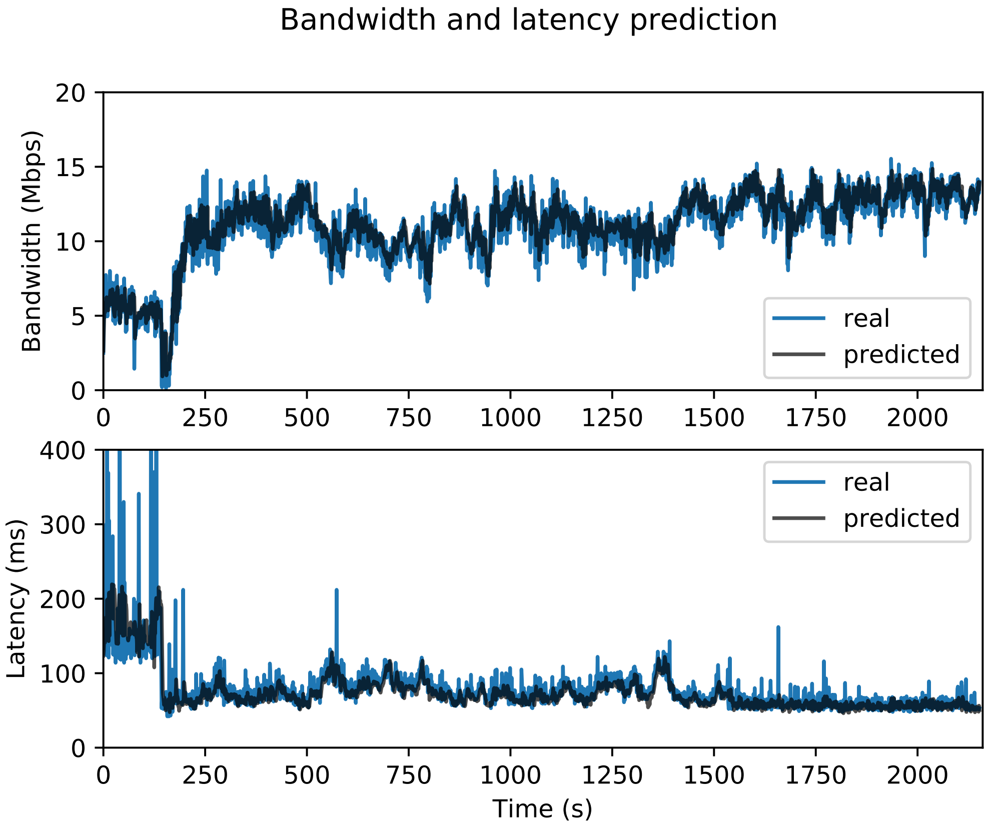


Figure 4.5: Bandwidth and latency prediction: validation of the LSTM model.

Once the model was validated, we generated the final model by training it with all the training dataset (10800 samples). Then, the final model was employed to predict values of the testing dataset. We limited the test to 2160 samples to foster a fair comparison with the validation results with a similar number of samples. For this subset, we compared the obtained values of MAE and RMSE with the ones coming from the previous validation. Figure 4.6 shows the results of the testing process. The bandwidth MAE and RMSE were equal to 0.94 Mbps and 0.51 Mbps, respectively, which are definitely close (and even better) to the values obtained during the validation. On the contrary, the latency MAE and RMSE were 31 ms and 97 ms, respectively, making evident that the latency is harder to accurately predict. From the Figure 4.6, it is clear that latency produces higher outliers than the bandwidth, which are difficult to predict.

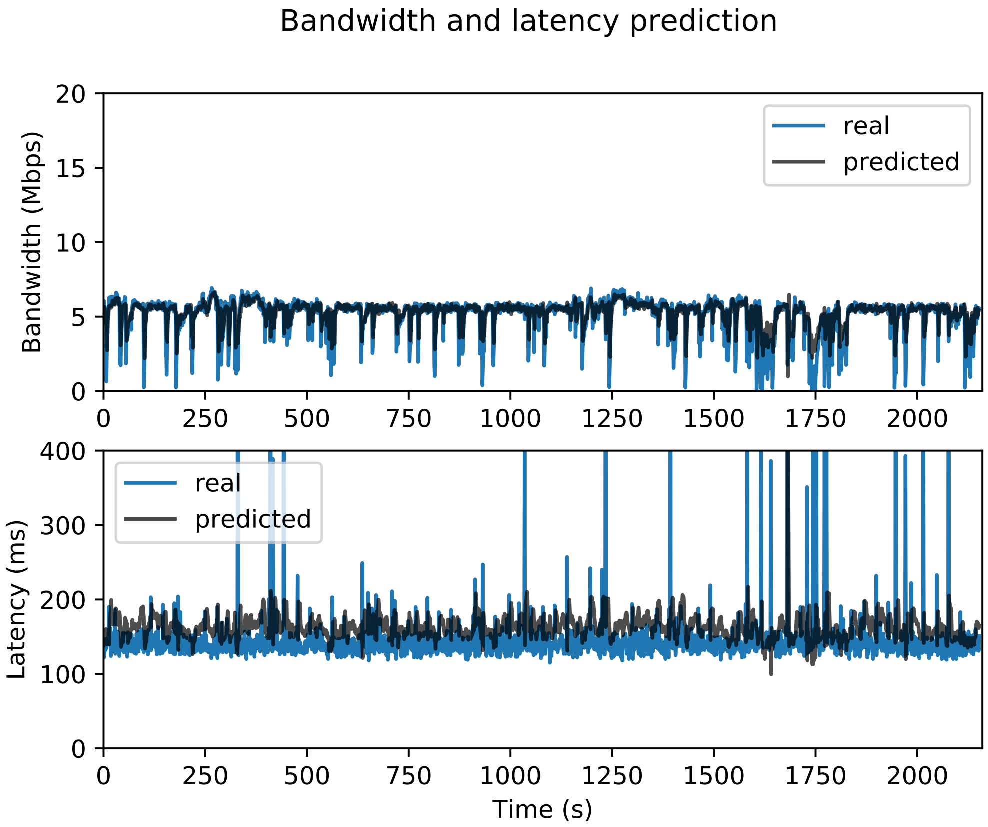


Figure 4.6: Bandwidth and latency prediction: testing of the trained LSTM model.

4.2.5.2 **QoS performance comparison**

INFLOW aims to manage the QoS performance and business cost trade-off. To this end, we identified different performance metrics for both the parameters to evaluate and balance them. We carried out the QoS evaluation by collecting the representation bitrate selected by the adaptation algorithm of the video players. Moreover, we compared the representation bitrate with the measured network bandwidth and latency to evaluate the efficiency of the utilization of the CDN resources as the efficiency increases as the overall throughput of a CDN approaches the available CDN bandwidth.

Table 4.2: Average value and standard deviation of the measured latency by the players.

Strategy	CDN1		CDN2		CDN3	
	$\mathbf{l}_{avg}$ **(ms)**	$\mathbf{l}_{dev}$ **(ms)**	$\mathbf{l}_{avg}$ **(ms)**	$\mathbf{l}_{dev}$ **(ms)**	$\mathbf{l}_{avg}$ **(ms)**	$\mathbf{l}_{dev}$ **(ms)**
Single CDN	89	39	-	-	-	-
Equal selection	89	42	132	35	42	25
Progressive selection	85	35	127	23	51	157
INFLOW selection	114	68	125	54	94	75

Table 4.3: Average value and standard deviation of the measured bandwidth by the players.

Strategy	CDN1		CDN2		CDN3	
	$\mathbf{bw}_{avg}$ **(Mbps)**	$\mathbf{bw}_{dev}$ **(Mbps)**	$\mathbf{bw}_{avg}$ **(Mbps)**	$\mathbf{bw}_{dev}$ **(Mbps)**	$\mathbf{bw}_{avg}$ **(Mbps)**	$\mathbf{bw}_{dev}$ **(Mbps)**
Single CDN	2.54	0.44	-	-	-	-
Equal selection	2.66	0.41	15.50	7.16	11.42	2.76
Progressive selection	2.90	0.34	21.84	4.79	10.65	2.17
INFLOW selection	3.52	2.43	5.23	3.85	5.88	2.82

We tested our solution by comparing it with other CDN selection strategies. The final set of experiments utilized the *single CDN (SC)*, *equal selection (ES)*, *progressive selection (PS)*, and *INFLOW selection (INFLOW)* strategies. As mentioned in the previous section, the minimum bandwidth for *INFLOW selection* algorithm was set to 4 Mbps, and the max amount of running players for each experiment was 100.

Table 4.2 shows the network latency for the video players. The results for each CDN while employing *SC*, *ES*, and *PS* strategies are close to each other. Each CDN presents similar latency independently of the strategy. On the contrary, *INFLOW* strategy presents a higher latency of up to +124% (CDN3) as switching the connection from a CDN to another inevitably implies the addition of delay. Furthermore, if the experienced latency is still in the order of hundreds of milliseconds, then it does not affect the video players, which have a playback buffer equal to one segment duration (2 seconds).

Table 4.3 shows the available network bandwidth for the video players while video content is being downloaded from the CDNs. Table 4.4 presents the selected bitrate from the client-side algorithm.

SC strategy provides only information for CDN1 as the other two are never used. This strategy provides the worst results when compared to the others because the players are experiencing a highly congested CDN communication. The average measured bandwidth is 2.54 Mbps, and average representation bitrate is 1.67 Mbps.

Table 4.4: Average value and standard deviation of the selected bitrate by the players.

Strategy	CDN1		CDN2		CDN3	
	R_{avg}(Mbps)	R_{dev}(Mbps)	R_{avg}(Mbps)	R_{dev}(Mbps)	R_{avg}(Mbps)	R_{dev}(Mbps)
Single CDN	1.67	0.62	-	-	-	-
Equal selection	1.78	0.67	4.46	2.10	4.77	1.71
Progressive selection	1.96	0.61	5.10	1.78	4.48	1.98
INFLOW selection	1.96	1.23	2.44	1.57	2.82	1.43

As expected, *SC* results improve when the multi-CDN strategy comes into place, therefore allowing for load balancing. *ES* and *PS* strategies limit to 33 (100 players / 3 CDNs) the number of players connected to CDN1. Both strategies produce similar results for the different CDNs. For CDN1, the average measured bandwidth of *ES* and *PS*, when compared to *single CDN* strategy, improves +4.7% and +14.1%, respectively. These results mean a higher bitrate selection, +6.6% and +17.4%, respectively. Moreover, the selected bitrates are considerably higher for CDN2 and CDN3 as they provide a higher performance. These CDNs provide more network resources serving higher bandwidths for video players. The measured results range between 15.50 Mbps (*ES*) and 21.84 (*PS*) for CDN2 and 10.65Mbps (*PS*) and 11.42 (*ES*) for CDN3. Thus, distributing video players across the available CDNs by just considering the number of players per CDN (33 players) is not fair as video players connected to CDN2 and CDN3 can select higher representation bitrates. Video players select a representation bitrate up to +160% higher if connected to CDN2 and +168% higher if connected to CDN3.

INLFOW uses a different approach. Here, the maximum number of players for each CDN is constantly updated through Equation 4.1. Then, video players are dynamically switched at any time. CDN1 still underperforms compared to CDN2 and CDN3, but the fairness is lower. The residual performance bias is due to the fact that CDN1 is still the preferred CDN, i.e., the other two are not used until CDN1 is congested. Accordingly, CDN3 is not used until CDN2 is congested too. Here, the higher average measured bandwidth is 5.88 Mbps at CDN3, which is +67% higher than the result at CDN1 (3.52 Mbps). For *ES* and *PS*, the variations are +483% (CDN2 compared to CDN1) and 653% (CDN2 compared to CDN1), respectively. In terms of the selected representation bitrate, *INFLOW* keeps the results obtained by the other multi-CDN strategies at CDN1 but underperforms at CDN2 and CDN3. This is because INFLOW aims to reduce the number of employed CDNs and the OPEX at any time, while guaranteeing at least 4 Mbps for the measured network bandwidth. Thus, video sessions to CDN2 or CDN3 can be retired, therefore saving the CDN OPEX. The other multi-CDN strategies do not reduce CDN usage, i.e., when the player is connected to a CDN, the connection is maintained until the session expires. In terms of fairness, *INFLOW* outperforms the other multi-CDN strategies as the average representation bitrates of each CDN are close to each other. Video players connected to CDN2 present the best representation bitrate (2.44 Mbps), which is +24% higher than those of video players connected to CDN1. Moreover, standard deviation for measured bandwidth and representation bitrate achieved with INFLOW also demonstrates that it is the fairest solution since the players experience almost the same variation independently of the CDN. Standard deviation for CDN2 is +58% higher than CDN1 if we consider measured bandwidth and +27% if we consider representation bitrate.

Concerning communications overheads to proactively enforce QoS, the traffic overhead is 893 MB from the total traffic (53592 MB). Thus, overhead causes an increase of +1.6% in the transmitted data. INFLOW exploits MPD update mechanism with an update period is equal to the segment duration. In our case this means 9040 Bytes requested every 2 seconds by each player. In the other strategies, MPD update mechanism is not used, then there is not an additional overhead.

The predictions performed by the INFLOW Forecast service during the MPD requests causes also higher MPD delivery delay compared to the other strategies. As a result, INFLOW strategy adds 53 ms of delay while delivering the MPD. Nevertheless, this delay does not affect the playback since the player employs the previous MPD until a new one is received and parsed. Thus, if it is necessary to perform a segment request during an MPD update, it is performed in any case, as the two operations are executed in different threads and do not interfere with each other. In terms of resource utilization, the node running the INFLOW Forecast service shown 3.2 GB RAM, 33% CPU and 27% GPU peak utilization rates. Thus, its hardware configuration could absorb a larger number of users.

4.2.5.3 **Business cost comparison**

Regarding business cost, OPEX consists of ongoing expenses that a business incurs inherent to the operation of the assets. In our case, we were interested in OPEX, as we wanted to evaluate the cost of the CDN resources due to their ongoing utilization, i.e., it depends on the utilization of the CDN resources at any time [113].

There is no common formula for evaluating the OPEX, as in many cases, the provider does not publish publicly its pricing plans, but it offers personalized plans to each customer. Nevertheless, [114], [115] and [116] reveal that the OPEX for CDN resources depends on a set of factors such as the employed network, storage, and time resources. Then, we express monthly OPEX through Equation 4.2.

$$
\begin{aligned}
OPEX_{month} = \sum_{i=1}^{K} \alpha_{loc_i} * Tr_i + \beta_{loc_i} * K_{req_i} + \\
+ \gamma_{loc_i} * T_i + \delta_{loc_i} * St_i + \epsilon_{loc_i}
\end{aligned}
\tag{4.2}
$$

In the equation, Tr_i is the traffic volume in a month, K_{req_i} is the number of HTTP requests producing such demand, T_i is the utilization time for a CDN that has active sessions from video players of a service, and St_i is the employed storage at the CDN. Therefore, α_{loc_i}, β_{loc_i}, γ_{loc_i}, δ_{loc_i}, and ϵ_{loc_i} are multiplicative coefficients established by a particular CDN provider and that depend on the location of the resources (cost of the servers depends on the country where they are located). The addition indicates that we are in multi-CDN environment. Then, we need to sum over the K available CDNs.

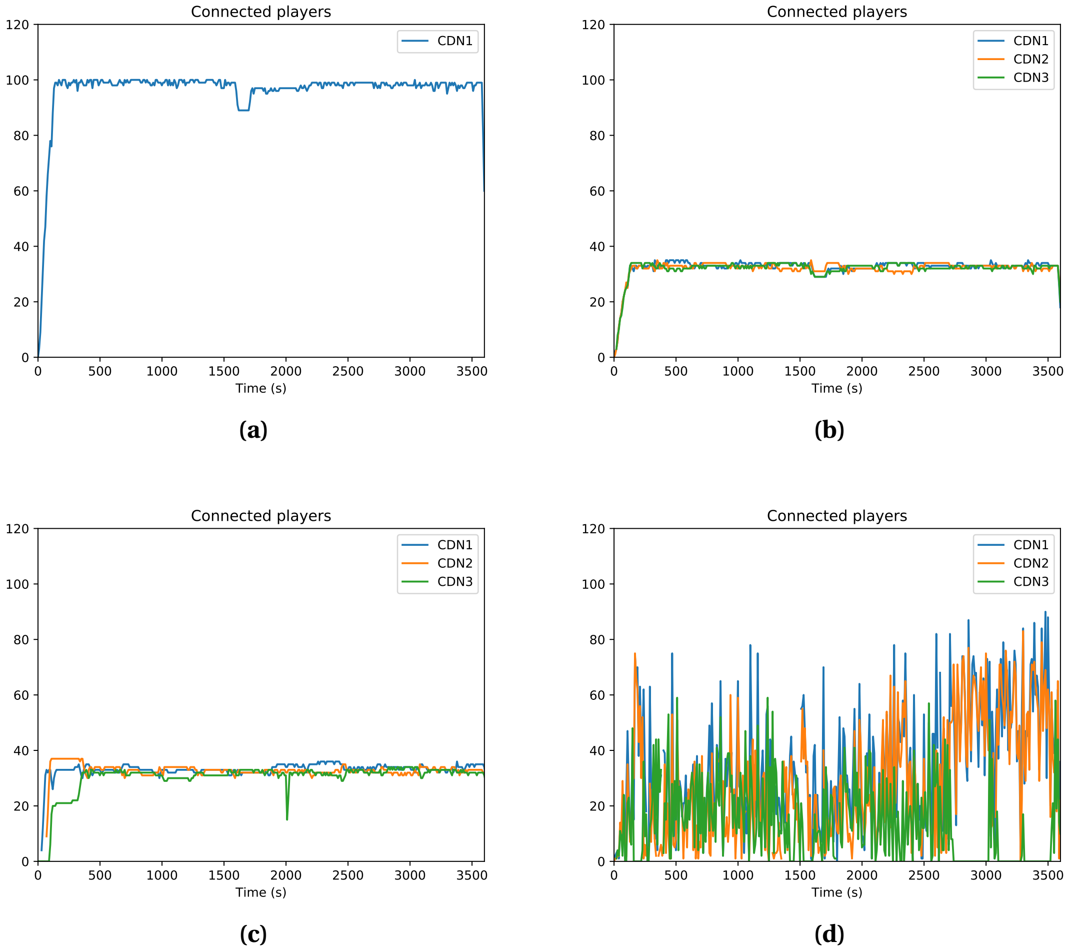
The values for the coefficients are closely related to the business model and the pricing plan of each CDN provider. Accordingly, the monthly OPEX is tailored to the employed CDNs. In any case, we evaluate the variables independent of the CDN vendor, which depend on the resources we are employing during the tests (Tr_i, K_{req_i}, T_i and St_i) and directly impact the OPEX.

To simplify the evaluation of OPEX, we have made some assumptions. First, St_i is fixed for each experiment as the amount of employed storage depends on the content size, and it is permanently stored, even if it is never requested. Second, K_{req_i} is almost constant as, in any case, the experiments run 100 players, which request a media segment and an MPD every 2 seconds (segment duration). Third, Tr_i is directly proportional to the selected representation bitrate. It can be roughly calculated by multiplying the mean bitrate of the sessions from video players and the duration of the experiment. As the selected bitrate is already being captured for the QoS evaluation, we can assess the traffic volume. Finally, T_i is the variable that really changes across every experiment depending on the cost-effective strategy, leading to different utilization rates for each available CDN. Then, we employ T_i as the main metric for comparing the OPEX achieved by the different strategies.

Table 4.5: Utilization time of the CDNs.

Strategy	CDN1	CDN2	CDN3
	T_i (minutes)	T_i (minutes)	T_i (minutes)
Single CDN	60	-	-
Equal selection	59	59	58
Progressive selection	58	58	57
INFLOW selection	52	48	31

Table 4.5 shows the usage time of each CDN while applying the different strategies. *ES* strategy is the most expensive solution as all the CDN are utilized almost all the time. In this case, the overall usage time is close to 3 h (1 h per each CDN). The actual result is 176 min. On the contrary, *SC* results in lower business costs as CDN2 and CDN3 are never employed. In this case, the usage time is just 60 min. *PS* is quite like *ES*. Figure 4.7 shows the number of players connected to each CDN for one hour and it is clear that *ES* and *PS* differ only in the first minutes. In *ES*, the three curves increase almost together, while in *PS*, the curves separately increase because CDN2 is not employed until CDN1 reaches 33 players and CDN3 is only employed after both CDN1 and CDN2 reach 33 players. The overall usage time is 173 min, which corresponds to a reduction of -2% compared to *ES*. From Table 4.5, the usage time of each CDN while employing PS strategy is similar to ES one. Finally, *INFLOW* graph presents a completely different behavior. Here, the number of players connected at each CDN is much more variable owing to the switching mechanism. The number of players of each CDN ranges between 0 (the CDN is not being used) to 100 (CDN serving all the players). The number of players for the other multi-CDN strategies is always around 33 players. Nevertheless, INFLOW can retire the sessions from a CDN, which is not necessary after migrating the clients. This results in 131 min of overall usage time; then the reductions for *ES* and *PS* are -26% and -24%, respectively. Compared to *SC* strategy, *INFLOW* employs +118% more CDN usage time, while the value increases to +193% and +183% for *ES* and *PS*, respectively If we focus on the usage time of each CDN, Table 4.5 clearly shows that INFLOW reduces the usage of CDN2 and CDN3.



(a)

(b)

(c)

(d)

Figure 4.7: Distribution of the players among the available CDNs: single CDN (**a**), equal selection (**b**), progressive selection (**c**) and INFLOW selection (**d**) strategies.

In summary, the proposed INFLOW solution improves the CDN resource management by dynamically selecting the CDN for each video player at any time. It allows for business cost saving by decreasing the usage time of the available CDNs, while maintaining a minimum bandwidth level. Moreover, the resources are more efficiently exploited because the players are distributed depending on the real capabilities of each CDN, such as the experienced network resources. Consequently, the selected bitrate is fairer among the players.

4.2.6 Conclusions and Future Work

The trend for the following years is an increasing consumption of media content, where the content is mostly delivered through CDN infrastructure. Here, the CP strives to guarantee the necessary QoS for its media service while reducing the business costs associated with the CDN.

Toward this goal, we introduce a novel solution called INFLOW for CDN selection in a multi-CDN delivery environment. INFLOW enables the media server with a forecast service so that metrics collected by the player are processed as MPEG-DASH streams are served. The forecast service executes time series analysis through an LSTM model for prediction of the future values of network bandwidth and latency. The predictions are exploited by the media server to act while the player requests an MPD update. The media server can decide to keep the same MPD or change it to switch the player to another available CDN from which content can be downloaded.

The proposed solution has been implemented and validated in a distributed and heterogeneous testbed employing real network nodes. The evaluation includes a comparison with other CDN selection strategies in terms of QoS and business cost. The results highlight the advantages of INFLOW for reducing the overall usage time of the available CDNs, while guaranteeing a minimum level of network bandwidth to every player.

Future work includes the exploitation of new metrics to improve the predictions made by the forecast service. Moreover, the collected metrics can be further exploited to obtain an estimation of a user's QoE, and actions that also take into consideration the user's expectations can be taken.

Acknowledgment

This work was fully supported by the EC projects Fed4Fire+, 732638 (H2020-ICT-13-2016, Research and Innovation action) and Open-VERSO project (Red Cervera program, Spanish government's Centre for the Development of Industrial Technology).

CHAPTER

5

MEC-enabled content delivery

5.1 Context

MEC is a new network architecture concept included in the 5G ecosystem to boost the performance of network services. It enables computing capability close to the RAN such to run algorithms and/or services that empower specific applications. Moreover, RNI Service (RNIS) integrated into MEC platform allows to access RAN information or RNI, a set of objective metrics (QoS) concerning the status of the UE connected through radio interface. When considering video streaming, having only QoS metrics is not enough. It is useful to assess or estimate the QoE experienced by the users and take actions to improve their individual QoE and the fairness among them in the exploitation of network resources. Maximizing customer satisfaction through QoE-based networking is a crucial challenge for video streaming services. ETSI working documents also support the application of analytics at the MEC to optimize the video streaming traffic.

Section 5.2 proposes a MEC-enabled MPEG-DASH video streaming proxy that estimates the users' QoE according to ITU-T P.1203. It is derived from monitored QoS metrics in a dense client radio cell and the information acquired by accessing the MPEG-DASH manifest (MPD). It does not need an explicit out of band messaging from video players to MEC system. Thus, the implemented MEC proxy is independent of video servers and players. Knowledge of QoE can enable advanced solutions that realize the network/application symbiosis, as it provides the information to subsequently decide streaming qualities in a coordinated manner in a dense client cell. The major contributions of this paper can be summarized as follows:

- A novel mechanism to estimate ITU-T P.1203 QoE scores from network dynamics at the cell and parameters parsed from MPEG-DASH MPD without an explicit out of band messaging from video players to MEC system. It aims to assess information of video representation bitrates, number of representation switches, number of stalling events and their duration.
- An implementation of a MEC proxy, independent from video servers and players, to monitor and assess ITU-T P.1203 QoE scores for each local session.
- The analysis of the accuracy of the proposed solution to assess the ITU-T P.1203 QoE of individual players in a Wi-Fi-based experimentation setup and a dense client cell. To this end, two scenarios with different levels of concurrency and congestion are performed.

Concerning stall assessment, the MEC proxy cannot detect all the stalls experienced by the player. In any case, the total duration of stalls tends to converge to the actual accumulated value experienced by the player. The MEC proxy cannot detect micro-stalls and tends to concentrate several of them into a macro-stall. This behavior is due to the sampling time to check for stalls (segment duration of 6 seconds). If two micro-stalls are experienced during the same segment, MEC proxy will conjecture that just one longer stall happened since it checks only at the end of each segment download. The results in terms of QoE scores obtained at the MEC are compared with the actual ones achieved at the player side. Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) rates show that the scores obtained at the MEC tend to be like the ones at the player. It demonstrates the capability of the MEC Proxy to estimate ITU-T P.1203 QoE scores close to actual ones.

In Section 5.3, information available at the MEC location is exploited to develop a MEC solution, called MEC4CDN, that allows two different operations. First, MEC4CDN caches popular MPEG-DASH segments at network edge to reduce CDN usage. To this end, it is able to identify recurrent requests. Second, it shields from identified or predicted CDN malfunction by switching the download of MPEG-DASH segments to an alternative CDN in order to ensure QoE rates. Thus, it enables a CDN dynamic selection based on live connectivity statistics. In both operations, the information of MPEG-DASH MPD is essential to know the available representations, as well as the location of the remote CDN servers. This paper comprises the following relevant contributions:

- Local cache at the MEC of the identified contents to minimize the traffic between the CDN and the network edge. MEC4CDN decides to locally cache already served responses to serve near future requests and proactively cache the identified future segments in case of VOD streams.

