Towards pragmatic solutions to improve the quality of video streaming in current and future networks

Mathias Lacaud

THESE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE

L'UNIVERSITÉ DE BORDEAUX

École Doctorale Mathématiques et Informatique

Spécialité Informatique

Mathias LACAUD

VERS DES SOLUTIONS PRAGMATIQUES POUR AMÉLIORER LA QUALITÉ DES DIFFUSIONS VIDÉO DANS LES RÉSEAUX ACTUELS ET LES RÉSEAUX DU FUTUR

TOWARDS PRAGMATIC SOLUTIONS TO IMPROVE THE QUALITY OF VIDEO STREAMING IN CURRENT AND FUTURE NETWORKS

Sous la direction de :

Daniel Négru

Soutenue le 8 Octobre 2020

Membres du jury :

Xavier Blanc Professeur, Université de Bordeaux ..... Président du jury

Yérom-David Bromberg Professeur, Université de Rennes ..... Rapporteur

Adlen Ksentini Professeur, Eurecom ..... Rapporteur

Sonia Ben Mokhtar Directeur de recherche, LIRIS-CNRS ..... Examinateur

Evangelos Pallis Professeur, Hellenic Mediterranean University ..... Examinateur

Daniel Negru Maître de Conférences, Bordeaux INP ..... Directeur de thèse

Résumé

Titre : Vers des solutions pragmatiques pour améliorer la qualité des diffusions vidéo dans les réseaux actuels et les réseaux du futur

La diffusion vidéo devrait atteindre 82% du trafic total sur Internet en 2022. Il y a deux raisons à ce succès : la multiplication des sources de contenu vidéo et la démocratisation des connexions haut-débit à Internet. Les principales plateformes de streaming vidéo dépendent d'infrastructures planétaires pour répondre à la demande croissante en qualité visuelle. Cependant, ces infrastructures sont coûteuses et peuvent souffrir de pannes ou indisponibilités locales. La diffusion pair-à-pair est une solution alternative permettant aux clients de récupérer les segments vidéo déjà téléchargés par d'autres utilisateurs. Un système permettant de diffuser simultanément les contenus vidéo depuis plusieurs pairs fournirait une haute fiabilité et une meilleure qualité d'expérience. Par ailleurs, les réseaux 5G sont progressivement déployés dans le monde, et soulèvent des problèmes concernant la diffusion vidéo. En effet, les architectures des réseaux sans fil d'intérieur comporteront des technologies hétérogènes. Il est ainsi nécessaire de trouver des solutions fiables et pragmatiques pour garantir la stabilité des diffusions vidéo sur de multiples chemins.

Cette thèse propose deux contributions pragmatiques afin d'améliorer la qualité d'expérience utilisateur pour la diffusion vidéo : (1) *PMS+*, une solution permettant la réception simultanée depuis de multiples pairs, et (2) *MSS/RRLH*, un système d'agrégation des flux sur plusieurs chemins pour les réseaux 5G.

Mots-clés : Streaming, Qualité d'Expérience, Multi-sources, Pair-à-pair, 5G

Laboratoire Bordelais de Recherche en Informatique (LaBRI)
Unité Mixte de Recherche CNRS (UMR 5800)
351 cours de la Libération, 33400 Talence, France

Abstract

Title: Towards pragmatic solutions to improve the quality of video streaming in current and future networks

Video streaming is expected to reach 82% of all Internet traffic in 2022. There are two reasons for this success: the multiplication of video sources and the pervasiveness of high quality Internet connections. Dominating video streaming platforms rely on large-scale infrastructures to cope with an increasing demand for high quality of experience and high-bitrate content. However, these infrastructures are expensive and can become overloaded or unavailable. Peer-to-peer streaming is an alternative solution enabling clients to download video segments from other peers. A streaming system able to simultaneously retrieve content from multiple peers would be highly reliable while providing a higher quality of experience. Meanwhile, 5G networks are being deployed globally, and raise several issues regarding video streaming. Indoor wireless networks architectures will include heterogeneous technologies. There is currently no reliable solution enabling high quality of experience video streaming over multiple heterogeneous paths.

This thesis proposes two pragmatic contributions to enhance end-user's quality of experience in video streaming: (1) *PMS+*, a solution enabling concurrent streaming from multiple peers, and (2) *MSS/RRLH*, a multiple-path stream aggregation system for indoor 5G networks.

Keywords: Streaming, Quality of Experience, Multiple-source, Peer-to-peer, 5G

Laboratoire Bordelais de Recherche en Informatique (LaBRI)
Unité Mixte de Recherche CNRS (UMR 5800)
351 cours de la Libération, 33400 Talence, France

Remerciements

Les travaux de recherche présentés donc ce document ont été possibles grâce à de nombreuses personnes. Je tiens à remercier en particulier Daniel Negru, mon directeur de thèse, qui m'a accompagné tout au long de cette aventure, depuis mon premier stage sous sa supervision. Merci également à Joachim Bruneau-Queyreix avec qui j'ai pu découvrir le monde de la recherche et commencer mes premiers travaux. Un grand merci à Simon Da Silva qui m'a apporté son aide, son soutien et souvent son café depuis notre premier jour dans le bureau 257.

Je souhaite remercier tous les collaborateurs de l'équipe PROGRESS au LaBRI pour m'avoir accueilli et m'avoir offert une place pour m'exprimer. Dans une même idée, je remercie toute l'équipe de JOADA pour leur soutien et leurs conseils avisés, ainsi que l'équipe de Viewsurf, client de l'entreprise, pour leur contribution à certaines expériences menées durant ces trois années. Merci aussi à Leo et Stefan qui ont été mon bureau, ainsi qu'à tous les doctorants du laboratoire qui ont été présent dans mon quotidien et avec lesquels nous avons eu quelques débats interminables dans la cuisine et autour de la machine à café.

I want to thank every collaborator of the Internet of Radio Light project for their contributions during the project leading to the elaboration of successful deliverables and promising results. In particular, I want to thank Adam and John, respectively the logistical and technical leaders.

Du côté de l'enseignement, je tiens à remercier l'équipe pédagogique des départements Telecom et RSI de l'ENSEIRB qui m'ont aidé à m'intégrer et à préparer les nombreuses heures de cours, TP et TD que j'ai donné. Merci également à tous les étudiants d'avoir prêté une oreille plus ou moins attentive à tout ce que j'avais à leur apporter. Merci surtout à Delphine pour m'avoir fourni régulièrement en madeleines et barres chocolatées, carburant essentiel pour tenir de longues matinées de cours en plein hiver.

Merci à mes amis qui se reconnaitront (Maxime, Ben, Arnaud, Jérémy, Yoann, Estelle et bien d'autres) toujours disponibles pour m'aider à me changer les idées et m'aérer l'esprit autour d'une petite partie de jeu vidéo pour les plus loins ou autour d'une petite bière pour les plus proches. Merci à ma famille, et à mes parents Eric et Mary, pour leurs messages toujours encourageants et pour la confiance qu'ils m'ont accordé quand je me suis lancé dans cette thèse. Merci aux joyeux musiciens de The Old Joe's Kaya pour avoir su s'adapter à un emploi du temps pas toujours très simple et pour m'avoir redonné le sourire dans des concerts endiablés.

Merci enfin à Lucie qui partage ma vie au jour le jour, pour toutes les heures passées avec moi, pour son soutien sans faille et sa patience au quotidien, en particulier durant le confinement.

En un mot : ***merci.***

Contents

List of Acronyms

5G PPP 5G Infrastructure Public Private Partnership

3GPP 3rd Generation Partnership Project

AS Autonomous System

CAP Candidate Assisting Peer

CDN Content Delivery Network

CMAF Common Media Application Format

DASH Dynamic Adaptive Streaming over HTTP

DTLS Datagram Transport Layer Security

GoP Group of Picture

HAS HTTP Adaptive Streaming

HeNB Home eNode B, cellular home network

HG Home Gateway

HIPG Home IP Gateway

HLS HTTP Live Streaming

IoRL Internet of Radio Light

ISP Internet Service Provider

MNO Mobile Network Operator

MPD Media Presentation Description

MS-Stream Multiple-Source Streaming

MSS/RRLH Multiple-Source Streaming over Remote Radio Light Head

NFV Network Function Virtualization

OR Overhead Reduction

OTT Over-The-Top

P2P Peer-to-Peer

POF Plastic Optical Fiber

QoE Quality of Experience

RF Radio Frequency

RRLH Remote Radio Light Head

RTMP Real Time Messaging Protocol

RTP Real-time Transport Protocol

RTSP Real Time Streaming Protocol

SCTP Stream Control Transmission Protocol

SDN Software-Defined Networking

SDP Session Description Protocol

SRTP Secure Real-time Transport Protocol

STUN Simple Traversal of UDP through NATs

TCP Transmission Control Protocol

TURN Traversal Using Relays around NAT

UDP User Datagram Protocol

vHG virtual Home Gateway

VLC Visible Light Controller

VM Virtual Machine

VNF-MANO Network Functions Virtualization Management and Orchestration

VoD Video on Demand

WebRTC Web Real-Time Communication

WLAN Wireless Local Area Network

List of Figures

List of Tables

Chapter 1

Introduction

"Show me who you are, and I will show you where you need to go."

Sander Cohen

1.1 Motivations

Many new trends appeared with the democratization of the Internet. Video contents began to increase both in quantity and in quality, thanks to platforms such as YouTube [Youtube, 2020], Vimeo [Vimeo, 2020] or Dailymotion [Dailymotion, 2020], all providing income through ads and sponsors. People look for specialized contents, which they can now consume on several devices (desktop and laptop computers, tablets, smartphones, smart TVs, etc.) whenever they want to. This facilitated the emergence of VoD services, rapidly followed by live streaming. The Over-The-Top (OTT) type of delivery, through websites, apps or set-top-boxes, is currently the most widespread way to deliver video content to customers.

Video streaming represented more than 60% of all Internet traffic [Sandvine, 2019] and 65% of worldwide mobile downstream traffic in 2020 [Sandvine, 2020]. According to Cisco [Cisco, 2018], video traffic is experiencing a tremendous growth and is expected to exceed 82% of the total Internet traffic by 2022, and live video will grow 15-fold to reach 17% of all video traffic by 2022. Two reasons account for this success: the multiplication of video sources (e.g., video streaming catalogs, online TV channels, personal videos sharing) and the pervasiveness of high-quality Internet connections. Most of the time, although traffic increase is rightfully forecast, core networks capacities are not upgraded due to the high cost of such an operation. Major issues consequently arise with respect

to the QoE of such services. Providing a high (and fairly shared among users) QoE is thus a rising issue, as servers and network links become overloaded.

1.1.1 Current networks: quality and scalability issues of industrial video platforms systems

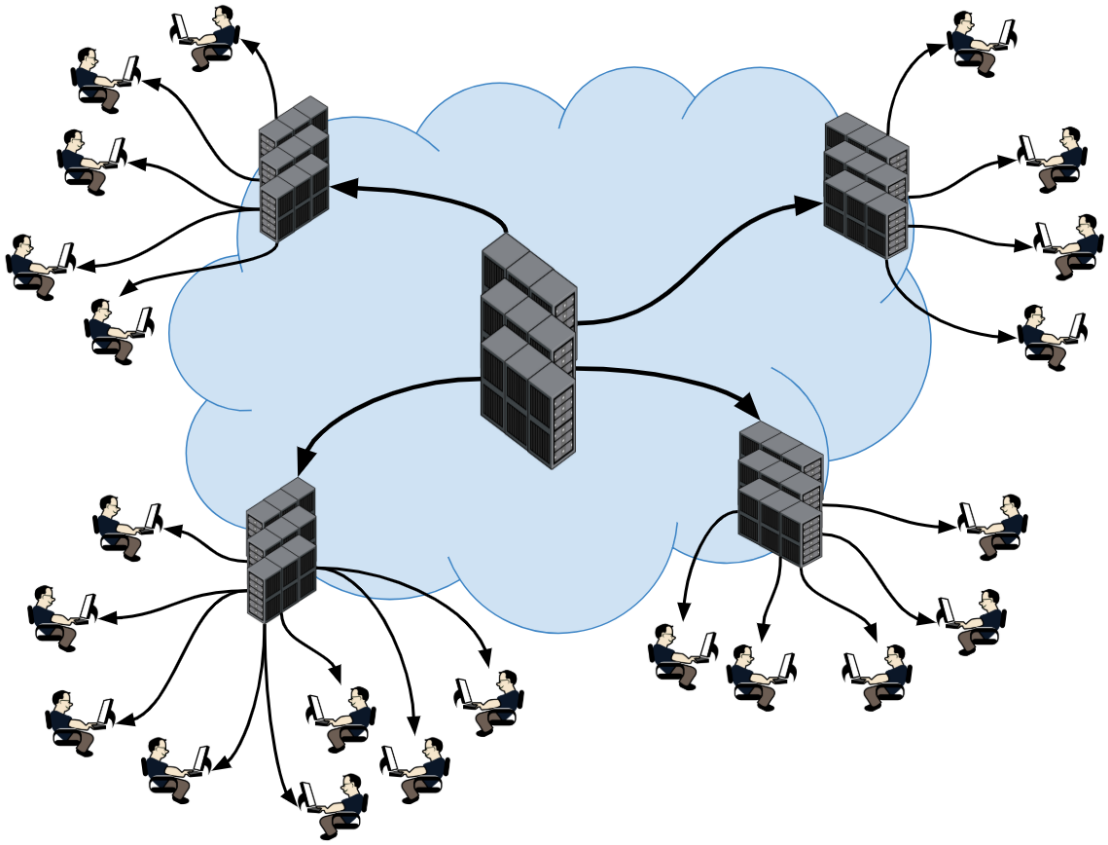

Figure 1.1: CDN illustration

Dominating video streaming platforms currently rely on large-scale infrastructures to cope with an increasing demand for high QoE and high-bitrate content. **CDN** are extensively used for the delivery of video content over the Internet (see Figure 1.1). For instance, YouTube [YouTube, 2020], Netflix [Netflix, 2020] or Twitch [Twitch, 2020] have set up planetary-scale proprietary CDN [Deng et al., 2017, Böttger et al., 2018, Adhikari et al., 2012d, Adhikari et al., 2012b]. They further deploy extra CDN nodes directly inside ISP networks (e.g., Google Global Caches) and negotiate special peering relations with their Autonomous Systems [Mok et al., 2018]. Other platforms usually rely on existing third-party CDN to serve content. Dailymotion [Dailymotion, 2020] is reported to use

the CDN of Orange, Akamai, and Limelight to scale video delivery in different parts of the world [Botta et al., 2018]. In such architectures, geographically distributed replica servers located as close as possible to the consuming clients are provisioned in advance with sufficient capacities using estimates of the expected workload. When accessing a content, consuming clients are automatically re-directed to the closest server to temper network congestion and achieve higher throughput.

In addition to CDN-based architectures, streaming services rely on HTTP Adaptive Streaming (HAS) solutions, such as the widely adopted Dynamic Adaptive Streaming over HTTP (DASH) standard, from MPEG, or Apple's HTTP Live Streaming (HLS). These solutions enable consuming clients to dynamically adjust the requested content bitrate according to the observed network conditions or to the client buffer occupancy.

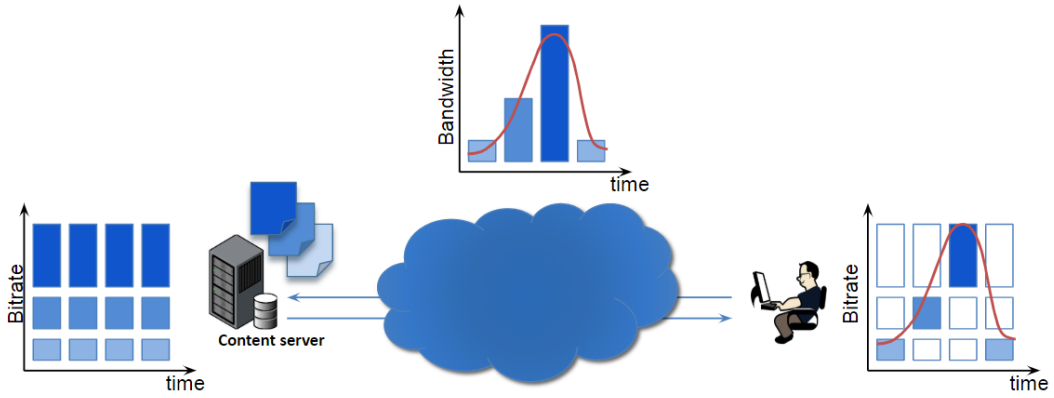

Figure 1.2: HAS illustration

Nevertheless, if a large amount of end-users located under the same geographic area is simultaneously consuming the same streamed content, the nearest server can rapidly become overloaded. Some users may consequently suffer throughput degradation or content unavailability, and may experience a poor or unfairly shared QoE as they compete for limited network and server resources. The latter issue is common in many situations today. Several examples can be found. For instance, YouTube suffered from a massive overload during the 2018 soccer world cup semifinal.[1] More recently, Altice, the French media network company owning *RMC Sport* TV channel, had to publicly apologize for

[1] *YouTube TV goes down during World Cup's England v. Croatia semifinal* – `https://www.cnet.com/news/youtube-tv-goes-down-during-world-cup-semifinal/`

content availability issues. Indeed, many clients were unable to access and watch the soccer competition and were demanding refunds.[2]

P2P-assisted Content Delivery Networks

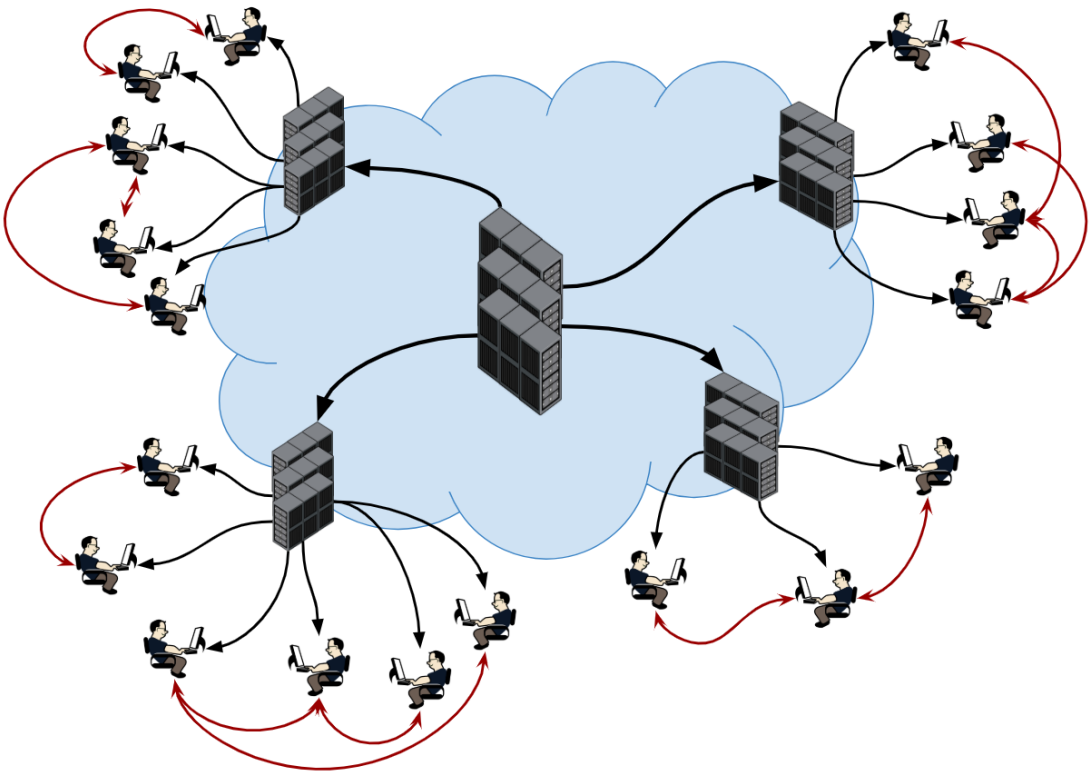

Figure 1.3: P2P-assisted CDN illustration

To solve this issue of CDN-side overload, the use of a P2P-assisted CDN (see Figure 1.3), or hybrid P2P/CDN, is an interesting alternative. It is particularly appealing for small players or platforms that do not wish to monetize their users' personal data to sustain their activity, such as the free and open PeerTube [PeerTube, 2020] network. More and more actors are investing resources to develop, improve and deploy hybrid P2P/CDN systems for larger purposes. P2P-assisted CDN complement core dedicated servers with the direct exchange of video content between end-users' devices. Examples of such platforms and actors using or providing an edge-assisted CDN are LiveSky [Yin

[2] *Communiqué de presse. Diffusion des premiers matchs de la ligue des champions via l'application RMC Sport* – `https://twitter.com/AlticeFrance/status/1042174185378918400/photo/1/`

et al., 2009a], Peer5 [Peer5, 2020], PeerTube [PeerTube, 2020], Streamroot [Streamroot, 2020], Hive Streaming [Hive Streaming, 2020] and Kankan [Zhang et al., 2014].

While hybrid P2P/CDN systems can solve current network issues, they are more complicated than simple adaptive players only relying on CDNs. The current state-of-the-art systems do not embed as efficient quality selection mechanisms as client-server ones and could be improved. At the same time and despite today's solutions still having room for improvements, new network architectures are already emerging with the early deployments of 5G infrastructure.

1.1.2 Future networks: reliability and high quality video streaming in 5G indoor networks

The 5G is the fifth generation of cellular network promising to drastically increase the bandwidth and reduce the latency of Internet connection. Deployment of 5G antennas have begun around the World using efficient digital modulation methods, customizable network functions, and multiple network paths. Considering the high capacities announced by the Internet Service Providers, the cellular network of the future is expected to extend its market not only to mobile users but to both desktop computers and smart objects. However, those devices are often found indoor and may require intra-building specific solutions to connect with 5G networks.

Wireless communications in buildings suffer from congestion and interference whilst modern building materials are often restricting the propagation of Radio Frequency (RF) waves within them. Building owners have been increasingly deploying private cellular home networks operating in licensed spectra in their premises to limit cross-talk and congestion. Unfortunately, these deployments require the permission of Mobile Network Operators (MNO) as they tend to interfere with the main mobile networks transmitted signals. However, MNO have only been able to analyse their largest customers' deployment requests, thereby losing a large market opportunity. To complicate matters further, each building requires a specific deployment for each MNO coverage, which is very costly and inconvenient for the building owner.

To solve the above mentioned issues, a European project called Internet of Radio Light (IoRL) proposes a broadband radio-light communications solution operating in unlicensed millimeter waves and visible light spectra. This system does not suffer from interference because of the specific Electro-Magnetic (EM) waves propagation characteristics in this spectrum, and provides universal broadband coverage within buildings from pervasively located radio-light access points within buildings' light roses. This technology

can also be applied to other indoor environments such as Tube Stations or Underground Pedestrian tunnels. The challenge addressed by the IoRL project is to design Remote Radio Light Head (RRLH) electronics that can be elegantly integrated into the myriad of different types of electric LED lighting systems. The primary benefit is enabling 10Gbps+ broadband communications services throughout buildings. The second one is geolocation, as user equipment can be located with a 10cm accuracy. Designing the radio-light communication system to fit into a light rose confined space requires (i) Network Function Virtualisation (NFV) solutions, for which cloud servers can be remote from the radio-light access points (elsewhere in the Home Cell Site or in the external Cloud network), and (ii) a Software Defined Network (SDN) to dynamically manage and route data to different parts of the radio-light network.