- Dynamic selection of the CDN based on live measurements in the context of a multi-CDN delivery. The player starts receiving the stream from a CDN server, that is favorable in terms of QoS connectivity metrics, and is dynamically switched to another one which provides better performance in case of CDN degradation or outage.

- Evaluation by delivering MPEG-DASH streams in a dense client cell and comparing the two proposed strategies (local cache and dynamic CDN selection) with a legacy stream delivery.

Concerning local cache strategy, the results show that MEC4CDN lets the player experience lower latency since the segments are closer to it, then the throughput is higher. Therefore, the players tend to request higher representation bitrates which improve the user's QoE. When considering dynamic CDN selection strategy, legacy MPEG-DASH delivery does not let the player take actions when the principal CDN suffers performance degradation. The player has to reduce the representation bitrate in order to continue playing. On the contrary, when MEC4CDN comes into play, it switches player's connection to a healthy CDN such that the representation bitrate is kept to higher levels. Therefore, MEC4CDN is able to enforce the delivery by switching to an alternative CDN which better performs. To sum up, both strategies present advantages over a legacy MPEG-DASH delivery. When network issues are experienced, they allow to keep the QoE rates (CDN switching) or even improve them (local cache at the edge).

5.2 Multi-access Edge Computing video analytics of ITU-T P.1203 Quality of Experience for streaming monitoring in dense client cells

- **Title:** Multi-access Edge Computing video analytics of ITU-T P.1203 Quality of Experience for streaming monitoring in dense client cells
- **Authors:** Roberto Viola, Mikel Zorrilla, Pablo Angueira and Jon Montalbán
- **Journal:** Multimedia Tools and Applications
- **Publisher:** Springer
- **Year:** (Submitted March 25, 2021)

Abstract: 5G promises unseen network rates and capacity. Furthermore, 5G ambitions agile networking for specific service traffic catalysing the application and network symbiosis. Nowadays, the video streaming services consume lots of networking assets and produce high dynamics caused by players mobility meaning a challenging traffic for network management. The Quality of Experience (QoE) metric defined by ITU-T P.1203 formulates the playback issues related to widely employed Dynamic Adaptive Streaming over HTTP (DASH) technologies based on a set of parameters measured at the video player. Monitoring the individual QoE is essential to dynamically provide the best experience to each user in a cell, while video players compete to enhance their individual QoE and cause high network performance dynamics. The edge systems have a perfect position to bring live coordination to dense and dynamic environments, but they are not aware of QoE experienced by each video player. This work proposes a mechanism to assess QoE scores from network dynamics at the cell and manifests of DASH streams without an explicit out of band messaging from video players to edge systems. Hence, this paper implements an edge proxy, independent from video servers and players, to monitor and estimate QoE providing the required information to later decide streaming qualities in a coordinated manner in a dense client cell. Its lightweight computation design provides real-time and distributed processing of local sessions. To check its validity, a WiFi setup has been exercised where the accuracy of the system at the edge is checked by assessing the ITU-T P.1203 QoE of individual players.

Keywords: 5G, MEC, MOS, MPEG-DASH, QoE.

5.2.1 Introduction

Popularity of video streaming platforms to consume live sports events and over-the-top (OTT) services like Netflix, when combined with mobility and city contexts, result in a complex traffic congestion scenario to be managed. Specifically, when video players share the same path, they try to individually enhance their Quality of Experience (QoE) considering instant available bandwidth and display setup. As the cell gets more subscribers and the Base Station (BS) serves more traffic sessions, dynamics turn higher and available bandwidth estimation at client side gets coarser, fuelling video player fluctuations on requested bandwidth which affect the QoE [382]. Therefore, when congestion at the radio link comes into place, a dense client cell means a pool of clients competing for the available network assets. This competition leads to player instability, unfairness between players, and bandwidth under-utilization [369].

Maximizing customer satisfaction through QoE-based networking is a crucial challenge for video streaming services. As most of the traffic flooding the networks comes from video streaming services, Mobile Network Operators (MNOs) need to configure the network according to cost-effective policies to cope with video demand while providing best QoE with the available resources. To this end, MNOs need to monitor the QoE without access to video streaming servers, Content Delivery Network (CDN) systems or video player applications.

In the upcoming horizon 5G aims to improve current Key Performance Indicators (KPIs), such as traffic density, volume and latency. It will flatten barriers to allow network operators and/or 3rd parties to provide dynamic networking performance to respond to specific traffic demands [383]. To this end, network management frameworks analyse network assets and traffic features to adapt network setup to concurrent traffic needs. This means a revolution on networks with a digital transformation which leaves behind the concept one-fits-all and moves to a specialized and mutable network which dynamically changes its configuration over a common bare-metal infrastructure.

In the line of catalysing synergies of network and traffic from services, edge computing is gaining relevance due to benefits of local processing. Enabled by the transition of networking assets to cloud infrastructures, the provision of hosting at the edge means new revenues for MNOs. Furthermore, the location of some parts or entire systems of a service closed to the clients enables advanced possibilities considering the user environment. The application of analytics at the edge to optimise the video streaming traffic is a use case compiled in the ETSI working documents as a key representative of Multi-access Edge Computing (MEC) application [39].

The proposed 5G MEC-enabled proxy estimates the end users' QoE according to ITU-T P.1203 [1] and derived from monitored QoS metrics in the radio cell to allow a decision by other edge services operating streaming qualities in a coordinated manner in a dense client cell. Thus, the proposed system catalyses the use of Video Streaming Analytics at the network edge to realise the network/application symbiosis [39].

The major contributions of this paper can be summarized as follows:

- A novel mechanism to estimate ITU-T P.1203 [1] QoE values from network dynamics at the cell and parameters parsed from manifests of DASH streams without an explicit out of band messaging from video players to edge system.

- A lightweight implementation of an edge computing proxy, independent from video servers and players, to monitor and assess in real-time ITU-T P.1203 QoE values for each local session.

- The analysis of the accuracy of the proposed system to assess the ITU-T P.1203 QoE of individual players in a WiFi-based experimentation setup and a dense client cell. To this end, two scenarios with different levels of concurrency and congestion are performed.

The rest of this paper is organized as follows. First, the related work is in Section 5.2.2. The MEC system model and architecture and the streaming analytics service is in Section 5.2.3. Furthermore, the system performance analysis, defined metrics, the evaluation setup and the results are described in Section 5.2.4. Finally, the Section 5.2.5 concludes the paper.

5.2.2 **Related Work**

QoE of streamed video contents is intrinsically related to customers' satisfaction. Mean Opinion Score (MOS) compiles the results of subjective evaluations over a variety of representative spectators/audience surveyed to score perceived quality, ranging 1 to 5 values [84]. However, this surveys process results in high expensive costs since it cannot be automated, as it is difficult to be repeated as updates on operational parameters are applied and it is complex to process the results in real-time.

Therefore, research community investigates models to map subjective distortion to objective QoS and application metrics which can be automatically monitored and computed in real-time producing actionable data [50]. These subjective quality models change as the streaming technologies evolves. Nowadays, HTTP Adaptive Streaming (HAS) technologies are widely used for live and on-demand video streaming services. Then, QoE models shift from pixel-level distance evaluation of received frames from the original served, such as Peak Signal to Noise Ratio (PSNR) based on Mean Squared Error (MSE) or Structural Similarity Index (SSIM), to content-agnostic models with sophisticated parametric equations comprising throughput, frequency of bitrate changes from available representations and buffering duration. These models better reflect quality level from a video streaming perspective, they are lightweight and can be processed in real-time. Per-title encoding techniques [384] are widely employed by major video services, adapting the employed bitrate to the image complexity to save bandwidth costs, so representation bitrate has a direct translation to image fidelity and pixel-level distortion is not required anymore.

Objective QoE models [385] are widely employed at media player side to enable decision making on representation/bitrate selection to accommodate the demand to the available network resources and maximise the local QoE. Some of the QoE models at player side are heavily dependent on buffer metrics [386]. At the server side, buffer information is not available and QoE models usually employ HTTP delivery information [387]. Some server-side models exploit also lower-level information, such as TCP connections [388], or combine both TCP and HTTP information [389]. The ITU-T P.1203, whose last version was published in October 2017, is a standardized model for QoE evaluation of adaptive audio and video streams including video and audio quality and buffering issues, where quality is closely related to representation bitrate of each played media segment, number of representation switches, number of stalling events and their duration.

Dense client cells are tough environments where the autonomous decisions from video players may produce unfair, biased, unsteady and inefficient use of the radio link. There, intrinsic ON-OFF periods of HAS technologies may cause oscillations on the experienced network throughput and, accordingly, on the selected bitrate [369]. Client-side algorithms, selecting representation to be downloaded and played, have been proposed to detect and compensate bitrate oscillations [390]. However, the lack of coordination across the video players may turn available network resources into stochastic and bitrate selections into erratic as the density of sessions or the audience dynamics mutate quicker. There, a server-assisted solution to compensate oscillations allows improving video player bitrate selection [391], it does not need modifications on client-side algorithms to work, but it does not solve uncoordinated behaviour across the players. Here, a more coordinated approach is needed to apply common policies to all the local subscribers. Evolved from legacy Real-Time Transport Control Protocol (RTCP), which periodically produces QoS statistics to allow the server to change operational ranges, server-side QoE estimation mechanisms [392] have been also employed to manage video delivery and provision platform resources.

However, once different representations are available at the server or CDN to match display preferences and changeable network conditions due to mobility, server-side approach is less scalable and slower to react than an edge system.

5G brings several technologies to leap density KPIs. At the Radio Access Network (RAN), massive multiple-input multiple-output (MIMO) technology, millimeter wave communications, and small-, micro-, pico-cell concepts have appeared to improve the spectrum efficiency raising communication throughput while saving energy [393]. Concerning the network backhaul and core, Software-defined Network (SDN) and Network Function Virtualization (NFV) paradigms adopted in the ETSI stack are the pillars to realise agile and smart networking for delivering traffic from different services to different users and under specific level of performance [394]. However, in the case of dense client cells, it is the MEC architecture which will make the difference to monitor QoE, apply policies to video delivery and coordinate video players' decisions transparently by managing content manifests [395].

Different systems have exploited the exceptional position of MEC architectures to host QoS/QoE monitoring services. Some of them to apply video transcoding operations [206], to perform in network caching [396] or to negotiate the wireless interface to use [397] to satisfy a target QoE. However, they add new systems processing provided data at the network edge losing transparency for media servers and players meaning messaging overheads [206]. From the scalability perspective, proposed systems tend to perform heavy processing such as Random Neural Network (RNN) to assess the QoE at the viewer [206] or perform statistical mapping of MOS from QoS metrics [397]. More focused in 5G MEC architectures, lightweight monitoring and processing systems embodied in a Virtual Network Function (VNF) which infers Dynamic Adaptive Streaming over HTTP (MPEG-DASH) clients' QoE, independent from the media server and players, means a viable service to process real-time QoE metrics in a dense client environment [398]. Here, objective metrics which influence QoE, such as initial playout delay and the rebuffering number and duration metrics, are estimated from requests timestamps, as the representation bitrate switches can be directly obtained by the requested URL from the media manifest. This approach still relies on a LTE network for testing, meaning that the MEC service is not actually running at network edge, but at the LTE Core (Evolved Packet Core) location. Our approach is similar, expanding QoE evaluation to include subjective results estimation through the QoE model defined by ITU-T P.1203 [1] and assessing actual bitrate of each requested segment, as the nominal bitrate of the manifest is coarse affecting inference accuracy. We add a novel mechanism to estimate ITU-T P.1203 QoE by analysing network dynamics and parsing manifest parameters of DASH streams without an explicit out of band messaging from video players to edge system. Moreover, we deploy our MEC service on a WiFi access point, according to MEC architecture defined by ETSI [41].

5.2.3 Edge Analytics Service

5.2.3.1 Architecture

The analysis of QoE is crucial for decision making systems enforcing or boosting video streaming services. Video players apply algorithms to adapt bitrate demand to the changeable network performance to autonomously enforce its QoE. However, in dense client cells a more coordinated approach across the sessions sharing the radio link is needed to avoid individual competition for available network assets [369], thus fostering steady and smooth playback. Edge systems have an exceptional position to enforce or enhance QoE of video streaming services applying common policies to all the users in a cell while bringing scalable, transparent and zero latency processing of local metrics [39].

This section describes the proposed MEC proxy according to ETSI standardized MEC architecture [39]. The Proxy performs the estimation of the standard ITU-T P.1203 [1] parameters as depicted in Figure 5.1. To keep video players and server architectures, no additional communications are introduced inside the delivery chain to capture and report players' metrics. Thus, communication overheads are avoided. This means that the MEC systems do not have direct access to all the factors which comes into play in the equation of the QoE which are measured at the media player. This manner, it is necessary to estimate the QoE by inferring the values of involved factors.

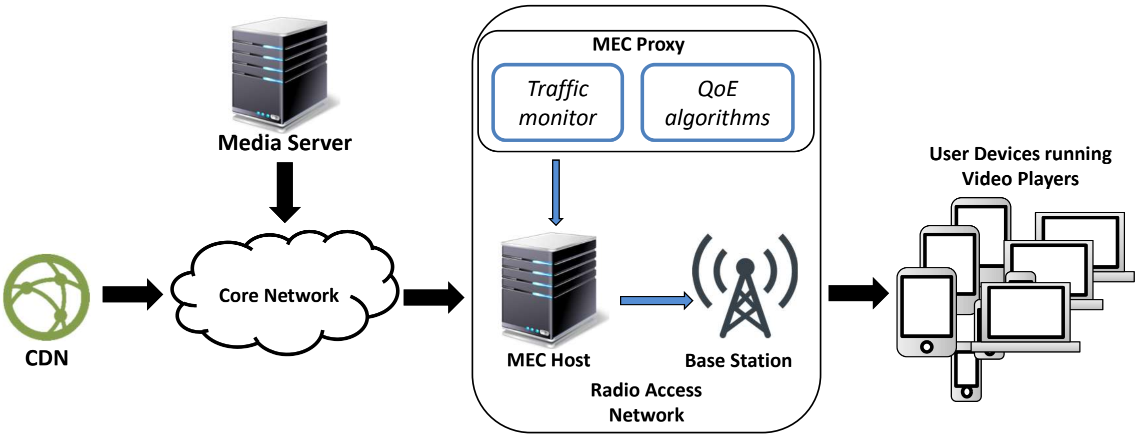


Figure 5.1: Architecture of MEC-based QoE estimation approach.

To achieve it, the MEC Proxy is deployed at the MEC Host and monitors all the traffic exchanged at RAN between the video players, media server and CDN [39]. During this operation, it can assess specific metrics related to the streaming session from two different activities:

- download of video manifest from media server, which includes different metadata for each representation such as resolutions, nominal bitrates, or language for the different media streams;
- player's representation selection, when matching the HTTP segment requests with representations available in the manifest.

When a streaming session is started, the MEC Proxy downloads the video manifest and serves it to the player. The MEC Proxy can locally analyse the manifest to list the available representations and their features since the manifest includes information for all the available video, audio and subtitles tracks. Later, when the session is already playing, the video player must retrieve segments from the media server or CDN through HTTP requests. In this case, the MEC Proxy can act in two different ways:

1. MEC Proxy can parse the information contained inside the HTTP URL and header as well as obtain request/response timestamps;

2. MEC Proxy can analyse all the HTTP packets payload, meaning an active analysis of the media stream.

Our solution is on top of first option to reduce the processing demands of the Proxy at the edge and favour a more scalable solution. In any case, it is security/privacy sensitive since standard encryption mechanisms of HAS standards just encrypts the payload of media segments to avoid unauthorized playback, so parsing the media manifest and process the HTTP requests endpoints are allowed.

The general communication is presented in Figure 5.2. First, the HTTP Proxy needs to identify each streaming session to link captured timestamps and QoE factors from each request and response to a specific item to infer its QoE metrics. To achieve it, the MEC Proxy generates a random ID when the first HTTP request from an IP requesting a manifest is received. Therefore, the MEC Proxy maintains a list with all the active streaming sessions. A session is considered expired when no HTTP requests are received during the duration of two segments. Then, inactive sessions are removed from the list.

When the manifest is received by the video player it decides which representation better fits with the display features, the user preferences, the service subscription, and the network performance. This decision is revised by the video player for each segment to request. Accordingly, it starts to request a specific representation bitrate. The MEC Proxy stores the timestamps to later process request pace to detect any buffering issue when comparing nominal segment duration and time elapsed between requests. Then the MEC Proxy performs the CDN request and analyse some high-level features from the provided segment to accurately estimate the bitrate which may differ from the nominal value declared at the manifest. From that information, the MEC Proxy is can infer an estimated ITU-T P.1203 QoE. Last, the segment is delivered to the video player.

The list of sessions and the estimated QoE, updated for each segment request, is available for other systems at the edge or in the cloud to facilitate the enforcement or enhancement of video streaming in a dense client cell concerning QoE.

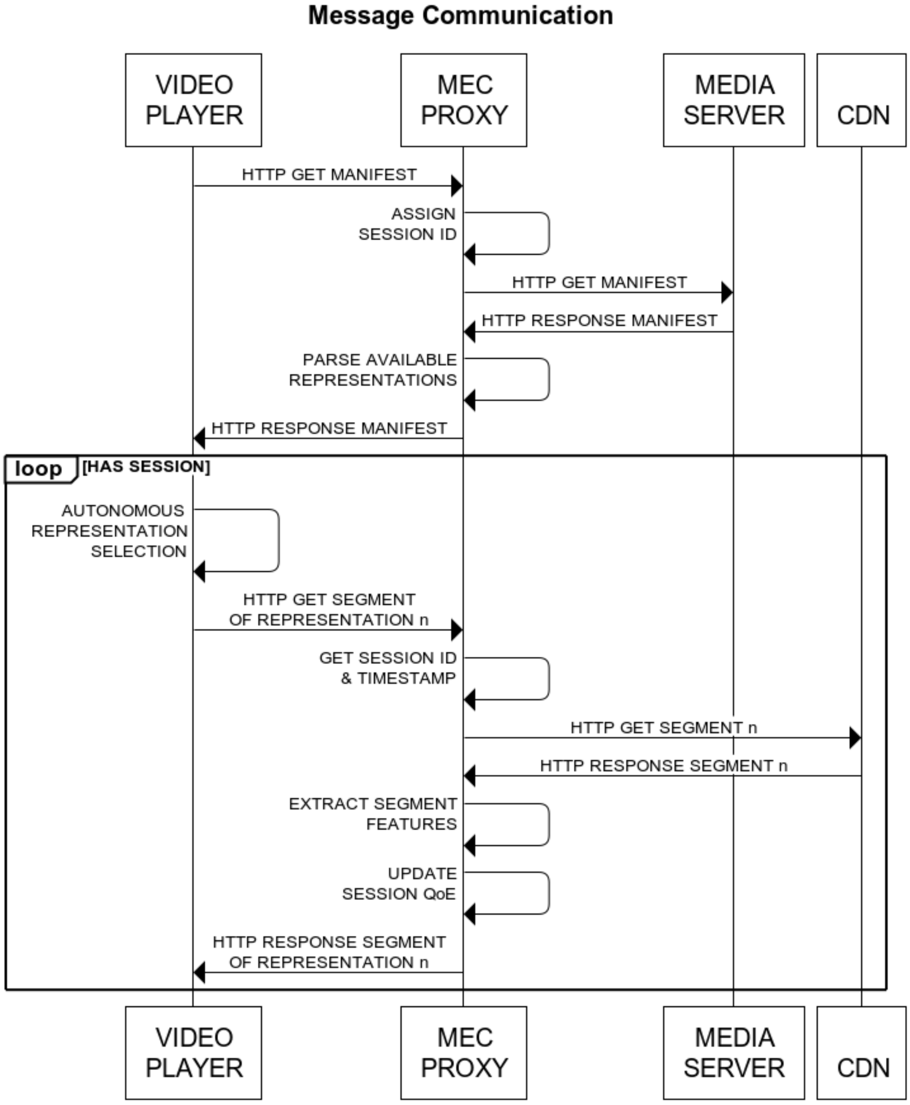


Figure 5.2: Message communication.

5.2.3.2 **QoE Estimation Algorithm**

ITU-T P.1203 [1] describes a model for monitoring media session quality while delivering content through HAS technologies. The building block of the ITU-T P.1203 model is presented in Figure 5.3.

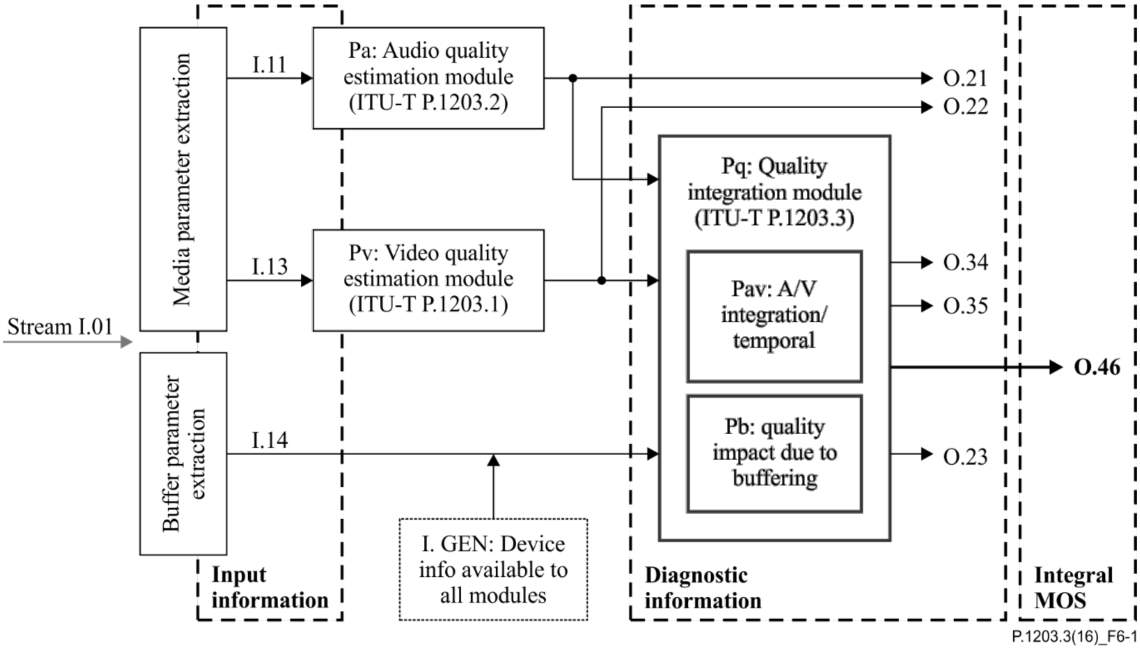

Stream I.01
Media parameter extraction
Buffer parameter extraction
I.11
I.13
I.14
Pa: Audio quality estimation module (ITU-T P.1203.2)
Pv: Video quality estimation module (ITU-T P.1203.1)
I. GEN: Device info available to all modules
Input information
O.21
O.22
Pq: Quality integration module (ITU-T P.1203.3)
Pav: A/V integration/ temporal
Pb: quality impact due to buffering
O.34
O.35
O.23
O.46
Diagnostic information
Integral MOS
P.1203.3(16)_F6-1

Figure 5.3: Building blocks of the ITU-T P.1203 model. Source: [1], Figure 1.

The ITU-T P.1203 model receives media stream information and playback device features to generates the inputs (*I.11, I.13, I.14, I.GEN*) for the internal modules (*Pa, Pv, Pq, Pav, Pb*). The model generates the following input signals:

- I.GEN: Playback display resolution and device type.
- I.11: Information on played audio segments, including audio codec and representation features.
- I.13: Information on played video segments, including video codec and representation features.
- I.14: Stalling event information, including stalling start time and its duration.

The inputs may be extracted or estimated in different ways since the ITU-T P.1203 does not provide information on *Buffer parameter extraction* and *Media parameter extraction* modules. The internal modules process the inputs signals to achieve several output QoE scores:

- O.21 and O.22: *Pa* and *Pv* modules provide one score per sampling interval for audio and video, respectively.
- O.34, O.35 and O.23: *Pav* and *Pb* modules provides cumulative scores for audio-visual and buffering, respectively.
- O.46: *Pq* module integrate audio-visual and buffering scores to provide the overall score.

All the outputs have 1-5 quality scale, where "1" means "bad" quality and "5" means "excellent" quality, according to MOS specifications [84].

The ITU-T P.1203 also establishes 4 modes of operation (mode 0 to 3) [1]. Mode 0 employs only content metadata. All the other modes work only with unencrypted content to acquire information from the media stream. Modes 2 and 3 also require decoding it. Consequently, if we employ mode 1-3 at the MEC Proxy, it may cause security issues. For this reason, we employ mode 0 for our MEC Proxy. Moreover, mode 0 is also the less intensive in terms of processing.

A software implementation of ITU-T P.1203 standard internal modules is provided in [399]. The software implements the internal modules [99] according to the ITU-T P.1203 and provides customized *Media parameter extraction* and *Buffer parameter extraction* modules (the ITU-T P.1203 does not specifies these modules). These customized modules are useful to generate compliant inputs signals and evaluate the internal modules, but they are limited for working with locally stored video files. So, they could not be used while streaming a content. All the modules provided by this software implementation are also capable to analyse the media content through any of the four available modes [400].

To feed the ITU-T P.1203 while streaming a content, we have designed and implemented our custom solutions to generate the inputs. *I.GEN* can be easily known by analysing the header of the HTTP requests since it contains a User-Agent field [401] that allows the recognition of the HTTP client type (mobile or desktop device, browser or application, etc.), while keeping anonymous the identity of the user. To extract the remain input signals, we design both *Media parameter extraction* and *Buffer parameter extraction* modules which execute Algorithm 4 and Algorithm 5, respectively.

Algorithm 4 Media parameter extraction

```
  function MEDIAPARAMETER(Manifest, segment$_n$, {segment}$_{n-1}$)    ▷ for each downloaded segment
Input: Manifest                                      ▷ media manifest
Input: segment$_n$                                   ▷ current downloaded segment
Input: {segment}$_{n-1}$                             ▷ session information including last segment
Output: {segment}$_n$                                ▷ session information updated with current segment
      d$_n$, res$_n$, fps$_n$, codec$_n$ ← parseManifest(Manifest, segment$_n$) ▷ segment duration, resolution, framerate and codec
      size$_n$ ← getBitSize(segment$_n$)                ▷ current segment size
      bitrate$_n$ = $\frac{size_n}{d_n}$                 ▷ current segment bitrate
      {segment$_n$} = {d$_n$, res$_n$, fps$_n$, codec$_n$, bitrate$_n$}   ▷ current segment information
      {segment}$_n$ ← {{segment}$_{n-1}$, {segment$_n$}}   ▷ updated session information
```

The Algorithm 4 is executed for each segment downloaded by the MEC Proxy. It receives the media manifest, the most recent downloaded segment, and the accumulated metadata for the past downloaded segments of a specific session. In the case of employing MPEG-DASH video streaming technology, the manifest would be a Media Presentation Description (MPD). First, specific metadata is captured by matching the manifest and the segment URL, identifying the specific selected representation by the video player for each segment time slot. Second, a more accurate metric in terms of the actual bitrate is captured from the segment size and its nominal duration. Usually, the duration of the segments is fixed as the Group of Pictures (GOP) size is fixed at the encoders to start the segment with a keyframe, so the nominal duration declared in the manifest is accurate. Once all the metadata for the current segment ({segment$_n$}) are collected, i.e., bitrate, duration, resolution, framerate and employed codec, this information is stored for all the session long ({segment}$_n$), updating the series for the last downloaded segment ({segment}$_{n-1}$).

The Algorithm 5 is executed at the MEC Proxy once each segment download is completed by the video player. It estimates stall occurrence and duration without access to video player buffer or playback issues. The MEC Proxy is remotely inferring the playback issues from the timestamps on video player requests. As the video players download a

Algorithm 5 Buffer parameter extraction

```
  function BUFFERPARAMETER(n, d, $t_0$, $t_n$, $\{stall\}_{n-1}$)    ▷ for each downloaded segment
Input: n    ▷ segment index
Input: d    ▷ segment duration
Input: $t_0$    ▷ first segment download time
Input: $t_n$    ▷ current segment download time
Input: $\{stall\}_{n-1} = \{\{start_0^{stall}, d_0^{stall}\},...,\{start_k^{stall}, d_k^{stall}\}\}$    ▷ session stalls including last segment
Output: $\{stall\}_n$    ▷ session stalls updated with current segment
    $k \leftarrow \{stall\}_{n-1}$    ▷ number of estimated stalls along the session
    $D^{stall} = \sum_{i=0}^{k} d_i^{stall}$    ▷ total duration of estimated stalls along the session
    $t_{playback} = (t_n - t_0) - D^{stall}$    ▷ playback time
    $d_{downloaded} = (n-1)*d$    ▷ total duration of downloaded segments
    $d^{stall} = t_{playback} - d_{downloaded}$    ▷ candidate stall duration
    if ($d^{stall} > 0$) then    ▷ check if stalling in the recent segment
        $d_{k+1}^{stall} = d^{stall}$    ▷ record stall duration
        $start_{k+1}^{stall} = t_n - d^{stall}$    ▷ stall start time
        $\{stall_{k+1}\} = \{start_{k+1}^{stall}, d_{k+1}^{stall}\}$    ▷ estimated parameters of new stall (start time and duration)
        $\{stall\}_n = \{\{stall\}_{n-1}, \{stall_{k+1}\}\}$    ▷ update stall series
        $k$++    ▷ new stall when playing the current segment
    else
        $\{stall\}_n = \{stall\}_{n-1}$    ▷ unchanged stall information
```

new segment once another has been played, the inter-arrival on video player requests should follow segment duration. Any identified drift likely means a buffering issue. To this end, this function, also executed for each downloaded segment, gets the duration of the segment, the download timestamps of the video player from the segment provided by the MEC Proxy, and the records of past estimations of stalls for previous segment time slots in the session. First, the total duration of estimated stalls along the session is calculated. Then, the current playback time is measured from the elapsed time from the player's start-up to the last downloaded segment, including the total estimated stall for all the session. The MEC Proxy assumes the first downloaded segment time as the start-up time. It is a reasonable choice since the player starts decoding and displaying the content only when the first segment is completely downloaded. With this assumption, an error could be introduced on the start-up time selection since the proxy cannot measure decoding delay of the first frame. In any case, decoding operation is done in real time, meaning that the maximum error is in the order of tens of milliseconds (1/framerate). Downloading time duration and player's internal buffer size are in the same order of magnitude of segment duration (seconds), which is 100 times higher than the decoding delay. Thus, the error due to decoding delay has a negligible impact on the overall measurement of the start-up time. To calculate the duration of content available at the video player, the algorithm only includes the previously downloaded segments ($n-1$) excluding the most recent one (n). As this function is evaluated just once the segment has been downloaded by the video player, the most recent segment has not been decoded yet. If the current playback time ($\mathrm{t}_{playback}$) is lower than the duration of content available at the video player ($\mathrm{d}_{downloaded}$), the video player has buffered content to be played, so stall should not happen. If the playback time is ahead the duration of available content, the stall duration is estimated (d^{stall}), and the start time of this stall is assumed shifting current segment download time (t_n). Last, estimated timestamp and duration of detected stall is stored.

5.2.4 **Results**

5.2.4.1 **Experimental setup**

To demonstrate the effectiveness of the proposed approach to assess the ITU-T P.1203 [1] QoE scores, we deployed the testbed shown in Figure 5.4. The experimental setup comprises the following:

- Media server: we use a mirror server located at Polytechnic University of Turin and belonging to D-DASH dataset and infrastructure [378] which provides a Dynamic Adaptive Streaming over HTTP (DASH) standard content.

- MEC proxy and access point: this is a unique node deployed through a Node.js [363] proxy application running on a Raspberry Pi 3 single-board computer [402]. Raspberry Pi 3 includes both computing capabilities to execute edge processing and a WiFi internal module to deploy a local wireless network such to provide connection to WiFi clients. Thus, this design guarantees ultra-low latency, proximity and high bandwidth [41]. This node has Internet access to download the MPD and the corresponding media segments stored at the Media Server which are served to the clients on demand.

- UE node: client node featuring WiFi interface and running several DASH players, based on GStreamer multimedia framework [353], which download the video stream. Players run headless (without GUI) since we are not interested in displaying the content and allowing the client to save computing capabilities.

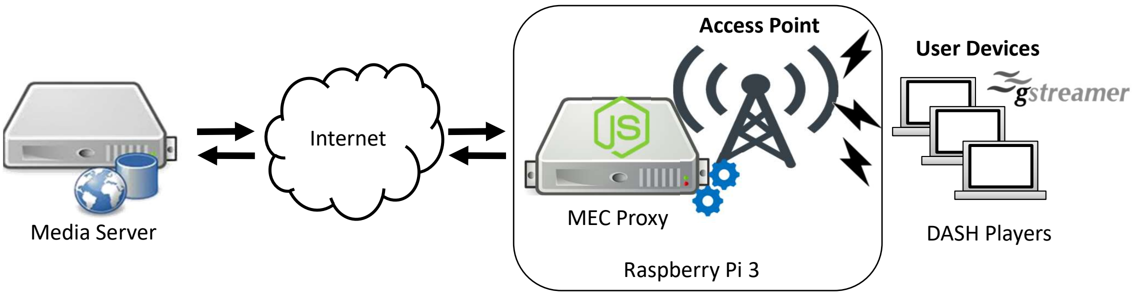


Figure 5.4: Testbed.

Node.js application at MEC node employs *http-server* module for acquiring HTTP request and response time and *xml2js* module for parsing the MPD to feed Algorithms 1 and 2. During the experiments, Algorithms 1 and 2 are also implemented inside Node.js application and run in real time to generate the inputs of ITU-T P.1203 model. However, ITU-T P.1203 QoE evaluation is performed offline, after that all the metrics are collected.

Thus, it simplifies the process of metrics collection and QoE evaluation, while the QoE results remain valid as the input metrics for the ITU-T P.1203 model do not change.

The dataset at the Media Server includes the Red Bull Playstreet video sequence, which is owned by Red Bull Media House and licensed for scientific purposes. This sequence is encoded for 17 video representations through advanced video coding (H.264/AVC) and 4 dual channel audio representations through advanced audio coding (AAC). Both audio and video are segmented with different segment lengths of 2, 4, 6, 10, and 15 seconds, and multiplexed in ISO MPEG4 files (ISO/IEC 14496-12 - MPEG-4 Part 12). For our experiments, we employed 6 seconds segments as such duration favour accuracy of the proposed solution when delivering segments with enough size in bytes to get a more solid assessment of the network performance while downloading, while the serving latency is still valid for streaming of live events.

The representations range from a resolution of 320 x 240 and 30 fps at 100 kbps to a resolution of 1920 x 1080 and 30 fps at 6000 kbps. As the client-side bitrate adaptation mechanism does not target a specific resolution, then each client struggles to achieve the highest representation bitrate.

We use the outcomes of [131] to model players behaviour. The authors provide an extensive analysis on user behaviour while accessing streaming services. The player inter-arrival time fits a modified version of the Poisson distribution. This means that the players are starting and stopping their sessions (joining and leaving the cell/hotspot) along the experiment according to a Poisson distributed inter-arrival time. Moreover, the duration of streaming sessions of the players is variable and follows the declared sections of 5 (37.44%), 10 (52.55%), or 25 min (75.25%). As a result, the effective number of players employed during the experiments is variable since the model is aleatory. Modelling players inter-arrival time and their streaming session duration according to a real distribution allows to emulate a real media traffic scenario.

To test with different network loads, we also consider two different scenarios where we change the limit of the maximum number of concurrent players:

- Scenario 1: 10 players at a time. Here, no more than 10 players at a time are connected to the WiFi access node and downloading the content.
- Scenario 2: 20 players at a time. The number of players concurrently consuming the video streaming is increased to 20.

We perform a test over each scenario for one hour. It results in 66 players taking part in the first scenario and 144 players participating in the second scenario.

5.2.4.2 **Evaluation metrics**

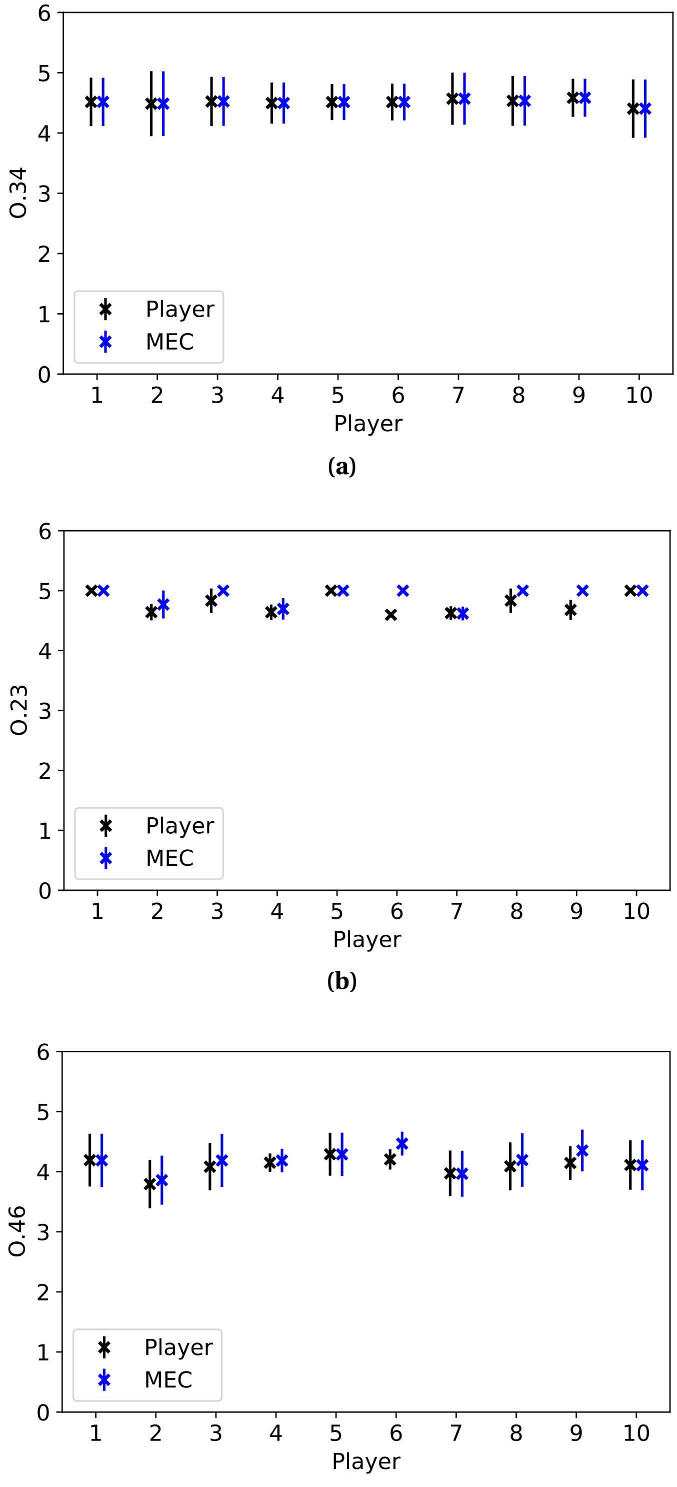
To evaluate the proposed solution, we employ the outputs of the ITU-T P.1203, which provide subjective evaluation of the streaming sessions. Since we are focused on evaluating QoE of video representation, we target the following outputs from the ones analysed by ITU-T P.1203:

- O.34: it provides a video quality score per output sampling interval. The default interval of the ITU-T P.1203 implementation is 1s, so we have 6 new values per every downloaded segment.
- O.23: it provides overall score considering stalling events. We update this value every time the player downloads a new segment.
- O.46: it provides overall score overall quality score, considering video and stalling events. We update this value every time the player downloads a new segment.

To check the accuracy of the values obtained by the MEC Proxy, we compare the outcomes at the MEC with the outputs at the player side. At the player it is not necessary to design algorithms to extract, estimate or infer the target signals of the ITU-T P.1203 model since all of them are directly available at the video player. The player establishes the video representation, then it knows the features of the downloaded segment after selecting one. Moreover, to know if a stall is experienced, it is enough to check when the internal playback buffer goes empty.

5.2.4.3 **QoE estimation at the Edge results**

As described in Section 5.2.4.1, the player inter-arrival time and session duration are modelled as described at [131]. While testing the proposed solution for one hour, it resulted in 66 video players for Scenario 1 and 144 for Scenario 2. In terms of computational capabilities consumption, Raspberry Pi 3, which runs the MEC Proxy (Algorithms 1 and 2) and the access point, experiences 13% CPU and 122MB RAM usage during Scenario 1 and 17% CPU and 156MB RAM usage during Scenario 2.



(a)

(b)

(c)

Figure 5.5: Average and standard deviation of ITU-T P.1203 QoE scores for 10 players from Scenario 2: O.34 (a), O.23 (b) and O.46 (c).

Table 5.1: Scenario 1: number (N_{stall}) and total duration (T_{stall}) of stalls.

	$\mathbf{N_{stall}}$	$\mathbf{T_{stall}}$
Player	661	289s
MEC	160	263s

Table 5.2: Scenario 2: number (N_{stall}) and total duration (T_{stall}) of stalls.

	$\mathbf{N_{stall}}$	$\mathbf{T_{stall}}$
Player	1623	757s
MEC	334	765s

Figure 5.5 shows average value and standard deviation along the streaming session of the considered ITU-T P.1203 outputs. It shows only the results for 10 players randomly chosen among the executed ones of the Scenario 2 since it is the most demanding scenario, where more stalls and quality changes are experienced. Scenario 1 has similar results, but as expected the average values are higher due to the lower competition for the available network resources. The results for O.34 (Figure 5.5a) obtained at the MEC are like the ones obtained at the player. It is reasonable since most of the video information to provide to the ITU-T P.1203 model comes from the MPD, which is available at the player, as well as at the MEC Proxy. On the contrary, the results for O.23 (Figure 5.5b) shows some differences since the Algorithm 5 may not detect all the stalling events at the player side from the MEC Proxy. The sampling time to check for stalls at the MEC Proxy is equal to the segment duration. Then, a stall experienced by the player, whose duration is short, and it is recovered before the next check at the MEC, may not be detected correctly. Consequently, estimated values at the MEC are higher that the captured ones at the player. Tables 5.1 and 5.2 also provides more in-depth details of the stalls. It is clear from the tables that the MEC cannot detect all the stalls. Anyway, the total duration of stalls tends to actual one experienced by the player. It means that MEC Proxy cannot detect micro-stalls and tends to concentrate several of them into a macro-stall. Again, this is due to the sampling time to check for stalls. If two micro-stalls are experienced during the same segment, the MEC Proxy will consider that they are one longer stall since it checks only at the end of each segment download. Finally, O.46 scores (Figure 5.5c) are directly influenced by the results obtained by the other two scores (O.34 and O.23). It has a significant standard deviation due to video

scores (O.34), but lower average value due to the impact of buffering score (O.23).

Tables 5.3 and 5.4 resume the scores of all the players executed under the Scenarios 1 and 2, respectively. The Tables shows the results by averaging the values obtained by the players and evaluating the standard deviation for each considered score. The Tables confirms that O.34 has the same values when estimated at the MEC and captured at the player, while O.23 scores assessed at the MEC tends to be higher than the real values issued at the player. Again, O.46 scores, which are influenced by the other two scores, shows a standard deviation coming from O.34 and lower average values inherited from O.23.

Table 5.3: Scenario 1: average and standard deviation of ITU-T P.1203 QoE scores.

	$\mathbf{O.34_{avg}}$	$\mathbf{O.34_{std}}$	$\mathbf{O.23_{avg}}$	$\mathbf{O.23_{std}}$	$\mathbf{O.46_{avg}}$	$\mathbf{O.46_{std}}$
Player	4.71	0.28	4.81	0.20	4.38	0.28
MEC	4.70	0.28	4.84	0.19	4.39	0.28

Table 5.4: Scenario 2: average and standard deviation of ITU-T P.1203 QoE scores.

	$\mathbf{O.34_{avg}}$	$\mathbf{O.34_{std}}$	$\mathbf{O.23_{avg}}$	$\mathbf{O.23_{std}}$	$\mathbf{O.46_{avg}}$	$\mathbf{O.46_{std}}$
Player	4.53	0.31	4.76	0.19	4.22	0.27
MEC	4.53	0.31	4.85	0.20	4.28	0.28

As expected, Scenario 1 presents higher values for any of the considered metrics since the network resources are the same as Scenario 2, but the number of players sharing them is lower. The resulting standard deviations are similar for both scenarios.

Tables 5.5 and 5.6 compare the results obtained at the MEC and at the player side by providing the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) between the different scores. The results show that the scores obtained at the MEC tends to be like the ones obtained at the player, achieving more accurate results in the Scenario 1, as expected.

Table 5.5: Scenario 1: MAE and RMSE for ITU-T P.1203 QoE scores.

	O.34	**O.23**	**O.46**
MAE	0.21	0.11	0.18
RMSE	0.39	0.21	0.36

Table 5.6: Scenario 2: MAE and RMSE for ITU-T P.1203 QoE scores.

	O.34	**O.23**	**O.46**
MAE	0.14	0.14	0.19
RMSE	0.43	0.24	0.36

Note that O.34 has higher MAE and RMSE than O.23 and O.46. It could be considered a contradictory to the fact that the values obtained at MEC Proxy compared to the values captured at the player for O.34 are more accurate than O.23 and O.46. Anyway, it is important to remember that these metrics have different sampling rates. O.34 provides a value every second, while O.23 and O.46 every segment (6s). Then, O.34 has higher values for MAE and RMSE since they are evaluated over a number of scores which is 6 times bigger than O.23 and O.46. Finally, if we compare the two scenarios, the results tend to be similar since the MAE and RMSE are relative values, they are not conditioned by the average values of the metrics. It means that the MEC Proxy performs similarly in both scenarios.

In summary, the proposed MEC Proxy can monitor network dynamics and estimate the individual QoE of each player according to Recommendation ITU-T P.1203. The results show that video related scores (O.34) estimated follow the actual ones captured at the player, while buffering scores (O.23) tend to miss some small stalls unperceived from segment to segment. Anyway, the overall scores (O.46) are still valid showing a small difference of the estimated values at the MEC from the actual values captured at the player.

5.2.5 Conclusion

The trend for the following years is an increasing consumption of media content due to the popularity of video streaming platforms. To cope with this increasing demand, MNOs need to manage the network according to cost-effective policies, requiring monitoring solutions which provide actionable data to management systems to provide the best QoE with the available network resources.

The proposed 5G MEC-enabled proxy aims to estimate the ITU-T P.1203 QoE metrics to enable edge services operating coordinated decisions on the streaming qualities. The MEC Proxy assesses QoS metrics at the radio cell and parses manifests of requested

DASH streams to estimate the parameters employed to evaluate ITU-T P.1203 QoE scores. Consequently, there is no need for explicit out-of-band messaging from video players to send playback statistics to the MEC.

The solution has been implemented in a real testbed where WiFi was employed as access network between MEC and players and tested in two scenarios with different demands and traffic loads over the network. The results demonstrate the capability of the MEC Proxy to estimate ITU-T P.1203 QoE scores close to actual ones.

Acknowledgment

This work was fully supported by the 5G-TEST project (Gipuzkoa's research and innovation programme) and Open-VERSO project (Red Cervera program, Spanish government's Centre for the Development of Industrial Technology).

5.3 MEC Proxy for efficient cache and reliable multi-CDN video distribution

- **Title:** MEC Proxy for efficient cache and reliable multi-CDN video distribution
- **Authors:** Roberto Viola, Ángel Martín, Mikel Zorrilla and Jon Montalbán
- **Proceedings:** 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)
- **Publisher:** IEEE
- **Year:** 2018
- **DOI:** `10.1109/BMSB.2018.8436904`

Abstract: The massive consumption of media contents needs of network accelerators, which boost the media delivery and optimize the traffic volume crossing the network from servers to media players. Content Delivery Network (CDN) is the common network function to distribute in a cloud-manner the contents, enhancing media availability and distribution performance. However, high concurrency rates of media sessions can produce CDN performance degradations and outages that impact negatively the Quality of Experience (QoE). 5G Multi-access Edge Computing (MEC) architecture envisions a QoE-aware system at the network edge which performs analytics to enhance or boost media services. This paper provides a novel MEC proxy to expand and enforce caching infrastructures for efficient and reliable content distribution. First, the proxy caches contents on the network edge to reduce the Capital Expenditure (CAPEX) of the CDN for the OTT service provider. To this end, the proxy is able to identify recurrent requests. Second, the proxy shields from identified or predicted CDN malfunction. Here, the proxy switches the download sessions to an alternative CDN in order to ensure QoE rates, enabling a CDN dynamic selection based on live connectivity statistics. The proposed solution is evaluated by delivering Dynamic Adaptive Streaming over HTTP (MPEG-DASH) streams in a dense client cell while applying different caching strategies.

Keywords: 5G, CDN, distributed cache, MEC, QoE, reliability & streaming analytics.

5.3.1 **Introduction**

The increase of video streaming users and the service requirements are driving the evolution of media services over the Internet. Mobility and quality improvements are the key catalysers in this evolution. Cisco estimates that video traffic will cover the 82% of all Internet traffic by 2021 [403]. Moreover, mobile traffic growth is estimated to be twice as fast as fixed IP traffic from 2016 to 2021 [404].

The delivery performance from the network has a big impact on the perceived QoE of the end user of media services. However, video streaming services work on top of unmanaged networks, where the traffic is managed and forwarded on a best-effort basis [23]. This operational context is targeted by the video streaming solution MPEG-DASH. MPEG-DASH inherits many benefits from Hypertext Transmission Protocol (HTTP). First, it enables a pull-based streaming [367] to easily traverses network functions such as firewalls and NAT devices. Second, MPEG-DASH streams can be played anywhere as any connected device supports HTTP. Third, MPEG-DASH has a Content Delivery Network (CDN)-ready design enabling the exploitation of existing HTTP caching infrastructures, which enhance the availability and the responsiveness of the content distribution.

Beyond the current network functions to boost and enhance the traffic delivery, the capacity and performance promised by 5G networks will make a significant leap towards higher data rates, heavier user densities and ultra-reliable and low-latency communications. The European Telecommunications Standards Institute (ETSI) specifies a set of component technologies which will be essential part of 5G systems, such as, Network Functions Virtualization (NFV), Millimetre Wave Transmission (mWT), Next Generation Protocols (NGP) and Multi-access Edge Computing (MEC) [14]. MEC architecture allows Mobile Network Operators (MNOs) to supply video delivery analytics-based intelligence at the network edge. Thus, MEC plays a significant role to achieve specific media traffic goals with zero-latency in a distributed manner. Furthermore, MEC enables coordinated operations at the network edge such to be transparent to the media server and players.

This work proposes a novel network proxy, called MEC4CDN, which complies with MEC architecture. MEC4CDN provides two features. First, it performs a local cache of identified trending contents to minimize the traffic between the CDN and the network edge. Second, it applies a dynamic selection of the CDN based on live measurements in the context of a multi-CDN delivery. The proposed solution allows the client to start playing from a CDN, that is favourable in terms of Quality of Service (QoS), and transparently and dynamically switch to another one which provides better performance in case of CDN degradation or outage. To this end, the proxy collects and process L3 connectivity metrics, L7 MPEG-DASH Media Presentation Description (MPD) files and L7 QoE scores. So, the proxy handles multiple CDN endpoints, deciding to locally cache a response for near future requests, or to conduct the request to a healthy cloud CDN in case of detected issues.

The paper is structured as follows. First, section 5.3.2 reviews the related work to improve the network delivery. Then, section 5.3.3 describes the proposed MEC4CDN network proxy. Section 5.3.4 presents the implementation of the solution. To validate the solution, section 5.3.5 compiles the testbed and the results. Finally, section 5.3.6 gathers the conclusions of this work.

5.3.2 Related Work

Apart from the content catalogue, the Quality of Experience (QoE) is a key aspect for user satisfaction and retention when rating streaming services. Hence, any system trying to enhance media delivery needs to consider QoE metrics.

QoS assesses the network delivery performance and it has a direct impact on human perception QoE. In this sense, stability of media playback is related to efficient utilization and fair resource sharing. Nevertheless, these key aspects are not guaranteed in unmanaged networks. This situation may lead to suboptimal results in terms of video playback, link utilization, and fairness among the clients [405].

Specific metrics for QoE of HTTP-based adaptive media streaming services, such as initial delay, stalling time, number of quality switches and inter-switching times, are fundamental parameters to get the estimated Mean Opinion Score (eMOS) [350]. This model quantifies the quality of video streaming services without a demographic perception study. Recently, the work [351] investigates a new model for MOS, called Ubiquitous-Mean Opinion Score for Video (U-vMOS), which makes initial buffering more dominant than [350] model.