The project intents to achieve the same goals as future 5G networks in terms of video delivery: being able to reliably send UHD live and on-demand videos with a good Quality of Experience to the end user. Various video delivery scenarios are identified, from classic 4K Video-on-Demand streaming to demanding low latency live streaming, but also immersive media consumption with a virtual reality headset. Because the IoRL network architecture is highly heterogeneous, the reliability of streaming sessions can be tough to insure using state-of-the-art single-path video delivery systems; video playback can be stalled in case of direct line-of-sight loss (e.g., when a specific light providing wireless connection is temporarily occluded).

1.2 Contributions

1.2.1 Thesis contributions

This thesis aims at putting ideas forward to solve two of the aforementioned issues for current and future networks, in terms of video delivery. First, current networks often lack easy and cost-effective availability and scalability of video content distribution. Second, future networks will often rely on multiple simultaneous paths, which pose reliability issues for streaming sessions. In both contributions, we also seek to improve the global Quality of Experience delivered to the end-users.

To this end, the two main contributions of the thesis are:

- **PMS+: a P2P streaming system for current networks** Towards quality and scalability issues in CDN, we propose a combination of a multi-source approach along with the P2P paradigm. Our contribution is called PMS+, a hybrid

P2P/CDN solution for live streaming enhanced with scalability and quality adaptation capabilities. PMS+ creates and manages small groups of peers downloading the same video quality. The peer clients organize themselves to retrieve as much data as possible from their neighboring peers. PMS+ also comes with innovative quality and peer selection algorithms. The system is implemented and deployed in a production environment to perform a very large scale evaluation with real end-users. The experiment and the overall evaluation are performed in partnership with a leading company of live streaming webcams for tourism.

- **MSS/RRLH: a multi-path streaming system for future networks.** Considering the plurality and heterogeneity of future intra-building 5G networks, we propose a system to reliably deliver high quality and low delay videos. It relies on a novel architecture combining edge computing resources and efficient multiple-path quality selection algorithms. This solution incorporates specific mechanisms for VoD and live streaming sessions such as path selection, buffer-based adaptation and low latency reliability estimation. MSS/RRLH is implemented, deployed and evaluated in the official demonstrator of the Internet of Radio Light (IoRL) European project.

The contributions presented in this document are based on important preliminary works on multiple-source streaming performed at the beginning of the thesis with other collaborators. We first introduced an HAS-evolving streaming framework, MS-Stream, advocating for a client-centric utilization of multiple servers concurrently. The MS-Stream client simultaneously requests several servers to deliver independently decodable video subsegments. The throughput available across all servers is aggregated to improve the global video quality. MS-Stream was also extended with `MUSLIN` as an answer to CDN-side network overload via multiple-source capabilities. `MUSLIN` proposes to retrieve feedback from video players to better assign content servers to users.

1.2.2 Thesis organization

The rest of the document is organized as follows:

- *Chapter 2 - Background: Evolution of video streaming* provides background about video streaming solutions, from old protocols to over-the-top video delivery and adaptive streaming.

- *Chapter 3 - Preliminary works: MS-Stream and* `MUSLIN` describes preliminary works on multiple-source streaming and servers allocation that have been performed at the beginning of the thesis in collaboration with other researchers and PhD students. These works are important keystones for the two main contributions presented in this document.
- *Chapter 4 - State of the Art: P2P and 5G video delivery* presents related research work and the State of the Art of the two domains of study. The first part is about research work performed in hybrid P2P/CDN delivery. The second part deals with research in 5G video delivery, and introduces the H2020 project Internet of Radio Light (IoRL).
- *Chapter 5 - PMS+: a pragmatic collaborative multi-source P2P streaming system to improve QoE and scalability* presents the design, development and large scale evaluation of PMS+, a hybrid P2P/CDN multi-souce streaming solution.
- *Chapter 6 - Multiple-Source streaming over Remote Radio Light Head: a pragmatic video streaming system for future light indoor 5G networks* details the creation, implementation, integration and evaluation of a multiple-path streaming system for 5G indoor networks, demonstrated through the 5G European project Internet of Radio Light (IoRL).
- *Chapter 7 - Conclusions and Perspectives* concludes the thesis, summarizes the contributions and presents insight on possible future research directions.

Chapter summary

There are two main challenges to tackle:

1. To provide better QoE and scaling capabilities in today's networks.
2. To put forward video streaming solutions to improve QoE and latency in future 5G networks.

This thesis aims at proposing practical and efficient video streaming systems, providing both a high Quality of Experience and strong reliability guarantees to its users at the lowest cost. The solutions are validated in large scale experiments, in an actual production environment and in extensive lab experiments within an official European project demonstrator.

Chapter 2

Background: Evolution of video streaming

"Why did you ask what when the delicious question is when ?"

Robert and Rosalind Lutece

In order to fully understand the contributions of the thesis, let us take a look at the evolution of video streaming from the last decades. In a first section, the first video streaming solutions are presented. The second section deals with HTTP Adaptive Streaming and quality adaptation mechanisms. The third section describes Content Delivery Networks. Finally, the last section discusses the current state of the art efforts to create multiple-source video streaming.

2.1 Definition of digital video

A digital video is a sequence of pictures. These pictures are then consecutively displayed to trick the human eye into seeing motion, which requires at least 24 frames per second. Nowadays, most movies are recorded at a rate of 60 frames per second, and playback can reach several hundreds of images per second for specific use-cases such as video gaming. Consequently, videos require a lot of data to be stored, delivered and displayed. This encourages the use of compression algorithms, often referred to as *codecs*. Most codecs techniques consist in reducing redundancy in videos by computing movement vectors between consecutive frames and using them instead of a whole new picture. They mostly use three types of frames (depicted in Figure 2.1): Key (I) frames, which are standalone full pictures; predicted (P) and bidirectional (B) frames, made of movement vectors

depending on previous and next frames [Varma, 2015]. A Group of Pictures (GoP) is the self-contained series of an I frame followed by several B and P frames. As the compression ratio depends on actual movement in the footage, the GoPs have different sizes. These fluctuations lead to Variable Bit Rate (VBR) videos, used by the H.264 standard [Schwarz et al., 2007], among others.

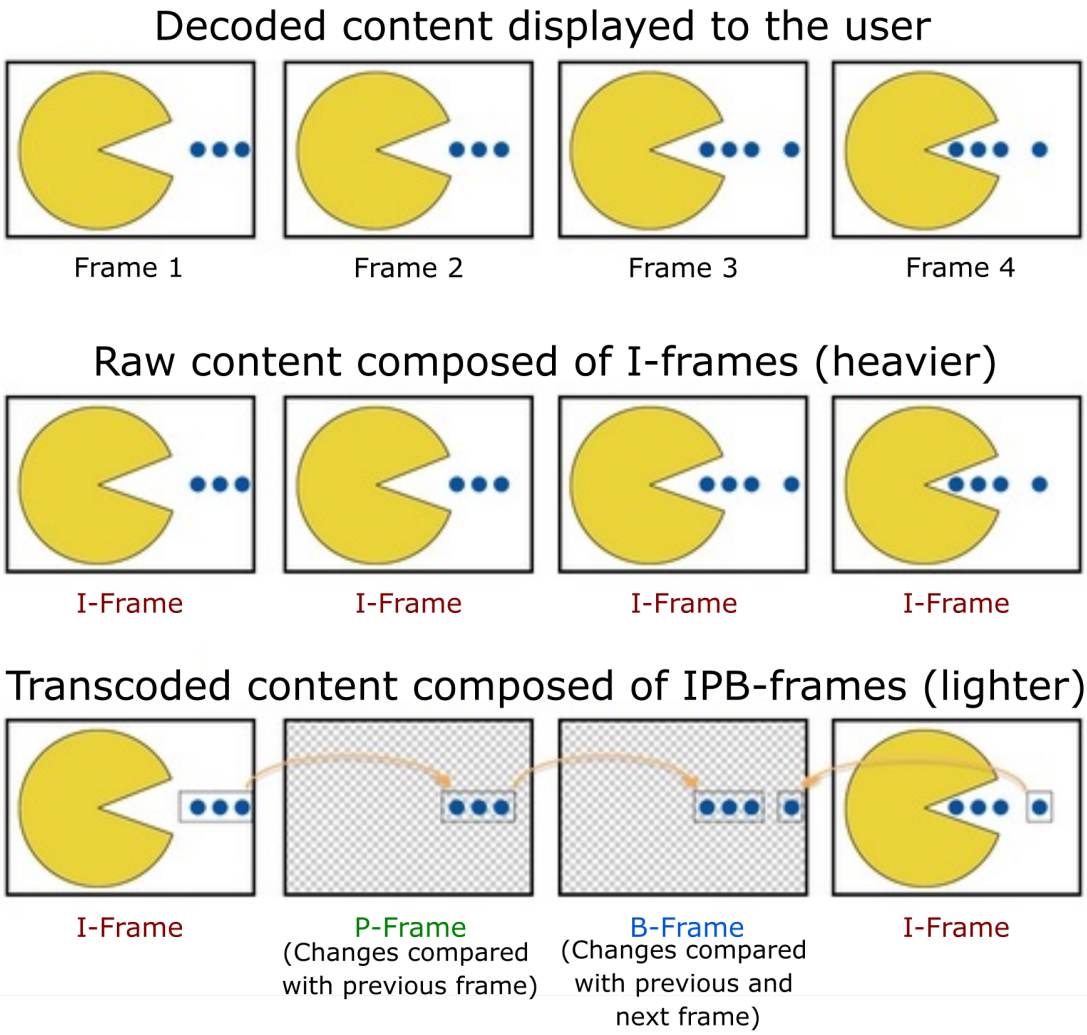

Figure 2.1: IPB frames in a transcoded video

The continuous delivery of a video is called "video streaming". Instead of downloading the complete video file before watching it, video streaming allows to watch the media content at the same time data are being received. This allows to consume video instantly

as well as live streams. Usually, video streams can be read GoP by GoP or frame by frame. Current challenge is to deliver video data in high-quality and without interruptions.

Within this chapter, the main video streaming solutions from 1990 to 2020 will be presented and analyzed in order to have a clear look on how video streaming techniques have evolved to reach high video quality delivery. Figure 2.2) summarizes them. Throughout this thesis manuscript, the term "bitrate" refers to the bitrate of the video. The terms "bandwidth" and "throughput" both relate to the capacity of the network or of a device (server or client) to transmit/receive data.

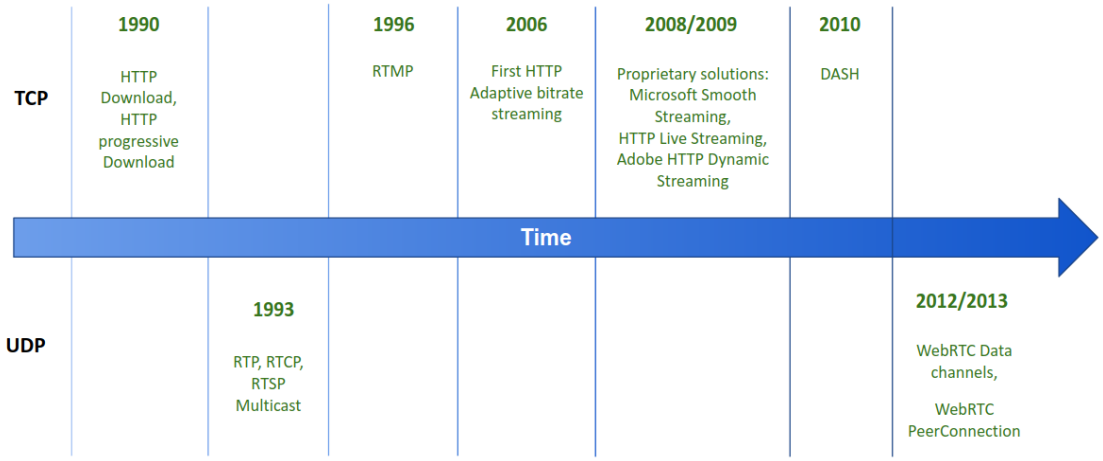

Figure 2.2: Evolution of video streaming protocols

2.2 Video streaming before HTTP Adaptive Streaming (HAS)

Before current HTTP Adaptive Streaming (HAS) techniques, several solutions had been envisaged, that can be splitted in the following three categories.

Traditional streaming. The Internet was not originally designed for the sustained delivery of modern bandwidth-intensive applications such as high-quality multimedia streaming. The fundamental difference between regular Internet traffic and video traffic is the real-time constraints of video traffic. Unlike email which can be delivered with a delay, video consumers usually expect to watch their videos as soon as they hit the play button and without interruptions. Most of the early work on packet video transmission focused on providing real-time transmission with techniques that support resource reservations and Quality of Service (QoS), such as Resource ReSerVation Pro-

tocol (RSVP) [RSVP, 1997] and Integrated Services (IntServ) [Diffserv, 2000]. Other protocols such as Real-time Transport Protocol (RTP) [RTP, 2003], Real Time Streaming Protocol (RTSP) [RTSP, 1998], Session Description Protocol (SDP) [SDP, 1998], Real Time Control Protocol(RTCP) [RTCP, 2003] were developed over the years in order to support real-time streaming over UDP and to control the server-side part of the system with functions to pause the stream, seek to another point of the playback or increase the playback speed. However, these techniques have encountered issues in traversing NATs and firewalls. They also require dedicated services and network infrastructures, thus increasing deployment and operating costs. On the other hand, TCP is a reliable protocol which guarantees the delivery of data. However, this reliability comes at the expense of a variable delay as senders wait for acknowledgments before continuing sending packets and re-transmitting lost ones. Since video is often delay intolerant and does not need high reliability to be acceptable, TCP was initially assumed unsuitable for multimedia delivery [Varma, 2015].

Unicast and Multicast Streaming. In the early 1990s, only a few users were able to enjoy video streaming on the Internet, mainly because of the low bandwidth network connections available at that time. Another reason was because a single streaming server was in charge of delivering all video requests, and unicast connections were established between the clients and this server. As the population of end-users consuming video streaming services drastically increased, the limited scalability of the unicast approach was rapidly reached. Subsequently, multicast protocols emerged and were more scalable under large number of clients consuming the same video content. IP multicast [Deering and Cheriton, 1990] [Sahasrabuddhe and Mukherjee, 2000] [Diot et al., 2000] is an extension of the IP protocol, with the objective of providing efficient multipoint packet delivery. Given that the network topology is best known within the network layer, multicast routing associated to this layer is also the most efficient. A common use of IP multicast is for Internet Protocol television (IPTV) applications. Television use case fits well with the multicast paradigm, especially a few years ago when the clients were all watching a limited number of channels. However, although IPTV uses the IP protocol, it is not limited to television delivered over the Internet. IPTV is widely deployed in subscriber-based networks with high-speed access channels at end-user premises via set-top boxes or other customer-premises equipment. IPTV is also used for the delivery of content in corporate and private networks. One major challenge for video multicast is the heterogeneity of end-user devices. With multicast, it is difficult to find a suitable video bitrate fitting different hardware capabilities and network resources of multiple clients

simultaneously. Although multicast seems an attractive solution for the delivery of video content, it was not largely adopted by the streaming actors [Quinn and Almeroth, 2001]. As a matter of fact, forwarding multicast traffic imposes a great deal of protocol complexity on network service providers. Moreover, core network infrastructures are particularly vulnerable to denial-of-service attacks along with IP multicast.

Traditional Adaptive Streaming. The Internet is the interconnection of multiple networks with best-effort traffic, therefore there is no guaranteed bandwidth for a real-time delivery of video packets. If the network bandwidth is not sufficient to support the video bitrate, then the video decoder at the client side consumes the video at a greater speed than the delivery rate of the data. Thus, the streaming client eventually runs out of video data to decode, which in turn results in screen freezes (video stalls or rebuffering events), which is known to impact the worst effect on viewing experience. In order to avoid such events without having to introduce costly and complex bandwidth reservation mechanisms, adaptive streaming solutions have been used to try to match the video bitrate to the available network bandwidth by [Varma, 2015]:

- Using a playout buffer embedded at the client side to pre-fetch data and store it locally in order to absorb the short-term variations of network throughput;
- Enabling on-the-fly video transcoding at server-side or in the network in order to adjust the bitrate (or resolution, frame rate, compression ratio) of the requested video to match the network capacities. This solution has a very high server-side processing footprint and requires complex hardware support.
- Establishing layered video coding (for example in [Schwarz et al., 2007]) to allow the encoding of a video into multiple dependent layers: a unique base layer (representing the least quality level) and several enhancement layers that improve the viewing quality. Hence, the encoded video can be adapted on-the-fly by adding or removing layers to the delivered content. However, such solutions require specialized servers and encoding schemes.
- Allowing stream switching adaptation permitting the offline encoding of the original video content into multiple different bitrates, resulting in multiple versions of the same content being available. A client-side adaptive algorithm is then used to select the most appropriate video bitrate according to the varying network conditions during transmission. Such solution does not require specialized servers, use

low processing power and provide high scalability due to the client-centric adaptation logic. However, more storage and finer granularity of encoded bitrates are required to enable the client to optimize the quality adaptation process.

Considering simplicity of implementation and deployment, the playout buffers and client-centric stream switching solutions were widely adopted by the industry.

Multicast streaming solutions also exploited adaptive bitrate techniques [Cable Television Laboratories, 2016] through three ways: single stream approaches, replicated stream approaches, and layered stream approaches. In the single stream approach, a single video stream is transmitted to the multicast group and feedback is received from all clients participating in the group. In the replicated stream approach, the same video is replicated in multiple streams (each with different bitrates) and the client can join a stream that fits its capability. In the layered stream approach, the server sends the video stream in multiple layers and each client can then subscribe to a subset of layers that fits its hardware capabilities and available network throughput.

2.3 HTTP Adaptive Streaming (HAS)

In the beginning of the 2000s, investigations started to see if TCP could be a possible candidate for delay-tolerant video transmission. An application layer playout buffer was hence introduced to absorb the rate fluctuations of TCP. The first implementations of video streaming over HTTP/TCP are called HTTP progressive download. The latter has been made possible because of new video container standards putting the metadata at the beginning of the file. In this scheme, the client simply downloads the entire video file with a constant video quality and starts the video playback as soon as enough video data have been received. One major drawback of this technique is that all end-users download the same video quality regardless of the heterogeneous network connections and capabilities of the end-users' devices. This can rapidly cause unwanted interruptions in the video playout if the clients' network connections do not reach the video bitrate.

The mid-2000s witnessed the rise of many proprietary HTTP Adaptive Streaming (HAS) solutions. Typically, HAS solutions rely on client-centric stream switching associated with an embedded buffer at the client side, using the HTTP/TCP protocol for content delivery. Hence, the client is able to request different video qualities to match the requested bitrate to the varying network conditions. The most notable differences between HAS and traditional streaming protocols lie in the fact that HAS is built on top of TCP instead of UDP, and that HAS clients request and receive video data in terms of

video segments containing few seconds of video playback instead of continuous streams of video packets.

Although some above-presented video streaming technologies are still in use for specific use cases -like retrieving data from a camera or uploading a live stream from a computer-, the video streaming industry has now adopted HAS protocols as their main techniques for streaming their videos over the Internet. As a matter of fact, using HTTP on top of TCP has several advantages:

- Clients use the standard HTTP protocol, which provides ubiquitous access of streaming video services on the Internet (through proxies, NAT and firewalls) [Popa et al., 2010].

- HAS servers can be commodity web servers, thus significantly reducing the operating costs and allowing the deployment of caches to improve performances and to reduce the network load.

- A client requests each video segment independently and maintains the playback session state, hence servers are stateless. Maintaining session state at the client side means clients can retrieve video segments from multiple servers, hence providing the means to balance the load of requests among servers [Liu et al., 2012a] [Liu et al., 2012b].

- TCP reliability and inter-flow friendliness improve the likelihood that streaming traffic consumes a fair fraction of the network bandwidth when competing with other non-video traffic.

These advantages enable streaming service providers to leverage existing and significantly cheaper HTTP infrastructures because every HAS system leverages on existing HTTP servers and proxy caches.

In the literature, many surveys reviewed the framework of most HTTP Adaptive Streaming solutions and more specifically, the Dynamic Adaptive Streaming over HTTP standard -i.e., DASH, also know as MPEG-DASH [Sodagar, 2011] [Bentaleb et al., 2019a] [Kua et al., 2017] [Sodagar, 2011], [Diallo et al., 2013] [Sani et al., 2017] [Garcia et al., 2014].

2.3.1 Dynamic Adaptive Streaming over HTTP

Dynamic Adaptive Streaming over HTTP (DASH, or MPEG-DASH) is the first HAS international standard. In a DASH-based solution, a video content is encoded into multiple versions -called representations- at different video bitrates. Each encoded video is then chunked into small video units called segments, each containing a few seconds of video playback. Segments from one bitrate are aligned in the video timeline to the segments from other bitrates so that the client can smoothly switch bitrates, if necessary, at the segment boundary. In practice, segments are independently decodable and embed one or several group of pictures. The DASH standard does not impose the way the content is delivered to the client. Many architectures exist for DASH-based solutions (client-centric, server-centric or network-assisted). Figure 2.3 represents a simple and common client-centric DASH-based client/server architecture.

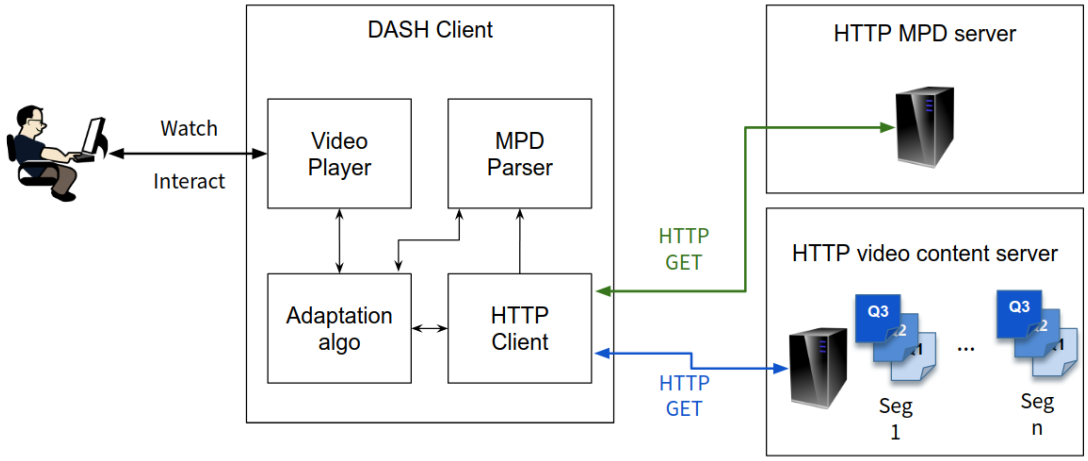

Figure 2.3: DASH

Content information such as video profiles, bitrates, resolutions, codecs, metadata, mimeType, server IP addresses, and segment URLs are described in the associated XML Media Presentation Description (MPD) files downloaded prior to the streaming session. The MPD describes a piece of video content within a specific duration as a period. In a period, there are multiple types of content available for adaptation such as video, audio and subtitles. They are referred to as adaptation set. In an adaptation set, there are multiple versions of the content, each known as a representation, each containing multiple segments (video segments for the case of a video adaptation set). Representations can represent multiple bitrate and resolutions available, allowing for adaptive consump-

tion of the content, but can also represent different languages for audio and subtitle tracks. In the second situation, the delivery is not adaptive and the player usually comes with a specific option for the end-user to select the appropriate representation. Figure 2.4 illustrates the structure of the MPD file. URLs pointing to the video segments in a MPD can either be explicitly described or be constructed via a template (client deriving a valid URL for each segment at a given quality). For example, an URL may include for every segment the resolution, the bitrate and the segment number in the form **$BASE_URL$/$RESOLUTION$/$BITRATE$/segment_$NUMBER$.m4s**. The format of MPEG-DASH video segment is derived from the MPEG ISO Base Media File Format (ISOBMFF) [ISOBMFF, 2015] container and MPEG-2 Transport Stream (MPEG-TS) [ISO MPEG-TS, 2019].