Caching is a common technique to get enhanced QoE for massive content consumption services. The CDN is the traditional network function provisioning cache features as a cloud service. Fueled by the CDN vendor proliferation, media services employ multi-CDN strategies to get a more reliable and cost-effective content delivery. Netflix case, as an example of multi-CDN solution, is studied in [28]. More recent work [30] also includes Hulu analysis for 3 CDN vendors. Here, with alternative CDNs available, the employed CDN is set by the streaming service for each session, where the choice is done by the server. This centralized architecture is difficult to scale hampering the orchestration of common policies or decisions to all users in an area.

When the employed CDN is not pre-set, transferring to clients the possibility to dynamically choose, each client should analyze repeatedly the network performance of each CDN. It means the introduction of a network overhead proportional to the number of clients and the number of CDNs. Hence, a client-side CDN selection is not an optimal solution.

Consequently, the European Broadcasting Union (EBU) aims to avoid both pre-set CDN and client's CDN selection by proposing the EBU Flow Multi-CDN [219]. It consists in a CDN switching service which selects the optimal CDN at any given moment in time. A similar solution is also provided by Cedexis Multi-CDN [406]. Both EBU and Cedexis proposals monitor network analytics at core network, then they are not aware of the state of connected clients at network edge.

On the contrary, if we focus at the network edge, the access point has knowledge of both the connected clients' activity in its cell and the CDN performance from its location. A proxy located at the network edge, retrieving access point information, can evaluate the performance of the delivery network just once and exploit it for each client (independently from the number of clients in the cell). This perfectly suits the telecommunication industry which proposes MEC as a new functional architecture to be integrated on the mobile network infrastructure. MEC allows Internet service provider (ISP) to provide Information Technology (IT) and cloud computing services at the edge of the network, closer to the clients. Hence, MEC opens the door for authorized Content Providers (CPs) to develop their own applications hosted on the MEC server. Therefore, CPs are able to tune the behavior of content delivery to end users in 4G and 5G contexts. Furthermore, next generation networks and 5G MEC architecture enable the deployment of distributed local caches at the network edge to efficiently minimize the volume of traffic passing through the network core and backhaul.

5.3.3 MEC Proxy for Multi-CDN Delivery

The proposed MEC4CDN proxy server, located on a MEC component, can exploit the knowledge of network analytics of different delivery paths. It aims to improve the overall throughput while minimizing the latency, in an environment where multiple clients are competing for network resources.

Figure 5.6 shows a scenario where a content is provided by means of two CDN vendors. For some reasons, such as sudden massive connections or geo-based cycles of human activity, the performance of *CDN A* starts to degrade. At the same time, the capacity of *CDN B* is improved because the *CDN B* starts to provision more resources for a specific area. The MEC4CDN system gets awareness of this unbalanced situation, based on L3 stats, and orchestrates all the media players subscribed to the managed cell to dynamically employ a CDN which ensures better QoS.

Therefore, for services delivered over multiple CDN providers, MEC4CDN can select an appropriate CDN for a RAN geo-position in real-time, according to L3 metrics. To this end, MEC4CDN gets alternative CDNs set from the MPD file, provided that it contains multiple definitions of *base URL* storing the path to the CDNs, or from the media service. Then, MEC4CDN employs the set of CDNs to dynamically switch the base URL

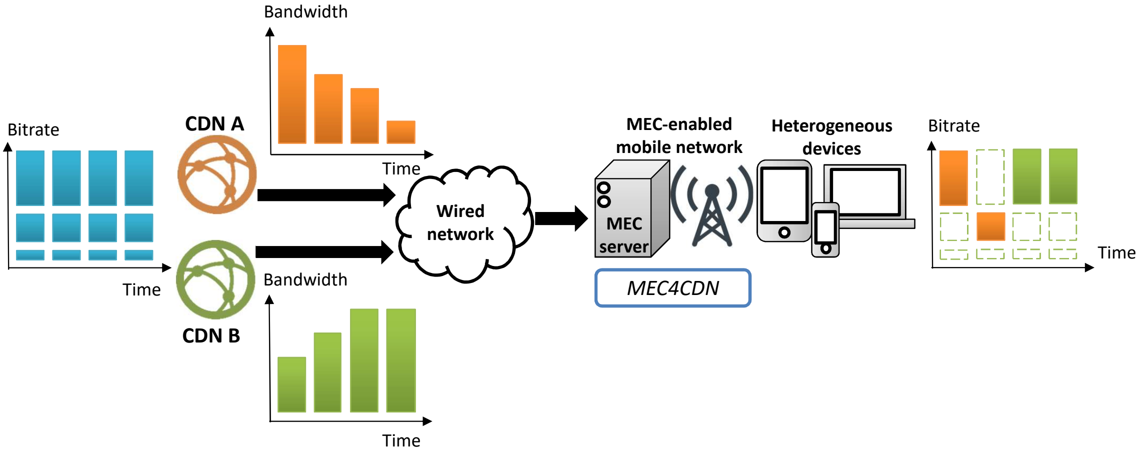


Figure 5.6: MEC4CDN multi-CDN selection.

of media sessions in the same RAN when necessary. So, in case of detected performance degradation, the MEC4CDN system replaces the base URL field of all the managed sessions to another known CDN endpoint, migrating all the managed clients at once to avoid outages.

The sequence diagram with the exchanged messages to allow the CDN switching is depicted in Figure 5.7. First, the MEC proxy server running MEC4CDN captures the HTTP GET requests from the User Equipments (UEs) to download the MPD file from the media server. Then, MEC4CDN retrieves the MPD file from the media server and appropriately parses it in order to retrieve the *base URLs* set before sending it to the UE. Then, the UE selects a representation bitrate (R_j) from the available ones according to display resolution, user preferences and connection capacity stats. The UE requests, through the MEC proxy, a specific segment file to the CDN accordingly. Once the MEC proxy server detects the HTTP GET requests from the UEs for downloading a segment file, it retrieves wired path stats. When network performance is not enough, the MEC4CDN switches the *base URL* field of the MPD file applicable for the next segment requests. Such operations are executed at the stream start and each time that the UEs asks for a new segment.

Furthermore, MEC4CDN ships the ability to significantly reduce the CDN traffic for a live content distribution in dense client cells, as depicted in Figure 5.8b. In a live media consumption scenario, each UE makes a request to get the video from a CDN. When MEC4CDN comes into play, it performs a local cache at the network edge in a proac-

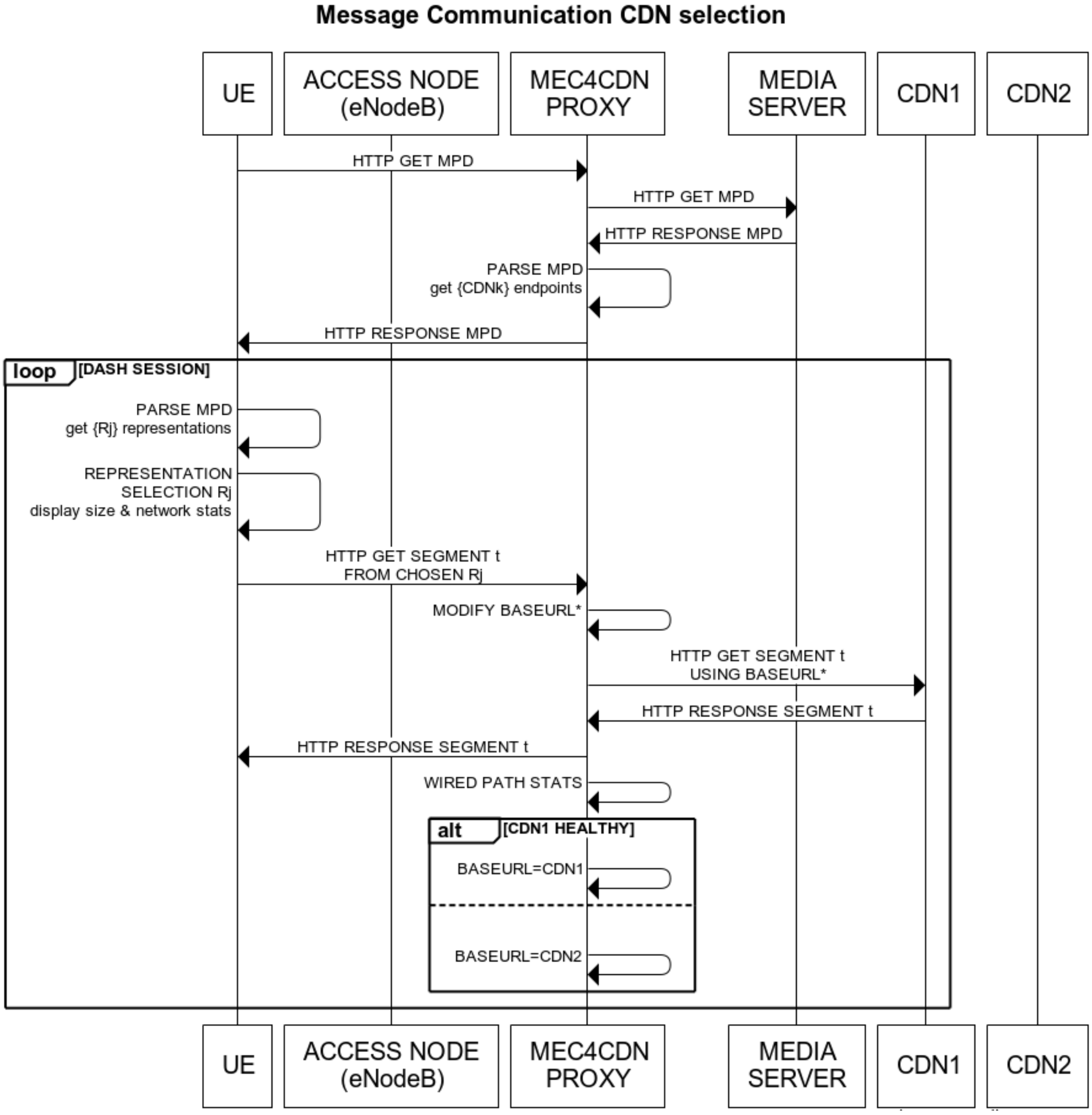
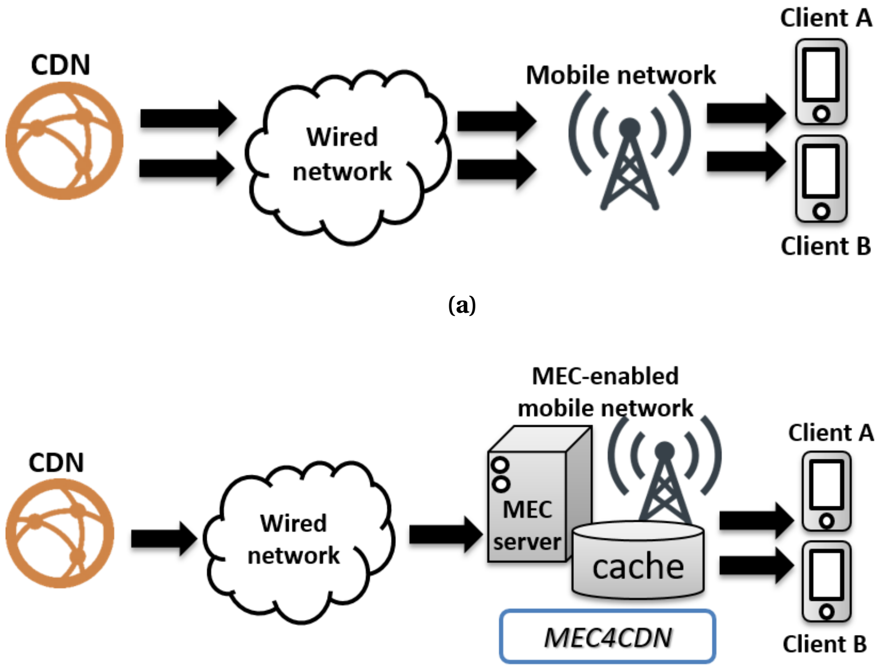


Figure 5.7: MEC4CDN multi-CDN sequence diagram.

tive manner. Triggered by the first content request, MEC4CDN downloads all the next segment representations from the CDN. Then, all the following requests are already downloaded and available in the MEC4CDN local cache for any representation bitrate chosen by the media players. Then, the greater the number of clients consuming a live content in a cell, the higher efficiency of this solution.



(a)

(b)

Figure 5.8: Legacy content delivery (a) and MEC4CDN cache-powered delivery (b).

The utilization of a CDN means a significant operational cost for media services. Moreover, when massive connections come concurrently to a CDN, the CDN infrastructure can start to collapse and the QoS could be negatively impacted. To prevent media delivery from this situation, it is necessary to turn broadcast of live content into an efficient distribution. To this end, MEC systems located at the network edge can cache recurrent contents, such as live sports events or concerts, in a proactive manner. Cache at network edge can improve the streaming experience caching all available representations. Thus, the selected representation will be locally available for any number of subscribers in a cell, enabling the video to start faster and reducing the buffering time, while minimizing CDN traffic. This approach preserves media players ability to decide the bitrate that fits with their contexts independently.

The sequence diagram with the exchanged messages to allow the local cache of a popular or live content is depicted in Figure 5.9. The proxy waits for a media content request and proceeds to download all the available representations. Then, the media players in the cell will select a specific representation (R_j) according to the device features, the assessed available bandwidth, the user preferences and subscription plan. Once the players start to request a representation for a specific time (t), the proxy requests the next segments ($Segments_{t+1} \forall R_j$) to be ready for the following requests.

It is also important to remark the zero-latency and distributed performance features by design coming from the MEC architecture fundamentals. The MEC systems are distributed and autonomously empower specific services for the co-located cell.

5.3.4 Implementation

The program of MEC4CDN for CDN selection is described in Algorithm 6. The outcome of MEC4CDN is to identify violations of bandwidth or latency performance in order to perform reactive switching to an alternative CDN provider. The inputs of the algorithm are the RTT and the bandwidth from the MEC4CDN proxy to the CDN and the current number of media playing sessions to this CDN. The output is the base URL field of the updated MPD file to be used by the media player.

Moreover, the program of MEC4CDN for local cache is described in Algorithm 7. The outcome of MEC4CDN is to proactively download what the users will play in order to locally cache next segments before the player needs to play them, enabling the video to start faster, decreasing stalls, reducing CDN traffic and enhancing the quality of experience. Thus, MEC4CDN reduces resource usage coming from the CDN provider by caching requested contents and exploiting their recurrent likelihood. The inputs of the algorithm are the current segment index employed by the media player and the available representations. The result is the temporal storage of next segments to be requested beyond the current one played by the user.

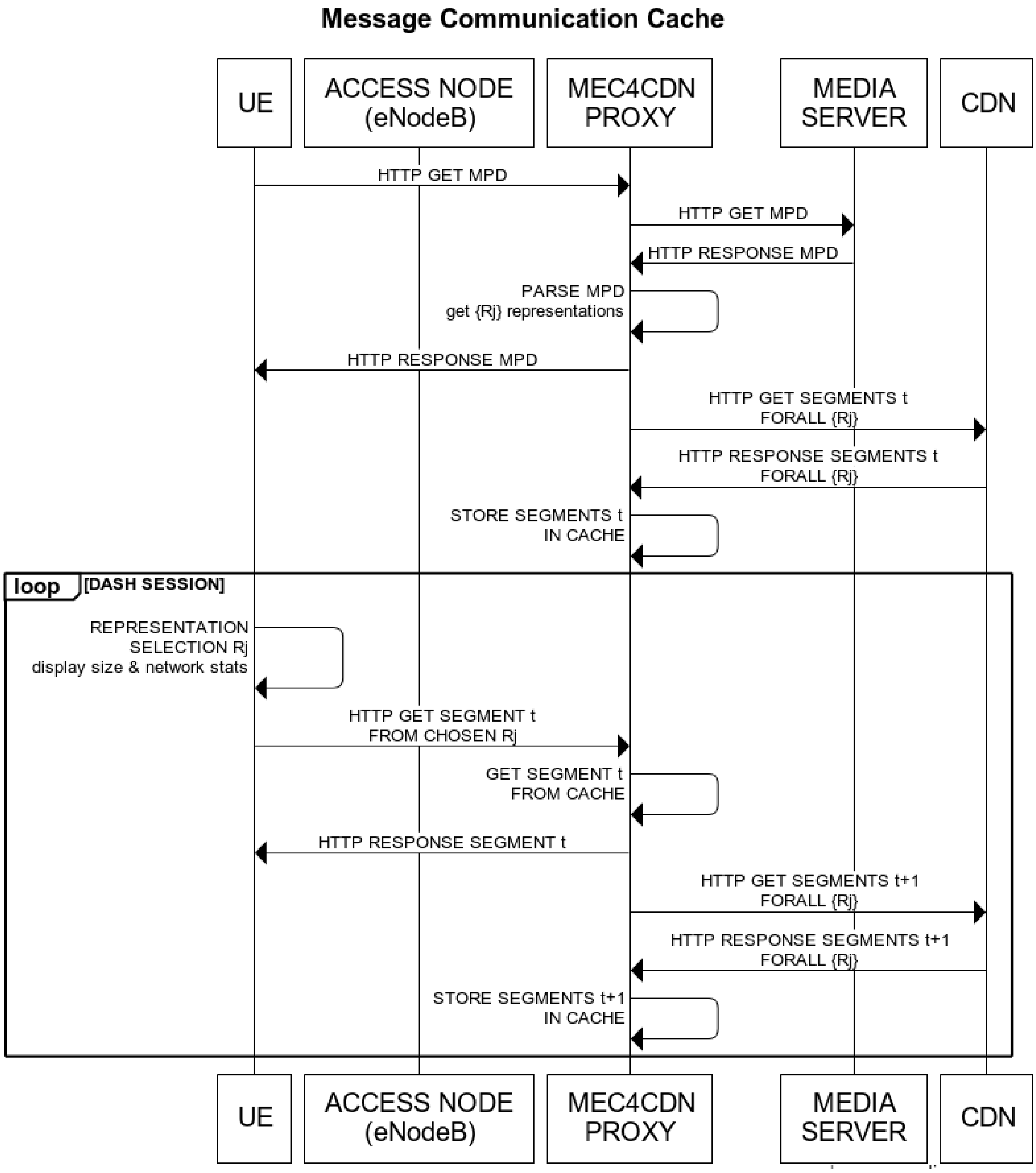


Figure 5.9: MEC4CDN cache sequence diagram.

Algorithm 6 CDN health check to switch all sessions at eNodeB

```
procedure CDNCHECK( )                                        ▷ listen to requests & switch CDN
    $rtt_{max}$                                              ▷ setup maximum latency
    $CDN_{list}$                                             ▷ set of alternative CDNs
    $n_p$                                                    ▷ number of players
    for all segment request do                               ▷ from the UE
        SegmentRequest()                                     ▷ to the $CDN_k$
        $s_{CDN_k} \leftarrow$ SegmentResponse()             ▷ from the $CDN_k$
        $rtt_{CDN_k} \leftarrow rtt(s_{CDN_k})$              ▷ L3 latency
        $bw_{CDN_k} \leftarrow 2\frac{size(s_{CDN_k})}{rtt(s_{CDN_k})}$    ▷ L3 BW
        $R_{max}^{bitrate} \leftarrow$ parse(MPD)            ▷ maximum representation bitrate
        if ($rtt_{CDN_k} > rtt_{max}$) || ($bw_{CDN_k} < (n_p$ x $R_{max}^{bitrate}$)) then
                                                             ▷ insufficient performance
            baseURL $\leftarrow$ alternative($CDN_{list}$,$CDN_k$)
                                                             ▷ change CDN
```

Algorithm 7 Cache proxy at eNodeB

```
procedure LOCALCACHE( )                                      ▷ listen to requests & cache segments
    $i_t$                                                    ▷ current segment index
    $\{R_j\}$                                                ▷ set of alternative representations
    for all $R_j \in \{R_j\}$ do                             ▷ download next segment
        $s_{i_t+1} \leftarrow$ Download($i_t + 1$,$R_j$)     ▷ from the CDN
        Cache($s_{i_t+1}$,$R_j$)                             ▷ at eNodeB proxy
```

5.3.5 Results

In order to test MEC4CDN solution, we deploy an experimental setup by exploiting a MPEG-DASH distributed dataset described by Lederer et al. [378] and provided for public experimentation of CDN-like infrastructures. This dataset consists in Red Bull Playstreet sequence stored at several geographically distributed mirror servers. The content is provided in 17 video representations encoded in H.264 Advanced Video Coding (H.264/AVC) and 4 dual channel audio representations encoded in Advanced Audio Coding (AAC). Both audio and video are segmented with different segment lengths of 2, 4, 6, 10, and 15 seconds and multiplexed in ISO MPEG4 files (ISO/IEC 14496-12 - MPEG-4 Part 12).

Table 5.7: Set of MPEG-DASH representations employed in the experiments.

index	bitrate	resolution	framerate
1	400kbps	480x360	30fps
2	900kbps	854x480	30fps
3	1500kbps	1280x720	30fps
4	2000kbps	1280x720	30fps
5	2500kbps	1280x720	30fps
6	3000kbps	1920x1080	30fps

Then, the overall experimental setup comprises both public network nodes and internal ones belonging to our network infrastructure:

- Three mirror servers: network nodes provided by Lederer et al. [378] which we use as CDN-like nodes for storing the media segments.
- A media server: a server in our infrastructure storing the MPD file which is requested by the client to play the content. The MPD file provides information for the player to retrieve the segments from the mirror servers. Moreover, the service employs segments with duration of 6 seconds and the video content is limited to six different representations, widely used by market services. Each representation is characterized by a specific bitrate as shown in Table 5.7.
- A proxy server running MEC4CDN: a server in our infrastructure provided by public Internet connection and acting as a network gateway for the wireless network edge deployed inside our infrastructure. Then, all the requests from the players

are processed by the proxy server before being transmitted, if necessary, to the media infrastructures on Internet. It executes the proposed MEC4CDN solution.

- A wireless access point: in order to provide wireless capabilities, an access point is used, which provides a wireless local area network (WLAN) using 2.4Ghz band. The access point is directly connected to the proxy server to provide a MEC-like architecture. The only role of the access point consists in forwarding all the incoming traffic on both directions (download and upload).

- A UE running MPEG-DASH players: a wireless network node connected to the access point, which is running GStreamer MPEG-DASH players.

The experimental setup is shown in Figure 5.10.

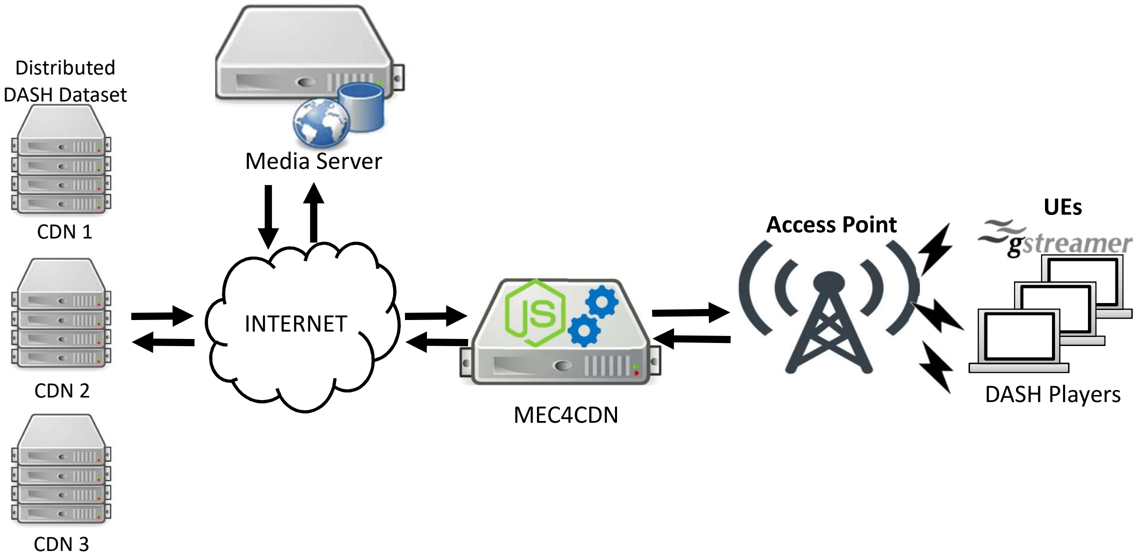


Figure 5.10: Testbed.

The tests include two different experiments aiming to verify separately the two main features of MEC4CDN:

- Multi-CDN experiment: the proxy server running MEC4CDN employs a multi-CDN infrastructure which allows CDN malfunction detection and switches the content download to an alternative CDN when necessary. In order to simulate

CDN malfunctions, we introduce a random latency between 0 and 500 milliseconds on the wired path between the MEC4CDN proxy and the three mirror servers storing the segments.

- Local cache experiment: the proxy server running MEC4CDN empowers the content delivery by caching segments at the network edge. This enables the reduction of CDN transactions, while improving the delivery with lower experienced latency.

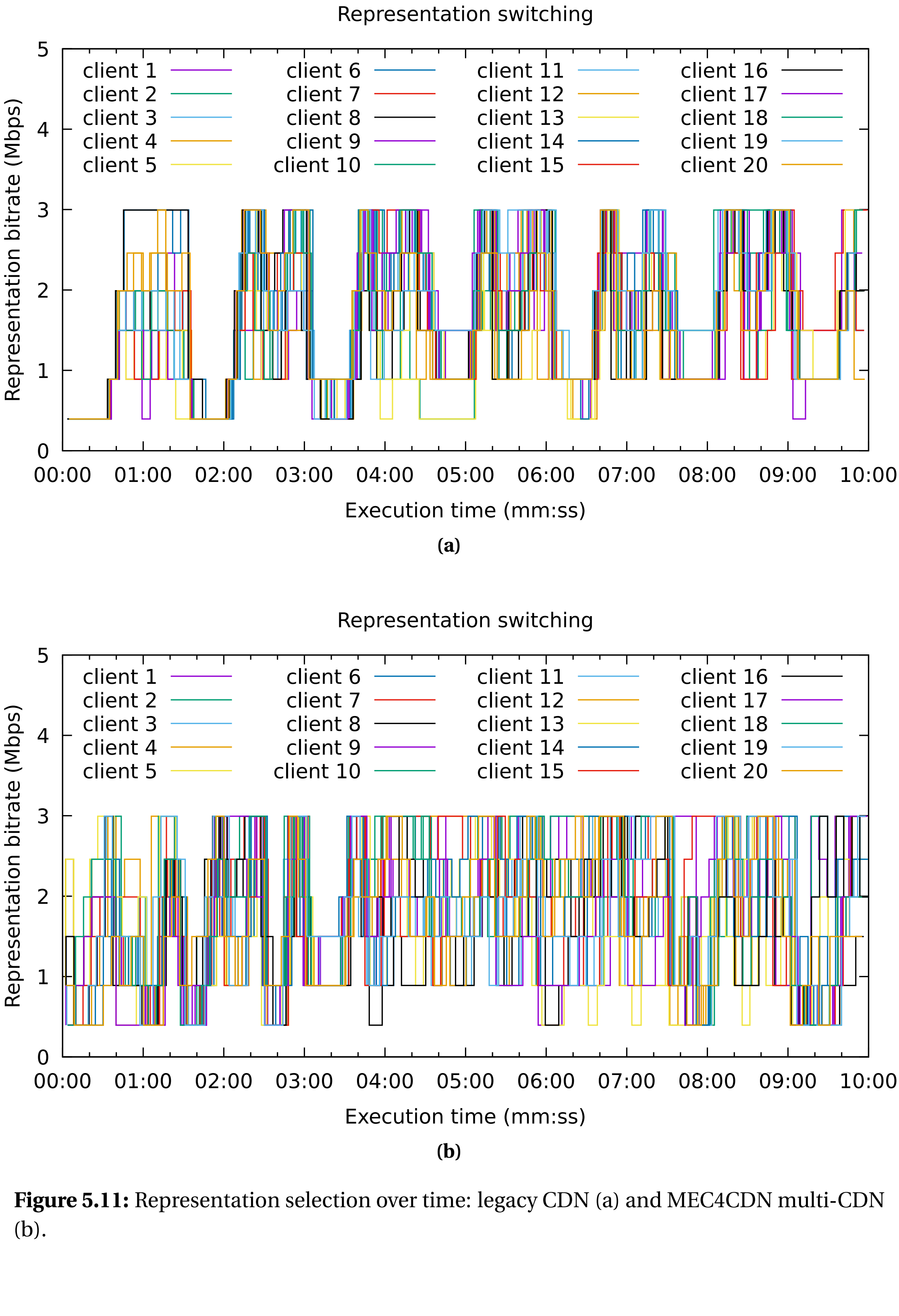
In both experiments, 20 players are sharing both core and network edge resources, competing for the shared wireless access point. The duration of each experiment is fixed to 10 minutes.

Moreover, to get evidence of the benefits of MEC4CDN solution, we carry out the two mentioned experiments where MEC4CDN comes into play and a baseline one without MEC4CDN. This basic setup provides a common delivery infrastructure where the network just forwards the requests to the first CDN. Here, GStreamer players are not able to identify CDN malfunctions, then they use always the same pre-set CDN even when it is suffering severe latency issues. Moreover, no cache is done at the edge, which means that CDN bandwidth is shared among the concurrent clients.

Concerning the first experiment, Figure 5.11 shows the representation switches along the multi-CDN experiment (10 minutes) where principal CDN suffers performance degradation. In Figure 5.11a the actors are not able to take decisions while in Figure 5.11b MEC4CDN apply strategies to switch to a healthy CDN. It is clear from the graphs that a legacy solution (5.11a) does not let the players to continue playing at the same representation level when a malfunction occurs while downloading the content. The players reduce the chosen representation bitrate in order to continue playing. In the case of a multi-CDN strategy (5.11b), some players access to higher representation bitrates since the MEC4CDN is able to enforce the delivery by switching to another CDN which better performs.

The improvements are also evident by the eMOS evaluation. The eMOS mean value among all the clients is 3.09 while downloading from a single CDN and 3.47 in case of multi-CDN delivery. This means a eMOS enhancement of +12.3%.

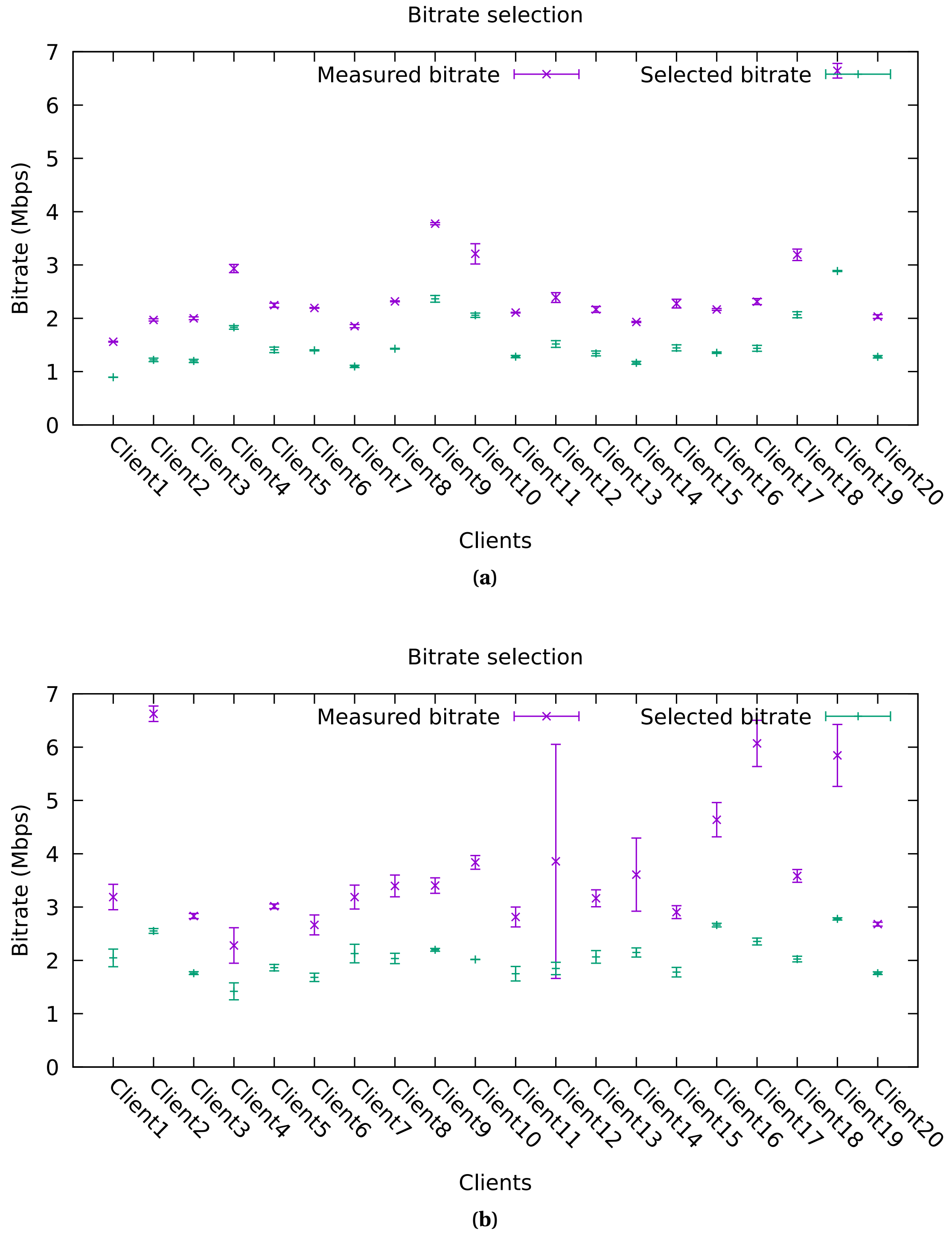
Regarding the second experiment, Figure 5.12 shows the mean value and the deviation of the measured bitrate and the representation bitrate. The results for a legacy



(a)

(b)

Figure 5.11: Representation selection over time: legacy CDN (a) and MEC4CDN multi-CDN (b).



(a)

(b)

Figure 5.12: Mean value and deviation of measured bitrate and selected representation bitrate: legacy content delivery (a) and MEC4CDN cache-powered content delivery (b).

content delivery are shown in Figure 5.12a and, when MEC4CDN with local cache delivery is added, the results improve as depicted in Figure 5.12b. This is clear since the cache lets the player to experience lower latency since the segments are closer to the clients, then the throughput is higher. Therefore, the clients tend to request higher representation bitrate which improves the user's QoE.

In this case, the eMOS mean value among all the clients is 3.08 in case of legacy delivery and 3.92 in case of cache-powered delivery, which means an eMOS increase of +27.3%.

5.3.6 Conclusion

This paper proposes a network proxy which enables multi-CDN video distribution and local cache at the network edge by exploiting the MEC architecture proposed by ETSI for future 5G networks.

The proposed proxy has two main outcomes. First, the proxy reduces the CAPEX of the CDN since it makes distributed cache at the network edge, then all the clients can play the contents received from the cache. Second, the proxy shields from CDN malfunction by switching the content download session to another available CDN such to keep QoE rates.

Finally, this proposal has been tested by performing two experiments on a real testbed. The first one exploiting a multi-CDN delivery and the second one employing local cache at network edge. In both the experiments, the results show that the MPEG-DASH players experience higher QoE compared to a legacy content delivery.

Part IV

Conclusions

CHAPTER

6

Conclusions and future work

6.1 Conclusions

This Ph.D. thesis has identified different research forks to address the main objective to improve QoS and QoE of media streaming, while reducing CP's business costs. In this sense, the exploitation of the information, acquired both from media content and from network analysis, has been decisive to increase the performance of video streaming services. The contributions have approached different stages and/or network functions involved in the video streaming workflow.

Concerning the first contribution area, *Network-aware video encoding*, strategies to encode and package the video content have been studied. Two different solutions have been designed to take into account network status when preparing the video content for streaming. The first solution enables to tune the video encoder at the SRT server accordingly to the network status. When the network throughput cannot cope with the demanded rate of the video content, the encoding bitrate and resolution are decreased to prioritize the playback smoothness over video quality. In the same way, when the throughput increase, encoding bitrate and resolution are also increased. The implemented SRT server works with compliant SRT clients without any modifications. The second solution evaluates the use of LL CMAF to reduce latency when delivering MPEG-DASH streams. Effectively, media players experience lower latency compared to a legacy MPEG-DASH solution. The latency and user's QoE trade-off is also evaluated by varying the encoding and packaging configurations, i.e., changing the GOP and fragment duration. When using an aggressive configuration with a small GOP and fragment duration, the playback is frequently affected by freezes which damage the QoE. Then, a more conservative configuration of LL CMAF is suggested to keep QoE scores.

The research on the second contribution area, *Network performance forecast for video delivery*, has investigated the use of ML algorithms to analyze network metrics and forecast performance. In a multi-CDN context, being able to forecast CDN performance means enabling a better CDN selection for the CP and reducing business cost for CDN usage. A solution that employs an LSTM model has been proposed and trained to provide CDN performance forecasts based on time series analysis of the collected network metrics. The integration of the LSTM model into the delivery chain and the exploitation of the information included in the MPEG-DASH MPD allowed the media server to take actions that enforce the delivery. The media server was able to modify the MPD to force the players to download the media segments from the more appropriate CDN that matches target QoS and CP's business requirements.

Finally, the research on the third contribution area, *MEC-enabled video delivery*, has led to de implementation of services to be employed on top of the novel 5G MEC architecture. The first solution consists in a MEC proxy that estimates the users' QoE according to ITU-T P.1203. QoE scores are derived by inferring monitored QoS metrics and the information acquired by parsing the MPEG-DASH MPD. It works independently of the video servers and players, as it does not need an explicit out of band messaging. The awareness of QoE values is an important enabler for advanced solutions to enforce the QoE at the MEC platform. In the second solution, MEC location is exploited to provide a service to enable a MEC-empowered delivery having two main advantages over legacy server-client communication. First, it proactively caches MPEG-DASH segments at network edge to reduce cloud CDN usage. Second, it shields from identified or predicted CDN malfunctions by switching the download of segments to an alternative CDN in order to ensure QoE rates. Thus, the implemented MEC service allows to keep the QoE scores by switching to healthy CDNs or even improve them by proactively caching the content at the edge.

In a nutshell, this research work provides progress beyond the state-of-the-art for video streaming. Architectures, systems and algorithms have been proposed to advance in three contribution areas. The feasibility of all the contributions has been demonstrated through the implementation of the proposed solutions and their deployment in operational and realistic setups. The results obtained have been also compared with legacy solutions to provide evidence of the improvements introduced in video streaming.

6.2 Future work

During the development of research activities, literature review, design and implementation of solutions, and analysis of results, several future research lines have been identified to complement or extend the research presented in the contribution areas of this thesis.

Regarding the first contribution, the identified future works are:

- *Business cost for encoding:* once the video codec has been chosen, the encoding operations may have operational costs that vary depending on the encoder choice, i.e., open-source or commercial, and where the encoder runs, i.e., cloud or on-premise encoding. Cloud encoding prices are established by cloud providers, while on-premise encoding depends on the hardware selection and maintenance. On the other side, on-premise encoding allows to have more control on the processed content compared to cloud encoding. In this context, the efficiency of the encoding operations could be furtherly increased by including considerations on business cost.

- *End-to-end latency:* while for VOD streams, HAS solutions, such as MPEG-DASH and HLS, are de facto standard, Live streaming still resists from a widely adoption of such solutions. This is still valid when enhancing HAS with LL CMAF. The reason is quite obvious as HAS cannot still compare in terms of latency with protocols originally designed for low latency applications. In this sense, WebRTC has raised in the last few years and is proliferating thanks also to the COVID-19 pandemic. As a drawback, WebRTC is not simple to scale since it employs specific signaling protocols, such as STUN and TURN, which add bootstrapping signaling and overheads when compared to HAS solutions. On one side, the future research will investigate how to improve LL CMAF and push its adoption. On the other side, it will try to increase WebRTC scalability.

Then, concerning the contributions of the second area, the future research activities should address:

- *Advanced network metrics:* this research has focused on employing network layer measurements (bandwidth and latency) to train a ML model and exploit the predictions jointly with application layer information (MPEG-DASH MPD). The model can be furtherly improved by collecting more complex metrics, including data link layer information, such as transmitted packets or packet losses. The higher the number of metrics processed, the more accurate the ML model would be.

- *Complex time series models:* research work in literature concludes that there is not an optimal time series model, as the selection depends on the particular application or physical network. New studies are investigating the possibility of combining different models at the same time. The idea is to exploit advantages of each model to provide better forecast results.

- *SDN integration:* the investigated solution takes actions to optimize the streaming process when the network assets' capabilities vary. Such optimization is limited to act on top of the network (changes are only operated at the media server), as no changes are applied at network layer. The traffic between server and clients are still transmitted on a best-effort basis. Here, it is interesting to move the optimization also to the network layer. It means enabling the direct management of the network assets and not just limiting monitoring them. In this context, the integration of SDN represents a further step. Employing a SDN controller to guarantee the necessary network layer capabilities between CDN and player and designing its cooperation with network functions (origin and media servers, players, etc.) create a more reliable and efficient end-to-end streaming system.

Finally, the future lines related to the third contribution are:

- *RNI standardization and processing algorithms:* the API to access RAN information or RNI has been recently standardized and its development is still ongoing.

When RNI Service (RNIS) implementations will be available, services running at the MEC host can be further optimized and embed more complex and precise algorithms. Improved algorithms will exploit RNI in order to adjust the operations and the performance of the overall system.

- *Business model and hardware acceleration:* MEC is still missing a business model equivalent to the one applicable in cloud computing infrastructures. However, unlike cloud computing, the decentralized location and utilization of shared resources between services makes the cost model more complex. Resource accounting and monitoring have to be determined in order to create a complete business model. The debate on the business model is even more intricate if hardware-acceleration assets, such as GPUs, are considered. Integrating GPU clearly provides capabilities to accomplish critical tasks where general-purpose hardware (CPU) has limitations. Once the use of GPU and the corresponding business model is clear, the debate on how to optimize resource and business cost trade-off could be raised.

- *Mobility:* the explosion in availability and type of mobile devices (e.g., smartphone and tablets) involves an increasing number of UEs to be served. Thus, mobility remains a major concern. The same way the connectivity is guaranteed when moving from a cell to another in a cellular network, migration support for MEC services is also required. Consequently, the investigation on a multi-MEC cooperation should be addressed in order to guarantee seamless migration of sessions across MEC hosts.

Part V

Appendix

APPENDIX

A

Other publications

Apart from the publications directly related with this thesis, the following list shows other publications carried out by the author of this work.

A.1 International journals

A.1.1 Journal paper OJ1

Title: Network Resource Allocation System for QoE-Aware Delivery of Media Services in 5G Networks
Authors: Ángel Martín, Jon Egaña, Julián Flórez, Jon Montalbán, Igor G. Olaizola, Marco Quartulli, Roberto Viola and Mikel Zorrilla
Journal: IEEE Transactions on Broadcasting
Pages: 561-574
Publisher: IEEE
Year: 2018
DOI: 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 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 and high efficiency video coding 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.

A.1.2 **Journal paper OJ2**

Title: MEC for Fair, Reliable and Efficient Media Streaming in Mobile Networks
Authors: Ángel Martín, Roberto Viola, Mikel Zorrilla, Julián Flórez, Pablo Angueira and Jon Montalbán
Journal: IEEE Transactions on Broadcasting
Pages: 264-278
Publisher: IEEE
Year: 2019
DOI: `10.1109/TBC.2019.2954097`
Abstract: *Beyond the advanced radio capabilities, 5G means a digital transformation, catalyzed by cloud technologies, making the networks agile and broader. However, high and quick dynamics in dense client cells consuming live broadcast contents can cause*

Quality of Experience (QoE) degradations. Here, inaccurate bandwidth assessment of media players drives to buffering times along with quality fluctuations. Moreover, massive recurrent requests can negatively impact on Content Delivery Network (CDN) performance. Complemented by capillarity and zero-latency features of multi-access edge computing (MEC) systems, 5G infrastructures will expand media services to take QoE to a new level. This paper investigates QoE gains of an MEC enabled infrastructure. The proposed MEC system applies three video delivery mechanisms. First, it enforces the QoE in a congested cell. Second, it shields from CDN degradation for a reliable content distribution. Third, it enhances network core and backhaul efficiency saving CDN traffic. Furthermore, our solution is deployed and tested on a LTE infrastructure. Results for live streams show that the MEC system makes the media players tend to a common and high quality bitrate, and it is able to quickly, transparently and coordinately switch to healthy CDN infrastructures and reduce CDN traffic.

A.1.3 **Journal paper OJ3**

Title: LAMB-DASH: A DASH-HEVC adaptive streaming algorithm in a sharing bandwidth environment for heterogeneous contents and dynamic connections in practice
Authors: Ángel Martín, Roberto Viola, Josu Gorostegui, Mikel Zorrilla, Julian Florez and Jon Montalban
Journal: Journal of Real-Time Image Processing
Pages: 2159-2171
Publisher: Springer Berlin Heidelberg
Year: 2019
DOI: `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 criterion. 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.

A.2 International conferences

A.2.1 Conference paper OC1

Title: Realising a vRAN based FeMBMS Management and Orchestration Framework
Authors: Alvaro Gabilondo, Javier Morgade, Roberto Viola, Juan Felipe Mogollón, Mikel Zorrilla, Pablo Angueira and Jon Montalbán
Conference: 2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)
Pages: 1-7
Publisher: IEEE
Year: 2020
DOI: `10.1109/BMSB49480.2020.9379891`

Abstract: *FeMBMS is the first broadcast only profile standardized in 3GPP. Rel-14 enables large scale transmission of multimedia content to mobile portable devices including free to air reception of TV services. While the new specification already meets most of the 5G-Broadcast requirements it is also expected to be further evolved in future 5G/3GPP releases. Moreover, in parallel to 5G standardization, a transition in the Radio Access Network (RAN) infrastructure is also taking place, transition where the virtualization of radio access technologies through the use of commodity processing hardware promises to make an end-to-end cloud based 5G network infrastructure a reality. In this paper we investigate first the potential of vRAN based 5GBroadcast networks. Later, based on*

OpenAirInterface and the containerization of its components, we introduce the development and analysis of a Kubernetes based FeMBMS end-to-end network architecture. The results address, among others, the potential of vRAN to foster the broadcast industry requirements in a 3GPP ecosystem.

A.2.2 **Conference paper OC2**

Title: L3 and L7-driven Dynamic Throughput Balancing over Cellular Networks
Authors: Alvaro Gabilondo, Roberto Viola, Ángel Martín, Mikel Zorrilla and Jon Montalbán
Conference: 2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)
Pages: 1-6
Publisher: IEEE
Year: 2019
DOI: `10.1109/BMSB47279.2019.8971949`
Abstract: *Broadcast of live sports and events often requires the coverage of a wide area and portable transmission units for the mobile cameras. In this context, the mobile network aspires to be a professional tool companion for media production to boost mobility and alleviate costs, space and specialist maintenance of satellite equipment. Transmission of live high quality captured video and graphic design to a cloud or distant studio production infrastructure requires high uplink data rates. However, steady and reliable communications are challenging for the network in disperse, distant and sparse areas. This context may need bonding multiple cellular links to ensure a sufficient Quality of Service (QoS). Video uplink solutions at different network layers can shield from QoS degradation. Communications industry solution for IP bonding consists on having different Long-Term Evolution (LTE) network interfaces with several Subscriber Identity Module (SIM) cards on the device which transmits the live stream, then having network redundancy. This paper provides an innovative method to dynamically balance the throughput for each concurrently employed network interface in real-time at the live video transmitter. The solution exploits live measurements obtained from the network layer (L3), such as network bandwidth, latency and jitter, which are periodically*

assessed along the video transmission, and application layer (L7) state, such as the encoding Group Of Pictures (GOP) schema, frame type and framerate, to split the video packets in the different network interfaces. The evaluation of the solution is made for a head-end implementation by sending live video streams and measuring the QoS at the production infrastructure. To conclude the benefits when the solution comes into play, results are compared to a scenario without bonding solutions and another one where balance rates are initially fixed.

APPENDIX

B

Resume

Roberto Viola received his Computer and Telecommunication Engineering degree in 2014 and an advanced degree in Telecommunication Engineering in 2016 from University of Cassino and Southern Lazio (Italy). Currently, he is Research Associate, as part of Digital Media department of Vicomtech. He is involved in R&D projects dealing with multimedia services and network infrastructure. At the same time, he is working on his PhD degree on video streaming in 5G networks at the University of the Basque Country (UPV/EHU).

APPENDIX

C

Acronyms

3GPP	3rd Generation Partnership Project
5G	Fifth Generation
5GPPP	5G Public Private Partnership
6G	Sixth Generation
AES	Advanced Encryption Standard
AES-CBC	AES block cipher mode
AES-CTR	AES counter mode
ANN	Artificial Neural Network
API	Application Programming Interface
AR	Augmented Reality
C-RAN	Cloud-RAN
CAPEX	Capital Expenditure

CDN	Content Delivery Network
CMAF	Common Media Application Format
CN	Core Network
COTS	Commercial off-the-shelf
CP	Content Provider
CRM	Customer Relationship Management
CSI	Channel State Information
DASH	Dynamic Adaptive Streaming over HTTP
DNS	Domain Name System
DTN	Delay-tolerant Networking
ESN	Echo State Network
ETSI	European Telecommunications Standards Institute
FeMBMS	Further enhanced MBMS
GUI	Graphical User Interface
HAS	HTTP Adaptive Streaming
HLS	HTTP Live Streaming
HTTP	HyperText Transfer Protocol
HVS	Human Visual System
IaaS	Infrastructure as a Service
IBN	Intent-Based Network
IIoT	Industrial Internet of Things

IM	Instant Messaging
IoT	Internet of Things
IP	Internet Protocol
ISP	Internet Service Provider
ITU	International Telecommunication Union
KPI	Key Performance Indicator
L1	Physical layer
L2	Data link layer
L3	Network layer
L4	Transport layer
L7	Application layer
LL CMAF	Low Latency CMAF
LL-DASH	Low Latency DASH
LL-HLS	Low Latency HLS
LSTM	Long short-term memory
LTE	Long-Term Evolution
M3U8	HLS playlist
MANO	Management and Orchestration
MBMS	Multimedia Broadcast/Multicast Service
MEC	Multi-access Edge Computing
MLP	Multi-layer Perceptron

MMS	Multimedia Messaging Service
MOS	Mean Opinion Score
MP4	MPEG-4 Part 14
MPD	Media Presentation Description
MPEG	Moving Picture Experts Group
MPTCP	Multipath TCP
Multi-RAT	Multiple Radio Access Technology
NAT	Network Address Translation
NFV	Network function virtualization
NFV-RA	NFV resource allocation
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NO	Network Operator
NS	Network Service
O-RAN	Open RAN
ONAP	Open Network Automation Platform
OPEX	Operational Expenditure
OSI	Open Systems Interconnection
OSM	Open Source MANO
OTT	Over-the-top
P2P	Peer-to-peer

PoP	Point of presence
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RNI	Radio Network Information
RNIS	RNI service
RNN	Recurrent Neural Network
RTCP	Real-time Transport Control Protocol
RTMP	Real-time Messaging Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SCTP	Stream Control Transmission Protocol
SDN	Software-defined network
SDR	Software-defined radio
SLA	Service Level Agreement
SON	Self-Organizing Network
SRT	Secure Reliable Transport
STUN	Session Traversal Utilities for NAT
SVA	Streaming Video Alliance
SVM	Support Vector Machine
SVR	Support Vector Regression

TCP	Transmission Control Protocol
TURN	Traversal Using Relays around NAT
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UE	User Equipment
UNESCO	United Nations Educational, Scientific and Cultural Organization
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
VNF-CC	VNF Chain Composition
VNF-FG	VNF Forwarding Graph
VNF-FGE	VNF Forwarding Graph Embedding
VNF-SCH	VNF Scheduling
VNFI	VNF Instance
VNFM	VNF Manager
VOD	Video-on-Demand
VR	Virtual Reality
vRAN	Virtual RAN
WebRTC	Web Real-Time Communication
WSN	Wireless Sensor Network

Part VI

Bibliography

Bibliography

[1] ITU. Recommendation itu-t p.1203: Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport, 2017. xiv, 46, 47, 166, 169, 170, 173, 174, 177

[2] Cornelia Wolf and Anna Schnauber. News consumption in the mobile era: The role of mobile devices and traditional journalism's content within the user's information repertoire. *Digital journalism*, 3(5):759–776, 2015. 3

[3] Cisco. Cisco annual internet report (2018–2023) white paper, 2020. 4, 11, 26

[4] Andra Lutu, Diego Perino, Marcelo Bagnulo, Enrique Frias-Martinez, and Javad Khangosstar. A characterization of the covid-19 pandemic impact on a mobile network operator traffic. In *Proceedings of the ACM Internet Measurement Conference*, pages 19–33, 2020. 4, 26