In each representation, there is a single initialization segment containing meta data, and many video segments. Concatenating the initialization segment with regular video segments results in a continuous video stream. Video segments are served to clients by using the HTTP protocol.

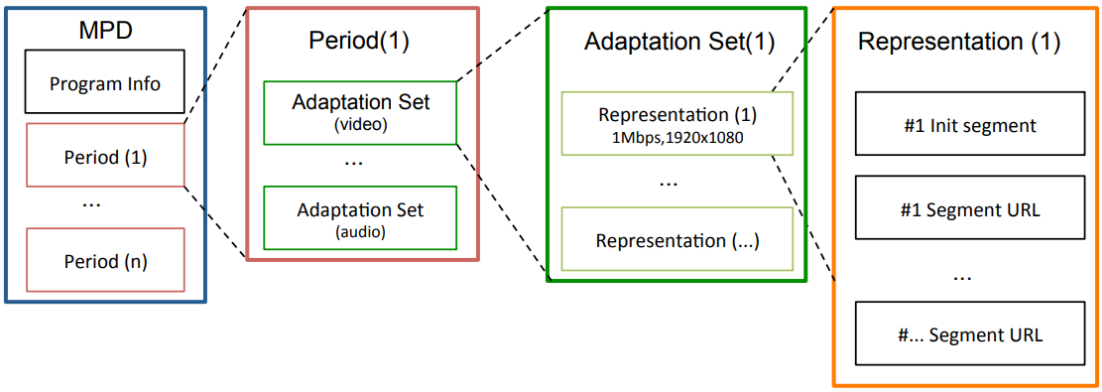

Figure 2.4: High level overview of the structure of a DASH Media Presentation Description (MPD)

Unlike traditional streaming strategies, the DASH standard does not enforce any specific implementations, adaptation mechanisms or segment scheduling policies. In its most basic form (and also the most widely implemented form), when a DASH streaming session starts, the client obtains and parses the MPD file associated with the requested content and starts requesting video segments as fast as possible to fill the playout buffer. Then, the player enters a steady state where it periodically downloads new segments according to the implemented adaptation logic. It does not usually try to get every single segment in order to save memory. Instead, once in the steady state, the player

alterns between a ON and a OFF state [Akhshabi et al., 2012a]. During a ON state, the player downloads a new segment. During the OFF state, the player does not and waits for its buffer to be below a threshold. The client typically keeps a few segments in the buffer as a trade-off to maintain video playback without using too much memory. In order to select a suitable video bitrate for the next segment to be downloaded, the video player uses various feedback signals observed for each segment. In a typical scenario, the achieved network throughput is used as basic criteria for bitrate selection decisions. For example, if the available network throughput gets higher, the DASH client selects a higher video bitrate to provide better QoE to the end-user. On the other hand, if the throughput drops, the client dynamically switches to a lower video bitrate in order to avoid buffer starvation and video freezing event that would cause major degradation of the end-user's QoE. An example of such a HAS streaming session is depicted in Figure 2.5. A "good" adaptation and consumption algorithm is expected to smoothly adapt the video bitrate to provide better QoE [Tian and Liu, 2012].

Proprietary commercial systems such as Microsoft's Smooth Streaming [Silverlight, 2017], Adobe's HTTP Dynamic Streaming (HDS) [Adobe HDS, 2010] or Apple's HTTP Live Streaming (HLS) [HLS, 2017] are following the same principles. The clients download a manifest and then video segments. The specifications does not include specific information about how the quality adaptation should be performed. Moreover, every solution is compatible with stream encryption. However, a few differences can be pointed out.

Manifests. Every streaming session begins with the download of a manifest. In Microsoft's Smooth Streaming, the manifest is a specific XML file. In HDS, the manifest is called the Adobe Media Manifest, or F4M. In HLS, manifests are called M3U8 and are specific text files. The master M3U8 file contains global information about the media and a list of M3U8 playlist per quality. M3U8 playlists includes the name and duration of every segment for one quality.

Video container and video codec. Some proprietary systems are limited by specific video containers. For example, segments in HDS should be in put into F4V containers. Until recently, HLS clients were consuming MPEG-TS segments. Nowadays, HLS can be used with fragmented MP4. Moreover, unlike DASH which is codec-agnostique and only limited by the external decoders available, proprietary systems may be codec-specific for

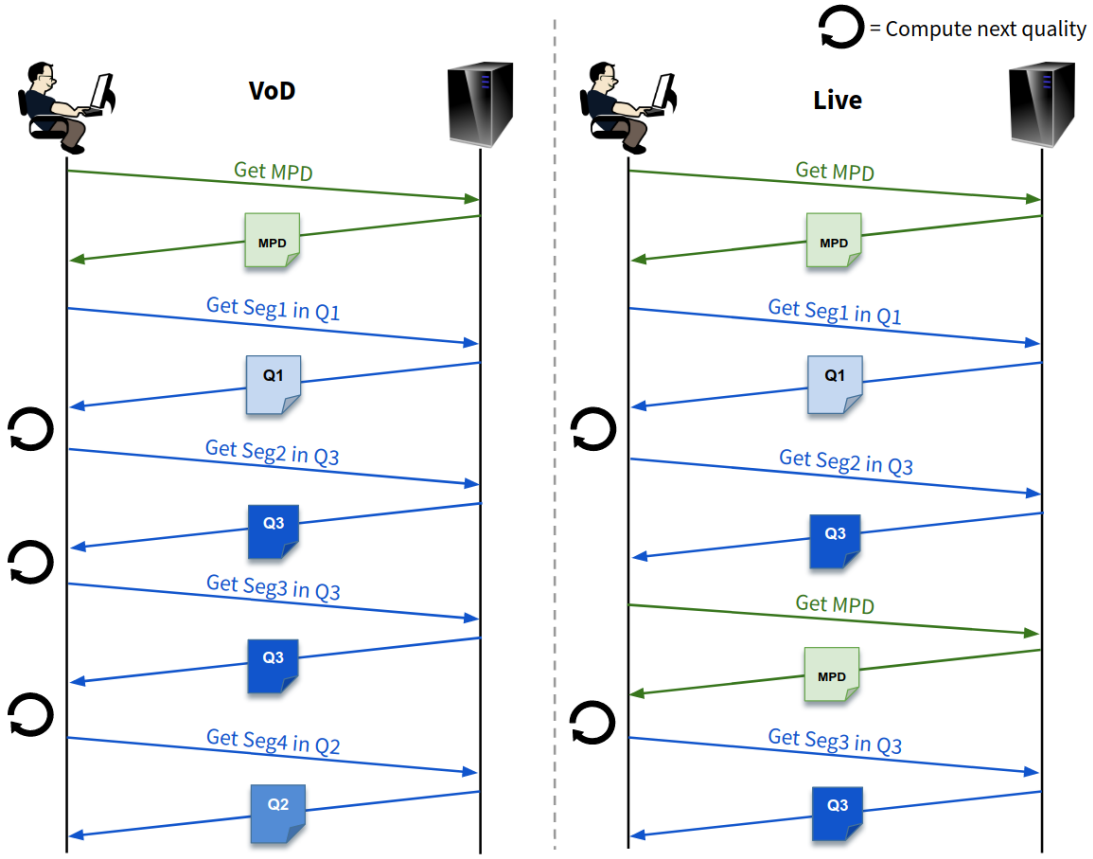

Figure 2.5: Example of client-centric HTTP Adaptive Streaming session for VoD and Live content

both video and audio content. As an example, the three HAS systems discussed in this section are not compatible with the open-source codecs VP8 and VP9.

Proprietary implementations. The official implementations from the technology owners may be the only solution to watch content if the standard is not compliant enough for open source or concurrent projects to emerge. In the early years, both HDS and Microsoft's Smooth Streaming were limited by the proprietary solution. Nowadays, the implementation of HLS in iOS can not be modified and it is not possible for a developer to use his own implementation. Because of this limitation, proprietary HAS solutions may be constrained by the non-optimal adaptive mechanisms carried by the official implementations.

Standardization. Move Networks was the first industrial actor on HAS and patented their technology [Patent US8868772, 2014] on adaptive streaming at the United States Patent and Trademark Office (USPTO) in 2010. The patent covers the structure of video content and the intelligent requests sent by clients to adapt video bitrate over IP networks. DASH is based on the work in 3GPP Release 9 [3GP-DASH, 2012] and Open IPTV Forum Release 2 [OIPF, 2014]. MPEG-DASH was first a Draft International Standard in January 2011, and became an ISO/IEC (International Standardization Organization/International Electrotechnical Commission) standard for adaptive streaming later in this year. The MPEG-DASH standard was published in April 2012 [MPEG-DASH, 2012]. The standard defines guidelines for media presentation, segmentation, and a collection of standard XML formats for the manifest file (MPD). However, specific client implementation, consumption and adaptation algorithms are not part of the standard [Stockhammer, 2011]. This field is left for researchers and industrial actors to explore and define their own solutions for content quality and delivery adaptation. The standard is also codec agnostic to favor and support future improvements in the field of media coding. DASH has been adopted by other standardized multimedia streaming systems such as in the IPTV (Open IPTV [OIPF, 2014]) standard and Digital Video Broadcasting (DVB) [DVB-DASH, 2020] for video delivery over the Internet.

The DASH Industry Forum (DASH-IF) [DASH-IF - DASH Industry Forum, 2020] is a group of companies and researchers leading adoption and research initiatives in current adaptive streaming systems. DASH-IF provides specific implementation guidelines and regular documentation of interoperability [DASH-IF Guidelines, 2020]. The community has developed an open-source dash.js [DASH-IF Player, 2020] reference player, which employs the Media Source Extensions of web/HTML5-based browsers. DASH-IF also provides a comprehensive list of publicly available test datasets [DASH-IF Tests, 2020], network profiles [DASH-IF Cases and Vectors, 2020] and client software for test, content preparation and validation [DASH-IF Software, 2020].

Because the standard does not define a specific content adaptation algorithm, intensive research has been conducted to produce better quality selection mechanisms with the objective of improving the Quality of Experience (QoE).

2.3.2 QoE of HTTP Adaptive Video Streaming.

Unlike Quality of Service (QoS) which defines the quantitative performances of network connections and is not directly related with video streaming, QoE refers to the subjective quality perceived by end-users. In the case of non quality-adaptive streaming, the

essential criteria for QoE can be grouped into two categories: the initial start-up delay and the video stalling due to rebuffering [Hoßfeld et al., 2011]. When considering quality-adaptive streaming protocols that perform trade-offs between video quality and video rebuffering events, the introduction of video bitrate modifications during the video playback influences the end-users' perceived quality. Therefore, the QoE for video streaming should also include the average video bitrate displayed, the number of quality switches observed during the playback, and the amplitudes of those switches. Several surveys and studies can be found in the literature discussing the QoE of HAS [Seufert et al., 2015] [Oyman and Singh, 2012] [De Vriendt et al., 2014] [Essaili et al., 2013] [Yitong et al., 2013] [Hoßfeld et al., 2014]. The authors in [Hoßfeld et al., 2013], [Hossfeld et al., 2012], and [Seufert et al., 2015] all reported that the perceived end-users' dissatisfaction during the playback of video is highly correlated with the number of video stalls. The authors in [Hossfeld et al., 2012] explained that rebuffering plays a more important role in the end-user dissatisfaction than the initial delay. With the same idea, the authors in [Seufert et al., 2015] concluded that the number of video stalling events should be minimized in every situation, even if it means reducing the initial delays and the average displayed bitrate. Additional studies on QoE for adaptive streaming services [Yitong et al., 2013] [Mok et al., 2012] also pointed out that limiting the amplitude of bitrate variations and minimizing the gap between two consecutive video quality switches reduces the negative effects on the perceived video quality. Finally, the work [Ni et al., 2011] showed that the number of quality switches has a minor impact on the overall observed QoE level. In the contributions of this thesis, we rely on the previously-mentioned studies and mostly focus on the number of video stalling events as well as on the mean displayed bitrate in order to characterize the QoE of our systems.

2.3.3 Research works on content adaptation

This subsection deals with the most significant quality adaptation contributions proposed in the literature.

To better manage the complexity of client-centric adaptation in HAS solutions, the authors of [Jiang et al., 2012] propose a general framework exposing the three major functional components of HTTP adaptive streaming: (1) resources estimation, (2) quality adaptation, (3) segment request scheduling. The segment scheduling function takes as inputs the history of segment download completion times as well as streaming session related information such as the current buffer level, and is responsible for deciding how and when the next video segment is to be requested. Then, the adaptation module

decides on the video bitrate of the next segment to be downloaded based on inputs given by the resource estimation module.

Although for the case of client-centric adaptation, the latter three modules are embedded in the client, they are not necessarily co-located. Any of these modules can be on a separate system (at the server side or in the network). In this section, we focus on the client-centric HAS solutions that embed these three components.

Resource estimation. Resource availability directly affects the capabilities of HAS clients to provide smooth streaming and high QoE to the end-users. Therefore, it is crucial to understand how resources are measured and estimated, and how they affect the content adaptation mechanisms. Typically, a resource estimation function monitors the resource used for the adaptation functions of the HAS client. Because content providers target a large number of clients, and since most of the streaming session's parameters can be better observed from the client's point of view (e.g., last-mile bandwidth, buffer occupancy, etc.), resource estimation is usually implemented at the client-side, hence providing a more scalable solution than its server-side or in-network alternatives. However, solely relying on client-side observations can result in selfish behaviors [Huang et al., 2013]. This lead to hybrid proposals combining server-centric [Liu et al., 2012b] or in-network solutions [Cofano et al., 2014].

The choice of resources to be considered as an input for adaptation and scheduling function is also context dependent [Miller et al., 2012]. The first generation of quality adaptation scheme mostly relies on throughput estimation and always selects the highest video bitrate that fits the measured throughput [Akhshabi et al., 2011] and [Thang et al., 2012]. It was assumed that this strategy can avoid rebuffering while at the same time providing the highest possible video quality. Later, it became obvious that throughput estimation alone is not a sufficient parameter for efficient adaptation scheme [Jiang et al., 2014] [Huang et al., 2012]. To avoid video stalls, the video clients should receive and store video data in advance in a buffer. In HAS players for example, video segments are pre-fetched and stored into a local buffer. This ensures that the client will continue displaying video from the buffered video for at least the duration of the buffered content. Hence, there is an inverse relationship between buffer occupancy and the probability of video stalls (i.e., the bigger the size of a buffer, the longer it takes to run out of content). Therefore, the buffer occupancy was introduced as another parameter for the design of efficient client-side quality adaptation algorithms.

Reliability of throughput estimation. An open challenge in the adaptation schemes of HAS is the reliability of the throughput estimation. Any throughput measurement done at the application layer can only consider the throughput calculated by the underlying TCP protocol. However, the authors of [Jain and Dovrolis, 2004] argue that equating the available bandwidth with the TCP throughput is error-prone since TCP throughput depends on many factors (including socket buffer sizes at the sender and receiver, the nature of the competing traffic, RTT, packet loss rate, the nature of TCP congestion control etc.). Similarly, an argument against matching the TCP throughput observed at the application layer with the available bandwidth is presented in [Li et al., 2014]. The paper showed that when clients compete on the same bottleneck, the presence of competing applications and the ON-OFF nature of the HAS downloads make it difficult for a client to correctly perceive its share of the available bandwidth. This results in an under-utilization of the available bandwidth leading to video quality flickering, which is known to negatively impact the end-users QoE [Akhshabi et al., 2012a] [Seufert et al., 2015]. To tackle this problem, the paper proposes a "Probe ANd Adapt" (PANDA) technique. The algorithm somehow copies the congestion control of TCP at the application layer. The TCP throughput is then used as input when it represents an accurate indicator of the fair-share of bandwidth, which is argued to happen when the network is congested. Otherwise, the algorithm probes the network by incrementing the sending rate and stops when a congestion is detected.

Adaptation functions. The adaptation function is the element within the HAS framework that decides the representation of a segment to be requested in terms of video bitrates, resolutions, framerates, codecs, etc. Although HAS permits to adapt the content according to a large panel of criteria, research contributions principally focus on adjusting the video bitrate (which can then be turned into other parameters) as it is the fundamental parameter that should best match the available network resources if video freezing events are to be prevented. Most adaptation logics usually take as input information regarding the available resources and the set of all the possible content representations in order to return the quality of the next segment to be downloaded. In the literature, quality adaptation contributions can be divided into four main categories: (1) heuristic based adaptation, (2) control theory based adaptation, (3) optimization based adaptation, and (4) layered-coding based adaptation.

Heuristic based adaptation. Most of the early adaptation proposals are based on heuristics such as pragmatic throughput estimation and buffer occupancy. For instance,

the authors of [Mok et al., 2012] proposes a QoE-aware adaptation algorithm called QDASH (QoE-aware DASH). The client reduces the video bitrate in a step-wise manner when the achievable throughput drops. Although this may result in suboptimal choices, the QoE is improved by enhancing the stability of the delivered quality. The conducted experiments in [Akhshabi et al., 2012b] showed that the Microsoft Smooth Streaming HAS player is using a similar approach. Although the switch-up transitions are faster than the downward transitions, the quality switching is not immediately performed to the video quality that matches the network throughput.

Huang et al. ([Huang et al., 2014], [Huang et al., 2012], [Huang et al., 2013]) are among the first to contribute to content adaptation assisted with buffer-based adaptive strategies. Indeed, only the buffer state is used to determine the video bitrate of the next segment to be downloaded. Nevertheless, when the buffer level is too low and prevents decision making, throughput estimation is performed by probing the network. The proposed algorithm was experimentally evaluated with real end-users on the Netflix streaming platform. Results showed that this approach reduce the amount of rebufferring events by 10 to 20% compared to Netflix's algorithm, while delivering a similar average video bitrate.

Miller et al. [Miller et al., 2012] proposed an algorithm that uses three threshold levels for the playout buffer, such that $0 < B_{min} < B_{low} < B_{high}$. The target interval B_{target} is between B_{low} and B_{high}, and the optimum buffer level $B_{optimum}$ is at the middle of the target interval. The algorithm attempts to keep the buffer level close to $B_{optimum}$. It allows the designer to explicitly control the trade-off between the variations in buffer occupancy and the fluctuations in video bitrate by controlling the B_{low} and B_{high} thresholds. Based on experiments conducted in a WiFi environment with and without throughput limitation at the server side, the authors showed that the algorithm presents a stable and fair behavior when multiple clients compete on a common network path. Other players that employ heuristic based adaptation logics exist in AdapTech Streaming [Akhshabi et al., 2012b], and in the Akamai HD Video Streaming services [De Cicco and Mascolo, 2010].

Control theory based adaptation. There have been many attempts to design adaptive bitrate strategies based on predictive and descriptive models. Control theory is used to model dynamical systems that are stable, accurate and settle quickly into a steady state [Abdelzaher et al., 2008]. The controller manipulates the inputs of a system to produce the desired outputs. Typically, a controller computes the distance between a

measured variable and an output value as a process error. The goal is to reduce this error by adjusting the input parameters.

The authors of [De Cicco et al., 2011] propose an adaptation logic based on feedback control. The video rate adaptation controller takes a target buffer as an input and returns the video rate of the segment to be downloaded. The goal of the controller is to ensure that the buffer is always maintained at the target level. This is achieved by computing the error between the target buffer and the measured buffer level. The error is then passed to a proportional integral controller that outputs a video bitrate matching the estimated available throughput. Experiments confirm that the controller selects the highest video bitrate that the available bandwidth can sustain. In [Tian and Liu, 2012], a control theoretic client-side rate adaptation performs a trade-off between the stability of the video quality and bandwidth utilization. Other papers propose adaptation functions that are implemented using control theory [Yin et al., 2015], [Zhou et al., 2013a], [Zhou et al., 2013b], [Miller et al., 2012] and [Cofano et al., 2014].

Optimization based adaptation. [Qiu et al., 2010] tried a different approach by exploiting an optimization technique for bitrate adaptation called *Intelligent Bitrate Switching*. The authors modelled the adaptation logic as an optimization problem, which maximizes benefits -the quality level of each segment- while minimizing penalties. A maximum penalty is assigned to video stalls. The authors proposed an adaptable model where users can adjust the penalty score based on its viewing experience. An optimal solution is expected to select a segment with the highest video rate among all the segments that satisfy the given constraint of a minimum number of video interruptions.

According to Bouten et al. [Bouten et al., 2014] [Bouten et al., 2012], the support for coordinated management and global optimization is essential to improve QoE. The authors propose to control the allocated network resources among competing clients. They employ an integer linear programming (ILP) model to either maximize the QoE of all end-users or minimize the penalties incurred when resource allocation is not optimal.

The authors in [Spiteri et al., 2016] formulated video quality adaptation as a utility maximization problem and proposed an online control algorithm called BOLA, using the Lyapunov optimization functions to minimize rebuffering and maximize video quality. BOLA does not require any throughput estimation, and assumes that the buffer level is sufficient to provide all the information about past bandwidth variations. The authors evaluated BOLA on 12 test vectors defining network characteristics (bandwdith delay, packet loss), referred as network profiles, provided by DASH-IF [DASH-IF Cases and

Vectors, 2020] with 85 publicly available 3G mobile bandwidth traces. They compared the obtained results with an optimal offline algorithm that guarantees the maximum achievable time-average utility for any given network trace (having the prior knowledge of future bandwidth variations) and found that BOLA achieves between 84% and 95% of the optimal utility.

Layered coding content adaptation. In the literature, most HAS research work assume that every segment is self-contained and independently encoded. To some extent, this is a valid assumption since most video codecs, including the widely adopted H.264/AVC, H.265/HEVC, VP8 and VP9, propose the latter content format by default. However, for each representation, all segments have to be encoded and stored separately. This induces significant additional storage for a video streaming provider employing HAS-based techniques. In [Huysegems et al., 2012], the authors are able to show that the Microsoft Smooth Streaming services necessitates between 200% to 300% of storage overhead compared to having only the highest video representation available. The authors of [Sánchez de la Fuente et al., 2011] also confirm the suboptimal performance of self-contained segment based HAS in terms of caching efficiency and of additional bandwidth to transport the segments to the servers and caches in the network [Lin and Hwang, 2011].

The purpose of any quality adaptation logic is to enable clients to adjust the quality of the requested video to evolving conditions. In self-contained segment based HAS services, a segment must be completely delivered. A better solution argued by [Sánchez de la Fuente et al., 2011] is to use layered coding. Scalable Video Coding (SVC) [Schwarz et al., 2007] is an extension of the H.264/AVC standard for layered video coding. Layered coding allows the encoding of a video into a number of layers, composed of a unique base layer (representing the least quality level) and several enhancement layers, with each enhancement layer improving the viewing quality. With SVC, a video is encoded once into multiple layers, and is decoded based on frame rate, resolution, or fidelity requirements. In this way, a client can select the appropriate number of layers in order to adapt the content to the varying network conditions as well as the varying terminal capabilities [Schwarz et al., 2007].

A detailed discussion on SVC can be found in [Schwarz et al., 2007], [Schwarz and Wien, 2008], and in [Unanue et al., 2011]. Even with layered coding, a video file needs to be chopped into segments to suit HAS. Multiple segment creation strategies were proposed in the literature. In [Tappayuthpijarn et al., 2011], each segment is composed

of several blocks, each block representing a layer. The paper [Xiang et al., 2012] proposes a different approach: the encoded video is divided along the layers, and then split into segments. Therefore, each segment request refers to a specific layer. The authors of [Grafl et al., 2013] use multiple independent groups of segments, each one having the same class of base layer so that they represent a particular resolution. Layers within a segment are used for quality adaptation.

Because DASH is codec agnostic, SVC video segments can easily replace single-layer segments in traditional bitrate adaptation strategies. For instance, [Abboud et al., 2011] and [Famaey et al., 2013] use SVC with the open source version of the Microsoft Smooth Streaming adaptation algorithm. Quality adaptation is performed by selecting a base layer and the required enhancement layers matching the available resources. The authors of [Müller et al., 2012b] and [Sieber et al., 2013] successfully adapted SVC segments to the adaptation scheme proposed in the work of Muller et al. [Müller et al., 2012a], originally designed for single layered content.

Many researchers have investigated the performance of SVC in HAS [Sánchez de la Fuente et al., 2011]. In addition to better caching efficiency, and since SVC allows clients to abort segment downloads without much overhead, the use of SVC improves the responsiveness in HAS schemes to the variations of network conditions [Huysegems et al., 2012]. The authors of [Famaey et al., 2013] and [Basso et al., 2014] pointed out that due to the increased number of requests compared to single layered based HAS solutions, SVC based HAS proposals are more vulnerable to high RTTs. Indeed, when RTT augments, the achievable throughput decreases, and SVC-based HAS techniques are expected to perform badly in low throughput conditions. In addition, despite reducing the storage requirements of HAS, SVC-based solutions require at least 10% of encoding overhead [Schwarz et al., 2007] in terms of data storage, resulting in higher bandwidth requirements. The authors of [Kalva et al., 2012] found that the increased cost of bandwidth outweigh the reduced cost of storage. For all these reasons, SVC has still not succeeded in being adopted in the industry.

2.4 Content Delivery Networks

Content Delivery Networks (CDNs) have been considered as the main approach for video distribution over the Internet. CDN servers are geographically distributed replica, cache and edge servers positioned as close as possible to the consuming clients. When accessing a content, consuming clients are automatically redirected to one of the best

available servers based on proximity -or other parameters- to temper network congestion, reduce the network delay and achieve higher throughput. Figure 2.6 provides a high level understanding of CDNs.

With the ever-increasing amount of video traffic, the CDN approach has been facing several challenges related to managing and administrating the entire CDN infrastructure, such as the support for HTTP-based video delivery, the scalability problem [Balachandran et al., 2013], replica placement [Pathan and Buyya, 2008], content selection [Gao et al., 2015] [Jin et al., 2016] [Scellato et al., 2011], and content placement [Applegate et al., 2010].

CDNs have been deployed by companies like Limelight, Akamai, and Level 3, but recent years have seen the rise of CDN services hosted by big companies such as Google, Facebook, Amazon and Microsoft. To provide a cheap pay-as-you-go service to a broad variety of customers, some CDNs have adopted cloud technologies which became known in the literature as cloud CDNs [Limelight, 2020]. Furthermore, in order to gain a better control over the data services served to their end-users, many telecom operators (AT&T, Orange, Telefonica, Verizon, etc.) have deployed their private telco-CDNs [Frank et al., 2013]. Overall, Cisco has estimated that CDN traffic will handle more than two thirds of all Internet video traffic by 2022 [Cisco, 2018].

There have been a number of studies on CDN-based VoD systems (e.g. YouTube [Adhikari et al., 2010] [Adhikari et al., 2012e], Netflix [Adhikari et al., 2012c] and Hulu [Adhikari et al., 2012a]. Several reported that CDNs are being stressed by the demands during peak hours [Wendell and Freedman, 2011] [Liu et al., 2012b], pointing out the limitation on CDN scalability. As a matter of fact, CDNs does not solve server-side overload but only decentralizes it. Instead of overloading central servers, specific local servers are the one being stressed during peak hours.

2.5 Multiple path streaming

The classification of existing research contributions to content adaptation show two main categories [Diallo et al., 2013]: content adaptation and content delivery adaptation.

The authors in [Diallo et al., 2013] explain that content adaptation is the process of selecting, generating, or modifying content to suit the end-user's preferences, consumption style, computing and communication environments as well as usage context. Research on content adaptation has been discussed previously in section 2.3.3.

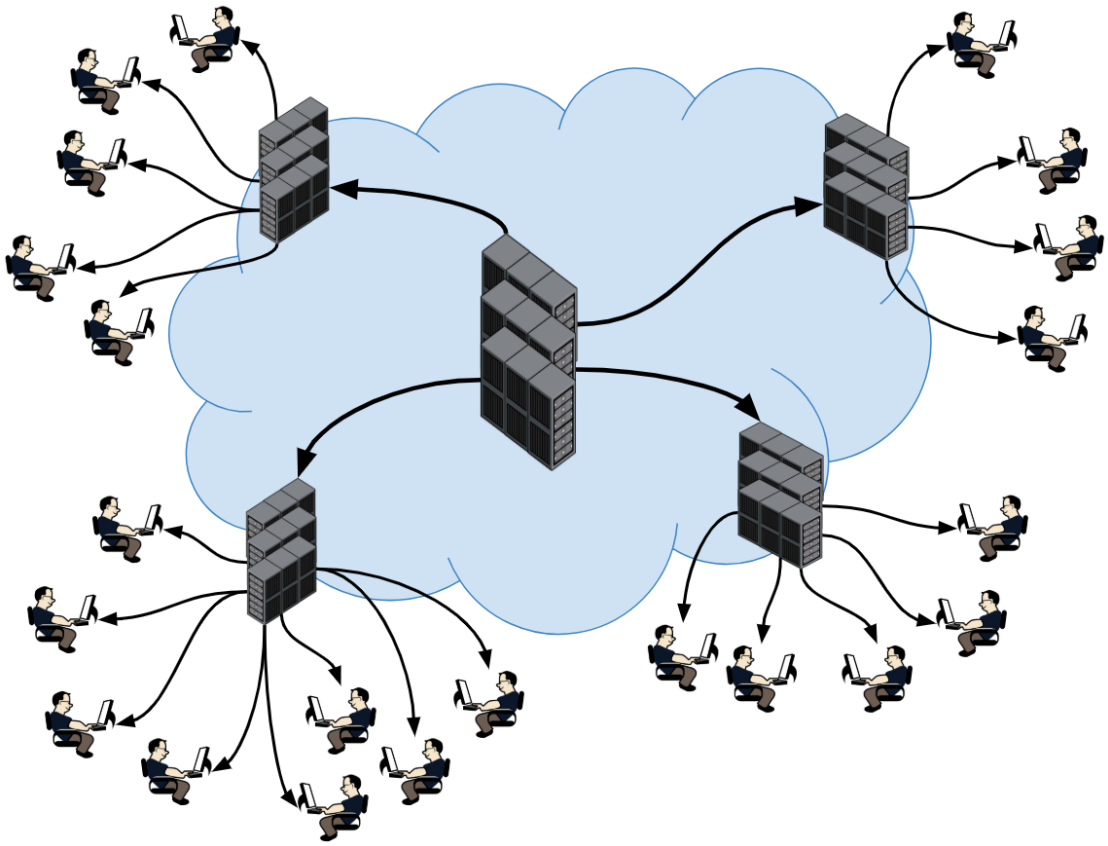

Figure 2.6: High level understanding of Content Delivery Networks

Differently from content adaptation, content delivery adaptation focuses on the service and network aspects of the content transmission only, instead of adapting the content quality. This directly relates to the choice of content delivery techniques as well as the flexibility offered to the clients and servers: unicast or multicast? From which server(s)? Possibility to handover the delivery to other servers? Through which access network? Possibility to use multiple network interfaces? Sequential or parallel segment downloads?. While most studies focus on content adaptation methods, fewer contributions address the adaptation of delivery means for HAS solutions by relying on range requests, queuing strategies, or multipath-TCP.

Reference [DASH-IF Cases and Vectors, 2014] introduces the DASH framework of Netflix, the largest DASH stream provider in the world and [Adhikari et al., 2012c] outlines that Netflix always binds one user to one server, regardless of the available throughput between the user and the server. The study performed in [Adhikari et al., 2012c] also indicates that QoE could greatly benefit from simultaneously using multiple

servers along with DASH-like protocols. All existing rate adaptation algorithms aim to either adapt the video bitrate to the available transport bandwidth so as to achieve high bandwidth utilization efficiency, or to ensure buffer completion level in order to provide continuous video playback. These adaptation methods have been designed for single-source server solutions.

A preliminary bitrate adaptation approach to be extended to multiple sources is proposed in [Tian and Liu, 2012]. Originally not suited for a multiple servers' purpose, the proposed bitrate adaptation decisions are asynchronously taken, imposing segments to be retrieved one after another, without considering completion time. Therefore, request completion is most of the time out of date due to channel heterogeneity in multiple-server environments. Reference [Zhang et al., 2015b] presents Presto, a streaming protocol designed to use several servers simultaneously in order to improve QoE by providing better fairness, efficiency and stability at the service provider's side. Nevertheless, the segments retrieved from the different servers are not independent from each other and not aggregatable, since they are divided into smaller non-decodable chunks spread over several servers.

In [Pu et al., 2011], an evolution of DASH is put forward using multiple servers assisted with Scalable Video Coding technique. Clients simultaneously request segment layers from several servers. However, dependency between layers makes segment scheduling a complex task and failing in retrieving the so-called "base layer" will block the consumer from watching the video.

DQ-Dash [Bentaleb et al., 2020b] proposes an interesting queuing system to request segments from multiple servers. In this system, the segments are retrieved one-by-one from different servers at the same time. If a download is too long, the request is aborted and is rescheduled with another source. Moreover, the server is blacklisted and no longer used for future reference during a period of time. As mentioned by the authors, this solution is limited to high buffer streaming sessions and the impact of the queuing system is reduced in low buffer sessions.

Another approach in multi-path streaming over the Internet is the use of multi-path TCP protocol (MPTCP). Unfortunately, various studies have shown that the performances of MPTCP are below the expectations, especially when the underlying paths have heterogeneous characteristics [Wu et al., 2016], [Singh et al., 2012]. MPTCP also requires implementations in both end-points but most of the clients' operating systems and most of the streaming providers' servers do not support it yet, as well as most of the current middleboxes. As an example, it is not possible to use MPTCP in the French mo-

bile networks. Overall, streaming providers are not keen to implement MPTCP in their servers, making the deployment of MPTCP restricted to a tiny fraction of the Internet.

The MPTCP packet scheduler is identified as the main weakness of the protocol. Packets that have to be transmitted are buffered into the MPTCP output queue. Schedulers select packets in the queue following a first-in first-out (fifo) process, and transfer them into the TCP sending buffer of one of the available links, according to the scheduler policy. However, packet losses occurring on one path can generate head-of-line blocking at the client side, potentially leading to streaming buffer dryness and rebuffering. Many papers suggested solutions to address this issue by focusing on: windows congestion restructuring [Kuhn et al., 2014], cross-layer scheduling [Wu et al., 2016], bandwidth and buffer management [Kurosaka and Bandai, 2015] and retransmission processes [Raiciu et al., 2012] at the transport layer. The majority of studies on MPTCP is not focused on HTTP adaptive streaming, leaving the adaptation logic to the application layer.

Regarding alternative video content multi-path delivery, the Multi-Path Real-time Transport Protocol (MPRTP) proposal [Singh et al., 2013] targets latency among others objectives, but it relies on RTP and UDP, which are not broadly used by today's content providers.

Chapter summary

From the first protocols to modern Over-The-Top solution, video streaming has been an important topic of research.
Recently, HAS protocols, such as MPEG-DASH and Apple's HLS, have seen extensive interest from the industry and the research community, mainly due to their capabilities to render smooth video playback. A lot of papers in this area focus on bitrate adaptation mechanisms and recent studies point out that multiple-source delivery could improve the quality of streaming systems.

Chapter 3

Preliminary works: MS-Stream and MUSLIN

"Because it does. Because it has. Because it will."

Elizabeth Comstock

The works presented in this section have been made at the beginning of the thesis, in strong partnership with two PhD students and other researchers. They are important keystones for the two main contributions presented in this thesis.

3.1 MS-STREAM

MS-Stream [Bruneau-Queyreix, 2017, Bruneau-Queyreix et al., 2017b, Bruneau-Queyreix et al., 2017a, Bruneau-Queyreix et al., 2018, Bruneau-Queyreix et al., 2016] is an evolution of HAS solutions (and more specifically, the Dynamic Adaptive Streaming over HTTP – DASH – standard) that simultaneously takes advantage of several servers for the download of one video segment (see Figure 3.1). A resource provider periodically advertises clients with potential usable servers. MS-Stream clients request several sub-streams (technically known as *descriptions*) from all -or a subset- of those servers. These descriptions are generated from the existing set of DASH content representations to handle network-path heterogeneity, mostly to fit fluctuating bandwidth. A very low CPU footprint overhead is necessary to create descriptions. When retrieved, the descriptions are merged to assemble a full segment and to display the requested content quality. In the event of description loss or out-dated delivery, content playback continuity is not affected, only content quality is. Additionally, if the considered servers or paths are un-

reliable, content-based adaptation and server switching avoid QoE degradation. Thanks to its codec agnosticism and DASH-compliance, this proposal represents an evolving solution that can be applied to many scenarios, such as P2P, CDNs, Clouds, Set-Top-Box overlays, as well as collaboration of resource providers to achieve higher QoE and create new businesses.

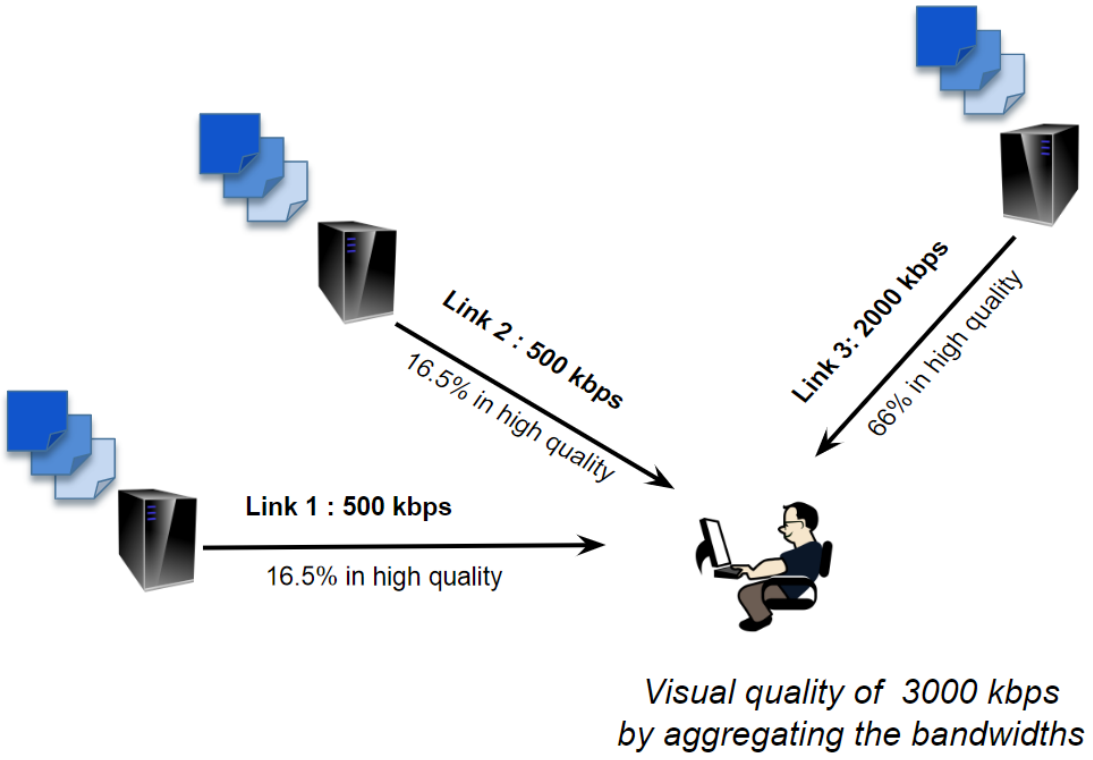

Figure 3.1: MS-Stream: Aggregating the bandwidth from three different servers

The MS-Stream overall client/server architecture is depicted in Figure 3.2 (additional modules — compared to DASH — are highlighted in blue). Before the streaming session, a manifest file provides the client with information about available MS-Stream servers and the number of Group of Pictures per video segment.

MS-Stream content delivery steps include: (1) the client instructs MS-Stream servers to generate and deliver substreams through the MS-Stream HTTP API; (2) the Description Creator generates the requested sub-stream from the existing set of content bitrate representations available in the DASH Storage; (3) descriptions transit on the network; (4) the Description Aggregator module merges the received descriptions to reconstruct the original content quality; Finally, (5) as content is being delivered over N

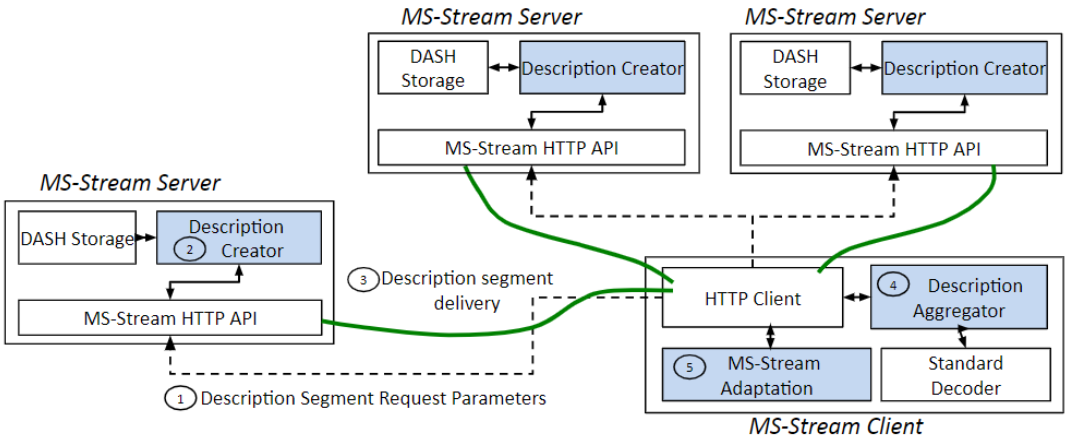

Figure 3.2: MS-Stream modular architecture

paths (N sources), a global and per-path adaptation process is required to deal with path heterogeneity, performed in the Adaptation module.

The MS-Stream client is an evolving DASH client, which incorporates a cost-effective description aggregation module and an adaptation engine capable of content and server adaptation. From a technical standpoint, MS-Stream ensures DASH-backward compliance: upon the delivery of a regular DASH manifest file, an MS-Stream client can use uni-source DASH protocol. MS-Stream is DASH-forward compliant because both the MPD and the video segments are standard. As authorized by the DASH standard, a MPD in MS-Stream contains a list of **BaseURL** objects embedding the base URL adresses of the multiple video servers.

In order to benefit from codec standard-compliance and from a large amount of sub-streams (also referred as subsegments or descriptions, c.f. related works), MS-Stream focus on a hybrid sub-stream generation solution based on temporal and compressed data domains.

A sub-stream is generated by interleaving the Group Of Pictures (GoPs) available at two different bitrates of the same segment. The number of descriptions for a given video segment may vary during the streaming session and is determined by the client according to the observed heterogeneous characteristics of the available servers, and to the targeted bitrate. Examples of GoP-based descriptions are depicted in Figure 3.3. They are composed of GoPs in high-quality and GoPs in low-quality. Reconstructing the original content quality is achieved by selecting the GoPs of higher size in the pool of available descriptions to recreate segments with only high-quality GoPs. Should some de-

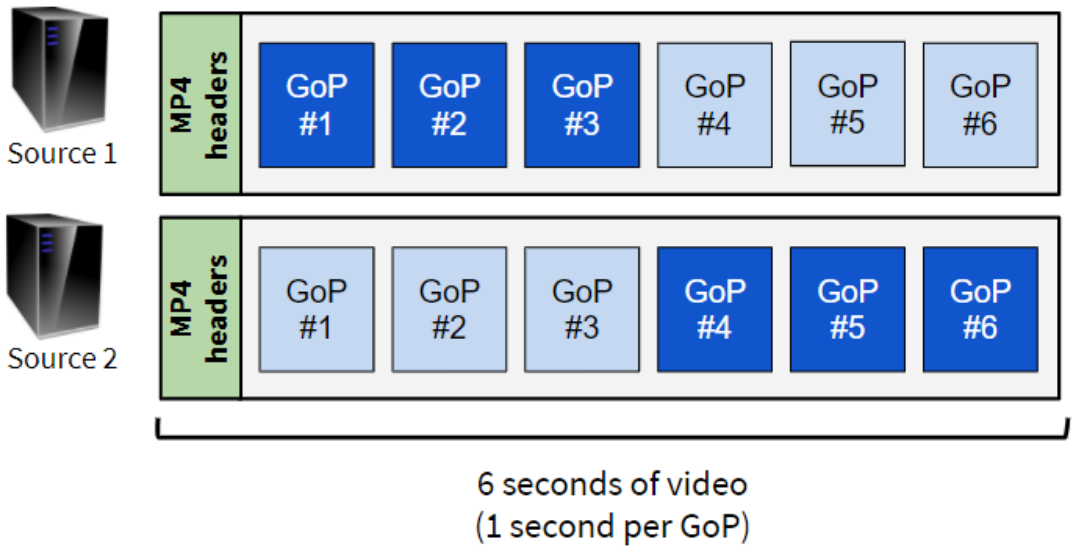

Figure 3.3: Two substreams with GoP in high and low quality from two different sources

scriptions be missing for content reconstruction at client side, the content is still playable with sub-optimal visual quality. Figure 3.4 shows an example of content reconstruction. In this example, the client receives Description 1 and Description 2, but not Description 3. The final segment contains the GoP in high-quality from Description 1 and 2, but the last two GoPs are missing. In this situation, the two last low-quality GoPs from the available descriptions are selected to complete the reconstructed segment.