[5] Anja Feldmann, Oliver Gasser, Franziska Lichtblau, Enric Pujol, Ingmar Poese, Christoph Dietzel, Daniel Wagner, Matthias Wichtlhuber, Juan Tapiador, Narseo Vallina-Rodriguez, Oliver Hohlfeld, and Georgios Smaragdakis. The lockdown effect: Implications of the covid-19 pandemic on internet traffic. *Proceedings of the ACM Internet Measurement Conference*, page 1–18, 2020. 4, 26

[6] Thomas Favale, Francesca Soro, Martino Trevisan, Idilio Drago, and Marco Mellia. Campus traffic and e-learning during covid-19 pandemic. *Computer Networks*, 176:107290, 2020. 4, 26

[7] Daniel L King, Paul H Delfabbro, Joel Billieux, and Marc N Potenza. Problematic online gaming and the covid-19 pandemic. *Journal of Behavioral Addictions*, 9(2):184–186, 2020. 4, 26

[8] Broadband commission agenda for action for faster and better recovery. 4, 26

[9] The affordability of ict services 2020. 4, 26

[10] Markets and Markets. Content delivery network market, 2020. 4

[11] Dan Rayburn. Cdn/media pricing see's big drop for largest customers: Pricing down to $0.0006, 2020. 5, 27

[12] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2014. 5

[13] Bo Han, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2):90–97, 2015. 5, 27

[14] Dario Sabella, Vadim Sukhomlinov, Linh Trang, Sami Kekki, Pietro Paglierani, Ralf Rossbach, Xinhui Li, Yonggang Fang, Dan Druta, Fabio Giust, et al. Developing software for multi-access edge computing. *ETSI white paper*, 20:1–38, 2019. 5, 10, 18, 67, 135, 187

[15] ETSI. Etsi gs mec 012: Mobile edge computing (mec); radio network information api, 2017. 5, 18, 68, 82

[16] Narjes Tahghigh Jahromi, Somayeh Kianpisheh, and Roch H Glitho. Online vnf placement and chaining for value-added services in content delivery networks. In *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, pages 19–24. IEEE, 2018. 6, 27, 40

[17] Mouhamad Dieye, Shohreh Ahvar, Jagruti Sahoo, Ehsan Ahvar, Roch Glitho, Halima Elbiaze, and Noel Crespi. Cpvnf: Cost-efficient proactive vnf placement and chaining for value-added services in content delivery networks. *IEEE Transactions on Network and Service Management*, 15(2):774–786, 2018. 6, 27, 40, 81

[18] Hyun Jong Kim and Seong Gon Choi. A study on a qos/qoe correlation model for qoe evaluation on iptv service. In *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, volume 2, pages 1377–1382. IEEE, 2010. 6, 27, 45

[19] Mohammed Alreshoodi and John Woods. Survey on qoe\qos correlation models for multimedia services. *arXiv preprint arXiv:1306.0221*, 2013. 6, 27, 32, 34, 45, 103

[20] Enrique Hernandez-Valencia, Steven Izzo, and Beth Polonsky. How will nfv/sdn transform service provider opex? *IEEE Network*, 29(3):60–67, 2015. 6, 12, 27, 80

[21] Michael Seufert, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hoßfeld, and Phuoc Tran-Gia. A survey on quality of experience of http adaptive streaming. *IEEE Communications Surveys & Tutorials*, 17(1):469–492, 2014. 7, 32, 34, 36, 131

[22] Roger Pantos and William May. Http live streaming. *rfc 8216, August*, 2017. 7, 38

[23] Iraj Sodagar. The mpeg-dash standard for multimedia streaming over the internet. *IEEE multimedia*, 18(4):62–67, 2011. 7, 38, 99, 100, 131, 186

[24] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003. 7, 103

[25] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012. 7

[26] Luis Torres and Murat Kunt. *Video coding: the second generation approach*. Springer Science & Business Media, 2012. 9, 16

[27] Anne Aaron, Zhi Li, Megha Manohara, Jan De Cock, and David Ronca. Per-title encode optimization, 2015. 9, 16, 46

[28] Vijay Kumar Adhikari, Yang Guo, Fang Hao, Matteo Varvello, Volker Hilt, Moritz Steiner, and Zhi-Li Zhang. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *2012 Proceedings IEEE INFOCOM*, pages 1620–1628. IEEE, 2012. 10, 17, 71, 74, 132, 134, 188

[29] Vijay Kumar Adhikari, Yang Guo, Fang Hao, Volker Hilt, and Zhi-Li Zhang. A tale of three cdns: An active measurement study of hulu and its cdns. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 7–12. IEEE, 2012. 10, 17, 71, 74, 132, 134

[30] Vijay K Adhikari, Yang Guo, Fang Hao, Volker Hilt, Zhi-Li Zhang, Matteo Varvello, and Moritz Steiner. Measurement study of netflix, hulu, and a tale of three cdns. *IEEE/ACM Transactions on Networking*, 23(6):1984–1997, 2014. 10, 17, 71, 74, 132, 134, 188

[31] Yiming Tan, Ce Han, Ming Luo, Xiang Zhou, and Xing Zhang. Radio network-aware edge caching for video delivery in mec-enabled cellular networks. In *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 179–184. IEEE, 2018. 10, 12, 18, 71, 76

[32] Angel Martin, Roberto Viola, Mikel Zorrilla, Julián Flórez, Pablo Angueira, and Jon Montalbán. Mec for fair, reliable and efficient media streaming in mobile networks. *IEEE Transactions on Broadcasting*, 66(2):264–278, 2019. 10, 12, 18, 69, 71, 76

[33] Michele A. Saad, Margaret H. Pinson, David G. Nicholas, Niels Van Kets, Glenn Van Wallendael, Ralston Da Silva, Ramesh V. Jaladi, and Philip J. Corriveau. Impact of camera pixel count and monitor resolution perceptual image quality. In *2015 Colour and Visual Computing Symposium (CVCS)*, pages 1–6, 2015. 11

[34] Poopathy Kathirgamanathan, Lisa M Bushby, Muttulingam Kumaraverl, Seenivasagam Ravichandran, and Sivagnanasundram Surendrakumar. Electroluminescent organic and quantum dot leds: the state of the art. *Journal of display technology*, 11(5):480–493, 2015. 11

[35] Cisco. New cisco annual internet report forecasts 5g to support more than 10 11

[36] Yue Chen, Debargha Murherjee, Jingning Han, Adrian Grange, Yaowu Xu, Zoe Liu, Sarah Parker, Cheng Chen, Hui Su, Urvang Joshi, et al. An overview of core coding tools in the av1 video codec. In *2018 Picture Coding Symposium (PCS)*, pages 41–45. IEEE, 2018. 12

[37] Naty Sidaty, Wassim Hamidouche, Olivier Déforges, Pierrick Philippe, and Jérôme Fournier. Compression performance of the versatile video coding: Hd and uhd visual quality monitoring. In *2019 Picture Coding Symposium (PCS)*, pages 1–5. IEEE, 2019. 12

[38] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al. Emerging mpeg standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):133–148, 2018. 12

[39] ETSI. Etsi gs mec 002: Multi-access edge computing (mec): Phase 2: Use cases and requirements, 2018. 12, 18, 69, 81, 166, 169, 170

[40] Afzal Badshah, Anwar Ghani, Shahaboddin Shamshirband, Giuseppe Aceto, and Antonio Pescapè. Performance-based service-level agreement in cloud computing to optimise penalties and revenue. *IET Communications*, 14(7):1102–1112, 2020. 12

[41] ETSI. Etsi gs mec 028 version 2.1.1 - multi-access edge computing (mec); wlan information api, 2020. 19, 169, 178

[42] Abdelnaser Adas. Traffic models in broadband networks. *IEEE communications Magazine*, 35(7):82–89, 1997. 32, 35, 55, 56

[43] Dan Chalmers and Morris Sloman. A survey of quality of service in mobile computing environments. *IEEE Communications surveys*, 2(2):2–10, 1999. 32, 33

[44] Jingwen Jin and Klara Nahrstedt. Qos specification languages for distributed multimedia applications: A survey and taxonomy. *IEEE multimedia*, 11(3):74–87, 2004. 32, 34, 43

[45] Huifang Feng and Yantai Shu. Study on network traffic prediction techniques. In *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005.*, volume 2, pages 1041–1044. IEEE, 2005. 32, 35

[46] Balakrishnan Chandrasekaran. Survey of network traffic models. *Waschington University in St. Louis CSE*, 567, 2009. 32, 35, 56

[47] Ahmed M Mohammed and Adel F Agamy. A survey on the common network traffic sources models. *International Journal of Computer Networks (IJCN)*, 3(2):103–115, 2011. 32, 35, 56

[48] Mohammad Ashraful Hoque, Matti Siekkinen, and Jukka K Nurminen. Energy efficient multimedia streaming to mobile devices—a survey. *IEEE Communications Surveys & Tutorials*, 16(1):579–597, 2012. 32, 34

[49] Sabina Baraković and Lea Skorin-Kapov. Survey and challenges of qoe management issues in wireless networks. *Journal of Computer Networks and Communications*, 2013, 2013. 32, 34

[50] Parikshit Juluri, Venkatesh Tamarapalli, and Deep Medhi. Measurement of quality of experience of video-on-demand services: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):401–418, 2015. 32, 34, 167

[51] Guan-Ming Su, Xiao Su, Yan Bai, Mea Wang, Athanasios V Vasilakos, and Haohong Wang. Qoe in video streaming over wireless networks: perspectives and research challenges. *Wireless networks*, 22(5):1571–1593, 2016. 32, 34

[52] Tiesong Zhao, Qian Liu, and Chang Wen Chen. Qoe in video transmission: A user experience-driven strategy. *IEEE Communications Surveys & Tutorials*, 19(1):285–302, 2016. 32, 34

[53] Zahid Akhtar and Tiago H Falk. Audio-visual multimedia quality assessment: A comprehensive survey. *IEEE access*, 5:21090–21117, 2017. 32, 34, 43

[54] Stefano Petrangeli, Jeroen Van Der Hooft, Tim Wauters, and Filip De Turck. Quality of experience-centric management of adaptive video streaming services: Status and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(2s):1–29, 2018. 32, 34

[55] Lea Skorin-Kapov, Martín Varela, Tobias Hoßfeld, and Kuan-Ta Chen. A survey of emerging concepts and challenges for qoe management of multimedia services. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(2s):1–29, 2018. 32, 34, 35

[56] Alcardo Alex Barakabitze, Nabajeet Barman, Arslan Ahmad, Saman Zadtootaghaj, Lingfen Sun, Maria G Martini, and Luigi Atzori. Qoe management of multimedia streaming services in future networks: a tutorial and survey. *IEEE Communications Surveys & Tutorials*, 22(1):526–565, 2019. 32, 34, 35

[57] Nabajeet Barman and Maria G Martini. Qoe modeling for http adaptive video streaming–a survey and open challenges. *Ieee Access*, 7:30831–30859, 2019. 32, 34

[58] Chuanji Zhang, Harshvardhan P Joshi, George F Riley, and Steven A Wright. Towards a virtual network function research agenda: A systematic literature review of vnf design considerations. *Journal of Network and Computer Applications*, 146:102417, 2019. 32, 35

[59] Jorge Navarro-Ortiz, Pablo Romero-Diaz, Sandra Sendra, Pablo Ameigeiras, Juan J Ramos-Munoz, and Juan M Lopez-Soler. A survey on 5g usage scenarios and traffic models. *IEEE Communications Surveys & Tutorials*, 22(2):905–929, 2020. 32, 35, 56

[60] Henning Schulzrinne, Stephen Casner, Ron Frederick, Van Jacobson, et al. Rtp: A transport protocol for real-time applications. *rfc 1889, January*, 1996. 36, 101

[61] Jon Postel et al. User datagram protocol. *STD 6, RFC 768, August*, 1980. 36

[62] Jon Postel et al. Transmission control protocol. *STD 7, RFC 793, September*, 1981. 36

[63] Lucian Popa, Ali Ghodsi, and Ion Stoica. Http as the narrow waist of the future internet. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 1–6, 2010. 37

[64] Henning Schulzrinne, Anup Rao, and Robert Lanphier. Real time streaming protocol (rtsp). *rfc 2326, April*, 1998. 38

[65] Michael Thornburgh. Adobe's rtmfp profile for flash communication. *Internet Engineering Task Force (IETF)*, 2014. 38

[66] Maria Sharabayko, Maxim Sharabayko, Jean Dube, Joonwoong Kim, and Jeongseok Kim. The srt protocol. *draft-sharabayko-srt-00*, 2021. 38

[67] Christer Holmberg, Stefan Hakansson, and G Eriksson. Web real-time communication use cases and requirements. *Request for Comments (RFC)*, 7478, 2015. 38

[68] Dan Wing, Philip Matthews, Rohan Mahy, and Jonathan Rosenberg. Session traversal utilities for nat (stun). *RFC5389, October*, 2008. 38

[69] Rohan Mahy, Philip Matthews, and Jonathan Rosenberg. Traversal using relays around nat (turn): Relay extensions to session traversal utilities for nat (stun). Technical report, RFC 5766 (Proposed Standard), Internet Engineering Task Force, 2010. 38

[70] K Hughes and D Singer. Information technology–multimedia application format (mpeg-a)–part 19: Common media application format (cmaf) for segmented media. *ISO/IEC*, 19:23000, 2017. 10, 38, 101, 113

[71] Kerem Durak, Mehmet N Akcay, Yigit K Erinc, Boran Pekel, and Ali C Begen. Evaluating the performance of apple's low-latency hls. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2020. 38

[72] Nassima Bouzakaria, Cyril Concolato, and Jean Le Feuvre. Overhead and performance of low latency live streaming using mpeg-dash. In *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, pages 92–97. IEEE, 2014. 38, 116

[73] Pete Chown. Advanced encryption standard (aes) ciphersuites for transport layer security (tls). Technical report, RFC 3268, June, 2002. 39, 104

[74] Divyashri Bhat, Amr Rizk, and Michael Zink. Not so quic: A performance study of dash over quic. In *Proceedings of the 27th workshop on network and operating systems support for digital audio and video*, pages 13–18, 2017. 39

[75] W3C. Quic api for peer-to-peer connections, 2020. 39

[76] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, et al. The quic transport protocol: Design and internet-scale deployment. In *Proceedings of the conference of the ACM special interest group on data communication*, pages 183–196, 2017. 39

[77] Lyndon Ong, John Yoakum, et al. An introduction to the stream control transmission protocol (sctp). Technical report, RFC 3286 (Informational), May, 2002. 39

[78] Alan Ford, Costin Raiciu, Mark Handley, Sebastien Barre, Janardhan Iyengar, et al. Architectural guidelines for multipath tcp development. *IETF, Informational RFC*, 6182:2070–1721, 2011. 39

[79] Quentin De Coninck and Olivier Bonaventure. Multipath extensions for quic (mp-quic). *draft-deconinck-quic-multipath-06*, 2020. 39

[80] Kristian Evensen, Tomas Kupka, Haakon Riiser, Pengpeng Ni, Ragnhild Eg, Carsten Griwodz, and Pål Halvorsen. Adaptive media streaming to mobile devices: challenges, enhancements, and recommendations. *Advances in Multimedia*, 2014, 2014. 40

[81] Madeleine Keltsch, Sebastian Prokesch, Oscar Prieto Gordo, Javier Serrano, Truong Khoa Phan, and Igor Fritzsch. Remote production and mobile contribution over 5g networks: scenarios, requirements and approaches for broadcast quality media streaming. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–7. IEEE, 2018. 40

[82] Federico Alvarez, David Breitgand, David Griffin, Pasquale Andriani, Stamatia Rizou, Nikolaos Zioulis, Francesca Moscatelli, Javier Serrano, Madeleine Keltsch, Panagiotis Trakadas, T. Khoa Phan, Avi Weit, Ugur Acar, Oscar Prieto, Francesco Iadanza, Gino Carrozzo, Harilaos Koumaras, Dimitrios Zarpalas, and David Jimenez. An Edge-to-Cloud Virtualized Multimedia Service Platform for 5G Networks. *IEEE Transactions on Broadcasting*, 65(2):369–380, June 2019. 40, 88, 90

[83] Chuanji Zhang, Harshvardhan P Joshi, George F Riley, and Steven A Wright. Towards a virtual network function research agenda: A systematic literature review of vnf design considerations. *Journal of Network and Computer Applications*, 146:102417, 2019. 40

[84] ITU. Recommendation itu-t p.800.1: Mean opinion score terminology, 2016. 45, 102, 117, 167, 174

[85] ITU. Recommendation itu-t p.800.2: Mean opinion score interpretation and reporting, 2016. 45

[86] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 46

[87] Abdul Rehman, Kai Zeng, and Zhou Wang. Display device-adapted video quality-of-experience assessment. In *Human Vision and Electronic Imaging XX*, volume 9394, page 939406. International Society for Optics and Photonics, 2015. 46, 48

[88] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy, and Megha Manohara. Toward a practical perceptual video quality metric, 2016. 46, 49

[89] Johan De Vriendt, Danny De Vleeschauwer, and David Robinson. Model for estimating qoe of video delivered using http adaptive streaming. *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 1288–1293, 2013. 46, 47, 48

[90] Xiaoqi Yin, Vyas Sekar, and Bruno Sinopoli. Toward a principled framework to design dynamic adaptive streaming algorithms over http. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, pages 1–7, 2014. 46, 48, 49

[91] Jingteng Xue, Dong-Qing Zhang, Heather Yu, and Chang Wen Chen. Assessing quality of experience for adaptive http video streaming. In *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2014. 46, 48

[92] Yao Liu, Sujit Dey, Fatih Ulupinar, Michael Luby, and Yinian Mao. Deriving and validating user experience model for dash video streaming. *IEEE Transactions on Broadcasting*, 61(4):651–665, 2015. 46, 48

[93] Abdelhak Bentaleb, Ali C Begen, and Roger Zimmermann. Sdndash: Improving qoe of http adaptive streaming using software defined networking. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1296–1305, 2016. 46, 48

[94] Zhengfang Duanmu, Kai Zeng, Kede Ma, Abdul Rehman, and Zhou Wang. A quality-of-experience index for streaming video. *IEEE Journal of Selected Topics in Signal Processing*, 11(1):154–166, 2016. 46, 48

[95] Huawei. Video experience-based bearer network technical white paper, 2016. 46, 48

[96] Zhengfang Duanmu, Wentao Liu, Diqi Chen, Zhuoran Li, Zhou Wang, Yizhou Wang, and Wen Gao. A knowledge-driven quality-of-experience model for adaptive streaming videos. *arXiv preprint arXiv:1911.07944*, 2019. 46, 48

[97] Ismael de Fez, Román Belda, and Juan Carlos Guerri. New objective qoe models for evaluating abr algorithms in dash. *Computer Communications*, 158:126–140, 2020. 46, 49

[98] ITU. Recommendation itu-t p.1204: Video quality assessment of streaming services over reliable transport for resolutions up to 4k, 2020. 46, 47

[99] Werner Robitza, Steve Göring, Alexander Raake, David Lindegren, Gunnar Heikkilä, Jörgen Gustafsson, Peter List, Bernhard Feiten, Ulf Wüstenhagen, Marie-Neige Garcia, et al. Http adaptive streaming qoe estimation with itu-t rec. p. 1203: open databases and software. In *Proceedings of the 9th ACM Multimedia Systems Conference*, pages 466–471, 2018. 47, 174

[100] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 1984. 49

[101] Tobias Hoßfeld, Lea Skorin-Kapov, Poul E Heegaard, and Martin Varela. Definition of qoe fairness in shared systems. *IEEE Communications Letters*, 21(1):184–187, 2016. 49

[102] SHI Huaizhou, R Venkatesha Prasad, Ertan Onur, and IGMM Niemegeers. Fairness in wireless networks: Issues, measures and challenges. *IEEE Communications Surveys & Tutorials*, 16(1):5–24, 2013. 49

[103] Tian Lan, David Kao, Mung Chiang, and Ashutosh Sabharwal. An axiomatic theory of fairness in network resource allocation. In *2010 Proceedings IEEE INFOCOM*, 2010. 50

[104] Bozidar Radunovic and Jean-Yves Le Boudec. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on networking*, 15(5):1073–1083, 2007. 50

[105] Jan Ozer. A video codec licensing update, 2019. 51

[106] Jan Ozer. The future of hevc licensing is bleak, declares mpeg chairman, 2018. 51

[107] Mozilla. Web video codec guide. 51

[108] Jan Ozer. A cloud encoding pricing comparison, 2016. 51

[109] Sreejata Basu. Cloud video encoding vs on-premise: Pros, cons and beyond, 2020. 51

[110] Alain Pellen. Cost comparison: On-premises vs cloud computing, 2020. 51

[111] Fabre Lambeau. Cloud-based per-title encoding workflows (with aws) – part 1: Establishing the architecture, 2021. 51

[112] Christian Timmerer. Mpeg-cmaf: Threat or opportunity?, 2016. 51

[113] Sofie Verbrugge, Didier Colle, Mario Pickavet, Piet Demeester, S Pasqualini, A Iselt, Andreas Kirstädter, R Hülsermann, F-J Westphal, and Monika Jäger. Methodology and input availability parameters for calculating opex and capex costs for realistic network scenarios. *Journal of Optical Networking*, 5(6):509–520, 2006. 52, 156

[114] DaCast. 2019 live streaming cdn pricing comparison, 2020. 52, 156

[115] CDNPerf. Cdn calculator. 52, 156

[116] Wowza. Wowza streaming cloud plans. 52, 156

[117] Roberto Viola, Angel Martin, Javier Morgade, Stefano Masneri, Mikel Zorrilla, Pablo Angueira, and Jon Montalbán. Predictive cdn selection for video delivery based on lstm network performance forecasts and cost-effective trade-offs. *IEEE Transactions on Broadcasting*, 2020. 52, 71, 75

[118] Martin Kennedy, Adlen Ksentini, Yassine Hadjadj-Aoul, and Gabriel-Miro Muntean. Adaptive energy optimization in multimedia-centric wireless devices: A survey. *IEEE communications surveys & tutorials*, 15(2):768–786, 2012. 53

[119] Kevin McClaning and Tom Vito. *Radio receiver design*. Noble Publishing, 2000. 53

[120] Ramona Trestian, Olga Ormond, and Gabriel-Miro Muntean. Energy–quality–cost tradeoff in a multimedia-based heterogeneous wireless network environment. *IEEE Transactions on Broadcasting*, 59(2):340–357, 2013. 54

[121] Longhao Zou, Ali Javed, and Gabriel-Miro Muntean. Smart mobile device power consumption measurement for video streaming in wireless environments: Wifi vs. lte. In *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–6. IEEE, 2017. 54

[122] Charles-Antoine Guérin, Henrik Nyberg, Olivier Perrin, S Resnick, H Rootzén, and C Stărică. Empirical testing of the infinite source poisson data traffic model. *Stochastic Models*, 19(2):151–200, 2003. 55, 56

[123] Thomas Karagiannis, Mart Molle, Michalis Faloutsos, and Andre Broido. A nonstationary poisson view of internet traffic. In *IEEE INFOCOM 2004*, volume 3, pages 1558–1569. IEEE, 2004. 55, 56

[124] Arup Bhattacharjee and Sukumar Nandi. Statistical analysis of network traffic inter-arrival. In *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, volume 2, pages 1052–1057. IEEE, 2010. 55, 56

[125] Wei Zhang and Jingsha He. Modeling end-to-end delay using pareto distribution. In *Second International Conference on Internet Monitoring and Protection (ICIMP 2007)*, pages 21–21. IEEE, 2007. 55, 56

[126] Muhammad Asad Arfeen, Krzysztof Pawlikowski, D McNickle, and Andreas Willig. The role of the weibull distribution in internet traffic modeling. In *Proceedings of the 2013 25th International Teletraffic Congress (ITC)*, pages 1–8. IEEE, 2013. 55, 56

[127] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007. 56

[128] Zhiyong Liu, Kun Wang, Wei Li, Qing Xiao, Denian Shi, and Guili He. Measurement and modeling study of iptv cdn network. In *2009 IEEE International Conference on Network Infrastructure and Digital Content*, pages 302–306. IEEE, 2009. 56

[129] Ashwin Rao, Arnaud Legout, Yeon-sup Lim, Don Towsley, Chadi Barakat, and Walid Dabbous. Network characteristics of video streaming traffic. In *Proceedings of the seventh conference on emerging networking experiments and technologies*, pages 1–12, 2011. 57

[130] Michael Zink, Kyoungwon Suh, Yu Gu, and Jim Kurose. Characteristics of youtube network traffic at a campus network–measurements, models, and implications. *Computer networks*, 53(4):501–514, 2009. 57

[131] Hongliang Yu, Dongdong Zheng, Ben Y. Zhao, and Weimin Zheng. Understanding user behavior in large-scale video-on-demand systems. *SIGOPS Oper. Syst. Rev.*, 40(4):333–344, April 2006. 57, 146, 179, 180

[132] Ning Liu, Huajie Cui, S.-H. Gary Chan, Zhipeng Chen, and Yirong Zhuang. Dissecting user behaviors for a simultaneous live and vod iptv system. *ACM Trans. Multimedia Comput. Commun. Appl.*, 10(3), April 2014. 57

[133] Rodrigo N Calheiros, Enayat Masoumi, Rajiv Ranjan, and Rajkumar Buyya. Workload prediction using arima model and its impact on cloud applications' qos. *IEEE transactions on cloud computing*, 3(4):449–458, 2014. 58, 59, 137

[134] Xin Dong, Wentao Fan, and Jun Gu. Predicting lte throughput using traffic time series. *ZTE Communications*, 13(4):61–64, 2015. 58, 59, 137

[135] Ayman Amin, Lars Grunske, and Alan Colman. An automated approach to forecasting qos attributes based on linear and non-linear time series modeling. In *2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, pages 130–139. IEEE, 2012. 58, 59, 137

[136] Ayman Amin, Alan Colman, and Lars Grunske. An approach to forecasting qos attributes of web services based on arima and garch models. In *2012 IEEE 19th International Conference on Web Services*, pages 74–81. IEEE, 2012. 58, 59, 137

[137] Congjie Wang, Zhihui Lu, Ziyan Wu, Jie Wu, and Shalin Huang. Optimizing multi-cloud cdn deployment and scheduling strategies using big data analysis. In *2017 IEEE International Conference on Services Computing (SCC)*, pages 273–280. IEEE, 2017. 58, 59

[138] Maciej Szmit, Anna Szmit, Sławomir Adamus, and Sebastian Bugała. Usage of holt-winters model and multilayer perceptron in network traffic modelling and anomaly detection. *Informatica*, 36(4), 2012. 58, 59, 60

[139] Ashraf A Shahin. Using multiple seasonal holt-winters exponential smoothing to predict cloud resource provisioning. *arXiv preprint arXiv:1701.03296*, 2017. 58, 59