The protocol presents a list of specificities included in our sub-stream generation scheme, easing its adoption by streaming providers: video-codec standard compatibility, tunable redundancy, low additional complexity and the possibility to create as many descriptions as needed. By leaving the encoded video data and its data structure unaltered, descriptions are compliant with video codec standards and do not require new decoder implementations. Low-complexity post-encoding and pre-decoding steps are required to respectively create and merge descriptions, thus not altering directly with the video encoding/decoding process. Descriptions are standalone (each sub-segment can be decoded and played independently when required); this is made possible by copying some common GoPs at a critically low bitrate (i.e., redundancy) into them. Such non-dependency between flows provides high reliability in heterogeneous unreliable networks, as any independent sub-segment can be lost without interrupting the streaming session. Nevertheless, the more the redundancy, the greater the network bandwidth consump-

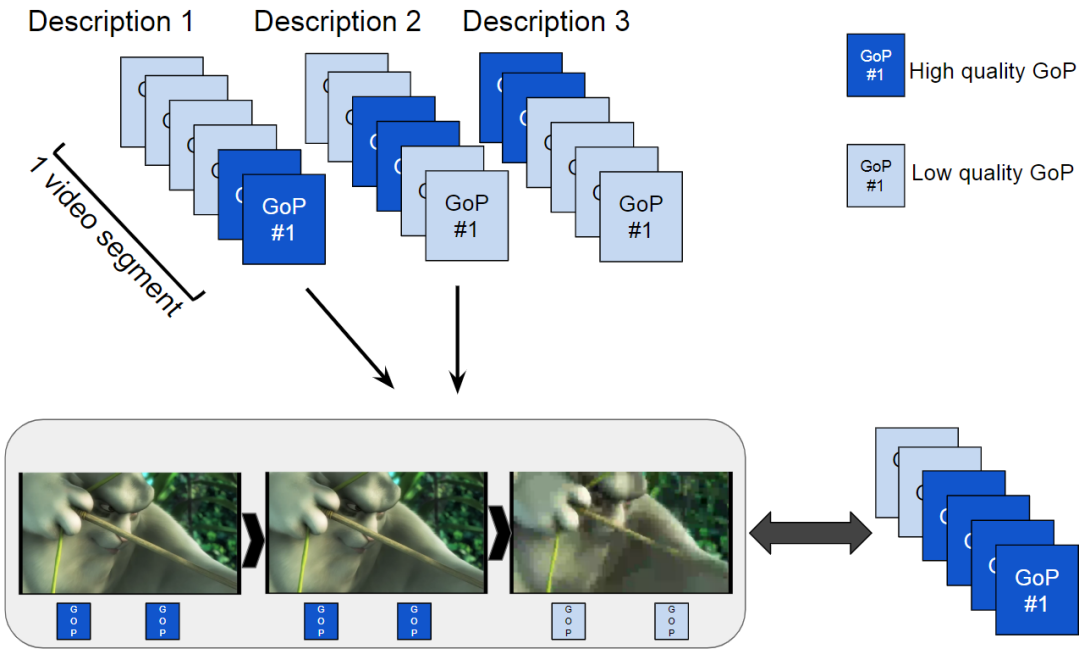

Figure 3.4: Description aggregation

tion overhead. Consequently, in order to match any specific scenario (e.g., from reliable sources with high throughput to volatile sources with low throughput), MS-Stream was extended in [Bruneau-Queyreix et al., 2017b] and [Bruneau-Queyreix, 2017] to embed mechanisms to control the degree of redundancy based on the observed network conditions and on the quality requested by the client. In this second version, video servers are able to send MP4 segments with missing GoPs, as illustrated in Figure 3.5. The segments with missing GoPs are not decodable and cannot be watched if they are the only segments received. The new synchronization algorithms take this into consideration when creating and monitoring the downloads in order to request redundant low-quality GoPs from the most reliable server. Moreover, in case the video buffer becomes critical and the video playback may be interrupted by the lack of some specific GoPs, the player reschedules the downloads of missing GoPs in low-quality to reliable sources.

MS-Stream summary

MS-Stream is a proposition that extends the DASH standard, wherein a client can simultaneously utilize multiple servers in order to aggregate bandwidth over multiple links while being resilient to network and server impairments. It reconstructs segments of the highest-possible bitrate supported by its download link, even when none of the

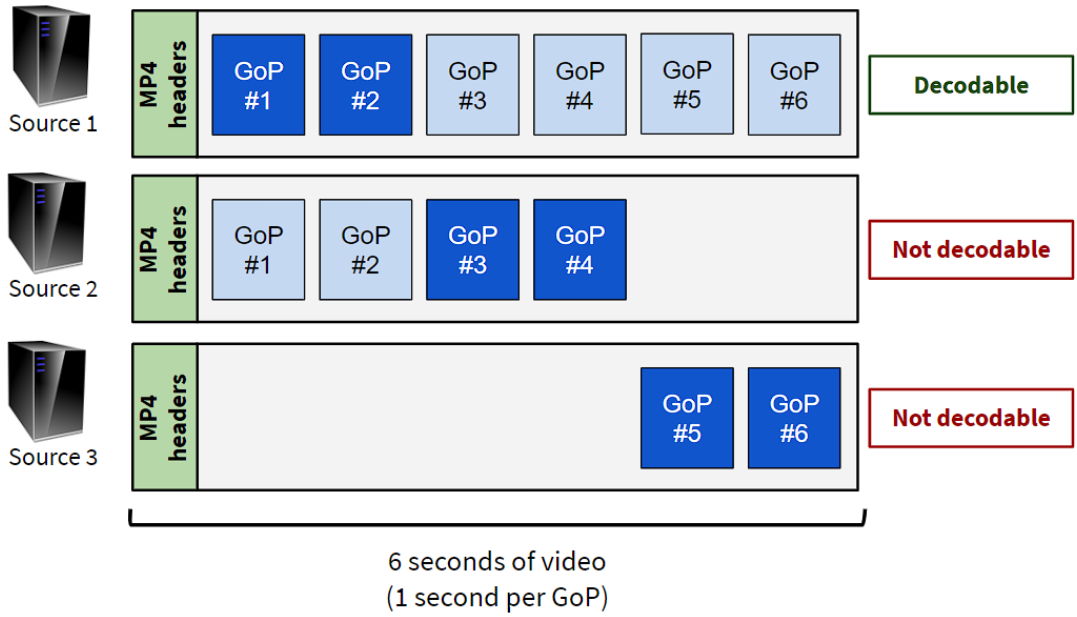

Figure 3.5: Three substreams with GoPs, one of them being decodable with all the GoPs in various qualities whereas the two other are not decodable and does not possess all the GoPs

sources is able to individually provide this quality. The MS-Stream client uses the redundant GoPs in low quality as a fallback if high-quality segments are not received on time. MS-Stream has been evaluated in [Bruneau-Queyreix, 2017, Bruneau-Queyreix et al., 2017a, Bruneau-Queyreix et al., 2018, Bruneau-Queyreix et al., 2016] using official network traces provided by DASH-IF [DASH-IF Guidelines, 2020]. Results from these papers shows that in a multi-source context, our contribution outperforms standard HAS solution and basic source switching technics in terms of QoE and reliability. However, because of the redundant low-quality GoPs introduced by design, the solution comes with an overhead. The bandwidth overhead depends on the bitrates used, and is typically lower than 10% with the help of the redundancy management algorithms added in [Bruneau-Queyreix et al., 2017b]. Besides, it ought to be noted that the generation and aggregation of sub-segments have very low processing footprints as they only require to assemble already encoded GoPs available at different bitrates.

As previously stated, streaming services usually rely on large-scale Content Delivery Network (CDN) infrastructures to host video content, and HTTP Adaptive Streaming (HAS) solutions (such as the DASH standard or MS-Stream) for delivery. When accessing

a video stream, consuming clients are automatically re-directed to the closest server to temper network congestion and achieve higher throughput. However, if a large amount of end-users located under the same geographic area is simultaneously consuming the same streamed content, the nearest server can rapidly become overloaded. Over-the-top video streaming services would greatly benefit from a mechanism to further increase QoE and decrease costs. Even though MS-Stream is resilient to server and network faults, some users can experience a video bitrate degradation when a server becomes overloaded, as they would only receive low quality descriptions from other sources. Therefore, we developed MUSLIN, a solution to tamper with CDN-side network and server congestion.

3.2 MUSLIN: Multi-Source Live Streaming

This section introduces MUSLIN [Da Silva et al., 2019b, Da Silva et al., 2018a, Da Silva et al., 2018b], a solution supporting a high and fairly shared end-users' Quality of Experience (QoE) for live video streaming services over the Internet. MUSLIN dynamically replicates content and improves server advertising to clients to enhance users' QoE and fairness while minimizing the required infrastructure scale. MUSLIN is based on MS-Stream for content delivery to aggregate throughput and further increase QoE at the lowest cost.

MUSLIN relies on periodic feedback from MUSLIN clients during streaming sessions and a ranking score for servers provisioning and advertising. As shown on Figure 3.6, the MUSLIN server provisioning module periodically estimates the required throughput to dynamically adjust the infrastructure scale according to real-world needs. The MUSLIN server selection module then advertises relevant content servers to clients depending on multiple criteria such as distance, bandwidth and server load. For content delivery, MUSLIN leverages MS-Stream, the above-mentioned multiple-source streaming solution based on the DASH standard, in which a client can simultaneously use several servers to aggregate throughput from multiple channels and offer a higher QoE for its users.

MUSLIN's goal is to provide a high and fairly shared QoE for live video content delivery. As QoE is subjective, it is a difficult challenge to evaluate the QoE of end-users. QoE depends on many criteria, such as stalls, video resolution, encoding quality factor, bitrate fluctuation over time, glitches, etc. The ITU-T recently provided automated methods to algorithmically assess streaming QoE according to multiple factors in the P.1203 recommendation [ITU-T, 2017]. As it is complex and costly to take all parameters into account, MUSLIN tackles the main reasons why end-users are not satisfied with

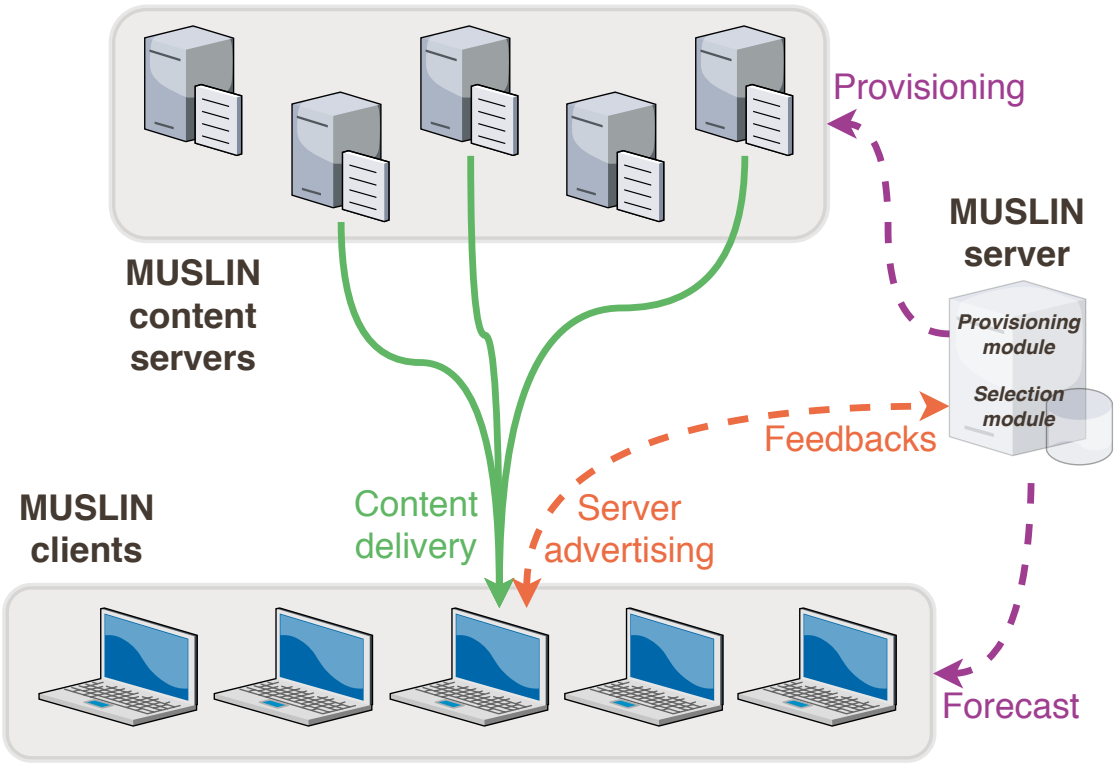

Figure 3.6: MUSLIN overview

their streaming experience, which are the number of rebuffering events, the average video bitrate displayed, and the number of quality changes during the session. Indeed, as mentioned before, rebuffering events are considered the main negative impact on perceived QoE [Hoßfeld et al., 2011], and both the average video bitrate and the quality changes have a significantly higher influence on QoE in adaptive streaming than other criteria [Seufert et al., 2014].

MUSLIN intends to solve the root causes for such QoE degradation, the two main reasons being (1) the server load and (2) the low bandwidth between the server and the client. Indeed, if a server is overloaded or if the network channel bandwidth to this server is low, clients requests to this server will timeout and cause rebufferings or visual quality degradation. Therefore, MUSLIN is able to monitor current delivery conditions to adapt its delivery schemes, thanks to a Ranking Score RS_{sc} in order to provision and advertise healthy servers to clients (as shown on Figure 3.7).

The MUSLIN system is composed of a MUSLIN server, MS-Stream clients, and MS-Stream content delivery servers augmented with an additional MUSLIN layer to handle

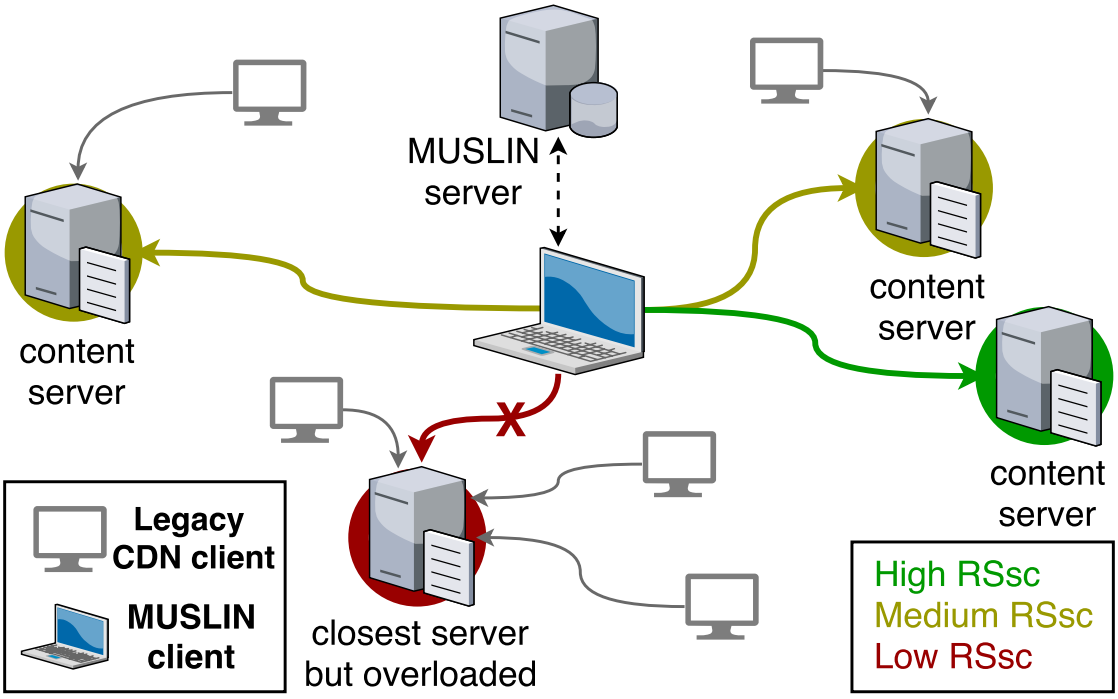

Figure 3.7: MUSLIN Process. If nearby content servers are overloaded, the MUSLIN server selects and advertises other content servers with a higher Ranking Score RS_{sc} to the client.

feedbacs and provisioning. MUSLIN clients send periodic feedbacs to the MUSLIN server, including the observed bandwidth from each server, the video sub-segment requests failure (timeout) rate, their average displayed video bitrate, the number of rebufferings they experience, and the number of quality changes. Then, based on these feedback, the MUSLIN server accordingly scales the underlying delivery platform to provide a higher QoE to end-users.

Fairness among users is mostly achieved thanks to the periodic feedback sent from the clients. They aim at monitoring the QoS metrics and the respective QoE each user is provided with, in order to improve server provisioning and selection accordingly. Server and Network Assisted DASH (SAND) [DASH-IF, 2018], introduced in MPEG-DASH Part 5, defines a standard for control messages exchanged between the servers and clients to report metrics. MUSLIN feedback messages are currently not compliant with SAND, as MUSLIN was developed prior to this standard, but will be in a future version for better interoperability. Besides, MS-Stream allows maximizing server throughput and reduce competition between clients when CDN servers resources are saturated, as each client depends on multiple servers and is not bound to a specific one.

MUSLIN is specifically effective for live streaming where churn rate can be very high and important audience fluctuations can happen within seconds. In a Video on Demand (VoD) use case, clients' buffers can be larger and there is less pressure to react in real-time.

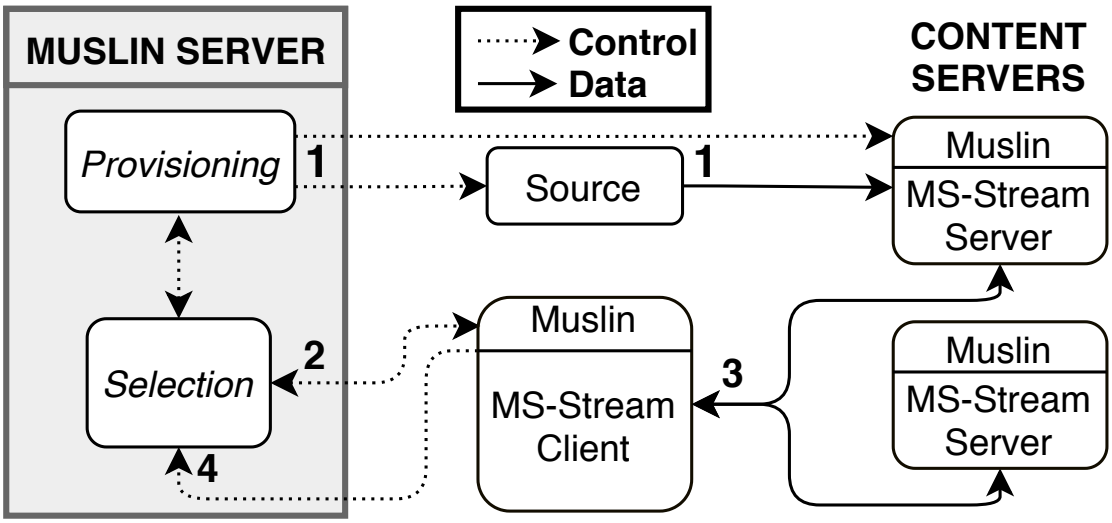

Figure 3.8: MUSLIN system architecture overview

As illustrated in Figure 3.8, (**1**) the MUSLIN server dynamically provisions content servers and replicates content to available MS-Stream content delivery servers, which then register themselves to the selection module; (**2**) when a client requests a Media Presentation Description (MPD) file, the selection module replies with a list of available servers; (**3**) the client can access live content and begin the streaming session with the MS-Stream protocol; (**4**) MUSLIN clients send periodic feedback.

The MUSLIN modules and MUSLIN content servers overlay are implemented in Java and run inside light-weight Docker containers. MUSLIN content servers are built on top of MS-Stream servers by adding the necessary glue code to manage the interaction with the MUSLIN provisioning and selection modules. All interactions with the MUSLIN modules fulfill the REST architecture style. MUSLIN clients are developed in pure JavaScript and run within any mobile or desktop Web browser. Clients extend MS-Stream clients by featuring periodic feedback reports to the MUSLIN server.

MUSLIN summary

We presented MUSLIN, a multi-source live streaming system which manages to reach higher QoE and fairness while lowering provisioning costs by combining dynamic provisioning with feedback-based servers selection and multiple-source content delivery. MUSLIN takes into account clients real-time feedback, dynamically replicates content and improves server advertising to clients to enhance users' QoE and fairness while minimizing the required infrastructure scale. We showed in extensive lab experiments MUSLIN [Da Silva et al., 2019b, Da Silva et al., 2018a] that thanks to the coupling of MS-Stream with the proposed MUSLIN system, end-users experienced almost no rebufferings, a higher video bitrate, and more evenly shared QoE, compared to existing state-of-the-art streaming systems setups.

Chapter summary

Two preliminary works are proposed as a starting point for the two main contributions of this thesis.
(1) MS-Stream is a multiple-source streaming solution that extends the DASH standard. MS-Stream clients can aggregate bandwidth from multiple servers to increase video quality and drastically reduce rebufferings, thanks to slight redundancy with a low network overhead cost.
(2) MUSLIN uses dynamic server provisioning and advertising based on real-time delivery conditions combined with MS-Stream to provide a better QoE at the lowest cost.

Chapter 4

State of the Art: P2P and 5G video delivery

> "The mind of the subject will desperately struggle to create memories where none exist."
>
> Robert Lutece

In this chapter, we overview the current state of the art related to the main contributions of the thesis. It is divided into two independent sections. Section 4.1 is about video delivery and quality adaptation in **peer-to-peer streaming systems**, and is related to our first contribution, PMS+ (see Chapter 5). The second section focuses on **media content delivery in future 5G networks**, particularly in the context of indoor networks, and is related to our second contribution, MSS/RRLH (see Chapter 6).

4.1 P2P streaming systems

This section overviews P2P-based systems for video streaming. Existing solutions consider pure P2P approaches but also hybrid ones in cunjunction with a deployed CDN, i.e., called hybrid P2P/CDNs. Variations in terms of architectures are presented, as well as related work on their evolution and on QoE improvement.

4.1.1 P2P streaming solutions overview

By not being dependent to a fixed-size streaming infrastructure, P2P systems [P2P Architecture, 2009] represent an interesting alternative to CDNs. Peers participate in

the streaming process as clients and servers, at the same time. They share their storage and network bandwidth with the rest of the community in order to increase content availability and streaming performance. Due to the inherent nature of P2P, clients aim to dedicate resources to serve requests retrieved from others. Hence, a P2P system can in principle scale automatically without the need of dedicated servers [Deng and Xu, 2013]. Therefore, unlike CDNs, P2P systems have the capacity to drastically reduce costs on infrastructure. Nevertheless, weaknesses and limitations exist related to P2P networks, the two main ones being: (1) peer churn: peers frequently and suddenly leaving or joining the system due to network failures or based on their own decisions [Stutzbach and Rejaie, 2006]. (2) peer (resources) heterogeneity: peers' available resources to deliver content varying importantly from one to the other, especially in terms of bandwidth due to the multiplicity of Internet connectivity types [Kaune et al., 2010]. When those two effects happen regularly in the network, the impact on the streaming performance is high. The startup delay increases, quality variations and video playback disruptions are more and more frequent [Huang et al., 2008].

4.1.1.1 P2P streaming as a research subject

The research community produced many studies related to the domain of P2P-based streaming. First, concerning the in-depth techniques for achieving structured and unstructured P2P networks, [Crowcroft et al., 2005] outlines a quite exhaustive survey. As well, [Xiao et al., 2009] overviews the peering mechanisms and the chunk scheduling techniques. Next, concerning the delivery methods, [Jurca et al., 2007] presents several adaptive streaming techniques conjuncted to P2P systems. Further work has also been conducted on the encoding part, such as layered video, along with Multiple Description Coding (MDC) techniques and Network Coding. The purpose was to propose adaptive streaming systems targeted to many users [Zhang et al., 2005b, Li et al., 2007], each one retrieving the adapted corresponding layer. Additionally, significant research work has been dedicated to pull-based P2P SVC streaming systems. The objective was to optimize the overlay structure in relation with the data scheduling in order to deliver the best possible video quality to each end-user [Medjiah et al., 2014, Moon et al., 2013, Xiao et al., 2009, Eberhard et al., 2010, Capovilla et al., 2010]. Finally, many studies focused on the real deployment of P2P-based streaming systems, such as e.g. PPLive VoD [Huang et al., 2008], Joost [Lei et al., 2010], CoolStreaming [Zhang et al., 2005b, Keung, 2007]. The outcomes show a direct inter-relation between the deployed infrastructure, on one

hand, and the capabilities in terms of (1) scalability, (2) fairness between peers, and (3) network auto-organization, on the other.

4.1.1.2 P2P streaming as a business opportunity

The business opportunity of P2P streaming systems have been identified by major video delivery providers. The main advantage of P2P is to scale while saving expensive infrastructure costs. Example of commercial systems which have developed P2P media streaming are NetSession from Akamai [Akamai, 2020], LiveSky [Yin et al., 2009a], Kankan [Zhang et al., 2014], and even Spotify [Spotify, 2020]. From a technical point of view, these systems often need the user to install a dedicated application. The P2P data exchange are coded with standard sockets inside the application. One important problem faced during the development is peer accessibility because a lot of end-users are connected behind NATs. To solve this issue, Akamai, Kankan and LiveSky take advantage of Interactive Connectivity Establishment (ICE). ICE uses two type of servers: a Session Traversal Utilities for NAT (STUN) which helps to discover the ports that can be used to traverse some NATs and a Traversal Using Relays around NAT (TURN) that can be used as a relay between the peers. Spotify has implemented another mechanism. The application relies on Universal Plug and Play (UPnP) that permits to dynamically open ports on routers. However, asking the user to install a specific application on his device is not always a good solution, especially when the user is only casually consuming media content. Moreover, with the plurality of devices —desktop computers, smartphones, set-top-box, and smart TV— the video provider might need to maintain several applications. A browser solution would help to attract casual users while being compatible with most devices.

With the rise of browser-based application and browser-based video players —in websites like Youtube, Daylimotion, Netflix, OTT TV— one tool has become a technical game changer for P2P streaming implementations. WebRTC [WebRTC, 2015], for Web Real-Time Communication, is an API standardized by W3C and IETF to communicate from one browser to another. The first work have started in 2011 but the API has been implemented in major browsers from 2013 to 2015. The goal of WebRTC is to provide ultra-low latency UDP sessions in order to support real-time scenarios like video conferencing or multiplayer gaming between users. It embeds security and NAT-traversal functionalities. The WebRTC API is composed of three main parts: **User Media**, **RTCPeerConnection** and **Data Channels**.

User Media. The User Media API offers tools to detect, select and use media equipment like cameras and microphones. With these functions, clients are able to create media streams from a webcam to be sent to other peers. The media used in a stream can be modified during live, to switch from the front to the back camera of a cell phone for example. The packets produced by the stream can be caught in order to save the video or transfer the data to an external server using any protocol.

RTCPeerConnection. As the name suggests, the goals of these functions are to establish a connection between two peers. A RTCPeerConnection takes some options as an input. For example, it is possible to provide servers for the NAT-traversal capabilities, define if a connection should be reliable, and if the packets should arrive in order. Then, a Session Description Protocol (SDP) string is created containing information to create a link between peers. The SDP should be sent to the corresponding peer but the method employed to transfer this information is not specified. The most common solution is to use a bidirectional websocket-based signaling server to ease the initialization process. Once the SDP is sent, the browser begins the ICE mechanism with the help of the STUN and TURN servers provided in the options. If a direct connection can be established with the help of a STUN server, this solution is prioritized. However, if the NAT can not allow such a link, a relay TURN server is selected to establish the session. Once the peer connection is in ready state, it becomes possible to send the media stream created with the **User Media** API to begin a simple ultra low latency video conference.

Data Channels. Once a RTC connection is established and if both peers support data transmission, a data channel can be created. Raw data can be sent from one peer to another through SCTP (Stream Control Transmission Protocol) over UDP channels. This feature brings a lot of flexibility because any kind of information can be transferred. It may be used to transfer custom control flow, specific text messages or even video segments. Because of these data channels, adding P2P capabilities to existing JavaScript DASH players has become possible. Instead of starting a standard HTTP request to get a segment from a server, a custom query can be sent to another peer through a reliable channel. Proprietary solutions offered by Peer5 [Peer5, 2020], Streamroot [Streamroot, 2020] or Hive Streaming [Hive Streaming, 2020] are using WebRTC data channels in their players. Similarly, the open source library P2P Media Loader [Novage, 2020] provide P2P capabilities on top of HAS video players with the help of WebRTC. The API also draws researchers attention to improve DASH segments delivery [Zhao et al., 2016].

4.1.2 Hybrid P2P/CDN streaming overview

The evolution in P2P domain led to the creation of hybrid P2P/CDN streaming solutions, the first goal being to alleviate as much as possible existing limitations of P2P networks —but still keeping the advantages— and benefit from CDNs reliability. Figure 4.1 illustrates the main approach behind hybrid P2P/CDN streaming. Content distribution is assured to be always effective thanks to geographically distributed high-end infrastructures, composed of reliable servers and configured to provide a high quality of service. The process is usually assisted with service-level agreements (SLAs) between CDN providers and content owners. However, even in such configuration, the scaling up of the infrastructure may induce important costs. Traditional CDNs require deployment and management of geographically distributed data centers, which can rapidly be very costly [Kim et al., 2015]. Thanks to the self-scaling property of P2P systems, such cost can be lowered [Menasche et al., 2009] [Qiu and Srikant, 2004], depending on the number of users capable of assuring a new re-transmission of the content they just received. A trade-off must hence be reached on the number of servers to be deployed, strongly depending on the reliability of users (in terms of presence and of resources) [Kaune et al., 2010]. Therefore, the hybrid P2P/CDN approach offers greater scalability thanks to the P2P paradigm while, at the same time, to compensate the bandwidth imbalance and/or unavailability of content access thanks to reliable CDN servers.

From an execution process point of view, in most common solutions ([Peterson and Sirer, 2009] [Roverso et al., 2011] [Zhao et al., 2013]]), a P2P overlay is first constructed from a subset of the peers. CDN servers deliver content to this overlay. After, the peers from the overlay start being servers of the received content to their neighbors up to their capabilities in terms of bandwidth. In case a peer is unable to deliver efficiently, i.e., it is overloaded, the impacted clients switch back to receive the streams from the CDN servers.

An analytical analysis of such concept can be found in [Xu et al., 2006] [Lv et al., 2012]. Simulations have been achieved in [Chellouche et al., 2012] [Huang et al., 2008], and large-scale deployment and evaluations in [Zhang et al., 2015a] [Yin et al., 2009b] [Yin et al., 2010]. A large number of OTT players have also performed and reported feasibility studies, such as BBC iPlayer [Karamshuk et al., 2015], Conviva [Balachandran et al., 2013], and MSN Video [Huang et al., 2007a]. Commercial CDNs have also started to deploy their own solutions. Those include Akamai [Zhao et al., 2013], ChinaCache [Yin et al., 2010], and Xunlei KanKan [Zhang et al., 2015a]. In general, results show a wide

range from 20 up to 80% of offloaded trafic from CDN servers to P2P, mostly depending on the architecture used and with various impacts on the QoE.

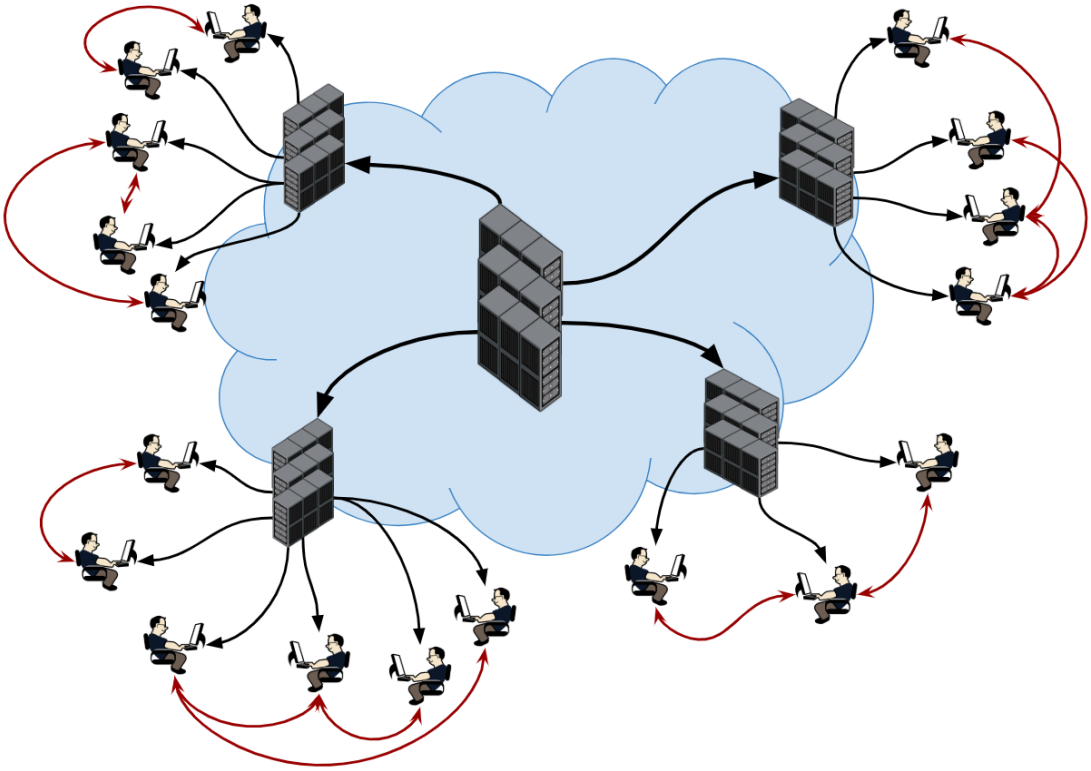

Figure 4.1: Illustration of hybrid P2P/CDN

4.1.3 P2P-based streaming architectures

In most cases, a P2P network can be classified based on the organization of its connected peers, namely onto a structured or an unstructured network. A tree-based architecture is deployed over a structured network whilst a mesh-based one relies on top of an unstructured network [Gho et al., 2013].

Tree-based architectures. Peers organization is based on a hierarchical (possibly multiple) tree structure. An example of tree-based topology is illustrated in Figure 4.2. The root node acts as the source of delivery for the first levels nodes, which in turn do the same for the next levels until reaching each leaf. Each node pushes data it receives to a number of other nodes. Each node in the tree can have as many receivers as its capacity allows, with respect to the streaming rate. The tree construction and the nodes

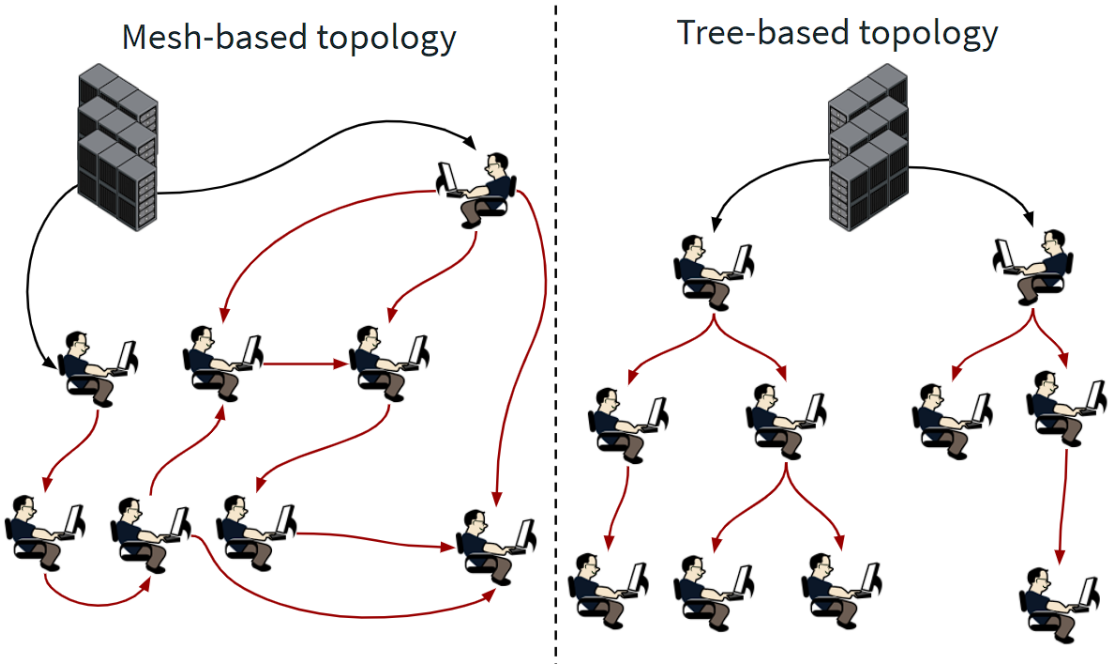

Figure 4.2: Mesh-based topology and tree-based topology

relations are determined by various factors. Although the tree-based approach is simple to deploy, high peer churn has a drastic effect on the streaming performance [Stutzbach and Rejaie, 2006, Gho et al., 2013].

Related to quality, limitations are imposed by each intermediate node in the tree. It is highly possible that a peer would not have enough bandwidth available to serve its linked child-clients in the quality they could have expected with respect to their capacities. This leads to an unsatisfactory situation. Multiple tree architectures address this problem by providing redundancy in network paths. However, designing and maintaining such systems is very challenging and may lead to solving contradictory issues such as minimizing tree depth, while simultaneously provisioning network path diversity [J. Ghoshal and Wang, 2007].

Mesh-based architectures. Peers organization do not rely on a static topology. Peers establish peering relationships dynamically according to the data availability. The mesh-based approach is indeed referred as the data-driven approach [Zhang et al., 2005a]. An example of mesh-based topology is provided in Figure 4.2. In contrast to structured network, which constantly repairs its structure in a highly dynamic P2P environment, the data availability among peers guides the peer relationships for mesh-based streaming

model. Participating peers form a randomly connected overlay. Each peer maintains a certain number of parent nodes and, at the same time, serves a specific number of neighbors. Peers upload or download the content from multiple neighbors upon data availability. In case of departure of a serving peer, client peers can still get the content from remaining ones. During the video streaming session, the peers continue to discover new potential serving peers to establish new peering relationships. Due to the fact that each peer maintains its own neighborhood dynamically, the mesh-based approach is robust against peer churn. For the discovery part, peers usually exchange information on which chunk of video is owned by which peer. It is called the "buffer maps" and it is performed during the "pull" process. Besides offering better resilient to node failures —since each node may rely on multiple peers to retrieve the desired content— mesh-based organization also has the advantage of low cost and simplicity of design and maintenance. Disadvantages towards tree-based consist in additional delays in pulling the data blocks resulting from prior control messages exchanges and the necessity of having large buffers [Seyyedi and Akbari, 2011, Magharei et al., 2007].

When focusing on the specific case of hybrid P2P/CDN architecture, the objective is to combine the advantages of both worlds while limiting their drawbacks. The challenge is to succeed in an efficient technical integration of both somehow contradictory concepts. To this end, two main categories of architectures have come out: centralized (as CDNs) and decentralized (as P2P).

In centralized hybrid P2P/CDNs, clients register themselves into the CDN system and store contextual information such as IP addresses, ports, consumed content, into a centralized database. Delivery servers from the CDN first transmit the first video segments to the clients while a "tracker" server also from the CDN is used to send periodically a list of randomly selected peers from the P2P overlay. This list is constantly refreshed to overcome problems due to high peer churn. In case of delivery failure from a peer, delivery servers from the CDN are used as backup. The centralized hybrid P2P/CDN architecture has several advantages over pure P2P systems: (1) in terms of reliability, it is much higher thanks to high content availability at replica servers, (2) in terms of management, especially related to the P2P overlay, it is much simpler since the CDN system is in charge of everything. Centralized architectures have been implemented in the vast majority of commercial hybrid P2P/CDNs [Zhao et al., 2013, Kim et al., 2011, Zhang et al., 2015a]. In decentralized hybrid P2P/CDN solutions, the P2P overlay is managed by elected tracker-peers (or super-peers) that are selected based on multiple criteria, such as their proximity to the CDN server, stability, network and processing capabilities.

When a client joins the decentralized hybrid P2P/CDN, it contacts an elected super-peer to obtain a list of active peers consuming the desired video content. The request is redirected to the nearest super-peers until a super-peer holding such a list is found. The super-peers also act as entry point for the content to arrive in the P2P overlay. When the amount of peers downloading a given content is insufficient, the super-peers are responsible for retrieving the consumed content from the CDN servers. Decentralized architectures have the advantage of minimizing the required CDN infrastructure by delegating most of the P2P overlay management functions to the clients. Nevertheless, they become more vulnerable to malicious attacks and are more difficult to manage in comparison with the centralized approach [Anjum et al., 2017].

4.1.4 QoE overview related to P2P-based streaming solutions

First of all, in case of P2P-based streaming solutions, QoE is directly impacted by (1) peer churn and (2) peer heterogeneity, as mentioned earlier.

Peer churn. Concerning peer churn, various work exist in the literature. [Chen et al., 2015] created Thunder Crystal, a crowd sourcing-based content distribution system that relies on smart Access Points (APs) distributed among clients. A smart AP is equipped with a large storage for caching high resolution videos and trades content with peers in exchange for some benefits obtained from the content provider (e.g., discounted subscription). Based on the same principle, Youku, one of the largest VoD services in China, deployed millions of dedicated smart peer routers (i.e., set-top-boxes) with 8GB storage capacity in consumers' homes and offices to assist content distribution [Ma et al., 2016]. [Nicolas et al., 2013] realized a trace-driven evaluation of YouTube traffic and, based on it, proposed P2PTube, a system exploiting set-top-boxes to assist YouTube content distribution. Results reported up to 46% of videos to be served from peers, 10% more efficient than previous result reported in [Zink et al., 2009]. [Gramatikov and Jaureguizar, 2016] elaborated a mathematical model to evaluate the impact of peer-churn on a hybrid P2P/CDN, when the users of set-top-boxes are not willing to share their resources. Finally, [Markakis et al., 2016] proposed a Home Box-assisted approach which relies on exploiting set-top-boxes as proxies between end-users and CDNs.

Peer heterogeneity. Related to peer heterogeneity, it is commonly known that the multiplicity and variability of Internet connectivity technologies are the main impacting points towards an efficient delivery. The asymmetry between upload and download

bandwidths of existing networks, such as ADSL2+, is an issue for all video streaming P2P applications. A factor up to 15 times can be found between the download capability compared to the upload one. In such scenarios, P2P live streaming applications, such as PPlive or Maze Networks, experience bottlenecks for high resolution video re-delivery from watching peers [Baccelli et al., 2013]. In order to cope with these issues, [Huang et al., 2007b] proposed two different peer selection policies: Water Levelling (WL) and Greedy Policy (GP). The WL approach intends to uniformly distribute the extra uploading bandwidth among all peers. The GP approach suggests that each client simply dedicates its remaining upload bandwidth to the nearest peer. Simulation results show that, in a hybrid P2P/CDN case, along with the greedy policy, the CDN servers bandwidth requirements could be reduced from 2.20 Gbps to 0.79 Gbps.

Nevertheless, the main issue is not only to reduce the cost of the CDN part but also to assure the best possible QoE to consumers. This challenge has been largely studied [Bobarshad et al., 2010] [Bobarshad et al., 2012] [Liu et al., 2008] [Passarella, 2012]. The majority of research efforts focused on strategies for improvements in terms of startup delay and minimization of video interruptions.

[Xu et al., 2006] proposes a three-phase streaming process to achieve low startup delay in case of an hybrid architecture. Peers first download some initial segments from a geographically close CDN server and only afterwards start to retrieve the remaining segments from the P2P network, leaving time for signaling to be established. [Lu et al., 2011] and [Ha et al., 2011] exploit the use of an effective buffer management at the client side, where the buffer is divided in two zones: normal and emergency. The concept of high priority is introduced and the highest is given to the first video segments to be displayed. When the buffer level is close to reach the emergency zone, the video segments switch to be retrieved from CDN nodes. In normal mode, the strategy is to maximize the retrieval from peers. The relative size of the two zones depends on the number of CDN nodes available in the system and on the number of participating peers. A new peer will first be server by a CDN server until its buffer is sufficiently filled (i.e., beyond the emergency zone), before switching to P2P data transfer.

On the same basis, [Lu et al., 2011] adds one region to the playback buffer, called startup region. Also, a jump into the emergency zone is triggered when a peer fails to download the required segments from neighboring peers before a given deadline. Simulations show improvements of 2.5 sec in the average startup delay in comparison to the traditional peer-to-peer networks.

Concerning video playback interruptions, several research works have been performed in order to minimize them [Xu et al., 2006] [Kim et al., 2011]. [Kim et al., 2011] proposes a Group Based decentralized CDN-P2P (G-CP2P) architecture, for which it deployed a location/content-aware peer selection mechanism. In the G-CP2P strategy, peers join different P2P groups based on their latency. Each group is controlled and managed by a super-peer, selected based on its closest distance to the CDN edge server. The super-peer exploits a distributed hash table algorithm (DHT) [Ratnasamy et al., 2001] to locate the content in the P2P network. At the moment of the super-peer election, the round trip time (RTT) is used as a key so that all super-peers are arranged and partitioned in increasing order of RTTs. This way, several separate clusters can be composed. Each new peer is linked to a super-peer which has the same order of RTT and joins the group within the same cluster. Results from simulation -and comparison with [Xu et al., 2006]- conclude that an average decrease of 0.5 sec in the start-up delay is gained thanks to location/content-aware mechanisms, as well as a reduction in the delivery time of each chunk from 0.5 to 4.5 sec.

In addition, [Tian and Liu, 2013] and [Lederer et al., 2012] are among the first to have proposed a joint combination of P2P and HAS techniques. A pragmatic standard-compliant solution to integrate peer-assisted streaming in conventional DASH client-server systems is outlined in [Lederer et al., 2012]. Nonetheless, the flexibility in using neighboring peers is restrained in downloading a given segment from one peer only. The P2P feature is considered as a secondary and beneficial add-on to the system, and the heterogeneity of peers' upload and download capacities in consuming and re-emitting their downloaded content is not taken into account. [Tian and Liu, 2013] applies DASH to a P2P architecture and exploits game theory, pricing models and the network resources of each peer to govern the quality adaptation decisions.

More recently, [Merani and Natali, 2016] designs a system focusing mainly on peers, where the servers are relegated to deliver the fewest segments possible to some dedicated peers in charge of spreading content in the P2P overlay. The QoE level at the consumer side is not considered, nor the variations of peers bandwidth capacities. Only the download rate, at the given time, is exploited by the bitrate adaptation logic.