[140] Mariyam Mirza, Joel Sommers, Paul Barford, and Xiaojin Zhu. A machine learning approach to tcp throughput prediction. *IEEE/ACM Transactions on Networking*, 18(4):1026–1039, 2010. 58, 60

[141] Paola Bermolen and Dario Rossi. Support vector regression for link load prediction. *Computer Networks*, 53(2):191–201, 2009. 58, 60

[142] Vin-sen Feng and Shih Yu Chang. Determination of wireless networks parameters through parallel hierarchical support vector machines. *IEEE Transactions on Parallel and Distributed Systems*, 23(3):505–512, 2011. 58, 60

[143] Xu Chen, François Mériaux, and Stefan Valentin. Predicting a user's next cell with supervised learning based on channel states. In *2013 IEEE 14th workshop on signal processing advances in wireless communications (SPAWC)*, pages 36–40. IEEE, 2013. 58, 60

[144] Mahmod Hosein Zadeh and Mir Ali Seyyedi. Qos monitoring for web services by time series forecasting. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 5, pages 659–663. IEEE, 2010. 58, 60, 137

[145] Alireza Khotanzad and Nayyara Sadek. Multi-scale high-speed network traffic prediction using combination of neural networks. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 2, pages 1071–1075. IEEE, 2003. 58, 60

[146] Salem Belhaj and Moncef Tagina. Modeling and prediction of the internet end-to-end delay using recurrent neural networks. *J. Networks*, 4(6):528–535, 2009. 58, 60, 137

[147] Hoang Duy Trinh, Lorenza Giupponi, and Paolo Dini. Mobile traffic prediction from raw data using lstm networks. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1827–1832. IEEE, 2018. 58, 60, 138

[148] Abdelhadi Azzouni and Guy Pujolle. Neutm: A neural network-based framework for traffic matrix prediction in sdn. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5. IEEE, 2018. 58, 60

[149] J-M Martinez-Caro and M-D Cano. On the identification and prediction of stalling events to improve qoe in video streaming. *Electronics*, 10(6):753, 2021. 58, 60

[150] Hongyan Cui, Yuan Yao, Kuo Zhang, Fangfang Sun, and Yunjie Liu. Network traffic prediction based on hadoop. In *2014 International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 29–33. IEEE, 2014. 58, 60

[151] Nicholas I Sapankevych and Ravi Sankar. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine*, 4(2):24–38, 2009. 57

[152] Muhammad Faisal Iqbal, Muhammad Zahid, Durdana Habib, and Lizy Kurian John. Efficient prediction of network traffic for real-time applications. *Journal of Computer Networks and Communications*, 2019, 2019. 57

[153] Yiannos Kryftis, Constandinos X Mavromoustakis, George Mastorakis, Evangelos Pallis, Jordi Mongay Batalla, Joel JPC Rodrigues, Ciprian Dobre, and Georgios Kormentzas. Resource usage prediction algorithms for optimal selection of multimedia content delivery methods. In *2015 IEEE international conference on communications (ICC)*, pages 5903–5909. IEEE, 2015. 57

[154] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. Machine learning strategies for time series forecasting. In *European business intelligence summer school*, pages 62–77. Springer, 2012. 59

[155] Amin Azari, Panagiotis Papapetrou, Stojan Denic, and Gunnar Peters. Cellular traffic prediction and classification: A comparative evaluation of lstm and arima. In *International Conference on Discovery Science*, pages 129–144. Springer, 2019. 59, 138, 142

[156] Amin Azari, Panagiotis Papapetrou, Stojan Denic, and Gunnar Peters. User traffic prediction for proactive resource management: learning-powered approaches. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019. 59, 138, 142

[157] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice.* OTexts, 2018. 59

[158] Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004. 59

[159] K-R Müller, Alexander J Smola, Gunnar Rätsch, Bernhard Schölkopf, Jens Kohlmorgen, and Vladimir Vapnik. Predicting time series with support vector machines. In *International Conference on Artificial Neural Networks*, pages 999–1004. Springer, 1997. 60

[160] Rishabh Madan and Partha Sarathi Mangipudi. Predicting computer network traffic: a time series forecasting approach using dwt, arima and rnn. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pages 1–5. IEEE, 2018. 60

[161] C Narendra Babu and B Eswara Reddy. Performance comparison of four new arima-ann prediction models on internet traffic data. *Journal of Telecommunications and Information Technology*, 2015. 60

[162] Prometheus. 61

[163] Syeda Noor Zehra Naqvi, Sofia Yfantidou, and Esteban Zimányi. Time series databases and influxdb. *Studienarbeit, Université Libre de Bruxelles*, page 12, 2017. 61

[164] Grafana. Grafana. 61

[165] Elastic. Elastic stack. 61, 62

[166] Board. Board. 61, 62

[167] Jamie Hoelscher and Amanda Mortimer. Using tableau to visualize data and drive decision-making. *Journal of Accounting Education*, 44:49–59, 2018. 61, 62

[168] Citrix. Citrix analytics. 61, 62

[169] Jerri L Ledford, Joe Teixeira, and Mary E Tyler. *Google analytics*. John Wiley and Sons, 2011. 61, 62

[170] Akamai. Media analytics. 61, 62

[171] Conviva. Conviva streaming analytics. 61, 62

[172] Amazon. Amazon kinesis. 61, 62

[173] András Varga and Rudolf Hornig. An overview of the omnet++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–10, 2008. 62, 63

[174] Teerawat Issariyakul and Ekram Hossain. Introduction to network simulator 2 (ns2). In *Introduction to network simulator NS2*, pages 1–18. Springer, 2009. 62, 63

[175] Thomas R Henderson, Mathieu Lacage, George F Riley, Craig Dowell, and Joseph Kopena. Network simulations with the ns-3 simulator. *SIGCOMM demonstration*, 14(14):527, 2008. 62, 63

[176] Xinjie Chang. Network simulations with opnet. In *WSC'99. 1999 Winter Simulation Conference Proceedings.'Simulation-A Bridge to the Future'(Cat. No. 99CH37038)*, volume 1, pages 307–314. IEEE, 1999. 62, 63

[177] Mininet: An Instant Virtual Network on your Laptop (or other PC). 62, 63

[178] Tetcos. Netsim. 62, 63

[179] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The one simulator for dtn protocol evaluation. In *Proceedings of the 2nd international conference on simulation tools and techniques*, pages 1–10, 2009. 62, 63

[180] iperf - the ultimate speed test tool for tcp and udp and sctp. 63, 64

[181] Miha Jemec. packeth–ethernet packet generator. 63, 64

[182] Robert Olsson. Pktgen the linux packet generator. In *Proceedings of the Linux Symposium, Ottawa, Canada*, volume 2, pages 11–24, 2005. 63, 64

[183] Paul Emmerich, Sebastian Gallenmüller, Daniel Raumer, Florian Wohlfart, and Georg Carle. Moongen: A scriptable high-speed packet generator. In *Proceedings of the 2015 Internet Measurement Conference*, pages 275–287, 2015. 63, 64

[184] Nicola Bonelli, Stefano Giordano, Gregorio Procissi, and Secchi Raffaello. Brute: A high performance and extensibile traffic generator. In *Int'l Symposium on Performance of Telecommunication Systems (SPECTS'05)*, volume 1, pages 222–227, 2005. 63, 64

[185] Joel Sommers and Paul Barford. Self-configuring network traffic generation. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 68–81, 2004. 63, 64

[186] Bharat Rahuldhev Patil, Minal Moharir, Pratik Kumar Mohanty, G Shobha, and S Sajeev. Ostinato-a powerful traffic generator. In *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pages 1–5. IEEE, 2017. 63, 64

[187] Cisco. Trex. 63, 64

[188] Stefano Avallone, S Guadagno, Donato Emma, Antonio Pescapè, and Giorgio Ventre. D-itg distributed internet traffic generator. In *First International Conference on the Quantitative Evaluation of Systems, 2004. QEST 2004. Proceedings.*, pages 316–317. IEEE, 2004. 63, 64

[189] Alessio Botta, Alberto Dainotti, and Antonio Pescapé. A tool for the generation of realistic network workload for emerging networking scenarios. *Computer Networks*, 56(15):3531–3547, 2012. 63, 64

[190] HP. Seagull – open source tool for ims testing, 2006. 63, 64

[191] Chengchao Liang, F Richard Yu, and Xi Zhang. Information-centric network function virtualization over 5g mobile wireless networks. *IEEE network*, 29(3):68–74, 2015. 65

[192] ETSI. Etsi gs nfv-man 001: Network functions virtualisation (nfv); management and orchestration, 2014. 65

[193] ETSI. Open source mano (osm). 67

[194] Linux Foundation. Open network automation platform (onap). 67

[195] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. OpenAirInterface: A Flexible Platform for 5G Research. *ACM SIGCOMM Computer Communication Review*, 44(5):33–38, October 2014. 67, 88, 90

[196] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D Sutton, Pablo Serrano, Cristina Cano, and Doug J Leith. srslte: An open-source platform for lte evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, pages 25–32, 2016. 67

[197] Van-Giang Nguyen, Anna Brunstrom, Karl-Johan Grinnemo, and Javid Taheri. Sdn/nfv-based mobile packet core network architectures: A survey. *IEEE Communications Surveys & Tutorials*, 19(3):1567–1602, 2017. 67

[198] Andres F Ocampo, Thomas Dreibholz, Mah-Rukh Fida, Ahmed Elmokashfi, and Haakon Bryhni. Integrating cloud-ran with packet core as vnf using open source mano and openairinterface. In *Proceedings of the 45th IEEE Conference on Local Computer Networks (LCN), Sydney, New South Wales/Australia (November 2020)*, 2020. 67

[199] Alvaro Gabilondo, Javier Morgade, Roberto Viola, Pablo Angueira, and Jon Montalbán. Realising a vran based fembms management and orchestration framework. In *2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–7. IEEE, 2020. 67, 71, 72

[200] Mao Yang, Yong Li, Depeng Jin, Li Su, Shaowu Ma, and Lieguang Zeng. Openran: A software-defined ran architecture via virtualization. *SIGCOMM Comput. Commun. Rev.*, 43(4):549–550, August 2013. 67, 88, 90

[201] Liljana Gavrilovska, Valentin Rakovic, and Daniel Denkovski. From cloud ran to open ran. *Wireless Personal Communications*, pages 1–17, 2020. 68

[202] Francesco Giannone, Pantelis A Frangoudis, Adlen Ksentini, and Luca Valcarenghi. Orchestrating heterogeneous mec-based applications for connected vehicles. *Computer Networks*, 180:107402, 2020. 69, 136

[203] Yue Li, Pantelis A Frangoudis, Yassine Hadjadj-Aoul, and Philippe 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. IEEE, 2016. 69

[204] Jordi Joan Gimenez, Jose Luis Carcel, Manuel Fuentes, Eduardo Garro, Simon Elliott, David Vargas, Christian Menzel, and David Gomez-Barquero. 5g new radio for terrestrial broadcast: A forward-looking approach for nr-mbms. *IEEE Transactions on Broadcasting*, 65(2):356–368, 2019. 70, 71

[205] Alexandros Doumanoglou, Nikolaos Zioulis, David Griffin, Javier Serrano, Truong Khoa Phan, David Jiménez, Dimitrios Zarpalas, Federico Alvarez, Miguel Rio, and Petros Daras. A system architecture for live immersive 3d-media transcoding over 5g networks. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 11–15. IEEE, 2018. 71, 72

[206] Sunny Dutta, Tarik Taleb, Pantelis A Frangoudis, and Adlen Ksentini. On-the-fly qoe-aware transcoding in the mobile edge. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2016. 71, 73, 168, 169

[207] Sepehr Rezvani, Saeedeh Parsaeefard, Nader Mokari, Mohammad R Javan, and Halim Yanikomeroglu. Cooperative multi-bitrate video caching and transcoding in multicarrier noma-assisted heterogeneous virtualized mec networks. *IEEE Access*, 7:93511–93536, 2019. 71, 73

[208] Chunyu Liu, Heli Zhang, Hong Ji, and Xi Li. Mec-assisted flexible transcoding strategy for adaptive bitrate video streaming in small cell networks. *China Communications*, 18(2):200–214, 2021. 71, 72, 73

[209] Tuyen X Tran and Dario Pompili. Adaptive bitrate video caching and processing in mobile-edge computing networks. *IEEE Transactions on Mobile Computing*, 18(9):1965–1978, 2018. 71, 73

[210] Yanting Wang, Yan Zhang, Min Sheng, and Kun Guo. On the interaction of video caching and retrieving in multi-server mobile-edge computing systems. *IEEE Wireless Communications Letters*, 8(5):1444–1447, 2019. 71, 73

[211] Qingmin Jia, Renchao Xie, Haijun Lu, Wenbin Zheng, and Hong Luo. Joint optimization scheme for caching, transcoding and bandwidth in 5g networks with mobile edge computing. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, pages 999–1004. IEEE, 2019. 71, 73

[212] John S Otto, Mario A Sánchez, John P Rula, Ted Stein, and Fabián E Bustamante. namehelp: Intelligent client-side dns resolution. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 287–288, 2012. 71, 74, 134

[213] Ruben Torres, Alessandro Finamore, Jin Ryong Kim, Marco Mellia, Maurizio M Munafo, and Sanjay Rao. Dissecting video server selection strategies in the youtube cdn. In *2011 31st International Conference on Distributed Computing Systems*, pages 248–257. IEEE, 2011. 71, 74, 135

[214] Utkarsh Goel, Mike P Wittie, and Moritz Steiner. Faster web through client-assisted cdn server selection. In *2015 24th International conference on computer communication and networks (ICCCN)*, pages 1–10. IEEE, 2015. 71, 74, 135

[215] Timm Böttger, Felix Cuadrado, Gareth Tyson, Ignacio Castro, and Steve Uhlig. Open connect everywhere: A glimpse at the internet ecosystem through the lens of the netflix cdn. *ACM SIGCOMM Computer Communication Review*, 48(1):28–34, 2018. 71, 74, 134

[216] SVA. Open caching. 71, 75

[217] Benjamin Frank, Ingmar Poese, Yin Lin, Georgios Smaragdakis, Anja Feldmann, Bruce Maggs, Jannis Rake, Steve Uhlig, and Rick Weber. Pushing cdn-isp collaboration to the limit. *ACM SIGCOMM Computer Communication Review*, 43(3):34–44, 2013. 71, 75, 136

[218] Matthias Wichtlhuber, Robert Reinecke, and David Hausheer. An sdn-based cdn/isp collaboration architecture for managing high-volume flows. *IEEE Transactions on Network and Service Management*, 12(1):48–60, 2015. 71, 75, 136

[219] EBU. Eurovision flow. 71, 75, 135, 188

[220] Citrix. Intelligent traffic management. 71, 75, 135

[221] Haivision. Haivision lightflow multicdn. 71, 75, 135

[222] Roberto Viola, Angel Martin, Mikel Zorrilla, and Jon Montalbán. Mec proxy for efficient cache and reliable multi-cdn video distribution. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–7. IEEE, 2018. 71, 76, 135

[223] Gino Carrozzo, Francesca Moscatelli, Gabriel Solsona, Oscar Prieto Gordo, Madeleine Keltsch, and Martin Schmalohr. Virtual cdns over 5g networks: scenarios and requirements for ultra-high definition media distribution. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–5. IEEE, 2018. 71, 76, 136

[224] Chang Ge, Ning Wang, Severin Skillman, Gerry Foster, and Yue Cao. Qoe-driven dash video caching and adaptation at 5g mobile edge. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, pages 237–242, 2016. 71, 76

[225] Yu Chen, Yong Liu, Jingya Zhao, and Qinghua Zhu. Mobile edge cache strategy based on neural collaborative filtering. *IEEE Access*, 8:18475–18482, 2020. 71, 76

[226] 3GPP. Overall description of lte-based 5g broadcast; version 16.0.0; technical report (tr) 36.976. *3rd Generation Partnership Project (3GPP)*, 2020. 70

[227] Huisheng Ma, Shufang Li, Erqing Zhang, Zhengnan Lv, Jing Hu, and Xinlei Wei. Cooperative autonomous driving oriented mec-aided 5g-v2x: Prototype system design, field tests and ai-based optimization tools. *IEEE Access*, 8:54288–54302, 2020. 72

[228] Gorka Velez, Ángel Martín, Giancarlo Pastor, and Edward Mutafungwa. 5g beyond 3gpp release 15 for connected automated mobility in cross-border contexts. *Sensors*, 20(22):6622, 2020. 72

[229] 5GINFIRE. D2.2-5ginfire experimental infrastructure architecture and 5g automotive use case(update). 72

[230] Trinh Viet Doan, Vaibhav Bajpai, and Sam Crawford. A longitudinal view of netflix: Content delivery over ipv6 and content cache deployments. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1073–1082. IEEE, 2020. 74

[231] Alexandros Biliris, Chuck Cranor, Fred Douglis, Michael Rabinovich, Sandeep Sibal, Oliver Spatscheck, and Walter Sturm. Cdn brokering. *Computer Communications*, 25(4):393–402, 2002. 75, 135

[232] ETSI. Etsi ts 128 313: 5g;self-organizing networks (son) for 5g networks, 2020. 76

[233] Osianoh Glenn Aliu, Ali Imran, Muhammad Ali Imran, and Barry Evans. A survey of self organisation in future cellular networks. *IEEE Communications Surveys & Tutorials*, 15(1):336–361, 2012. 77

[234] Honglin Hu, Jian Zhang, Xiaoying Zheng, Yang Yang, and Ping Wu. Self-configuration and self-optimization for lte networks. *IEEE Communications Magazine*, 48(2):94–100, 2010. 77

[235] Jessica Moysen and Lorenza Giupponi. From 4g to 5g: Self-organized network management meets machine learning. *Computer Communications*, 129:248–268, 2018. 77, 78

[236] Ali Imran, Ahmed Zoha, and Adnan Abu-Dayya. Challenges in 5g: how to empower son with big data for enabling 5g. *IEEE network*, 28(6):27–33, 2014. 78

[237] Paulo Valente Klaine, Muhammad Ali Imran, Oluwakayode Onireti, and Richard Demo Souza. A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Communications Surveys & Tutorials*, 19(4):2392–2431, 2017. 78

[238] HCL. Son. 78

[239] Nokia. Edennet. 78

[240] Ericsson. Son optimization manager. 78

[241] Ye Ouyang, Zhongyuan Li, Le Su, Wenyuan Lu, and Zhenyi Lin. Application behaviors driven self-organizing network (son) for 4g lte networks. *IEEE Transactions on Network Science and Engineering*, 7(1):3–14, 2018. 79

[242] Javad I Khan, Seung Su Yang, Qiong Gu, Darsan Patel, Patrick Mail, Oleg Komogortsev, Wansik Oh, and Zhong Guo. Resource adaptive netcentric systems: A case study with sonet-a self-organizing network embedded transcoder. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 617–620, 2001. 79

[243] Chetna Singhal and BN Chandana. Aerial-son: Uav-based self-organizing network for video streaming in dense urban scenario. In *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 7–12. IEEE, 2021. 79

[244] Juliver Gil Herrera and Juan Felipe Botero. Resource allocation in nfv: A comprehensive survey. *IEEE Transactions on Network and Service Management*, 13(3):518–532, 2016. 79

[245] Yanghao Xie, Zhixiang Liu, Sheng Wang, and Yuxiu Wang. Service function chaining resource allocation: A survey. *arXiv preprint arXiv:1608.00095*, 2016. 79

[246] Frederico Schardong, Ingrid Nunes, and Alberto Schaeffer-Filho. Nfv resource allocation: A systematic review and taxonomy of vnf forwarding graph embedding. *Computer Networks*, page 107726, 2020. 79

[247] Jordi Ferrer Riera, Eduard Escalona, Josep Batalle, Eduard Grasa, and Joan A Garcia-Espin. Virtual network function scheduling: Concept and challenges. In *2014 international conference on smart communications in network technologies (SaCoNeT)*, pages 1–5. IEEE, 2014. 80

[248] Oscar Adamuz-Hinojosa, Jose Ordonez-Lucena, Pablo Ameigeiras, Juan J Ramos-Munoz, Diego Lopez, and Jesus Folgueira. Automated network service scaling in nfv: Concepts, mechanisms and scaling workflow. *IEEE Communications Magazine*, 56(7):162–169, 2018. 80

[249] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Steven Latré, Marinos Charalambides, and Diego Lopez. Management and orchestration challenges in network functions virtualization. *IEEE Communications Magazine*, 54(1):98–105, 2016. 80

[250] Xiaoke Wang, Chuan Wu, Franck Le, Alex Liu, Zongpeng Li, and Francis Lau. Online vnf scaling in datacenters. In *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, pages 140–147. IEEE, 2016. 80

[251] Milad Ghaznavi, Aimal Khan, Nashid Shahriar, Khalid Alsubhi, Reaz Ahmed, and Raouf Boutaba. Elastic virtual network function placement. In *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pages 255–260. IEEE, 2015. 80

[252] David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Choudhury, and AK Qin. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2020. 81

[253] 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. IEEE, 2008. 81

[254] Xincai Fei, Fangming Liu, Hong Xu, and Hai Jin. Adaptive vnf scaling and flow routing with proactive demand prediction. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 486–494. IEEE, 2018. 81

[255] Rashid Mijumbi, Sidhant Hasija, Steven Davy, Alan Davy, Brendan Jennings, and Raouf Boutaba. A connectionist approach to dynamic resource management for virtualised network functions. In *2016 12th International Conference on Network and Service Management (CNSM)*, pages 1–9. IEEE, 2016. 81

[256] Xiaoxi Zhang, Chuan Wu, Zongpeng Li, and Francis CM Lau. Proactive vnf provisioning with multi-timescale cloud resources: Fusing online learning and online optimization. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017. 81, 85

[257] Huawei Huang and Song Guo. Proactive failure recovery for nfv in distributed edge computing. *IEEE Communications Magazine*, 57(5):131–137, 2019. 81

[258] FJ Moreno-Muro, M Garrich, C San-Nicolás-Martínez, M Hernández-Bastida, P Pavón-Mariño, A Bravalheri, AS Muqaddas, N Uniyal, R Nejabati, D Simeonidou, et al. Joint vnf and multi-layer resource allocation with an open-source optimization-as-a-service integration. In *45th European Conference on Optical Communication (ECOC 2019)*, pages 1–4. IET, 2019. 81

[259] Anselme Ndikumana, Saeed Ullah, Tuan LeAnh, Nguyen H Tran, and Choong Seon Hong. Collaborative cache allocation and computation offloading in mobile edge computing. In *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 366–369. IEEE, 2017. 82

[260] Michael Till Beck, Sebastian Feld, Andreas Fichtner, Claudia Linnhoff-Popien, and Thomas Schimper. Me-volte: Network functions for energy-efficient video transcoding at the mobile edge. In *2015 18th International Conference on Intelligence in Next Generation Networks*, pages 38–44. IEEE, 2015. 82

[261] Yang Sun, Tingting Wei, Huixin Li, Yanhua Zhang, and Wenjun Wu. Energy-efficient multimedia task assignment and computing offloading for mobile edge computing networks. *IEEE Access*, 8:36702–36713, 2020. 82

[262] Ejaz Ahmed, Arif Ahmed, Ibrar Yaqoob, Junaid Shuja, Abdullah Gani, Muhammad Imran, and Muhammad Shoaib. Bringing computation closer toward the user network: Is edge computing the solution? *IEEE Communications Magazine*, 55(11):138–144, 2017. 82

[263] Shaoshan Liu, Liangkai Liu, Jie Tang, Bo Yu, Yifan Wang, and Weisong Shi. Edge computing for autonomous driving: Opportunities and challenges. *Proceedings of the IEEE*, 107(8):1697–1716, 2019. 82

[264] Saman Biookaghazadeh, Ming Zhao, and Fengbo Ren. Are fpgas suitable for edge computing? In {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018. 82

[265] Ian Colbert, Jake Daly, Ken Kreutz-Delgado, and Srinjoy Das. A competitive edge: Can fpgas beat gpus at dcnn inference acceleration in resource-limited edge computing applications? *arXiv preprint arXiv:2102.00294*, 2021. 82

[266] Sagar Arora, Pantelis A Frangoudis, and Adlen Ksentini. Exposing radio network information in a mec-in-nfv environment: the rnisaas concept. In *2019 IEEE Conference on Network Softwarization (NetSoft)*, pages 306–310. IEEE, 2019. 82

[267] Lechosław Tomaszewski, Sławomir Kukliński, and Robert Kołakowski. A new approach to 5g and mec integration. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 15–24. Springer, 2020. 82

[268] Xiantao Jiang, F Richard Yu, Tian Song, and Victor CM Leung. A survey on multi-access edge computing applied to video streaming: Some research issues and challenges. *IEEE Communications Surveys & Tutorials*, 2021. 83

[269] Gürkan Gür, Pawani Porambage, and Madhusanka Liyanage. Convergence of icn and mec for 5g: Opportunities and challenges. *IEEE Communications Standards Magazine*, 4(4):64–71, 2020. 83

[270] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 78:680–698, 2018. 83

[271] Yinhao Xiao, Yizhen Jia, Chunchi Liu, Xiuzhen Cheng, Jiguo Yu, and Weifeng Lv. Edge computing security: State of the art and challenges. *Proceedings of the IEEE*, 107(8):1608–1631, 2019. 83

[272] Sonia Shahzadi, Muddesar Iqbal, Tasos Dagiuklas, and Zia Ul Qayyum. Multi-access edge computing: open issues, challenges and future perspectives. *Journal of Cloud Computing*, 6(1):1–13, 2017. 83

[273] Jorge Martín-Pérez, Luca Cominardi, Carlos J Bernardos, Antonio de la Oliva, and Arturo Azcorra. Modeling mobile edge computing deployments for low latency multimedia services. *IEEE Transactions on Broadcasting*, 65(2):464–474, 2019. 83

[274] Jingjing Yao, Tao Han, and Nirwan Ansari. On mobile edge caching. *IEEE Communications Surveys & Tutorials*, 21(3):2525–2553, 2019. 83

[275] Xenofon Foukas, Georgios Patounas, Ahmed Elmokashfi, and Mahesh K Marina. Network slicing in 5g: Survey and challenges. *IEEE Communications Magazine*, 55(5):94–100, 2017. 84

[276] Ibrahim Afolabi, Tarik Taleb, Konstantinos Samdanis, Adlen Ksentini, and Hannu Flinck. Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. *IEEE Communications Surveys & Tutorials*, 20(3):2429–2453, 2018. 84, 86

[277] Akihiro Nakao, Ping Du, Yoshiaki Kiriha, Fabrizio Granelli, Anteneh Atumo Gebremariam, Tarik Taleb, and Miloud Bagaa. End-to-end network slicing for 5g mobile networks. *Journal of Information Processing*, 25:153–163, 2017. 84

[278] Peter Rost, Christian Mannweiler, Diomidis S Michalopoulos, Cinzia Sartori, Vincenzo Sciancalepore, Nishanth Sastry, Oliver Holland, Shreya Tayade, Bin Han, Dario Bega, et al. Network slicing to enable scalability and flexibility in 5g mobile networks. *IEEE Communications magazine*, 55(5):72–79, 2017. 84

[279] Jose Ordonez-Lucena, Pablo Ameigeiras, Diego Lopez, Juan J Ramos-Munoz, Javier Lorca, and Jesus Folgueira. Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges. *IEEE Communications Magazine*, 55(5):80–87, 2017. 84

[280] Shunliang Zhang. An overview of network slicing for 5g. *IEEE Wireless Communications*, 26(3):111–117, 2019. 84

[281] Zbigniew Kotulski, Tomasz Nowak, Mariusz Sepczuk, Marcin Tunia, Rafal Artych, Krzysztof Bocianiak, Tomasz Osko, and Jean-Philippe Wary. On end-to-end approach for slice isolation in 5g networks. fundamental challenges. In *2017 Federated conference on computer science and information systems (FedCSIS)*, pages 783–792. IEEE, 2017. 84

[282] Qian Li, Geng Wu, Apostolos Papathanassiou, and Udayan Mukherjee. An end-to-end network slicing framework for 5g wireless communication systems. *arXiv preprint arXiv:1608.00572*, 2016. 84

[283] Pol Alemany, L Juan, Ana Pol, Anton Roman, Panagiotis Trakadas, Panagiotis Karkazis, Marios Touloupou, Evgenia Kapassa, Dimosthenis Kyriazis, Thomas Soenen, et al. Network slicing over a packet/optical network for vertical applications applied to multimedia real-time communications. In *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–2. IEEE, 2019. 85

[284] Qi Wang, Jose Alcaraz-Calero, Ruben Ricart-Sanchez, Maria Barros Weiss, Anastasius Gavras, Navid Nikaein, Xenofon Vasilakos, Bernini Giacomo, Giardina Pietro, Mark Roddy, Michael Healy, Paul Walsh, Thuy Truong, Zdravko Bozakov, Konstantinos Koutsopoulos, Pedro Neves, Cristian Patachia-Sultanoiu, Marius Iordache, Elena Oproiu, Imen Grida Ben Yahia, Ciriaco Angelo, Cosimo Zotti, Giuseppe Celozzi, Donal Morris, Ricardo Figueiredo, Dean Lorenz, Salvatore Spadaro, George Agapiou, Ana Aleixo, and Cipriano Lomba. Enable Advanced QoS-Aware Network Slicing in 5G Networks for Slice-Based Media Use Cases. *IEEE Transactions on Broadcasting*, 65(2):444–453, June 2019. 85, 88, 89

[285] Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J García Clemente, Fabrizio Ippoliti, and Gregorio Martínez Pérez. Dynamic network slicing management of multimedia scenarios for future remote healthcare. *Multimedia Tools and Applications*, 78(17):24707–24737, 2019. 85

[286] Claudia Campolo, Antonella Molinaro, Antonio Iera, and Francesco Menichella. 5g network slicing for vehicle-to-everything services. *IEEE Wireless Communications*, 24(6):38–45, 2017. 85

[287] Jie Mei, Xianbin Wang, and Kan Zheng. Intelligent network slicing for v2x services toward 5g. *IEEE Network*, 33(6):196–204, 2019. 85

[288] Fabian Kurtz, Caner Bektas, Nils Dorsch, and Christian Wietfeld. Network slicing for critical communications in shared 5g infrastructures-an empirical evaluation.