4.1.5 Conclusion

Advanced works have been made to improve the quality of experience of HTTP adaptive streaming. Over-the-top content delivery infrastructures such as CDNs as well as managed network infrastructure with end-to-end QoS improve reliability for streaming

services. However, these architectures are subject to server-side overloading and scalability issues. The result can be frustration for the end-user, but also bad publicity and loss of clients for the video provider.

Several propositions have been made to solve this problem. On one hand, multiple source solution such as MS-Stream and Muslin, our preliminary contributions from Chapter 4, propose to tackle this issue by using several sources at the same time. Despite showing good results by reducing the impact of server-side congestion and improving the reliability of streaming sessions, the system can not scale more than the number of physical resources available.

On the other hand, existing P2P technologies enable video content delivery to a very large number of end-users on heterogeneous end-systems over the Internet. In this section, we dealt with the architecture of P2P systems and compared the peer organization in mesh-based and tree-based topologies. Then, we discussed the advantage of mixing P2P and CDNs to improve the QoE and the scalability of video streaming platforms. We listed the research contributions to improve the end-users' QoE with content quality adaptation and content delivery adaptation for P2P streaming. The two most important issue faced by P2P solutions, namely peer churn and peer resource heterogeneity, have been presented. We exposed existing work on adaptive P2P streaming and hybrid P2P/CDN solutions that both benefit from P2P reduced scalability costs and CDNs high reliability. Nonetheless, the lack of mechanisms improving end-users QoE in such hybrid systems was also identified, especially when considering the advances made with classic DASH-like solutions.

Resulting from this study, the first main contribution of this thesis proposes to combine the best of both world to increase the quality of experience by leveraging on (1) P2P-compatible adaptive mechanisms and (2) multiple-source capabilities. This solution is presented in Chapter 5. The system has been developed over two alternatives. The first one, called PMS, consisted in creating different overlays for every video quality. By relying on both global and remote metrics, a multi-source adaptive quality algorithm is introduced, addressing the reliability issue by taking care of the health of every overlay before allowing a contributing peer to leave. The second one, called PMS+, considers small groups of peers collaborating to retrieve video content. Compared with PMS, PMS+ proposes a faster adaptive mechanism and a better P2P utilization with the help of local peer organization.

4.2 Media content delivery in future 5G networks

This section focuses on future 5G networks and, more specifically the related indoor wireless ones, as well as their evolution towards the leveraging of video streaming through them.

4.2.1 5G networks definition, requirements and architecture

This subsection is dedicated to the provision of background on 5G networks and on the most relevant underlying technologies.

4.2.1.1 5G networks

5G is the next generation of wireless mobile networks. It aims at providing a higher bandwidth and a lower latency than previous generations. In Europe, its development is led by the 5G Infrastructure Public Private Partnership, known as 5G-PPP [5G-PPP, 2020]. 5G-PPP is a joint initiative between the European Commission and European ICT industry (ICT manufacturers, telecommunications operators, service providers, SMEs and research institutions). Its goal is to deliver solutions, architectures, technologies and standards for the next generation ubiquitous communication infrastructures of the coming decade. 5G-PPP's main challenge is to secure Europe's leadership in the particular areas where Europe acts as leader, or where there is potential for creating new markets, such as smart cities, e-health, intelligent transport, education or entertainment and media.

Besides, another partnership towards 5G development exists. The 3rd Generation Partnership Project [3GPP, 2020] is a telecommunications standard development partnership producing reports and specifications that define 3GPP technologies. Even though the 3GPP was initially founded to specify the 3rd generation network, it is currently working on 5G specifications to coordinate regional 5G infrastructures efforts, such as 5G-PPP.

According to 5G-PPP, the key challenges for 5G are:

- Providing 1000× higher wireless area capacity and more varied service capabilities compared to 2010;
- Saving up to 90% energy per service provided, the main focus being in mobile communication networks where the dominating energy consumption comes from the radio access network;

- Reducing the average service creation time cycle from 90 hours to 90 minutes;
- Creating a secure, reliable and dependable Internet with a "zero perceived" downtime for services provision;
- Facilitating very dense deployments of wireless communication links to connect over 7 trillion wireless devices serving over 7 billion people;
- Ensuring for everyone and everywhere the access to a wider panel of services and applications at lower cost.

From an operational level, the following new network characteristics are foreseen:

- 1000× higher mobile data volume per geographical area;
- 10× to 100× more connected devices;
- 10× to 100× higher typical user data rate;
- 10× lower energy consumption;
- End-to-End latency of less than $1ms$;
- Ubiquitous 5G access, including within low density areas.

Figure 4.3 from the European Commission illustrates various use cases that should be facilitated by 5G. A lot of daily tasks are expected to be improved thanks to automation and massive connectivity of objects and services. 5G will be used in the health sector to assist surgeons and monitor patients, in the automotive sector with research focused autonomous vehicle and signaling, and in the power provisioning chain to balance the energy produced by renewable sources. More importantly for the subject of this thesis, the emergence of this new technology is also expected to have an important impact on video delivery techniques and on the way people consume high-quality media. As an example, new VR headsets, mobile devices, and smart TVs are expected to support bandwidth-consuming media such as 360 degree immersive video and 4K UHD content.

4.2.1.2 Network virtualization

In order to reach the ambitious goals of 5G, the flexibility of the network needs to be improved. Replacing specific hardware by neutral computing resources will allow real-time optimization, improvement and adaptation of the functionalities deployed in the

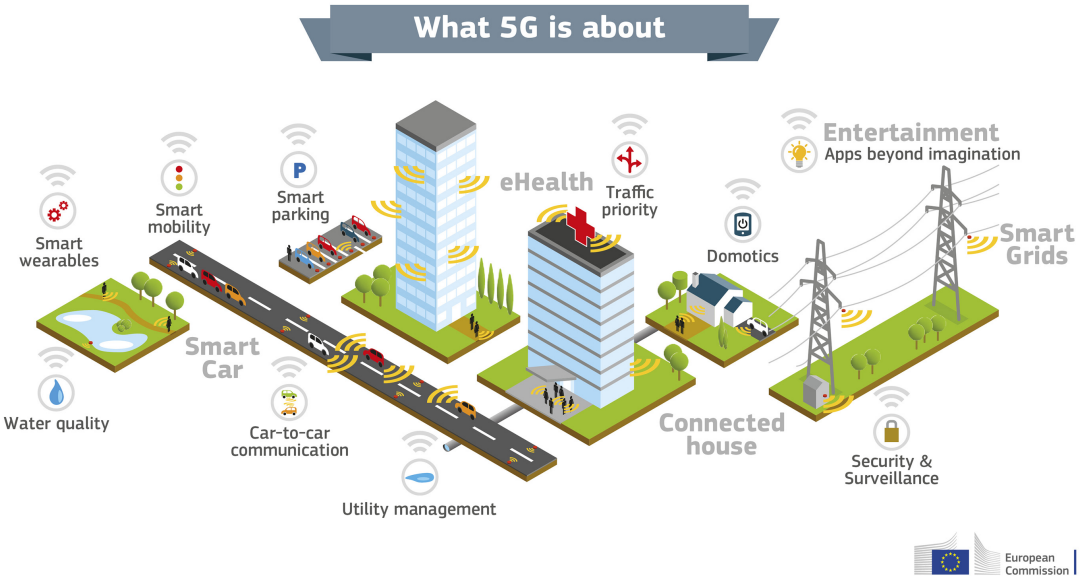

Figure 4.3: Expected use cases impacted by 5G — from the website of the European Commission, at https://ec.europa.eu/

network. Software-defined networking (SDN) is a new approach to design, build, and manage networks, that can be achieved by decoupling or disassociating the system that makes decisions on where traffic is sent (the control plane) from the underlying system that forwards traffic to the selected destination (the data plane) [Nunes et al., 2014]. Hardware SDN switches are widely used in large data centers, but they are relatively expensive and bulky, preventing them from being used in smaller installations. For that reason, softwarized virtual switches appeared. They connect Virtual Machines (VMs) or containers (e.g. Docker) inside dedicated servers. Virtual switch solutions can be both proprietary and open source. Such dynamic switches allow virtual paths and link management in order to optimize bandwidth through different paths or to reduce latency by routing directly the packets to a nearby function. Virtual switch are controlled by a virtual SDN controller in charge of applying functions.

Network Functions Virtualization (NFV) adds new capabilities to networking by allowing a set of management and orchestration functions to be added to the current model of operations, administration, maintenance and provisioning. Before the appearance of NFV, Network Function (NF) implementations were often dependent on the infrastructure on which they were installed. Now, NFV introduces software implementations of Network Functions from the computation, storage, and networking resources they use

by imposing a virtualization layer on the actual existing hardware. The decoupling between software and hardware network entities exposes a new set of functions, i.e., the Virtualization Network Functions (VNFs), and a new set of relationships between them. VNFs can be chained with other VNFs and/or Physical Network Functions to realize a Network Service. Since Network Services, Virtual Links, VNFs and the relationships between them did not exist before the emergence of NFV, their handling requires a new and different set of management and orchestration functions. The Network Functions Virtualization Management and Orchestration (NFV-MANO) architectural framework has the role to manage the Network Function Virtualized Infrastructure and orchestrate the allocation of resources needed by the Network Services and VNFs.

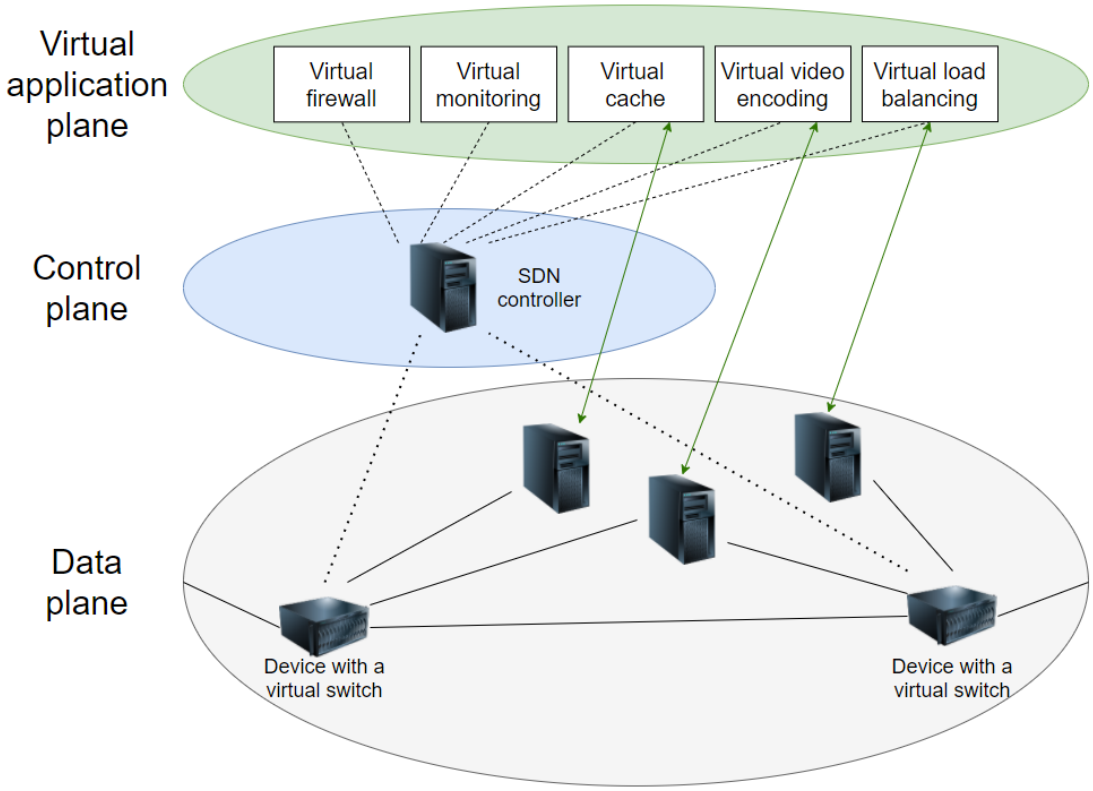

Figure 4.4: Simplified architecture of SDN/NFV

The architecture of SDN is illustrated in Figure 4.4. Devices embedding virtual SDN functions are connected to a controller. Virtual network functions can provide dynamic rules to the controller in order to route packets to appropriate Network Services.

SDN and NFV are widely considered for 5G use cases. The authors of [Bouras et al., 2017, Abdelwahab et al., 2016, Hakiri and Berthou, 2015] review the advances of SD-

N/NFV infrastructures for 5G. In [Ordonez-Lucena et al., 2017], the authors propose network slicing virtual functions to separate the data traffic of different 5G use cases. SDN/NFV systems come with opportunities for video streaming by allowing to dynamically deploy, move and run video streaming functions such as video transcoding and data caching. At the same time, the routing capabilities of virtual switches are useful to allocate specific virtual paths to resource demanding streaming services and redirect video data in live to appropriate streaming functions [Boros et al., 2019].

4.2.2 Intra-building wireless systems

The increased use of Wireless Local Area Networks (WLAN) communications in buildings is causing congestion and interference, whilst modern building materials are restricting the propagation of Radio Frequency (RF) waves within them. Therefore, building owners have been increasingly turning to the deployment of cellular home networks (HeNBs) in their buildings because they operate in licensed spectrum that can avoid interference and congestion. Unfortunately these deployments require the permission of Mobile Network Operators (MNOs) due to their potential to interfere with the main transmitted signal from the main mobile network. However, MNOs have only had the capacity to analyze their largest customers' deployment requests, thereby losing a large market opportunity. To complicate matters further, each building requires a HeNB deployment for each MNO that is providing coverage for it, which is very costly and inconvenient for the building owner.

4.2.2.1 Internet of Radio Light (IoRL)

The IoRL project [IoRL, 2020] attempts to solve this problem by providing a broadband radio-light communications solution that operates in unlicensed millimeter wave and visible light spectra. Such solution does not suffer from interference because of the propagation characteristics of Electromagnetic waves in this part of the spectrum and provides universal broadband coverage within buildings from radio-light access points that are pervasively located within the buildings light roses. This technology can also be applied to other indoor environments such as Tube Stations, Underground Pedestrian tunnels, etc. The challenge that is addressed by IoRL project is to design Remote Radio Light Head (RRLH) electronics that can be elegantly integrated into the myriad of different types of electric LED lighting systems. The main benefit is the availability of broadband communications services greater than 10 Gbits/sec ubiquitously available throughout buildings, from pervasively located radio-light RRLH access points situated

within light roses. A further advantage is that user equipment can be located close to an accuracy of less than 10cm. Designing the radio-light communication system to fit into the confined space of a light rose requires a NFV solution, for which cloud computers can be variously located remote from the radio-light access points, in the Home Cell Site or in the external Cloud network. As well, a SDN is deployed to intelligently manage and route data to the different parts of the radio-light network. Consequential outcomes of this architecture are that its common building electric light network resources can be more easily shared between MNOs by slicing. Additionally, the NFV solution provides an API, which allows third-party service providers to write specialized network applications to manage multi-MNO networks in homes, businesses and public space buildings, as well as tunnels, train stations and airports.

IoRL project will also significantly benefit 5G MNOs by considerably reducing electromagnetic interference. Indeed, no more will be generated by Home eNodeBs, thereby increasing throughput in the wider 5G mobile network and improving mobile access to in-building users. This will consequently permit to increase the value of their customers' buildings. The net result is that there will be a considerable reduction of transmission power and Electro-Magnetic Fields radiation levels. The user equipment will potentially require consuming 10 times less energy, resulting in 90% energy savings, and a ten times increase in battery lifetime during use in buildings. The combined effects of a reduction in delay spread, due to smaller room geometries, the adapted 3GPP 5G approach and the considerable improvement in propagation delays, are expected to result in a reduction in latency to within 1ms. 5G mobile network users will significantly take advantage of it since they will have the choice of a wider range of network services from third-party network and home services providers. They could for instance provide substantially higher bitrates using RRLH in in-door environments, whilst also significantly reducing the level of EM exposure.

4.2.2.2 Use cases identified in IoRL

Various use cases have been identified in home buildings, supermarkets, museums and train stations (Figure 4.5). This subsection introduces and describes these use cases.

Residential buildings. Modern buildings are increasingly being constructed with materials that seriously inhibit the propagation of radio signals within a building, such as: foil back plasterboard, insulation board, multi-foil board, flexible foil insulation, low e-glass internal wall insulation, cavity wall insulation, pitched roof insulation and loft

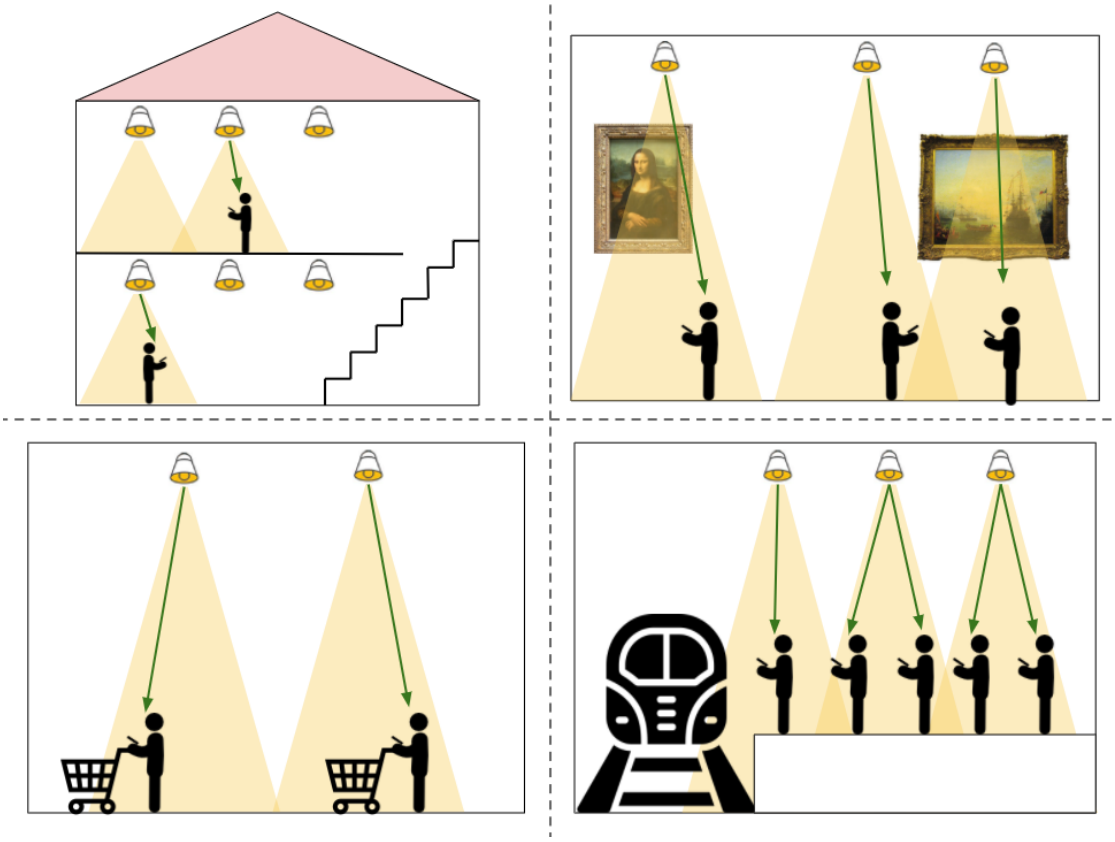

Figure 4.5: Use cases of IoRL: Home building, supermarket, museum and train station

insulation for heat insulation purposes. Therefore, mmWave and VLC at LED light access points in each room are attractive broadband communication solutions for the home. The Home use case scenario [Cosmas et al., 2018c] is situated in the BRE Smart Home Lab and considers the connectivity required at any place and at any time by humans and machines in home environments, including the transitions from indoor to outdoor environments and vice versa. The BRE Smart Home Lab showcases the latest technologies in Online Services, Office in the Home, Communications, Entertainment, Assisted Living and Tele-healthcare, and Smart Home. This is a commercially important use case because there are over 220 million households in Europe and the over-the-air TV transmission systems do not have sufficient bandwidths to deliver sufficient 4k/8k TVs channels. Therefore, 5G indoor technology can be the solution to provide this. The 21 billion Euros computer games market in the EU also makes this a very important use case because 5G can provide wireless VR headsets with lower latency, higher resolutions and higher location accuracy, providing gamers with a much higher quality of experience.

Business and supermarkets. Big shopping malls and grocery stores usually have LED lights on all day, thus providing an excellent environment for VLC application. There certainly exists the need to provide a better customer's shopping experience by using the location information of the shoppers as well as data-pushing functionality supported by VLC [Cosmas et al., 2017]. Price is less and less a competitive differentiator for retailers. Nowadays, selling services and good shopping experiences are the key points to attract and retain loyal customers. Because millions of people across Europe are turning to online shopping to order groceries direct to the home, retailers need to use 5G indoor technology to make shopping in supermarkets more than just an usual shopping experience. 5G services in supermarket include location-based data access, routing and monitoring to the nearest 10cm for shoppers using smartphones. Location-based data access could enable the delivery of product details instantly to customers when close to them. Location-based routing can be used to guide shoppers to their favorite items or products they may be interested in, through personalized profiling. Location-based monitoring can be exploited if customers are adhering to social distancing rules in a virus pandemic such as Covid-19. Video streaming could be used to display personalized advertisement, information on products or entertainment for children.

Public space museums. With international audiences of millions of visitors each year, museums are always seeking to find new and more efficient ways to engage visitors from around the world with their collection [Cosmas et al., 2018a]. The *Musée de la Carte à Jouer* in France has been chosen by the IoRL project to demonstrate the positive impact of indoor 5G solutions to deliver rich content. The museum consists of two buildings: the new one and the old one. The New Museum building is an excellent use case as it was built on the side of a hill and the basement areas suffered from an underground water course running through the building causing dampness. For this reason, the basement areas of this new building were encased in stainless steel containers to keep out the water; this also acts as a Faraday Cage that prohibits outside wireless signals from entering the building and vice versa. The museum regularly constructs new exhibits in the Exhibition Hall on different themes and it has to improve or reshape their permanent collections. 5G indoor technology is useful to provide accessibility, location based data access and high quality video delivery at peak hours. With efficient location, personalized video might be streamed to the users in order to provide more details about the collection or artifact they are looking at and thus assist the work of guides.

Public space tunnels. IoRL intends to demonstrate its radio-light technology in the Nuevos Ministerios Station in Madrid [Cosmas et al., 2018b]. The station corresponds to a large underground space that provides service to 6 lines of the regional railways. The railway station is also linked to 3 lines of the Metro system. This transportation hub connects the center of the city with representative locations such as the Madrid Adolfo Suarez Airport or the Puerta de Atocha High Speed Station. As a preliminary estimation, the whole system has more than 25 million users a year. The station also provides a place for other business and activities such as shops, cafes, restaurants and temporary exhibitions. The site represents a clear opportunity to test the results in an environment that comprises a transport infrastructure and public spaces in the same underground location. In addition, an underground station is a quite restrictive environment which implies a challenge, not only for the implementation of the new technology, but also for the large potential number of users. 5G services will include location based access and visualization of data to the nearest 10cm for maintenance staff using Augmented Reality glasses showing the presence of water, cracks, escape routes maintenance (cleaning), draining system maintenance, track geometry, tracks Kilometer Points and notice the presence of alien objects. It also includes video conference calls to maintenance staff and security guard first responders for any type of emergency incident (medical conditions of passengers, missing or suspect objects, etc.) that takes place at the station. This is an industrially important 5G use case as millions of people across Europe use public transport so safety and reliability of public underground transportation services are of paramount importance for its efficient functioning. Therefore, 5G indoor access technology is needed to provide accessibility, reliability and location based data and video calls are very important to happen with low latency and packet loss.