In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pages 393–399. IEEE, 2018. 85

[289] Alcardo Alex Barakabitze, Arslan Ahmad, Rashid Mijumbi, and Andrew Hines. 5g network slicing using sdn and nfv: A survey of taxonomy, architectures and future challenges. *Computer Networks*, 167:106984, 2020. 85

[290] Spyridon Vassilaras, Lazaros Gkatzikis, Nikolaos Liakopoulos, Ioannis N Stiakogiannakis, Meiyu Qi, Lei Shi, Liu Liu, Merouane Debbah, and Georgios S Paschos. The algorithmic aspects of network slicing. *IEEE Communications Magazine*, 55(8):112–119, 2017. 86

[291] Zubair Md Fadlullah, Fengxiao Tang, Bomin Mao, Nei Kato, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials*, 19(4):2432–2455, 2017. 86

[292] Chih-Lin I, Sławomir Kuklinskí, and Tao Chen. A perspective of o-ran integration with mec, son, and network slicing in the 5g era. *IEEE Network*, 34(6):3–4, 2020. 86

[293] Yiqing Zhou, Ling Liu, Lu Wang, Ning Hui, Xinyu Cui, Jie Wu, Yan Peng, Yanli Qi, and Chengwen Xing. Service aware 6g: an intelligent and open network based on convergence of communication, computing and caching. *Digital Communications and Networks*, 2020. 86, 87

[294] Yiming Wei, Mugen Peng, and Yaqiong Liu. Intent-based networks for 6g: Insights and challenges. *Digital Communications and Networks*, 6(3):270–280, 2020. 86

[295] Jean-Baptiste Monteil, Jernej Hribar, Pieter Barnard, Yong Li, and Luiz A DaSilva. Resource reservation within sliced 5g networks: A cost-reduction strategy for service providers. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2020. 86

[296] Dawei Chen, Yin-Chen Liu, BaekGyu Kim, Jiang Xie, Choong Seon Hong, and Zhu Han. Edge computing resources reservation in vehicular networks: A meta-learning approach. *IEEE Transactions on Vehicular Technology*, 69(5):5634–5646, 2020. 86

[297] Xinwei Wang, Jiandong Li, Lingxia Wang, Chungang Yang, and Zhu Han. Intelligent user-centric network selection: A model-driven reinforcement learning framework. *IEEE Access*, 7:21645–21661, 2019. 86

[298] Jagadeesha R Bhat and Salman A Alqahtani. 6g ecosystem: Current status and future perspective. *IEEE Access*, 9:43134–43167, 2021. 87

[299] Zaheer Allam and David S Jones. Future (post-covid) digital, smart and sustainable cities in the wake of 6g: Digital twins, immersive realities and new urban economies. *Land Use Policy*, 101:105201, 2021. 87

[300] Muhammad Waseem Akhtar, Syed Ali Hassan, Rizwan Ghaffar, Haejoon Jung, Sahil Garg, and M Shamim Hossain. The shift to 6g communications: vision and requirements. *Human-centric Computing and Information Sciences*, 10(1):1–27, 2020. 87

[301] Ahmad Rostami. Private 5g networks for vertical industries: Deployment and operation models. In *2019 IEEE 2nd 5G World Forum (5GWF)*, pages 433–439. IEEE, 2019. 87

[302] Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. Open, programmable, and virtualized 5g networks: State-of-the-art and the road ahead. *Computer Networks*, 182:107516, 2020. 87

[303] Olga Galinina, Alexander Pyattaev, Sergey Andreev, Mischa Dohler, and Yevgeni Koucheryavy. 5g multi-rat lte-wifi ultra-dense small cells: Performance dynamics, architecture, and trends. *IEEE Journal on Selected Areas in Communications*, 33(6):1224–1240, 2015. 87

[304] Pavlos Basaras, George Iosifidis, Stepan Kucera, and Holger Claussen. Multicast optimization for video delivery in multi-rat networks. *IEEE Transactions on Communications*, 68(8):4973–4985, 2020. 87

[305] Sem Borst, Aliye Özge Kaya, Doru Calin, and Harish Viswanathan. Dynamic path selection in 5g multi-rat wireless networks. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017. 87

[306] Ahmet Yazar, Seda Doğan Tusha, and Huseyin Arslan. 6g vision: An ultra-flexible perspective. *ITU Journal on Future and Evolving Technologies*, 1(1), 2020. 87

[307] Tongyi Huang, Wu Yang, Jun Wu, Jin Ma, Xiaofei Zhang, and Daoyin Zhang. A survey on green 6g network: Architecture and technologies. *IEEE Access*, 7:175758–175768, 2019. 87

[308] Daniela Renga and Michela Meo. From self-sustainable green mobile networks to enhanced interaction with the smart grid. In *2018 30th International Teletraffic Congress (ITC 30)*, volume 1, pages 129–134. IEEE, 2018. 87

[309] Hatem Abou-Zeid and Hossam S Hassanein. Predictive green wireless access: Exploiting mobility and application information. *IEEE wireless communications*, 20(5):92–99, 2013. 87

[310] Abbas Mehrabi, Matti Siekkinen, and Antti Ylä-Jääski. Energy-aware qoe and backhaul traffic optimization in green edge adaptive mobile video streaming. *IEEE Transactions on Green Communications and Networking*, 3(3):828–839, 2019. 87

[311] Chenren Xu, Lei Yang, and Pengyu Zhang. Practical backscatter communication systems for battery-free internet of things: A tutorial and survey of recent research. *IEEE Signal Processing Magazine*, 35(5):16–27, 2018. 87

[312] Saman Naderiparizi, Mehrdad Hessar, Vamsi Talla, Shyamnath Gollakota, and Joshua R Smith. Towards battery-free {HD} video streaming. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*, pages 233–247, 2018. 87

[313] Ali Saffari, Mehrdad Hessar, Saman Naderiparizi, and Joshua R Smith. Battery-free wireless video streaming camera system. In *2019 IEEE International Conference on RFID (RFID)*, pages 1–8. IEEE, 2019. 87

[314] Yazan Al-Issa, Mohammad Ashraf Ottom, and Ahmed Tamrawi. ehealth cloud security challenges: a survey. *Journal of healthcare engineering*, 2019, 2019. 87

[315] Jin Cui, Lin Shen Liew, Giedre Sabaliauskaite, and Fengjun Zhou. A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. *Ad Hoc Networks*, 90:101823, 2019. 87

[316] Minghao Wang, Tianqing Zhu, Tao Zhang, Jun Zhang, Shui Yu, and Wanlei Zhou. Security and privacy in 6g networks: New areas and new challenges. *Digital Communications and Networks*, 6(3):281–291, 2020. 87

[317] CogNet (Building an Intelligent System of Insights and Action for 5G Network Management). 88

[318] Imen Grida Ben Yahia, Jaafar Bendriss, Alassane Samba, and Philippe Dooze. Cog-Nitive 5G networks: Comprehensive operator use cases with machine learning for management operations. In *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pages 252–259, Paris, March 2017. IEEE. 88, 89

[319] Haytham Assem, Lei Xu, Teodora Sandra Buda, and Declan O'Sullivan. Machine learning as a service for enabling Internet of Things and People. *Personal and Ubiquitous Computing*, 20(6):899–914, November 2016. 88, 89

[320] SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). 88

[321] Alberto Huertas Celdran, Manuel Gil Perez, Felix J. Garcia Clemente, and Gregorio Martinez Perez. Enabling Highly Dynamic Mobile Scenarios with Software Defined Networking. *IEEE Communications Magazine*, 55(4):108–113, April 2017. 88, 89

[322] SliceNet (End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks). 88

[323] Pablo Salva-Garcia, Jose M. Alcaraz-Calero, Qi Wang, Miguel Arevalillo-Herraez, and Jorge Bernal Bernabe. Scalable Virtual Network Video-Optimizer for Adaptive Real-Time Video Transmission in 5G Networks. *IEEE Transactions on Network and Service Management*, 17(2):1068–1081, June 2020. 88

[324] SoftFIRE (Software Defined Networks and Network Function Virtualization Testbed within FIRE+). 88

[325] David Lake, Gerry Foster, Serdar Vural, Yogaratnam Rahulan, Bong-Hwan Oh, Ning Wang, and Rahim Tafazolli. Virtualising and orchestrating a 5G evolved packet core network. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–5, July 2017. 88, 89

[326] FLAME (Facility for Large-scale Adaptive Media Experimentation), January 2017. 88

[327] Kay Haensge, Dirk Trossen, Sebastian Robitzsch, Michael Boniface, and Stephen Phillips. Cloud-Native 5G Service Delivery Platform. In *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–7, November 2019. 88, 89

[328] 5GTango (5G Development and validation platform for global industry-specific network services and Apps). 88

[329] Manuel Peuster, Stefan Schneider, Mengxuan Zhao, George Xilouris, Panagiotis Trakadas, Felipe Vicens, Wouter Tavernier, Thomas Soenen, Ricard Vilalta, George Andreou, Dimosthenis Kyriazis, and Holger Karl. Introducing Automated Verification and Validation for Virtualized Network Functions and Services. *IEEE Communications Magazine*, 57(5):96–102, May 2019. 88, 89

[330] Thomas Soenen, Wouter Tavernier, Manuel Peuster, Felipe Vicens, George Xilouris, Stavros Kolometsos, Michail-Alexandros Kourtis, and Didier Colle. Empowering network service developers: enhanced nfv devops and programmable mano. *IEEE Communications Magazine*, May 2019. 88, 89

[331] 5G-Media (Programmable Edge-to-Cloud Virtualization Fabric for the 5G Media Industry). 88, 90

[332] David Breitgand, Avi Weit, Stamatia Rizou, David Griffin, Ugur Acar, Gino Carrozzo, Nikolaos Zioulis, Pasquale Andriani, and Francesco Iadanza. Towards

Serverless NFV for 5G Media Applications. In *Proceedings of the 11th ACM International Systems and Storage Conference*, SYSTOR '18, page 118, Haifa, Israel, June 2018. Association for Computing Machinery. 88, 90

[333] 5Growth (5G-enabled Growth in Vertical Industries). 88, 90

[334] Xi Li, Andres Garcia-Saavedra, Xavier Costa-Perez, Carlos J Bernardos, Carlos Guimarães, Kiril Antevski, Josep Mangues-Bafalluy, Jorge Baranda, Engin Zeydan, Daniel Corujo, et al. 5growth: An end-to-end service platform for automated deployment and management of vertical services over 5g networks. *IEEE Communications Magazine*, 59(3):84–90, 2021. 88, 90

[335] 5GCity – A distributed cloud & radio platform for 5G Neutral Hosts. 88

[336] Carlos Colman-Meixner, Hamzeh Khalili, Konstantinos Antoniou, Muhammad Shuaib Siddiqui, Apostolos Papageorgiou, Antonino Albanese, Paolo Cruschelli, Gino Carrozzo, Luca Vignaroli, Alexandre Ulisses, Pedro Santos, Jordi Colom, Ioannis Neokosmidis, David Pujals, Rita Spada, Antonio Garcia, Sergi Figerola, Reza Nejabati, and Dimitra Simeonidou. Deploying a Novel 5G-Enabled Architecture on City Infrastructure for Ultra-High Definition and Immersive Media Production and Broadcasting. *IEEE Transactions on Broadcasting*, 65(2):392–403, June 2019. 88, 90

[337] OpenAirInterface – 5G software alliance for democratising wireless innovation. 88, 90

[338] Mosaic5G. 88, 90

[339] Navid Nikaein, Chia-Yu Chang, and Konstantinos Alexandris. Mosaic5G: agile and flexible service platforms for 5G research. *ACM SIGCOMM Computer Communication Review*, 48(3):29–34, September 2018. 88, 90

[340] O-ran alliance. 88, 90

[341] James Nightingale, Qi Wang, Jose M Alcaraz Calero, Enrique Chirivella-Perez, Marian Ulbricht, Jesus A Alonso-Lopez, Ricardo Preto, Tiago Batista, Tiago Teixeira, Maria Joao Barros, et al. Qoe-driven, energy-aware video adaptation in 5g

networks: The selfnet self-optimisation use case. *IJDSN*, 12(1):7829305–1, 2016. 89

[342] Chia-Yu Chang, Navid Nikaein, Osama Arouk, Kostas Katsalis, Adlen Ksentini, Thierry Turletti, and Konstantinos Samdanis. Slice Orchestration for Multi-Service Disaggregated Ultra-Dense RANs. *IEEE Communications Magazine*, 56(8):70–77, August 2018. 89

[343] SRT Allicance. Secure reliable transport protocol, 2018. 99, 102

[344] Michael Luby, Lorenzo Vicisano, Jim Gemmell, Luigi Rizzo, M Handley, and Jon Crowcroft. The use of forward error correction (fec) in reliable multicast. Technical report, RFC 3453, December, 2002. 99, 102

[345] Sheng Wei and Viswanathan Swaminathan. Low latency live video streaming over http 2.0. In *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, pages 37–42, 2014. 101, 116

[346] Ali El Essaili, Thorsten Lohmar, and Mohamed Ibrahim. Realization and evaluation of an end-to-end low latency live dash system. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–5. IEEE, 2018. 101, 116, 122

[347] Roberto Viola, Alvaro Gabilondo, Angel Martin, Juan Felipe Mogollón, Mikel Zorrilla, and Jon Montalbán. Qoe-based enhancements of chunked cmaf over low latency video streams. In *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–6. IEEE, 2019. 101

[348] J Ott and C Perkins. Guidelines for extending the rtp control protocol (rtcp). Technical report, RFC 5968, September, 2010. 101

[349] Bolun Wang, Xinyi Zhang, Gang Wang, Haitao Zheng, and Ben Y Zhao. Anatomy of a personalized livestreaming system. In *Proceedings of the 2016 Internet Measurement Conference*, pages 485–498, 2016. 102

[350] Maxim Claeys, Steven Latre, Jeroen Famaey, and Filip De Turck. Design and evaluation of a self-learning http adaptive video streaming client. *IEEE communications letters*, 18(4):716–719, 2014. 103, 117, 122, 124, 188

[351] Carlos M Lentisco, Luis Bellido, Encarna Pastor, et al. Qoe-based analysis of dash streaming parameters over mobile broadcast networks. *IEEE Access*, 5:20684–20694, 2017. 103, 117, 188

[352] Peter A Sarginson. Mpeg-2: Overview of the systems layer. 1996. 104

[353] Gstreamer: open source multimedia framework. 104, 105, 108, 117, 119, 145, 178

[354] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014. 108

[355] Traffic control. 109

[356] Pravir Chawdhry, Gianluca Folloni, Stefano Luzardi, and Stefano Lumachi. European broadband user experience. european commission, joint research centre (jrc). [dataset]. 109

[357] Xiph.Org Foundation. Xiph.org video test media. 109, 122

[358] Cisco. Visual networking index: Forecast and methodology, 2017-2022, 2018. 113, 146

[359] Jordi Calabuig, Jose F Monserrat, and David Gomez-Barquero. 5th generation mobile networks: A new opportunity for the convergence of mobile broadband and broadcast services. *IEEE Communications Magazine*, 53(2):198–205, 2015. 113

[360] Thorsten Lohmar, Torbjörn Einarsson, Per Fröjdh, Frédéric Gabin, and Markus Kampmann. Dynamic adaptive http streaming of live content. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–8. IEEE, 2011. 113, 139

[361] Viswanathan Swaminathan and Sheng Wei. Low latency live video streaming using http chunked encoding. In *2011 IEEE 13th International Workshop on Multimedia Signal Processing*, pages 1–6. IEEE, 2011. 116

[362] Tobias Hoßfeld, Sebastian Egger, Raimund Schatz, Markus Fiedler, Kathrin Masuch, and Charlott Lorentzen. Initial delay vs. interruptions: Between the devil and the deep blue sea. In *2012 Fourth International Workshop on Quality of Multimedia Experience*, pages 1–6. IEEE, 2012. 117, 122, 124

[363] Node.js: asynchronous event driven javascript runtime. 119, 145, 178

[364] ETSI. Dynamic adaptive streaming over http (3gp-dash), 2018. 120

[365] Mozilla MDN Web Docs. Range http request header. 120

[366] Kyungmo Park, Namgi Kim, and Byoung-Dai Lee. Performance evaluation of the emerging media-transport technologies for the next-generation digital broadcasting systems. *IEEE Access*, 5:17597–17606, 2017. 131

[367] Ali Begen, Tankut Akgul, and Mark Baugher. Watching video over the web: Part 1: Streaming protocols. *IEEE Internet Computing*, 15(2):54–63, 2010. 131, 186

[368] Patrick Maillé and Galina Schwartz. Content providers volunteering to pay network providers: Better than neutrality? In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 484–489. IEEE, 2016. 131

[369] 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*, pages 9–14, 2012. 131, 165, 168, 169

[370] Simon Da Silva, Joachim Bruneau-Queyreix, Mathias Lacaud, Daniel Negru, and Laurent Réveillère. Muslin: A qoe-aware cdn resources provisioning and advertising system for cost-efficient multisource live streaming. *International Journal of Network Management*, 30(3):e2081, 2020. 132

[371] Zhi-Li Zhang. Feel free to cache: Towards an open cdn architecture for cloud-based content distribution. In *2014 International Conference on Collaboration Technologies and Systems (CTS)*, pages 488–490. IEEE, 2014. 134

[372] Sa'di Altamimi and Shervin Shirmohammadi. Qoe-fair dash video streaming using server-side reinforcement learning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(2s):1–21, 2020. 136

[373] Jing Wang, Jian Tang, Zhiyuan Xu, Yanzhi Wang, Guoliang Xue, Xing Zhang, and Dejun Yang. Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017. 138, 142

[374] Information technology — mpeg systems technologies — part 7: Common encryption in iso base media file format files. *ISO/IEC*, 2016. 139

[375] Emmanuel Thomas, MO Van Deventer, Thomas Stockhammer, Ali C Begen, and Jeroen Famaey. Enhancing mpeg dash performance via server and network assistance. In *IBC 2015 Conference*. IET, 2015. 140

[376] Lucas Nussbaum. An overview of fed4fire testbeds–and beyond? In *GEFI-Global Experimentation for Future Internet Workshop*, 2019. 144

[377] Nikos Makris, Christos Zarafetas, Spyros Kechagias, Thanasis Korakis, Ivan Seskar, and Leandros 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. IEEE, 2015. 144

[378] Stefan Lederer, Christopher Mueller, Christian Timmerer, Cyril Concolato, Jean Le Feuvre, and Karel Fliegel. Distributed dash dataset. In *Proceedings of the 4th ACM Multimedia Systems Conference*, pages 131–135, 2013. 145, 146, 178, 196

[379] Tensorflow: end-to-end open source platform for machine learning. 146

[380] Clinton Gormley and Zachary Tong. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.", 2015. 146

[381] Yuvraj Gupta. *Kibana essentials*. Packt Publishing Ltd, 2015. 146

[382] Koffka Khan and Wayne Goodridge. What happens when adaptive video streaming players compete in time-varying bandwidth conditions? *International Journal of Advanced Networking and Applications*, 10(1):3704–3712, 2018. 165

[383] ETSI. Etsi ts 122 261 version 15.6.0 - 5g; service requirements for next generation new services and markets, 2018. 165

[384] Jan De Cock, Zhi Li, Megha Manohara, and Anne Aaron. Complexity-based consistent-quality encoding in the cloud. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1484–1488. IEEE, 2016. 167

[385] Eirini Liotou, Dimitris Tsolkas, and Nikos Passas. A roadmap on qoe metrics and models. In *2016 23rd International Conference on Telecommunications (ICT)*, pages 1–5. IEEE, 2016. 167

[386] Weiwei Huang, Yipeng Zhou, Xueyan Xie, Di Wu, Min Chen, and Edith Ngai. Buffer state is enough: Simplifying the design of qoe-aware http adaptive video streaming. *IEEE Transactions on Broadcasting*, 64(2):590–601, 2018. 167

[387] Tarun Mangla, Emir Halepovic, Mostafa Ammar, and Ellen Zegura. Mimic: Using passive network measurements to estimate http-based adaptive video qoe metrics. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–6. IEEE, 2017. 167

[388] Tarun Mangla, Emir Halepovic, Mostafa Ammar, and Ellen Zegura. emimic: Estimating http-based video qoe metrics from encrypted network traffic. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–8. IEEE, 2018. 167

[389] M Hammad Mazhar and Zubair Shafiq. Real-time video quality of experience monitoring for https and quic. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1331–1339. IEEE, 2018. 167

[390] Christopher Mueller, Stefan Lederer, Reinhard Grandl, and Christian Timmerer. Oscillation compensating dynamic adaptive streaming over http. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2015. 168

[391] Saamer Akhshabi, Lakshmi Anantakrishnan, Constantine Dovrolis, and Ali C Begen. Server-based traffic shaping for stabilizing oscillating adaptive streaming players. In *Proceeding of the 23rd ACM workshop on network and operating systems support for digital audio and video*, pages 19–24, 2013. 168

[392] Lauri Koskimies, Tarik Taleb, and Miloud Bagaa. Qoe estimation-based server benchmarking for virtual video delivery platform. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017. 168

[393] Xiaohu Ge, Song Tu, Guoqiang Mao, Cheng-Xiang Wang, and Tao Han. 5g ultra-dense cellular networks. *IEEE Wireless Communications*, 23(1):72–79, 2016. 168

[394] Christian Quadri, Sabrina Gaito, Roberto Bruschi, Franco Davoli, and Gian Paolo Rossi. A mec approach to improve qoe of video delivery service in urban spaces. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 25–32. IEEE, 2018. 168

[395] Jose Oscar Fajardo, Ianire Taboada, and Fidel Liberal. Improving content delivery efficiency through multi-layer mobile edge adaptation. *IEEE Network*, 29(6):40–46, 2015. 168

[396] Ayman Younis, Tuyen X Tran, and Dario Pompili. On-demand video-streaming quality of experience maximization in mobile edge computing. In *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 1–9. IEEE, 2019. 168

[397] Xi Zhang and Jingqing Wang. Joint heterogeneous statistical-qos/qoe provisionings for edge-computing based wifi offloading over 5g mobile wireless networks. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2018. 168, 169

[398] Chang Ge and Ning Wang. Real-time qoe estimation of dash-based mobile video applications through edge computing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 766–771. IEEE, 2018. 169

[399] Itu-t rec. p.1203 standalone implementation. 174

[400] Alexander Raake, Marie-Neige Garcia, Werner Robitza, Peter List, Steve Göring, and Bernhard Feiten. A bitstream-based, scalable video-quality model for http adaptive streaming: Itu-t p. 1203.1. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2017. 174

[401] Mozilla MDN Web Docs. User-agent. 174

[402] Matt Richardson and Shawn Wallace. *Getting started with raspberry PI.* " O'Reilly Media, Inc.", 2012. 178

[403] Cisco. Visual networking index: Forecast and methodology, 2016-2021, 2017. 186

[404] Cisco. Visual networking index: Global mobile data traffic forecast update 2014-2019, 2016. 186

[405] 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*, pages 55–60, 2016. 188

[406] Cedexis. Multi-cdn architecture. 188