4.2.2.3 Architecture of the system built in IoRL

Radio Access Network. The goal of the IoRL project is the synthesis of license-free 60 GHz Radio Frequency (RF) and the Visible Light Communication (VLC) into one system. The symbiosis of the two technologies enables new application fields and the reuse of already existing infrastructure. However, to be accepted by end-users, the new system needs to offer new capabilities and applications, as well as a compatibility with existing system/standards like the IEEE 802.11, 802.15 and 3GPP. The IoRL Radio Access Network (RAN) is composed of multiple VLC and mmWave connected to Remote Radio Light Head Controllers (RRLH Controllers). An example of the IoRL RAN in home building is illustrated in Figure 4.6. Each room or floor area in a building can

be provisioned by a single RRLH Controller with its group of eight RRLHs and intra-building handover performed between these areas with the aid of VLC and mmWave location sensing application that continuously records the positions of UE in the building. The RRLH Controllers are connected to a local server called the Home IP Gateway (HIPG) via Plastic Optical Fiber (POF).

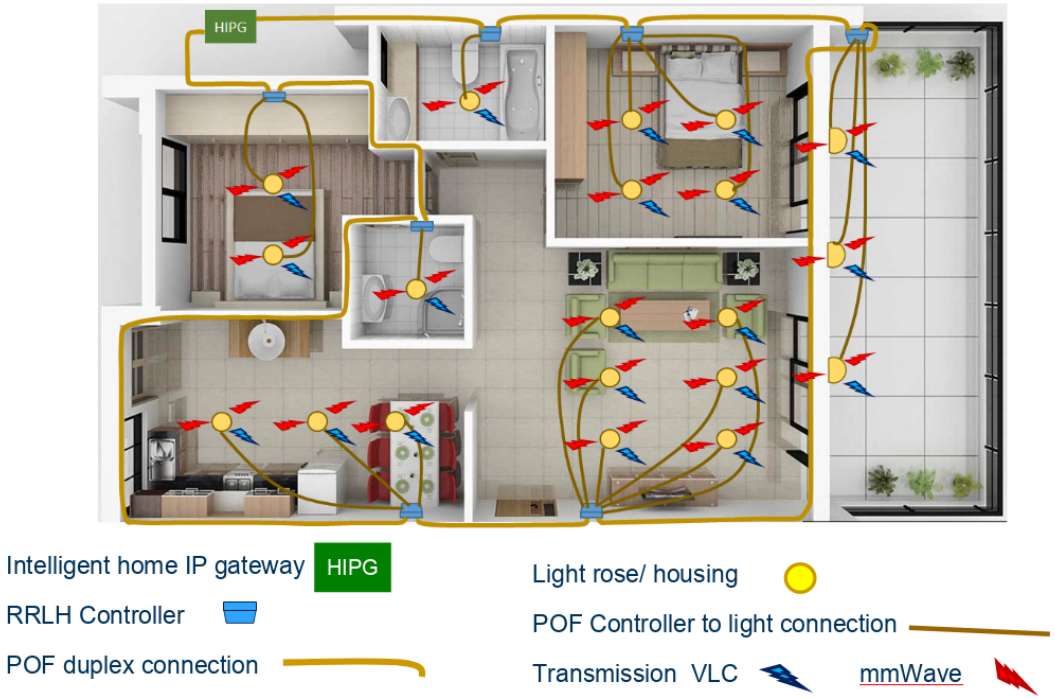

Figure 4.6: IoRL RAN in building

HIPG. Two possibilities exist concerning the Home Gateway and its services. First, a "dumb" bridge could be installed on the customer's premise, and all virtual Home Gateway (vHG) functionalities could be performed inside a remote cloud. The second option is that the vHG can be hosted fully on a server in the local network. This acts as the Home IP Gateway (HIPG). The proposed HIPG includes functionalities of today's Home Gateways and extends them with specific media-related functions, in order to instantiate a real multi-play device (e.g., multiple users, services, terminals, acting as streaming client and server), enhanced with cutting-edge technology and virtual capabilities. Additionally, HIPG offers context- and network-awareness for Media Services and provides inputs for enabling content-awareness. An example includes a packet inspection VNF that determines if the traffic embeds video data and sends it further to a

virtual transcoding unit VNF for quality adjustment. If data traffic is detected, packets are steered to a virtual security appliance acting as a virtual firewall, which sends them further to a virtual proxy VNF and a deep packet inspection VNF. Figure 4.7 shows the architecture of the functions deployed inside the HIPG inside the IoRL infrastructure. Security, location, transcoding and various VNF are connected to virtual switches. Data packets are routed to and from IoRL RAN on one side and the Internet on the other side.

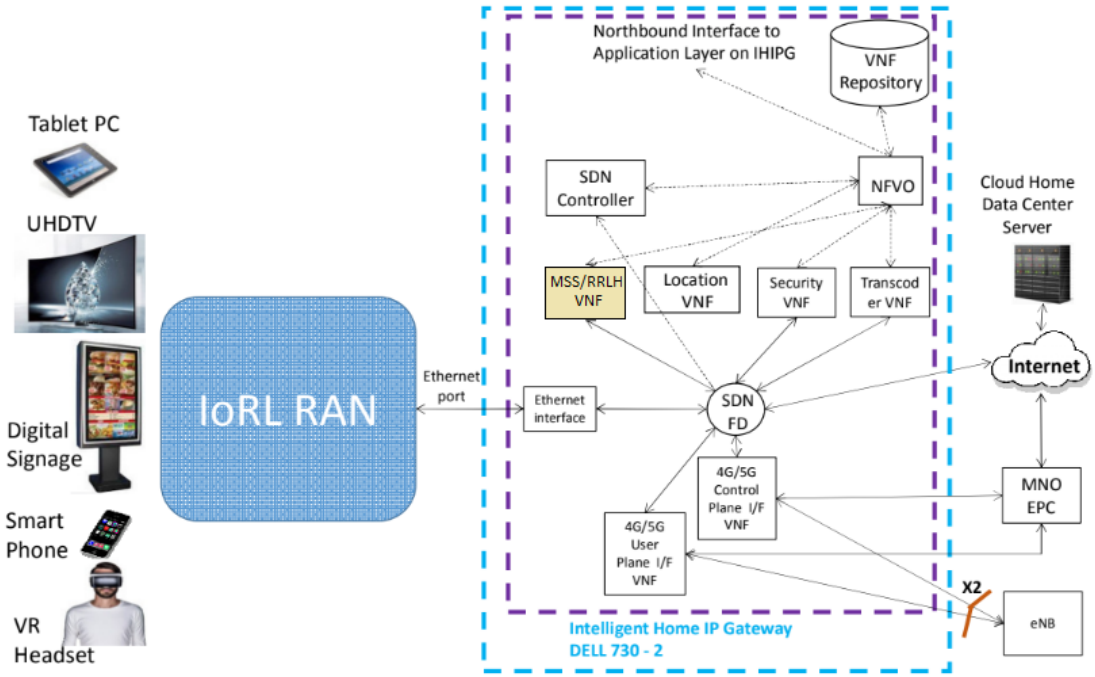

Figure 4.7: IoRL HIPG architecture

4.2.3 Video streaming solutions in 5G

High-quality video delivery in 5G networks is a trending topic in research as the consumer demands for video content keep getting higher [Cisco, 2018]. Future 5G networks should be able to keep with the emergence of new video formats requiring important bandwidth capabilities and offer pragmatic solutions to guarantee the QoE of the end-users. This section focuses on specific systems or problems emerging with future networks, new business, and 5G use cases.

4.2.3.1 Video streaming in future 5G networks.

Edge functionalities with SDN/NFV. Several research contributions discuss the challenges of future 5G networks in terms of media consumption. A lot of systems are developed using SDN and NFV capabilities. One trending topic is the use of edge servers for caching. The authors of [Ge et al., 2016] and [Munir et al., 2019] propose different systems to optimize video delivery by using servers and machines at the edge of 5G networks to be closer to the end-users. Some contributions go even further and consider the utilization of network side solution with SDN/NFV to assist the video delivery. In particular, [Boros et al., 2019] proposes to improve DASH by using real-time bandwidth information retrieved directly from the network and to enforce this information using SDN capabilities to route specific packets to specific network paths. The authors of [Nightingale et al., 2018] propose a framework running at the edge of the network to predict and evaluate the QoE of UHD streaming session. This kind of framework could be used to adapt the transcoding parameters in the edge server and improve the quality of video delivery. In the IoRL architecture, the HIPG can be used by design as an edge server. Moreover, by embedding a SDN/NFV system, this server is able to route video packets to specific virtual transcoding and caching functions.

Latency of live streaming. As low latency is a point of interest in 5G developments, video delivery systems are expected to have the same property. The video delay, or latency, in video streaming systems is defined as the duration between the moment when a frame is captured by a camera and the moment when the same frame is displayed on the user terminal. Although being able to deliver live streaming sessions, HAS solutions rely on segments and buffering to increase the QoE. Good practices are to maintain about 40 seconds of available buffer, and therefore the same level of delay, which is not always acceptable for live streaming use cases. Hence, today's video streaming systems are to be improved or at least adapted to fit with the low latency requirement.

The current solution in the industry to reduce live latency is to use chunk capabilities of Common Media Application Format (CMAF) containers [CMAF, 2018]. The main idea behind this proposition is to be able to deliver every video frame whenever they are ready, without having to wait for a full segment to be produced. The authors of [Bentaleb et al., 2019b] and [Bentaleb et al., 2020a] introduce a predictive algorithm to adapt the bitrate of low latency live streaming sessions according to the history of observed bandwidth. Despite showing very impressive results in terms of QoE and end-

to-end latency, this solution is not suited for multiple heterogeneous and potentially lossy networks.

Another well studied solution is to focus on HTTP extensions such as WebSockets. An HTTP/2 solution to improve low latency is proposed in [Yahia et al., 2019]. It relies on frame multiplexing and resetting capabilities of HTTP/2 in one TCP connection to select the appropriate quality on the fly. Another solution is discussed in [Wu et al., 2017] where HTTP/2 is used with WebSockets to implement network-assisted DASH. Such solutions need the adoption of server-side HTTP/2, and their performances are limited by design to one connection, thus impacting the reliability and QoE expectations compared to a multiple network paths one.

The last solution is to build a streaming system using WebRTC [WebRTC, 2015]. WebRTC has been introduced in section 4.1.1.2. Despite being usually associated with P2P scenarios, the API is also known to provide ultra-low latency sessions. WebRTC also comes with DataChannels to send raw data and extend its own capabilities. The authors of [Zhao et al., 2016] proposed a push-based algorithm derived from DASH using the above mentioned DataChannels along with WebSocket for the signaling messages. The solution achieves very good results in terms of latency by taking advantage of the direct connection and the use of RTP/UDP. However, WebRTC-based systems are optimizing speed over bandwidth capacity and, similar to the other discussed solutions, are only single source and not adapted to multi-path with lossy network connections.

4.2.4 Conclusion

To achieve 5G requirements in terms of media delivery, the IoRL project proposes a completely new way of distributing media content, video in particular, to all devices. Existing standards, such as MPEG-DASH or Apple's HLS, focus on the *1 server, 1 client* paradigm for a dedicated video and try to solve networking or client's problems by lowering the delivered video quality. Unlike them, the IoRL solution considers several servers and stream simultaneously through different paths. This new solution, called MSS/RRLH (Multiple-Source Streaming over Remote Radio Light Head), is presented in Chapter 6. It takes into account all devices capabilities and different networks paths to benefit from all the potential content sources. Each stream is independent and could be decoded separately, while remaining aggregatable with other streams in order to produce a better video quality. This new solution is particularly well suited for the IoRL concept, which considers multiple network technologies with totally distinct capabilities, namely: mmWave, VLC, WLAN, Mobile (4G/5G).

MSS/RRLH accommodates totally to each network's characteristics by sending a specific video stream through each of them, adjusted to the available bandwidth and delay on the end-to-end path for the required video. Hence, when a video is requested by a client, it is concurrently retrieved from several paths, which creates independent and aggregatable substreams based on the available resources on each link. The client will receive all substreams and merge them in order to get the best possible video quality. For example, a smartphone client may receive substreams from 2 different paths (1 Visible Light Coms, and 1 WLAN).

Last but not least, MSS/RRLH intends to go further than the already well-developed MS-Stream, introduced in section 1.5, to be able to use the SDN/NFV system deployed in the IoRL HIPG at its full potential. This novel version of the *n sources, 1 client* paradigm extends the concept of *description-as-1-segment* to *description-as-multiple-segments* and proposes new adaptation and path selection algorithms to improve video streaming sessions QoE and reduce latency.

Chapter summary

Hybrid P2P/CDN systems is an alternative to solve the server-side overload issue. Despite being a well studied topic, bitrate adaptation mechanisms in this context are still sub-optimal.
In order to extend the access to the 5th generation of networks, the IoRL project proposes solutions to offer 5G connectivity inside buildings. The considered architecture calls for innovative multi-path video streaming solutions to deliver high quality media through the complex and heterogeneous networks composing the system.

Chapter 5

PMS+: a pragmatic collaborative multi-source P2P streaming system to improve QoE and scalability

> "We are all family now. One people, one cause. You can stop this heart, bleed this old body, but you cannot end the Family."
>
> Grace Holloway

5.1 Introduction

Although the usage of a multi-source streaming protocol and CDN has the potential to reach high end-users' QoE, the scale of the considered infrastructure (and consequently, its total upload throughput capacity) is limited and costly. The P2P paradigm for video streaming permits to leverage the cooperation of peers, allowing to serve every video request with increased scalability and reduced costs. Hence, the contribution of each consuming client in transferring its downloaded segments to neighboring peers is a strong asset to further enhance the system's scalability at the best possible QoE.

We propose to combine the approaches of MS-Stream with the P2P paradigm in order to present a hybrid P2P/Multi-Server solution for live streaming gathering scalability and quality adaptation capabilities. Indeed, we wish to benefit from the QoE and scalability potential of both P2P and multi-source streaming approaches. We assume

the scenario of a video streaming provider delivering a single video content available in different bitrates and consumed by a population of peers. The provider's objective is to deliver the highest possible QoE by taking into account the heterogeneous and volatile connectivity capacity of consuming peers as well as the limited upload throughput capacity of the fixed-size CDN infrastructure. The contributions have been developed in two steps. The two resulting systems and their architectures are presented along with the streaming signaling, the peer selection and the proposed P2P network impairment resiliency mechanisms.

In the first system, called "P2P/Multi-Server" (PMS), peers are placed in mesh-based overlays, which are identified by the retrieved content quality, hence the one that can be re-emitted to neighbors. Each peer requests a part of the GoPs composing the current video segment from other peers in the P2P application group it belongs to —when available— and the missing GoPs from the server infrastructure. Additionally, we designed a distributed quality adaptation algorithm relying upon local and global indicators of the PMS functioning to control the transitions of peers from one overlay to another. This new quality adaptation strategy strives to enhance the end-users QoE while concurrently aiming at the successful functioning of all P2P overlays. To that end, the quality adaptation running at peer site takes into account the current network conditions, the capacity and efficiency of the P2P overlays, the peer's device resources and its contribution to the good functioning of overlays.

The second system, named PMS+, extends PMS by optimizing the data exchange between peers and accelerating the quality adaptation mechanism. Peers are placed in small groups belonging to an overlay. Inside a group, peers are self-organized and cooperate to exchange a maximum of video data and only download what is necessary from the servers. The peers sort themselves according to their position in the playback. The first ones download the video segments from the server while the others try to retrieve data from their predecessors. With the help of an effective data exchange protocol, the quality adaptation mechanism can ease the stabilization process and increase the global QoE of the solution.

PMS and PMS+ were evaluated in a large scale experiment. Metrics were collected during several months from a real production environment deployed in France in partnership with a public webcam provider. More than 10 000 anonymous users per day were watching live streams using PMS and PMS+ video players. The results highlight that our contribution shows significant improvements against standard HAS systems as

well as state-of-the-art P2P solutions in terms of QoE and P2P efficiency in a large scale deployment.

A first section of this chapter presents the architecture and main mechanisms of the first version of our P2P system: PMS. A second section explains the new architecture and the modified concepts of PMS+, the second version of the P2P system. PMS+ offers some improvements against PMS in terms of P2P delivery and quality adaptation. The third section is about the large scale evaluation. After a presentation of the experimental platform, the results obtains for PMS, PMS+ and state-of-the-art players are analyzed and discussed highlighting the benefits of our contributions.

5.2 First step towards a pragmatic P2P streaming system: PMS

In this section, we first present the hybrid streaming architecture of PMS before detailing the streaming signaling, the peer selection and the proposed P2P network impairment resiliency mechanisms.

5.2.1 System specifications and architecture

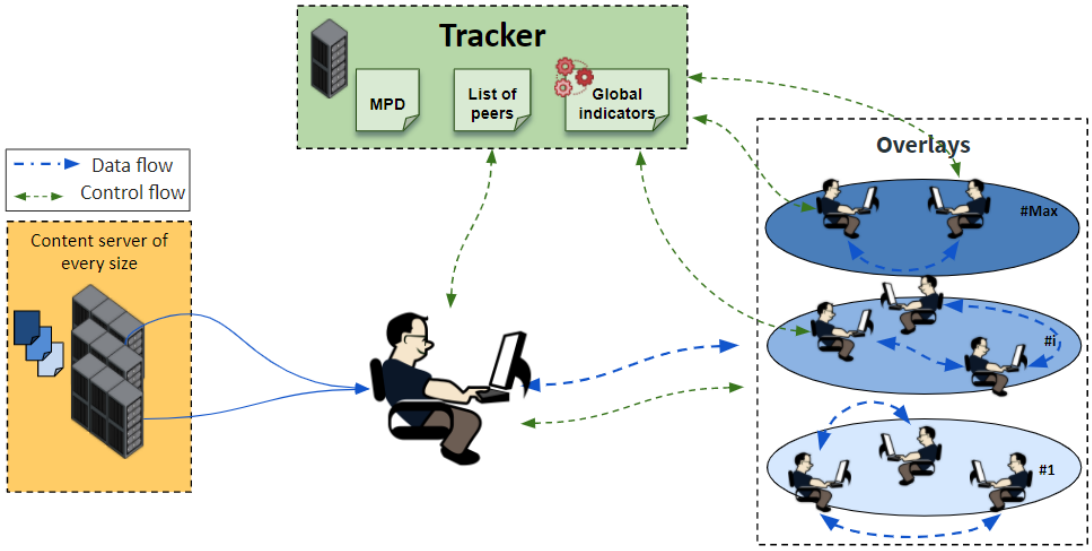

Figure 5.1: Architecture of PMS

As shown in Figure 5.1, the PMS system is composed of three major components. The first one is the tracker being in charge of delivering MPD files that list available

servers hosting the content in various qualities. It also acts as rendez-vous points for all clients to obtain lists of candidate neighbors and to notify themselves as available resources for other peers. Additionally, the tracker is periodically computing global indicators on the current health of each overlay, based on metrics reported by the peers. The latter indicators are forwarded to the peers and are then used in the decentralized quality and scalability adaptation mechanisms. The second major component is a group of content servers that are dynamically provisioned with the live video flow at multiple qualities. The third major component is a set of peers, which are organized in Q_{max} application-layer mesh-based overlays implementing a pull-based content consumption protocol, obtaining video data from storage servers and engaging in P2P data transfers. Within each distinct overlay i composed of $N_i(t)$ peers, the i-th quality (i.e., bitrate b_i) is actively being consumed and re-emitted by peers. At any given moment in time, a peer is part of one overlay only and can contribute to the streaming process up to its maximum upload throughput capacity. We rely on a multiple-overlay architecture in order to better understand and design the proposed quality and scalability adaptation algorithms. Ultimately, this multiple-overlay architecture does not overburden the classical design of application P2P networks.

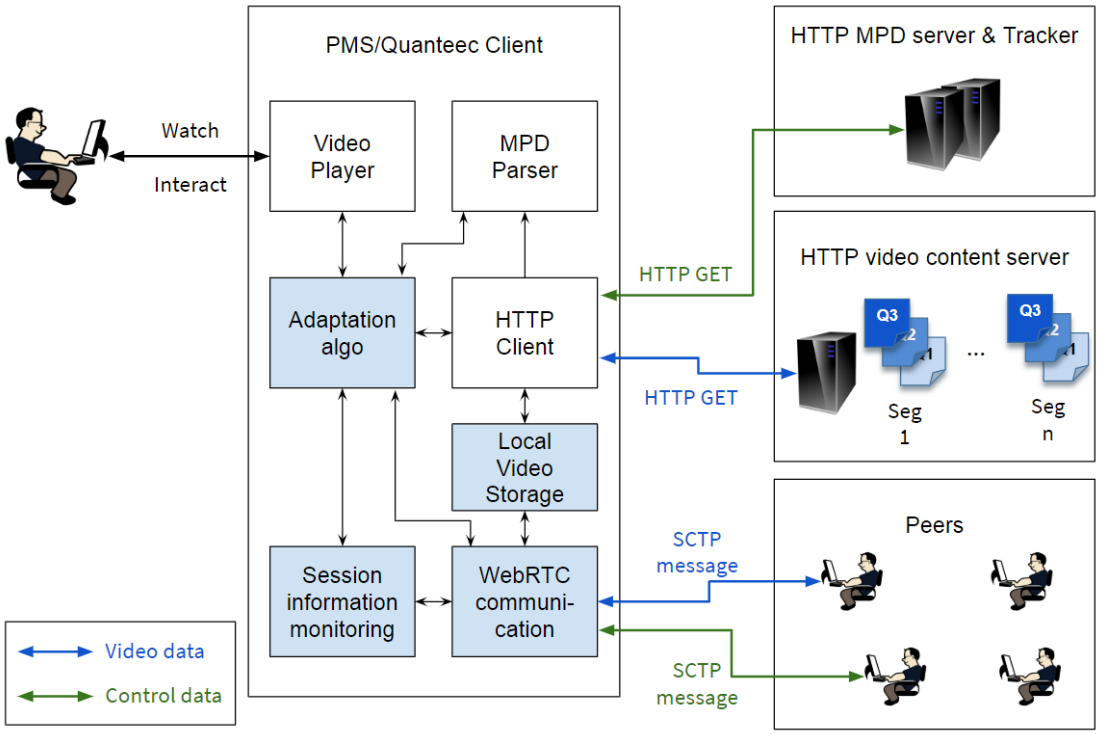

Figure 5.2: Peer's software architecture

The peer's functional software architecture is depicted in Figure 5.2. The additional modules compared to the previously proposed MS-Stream client/server architecture are highlighted in plain blue. The **adaptation algorithms** are modified in order to use the peers. A **local video storage** is used to save the requested segments. Once saved, the segments can be sent to other peers. The **session information monitoring** block manage the information received from the peers, like the video data available and their upload bandwidth. The HTTP Client is still exploited to retrieve data from the tracker and the video content servers, but the P2P messages are transmitted by the **WebRTC communication** module. This module uses WebRTC data channels to exchange packets with the other peers.

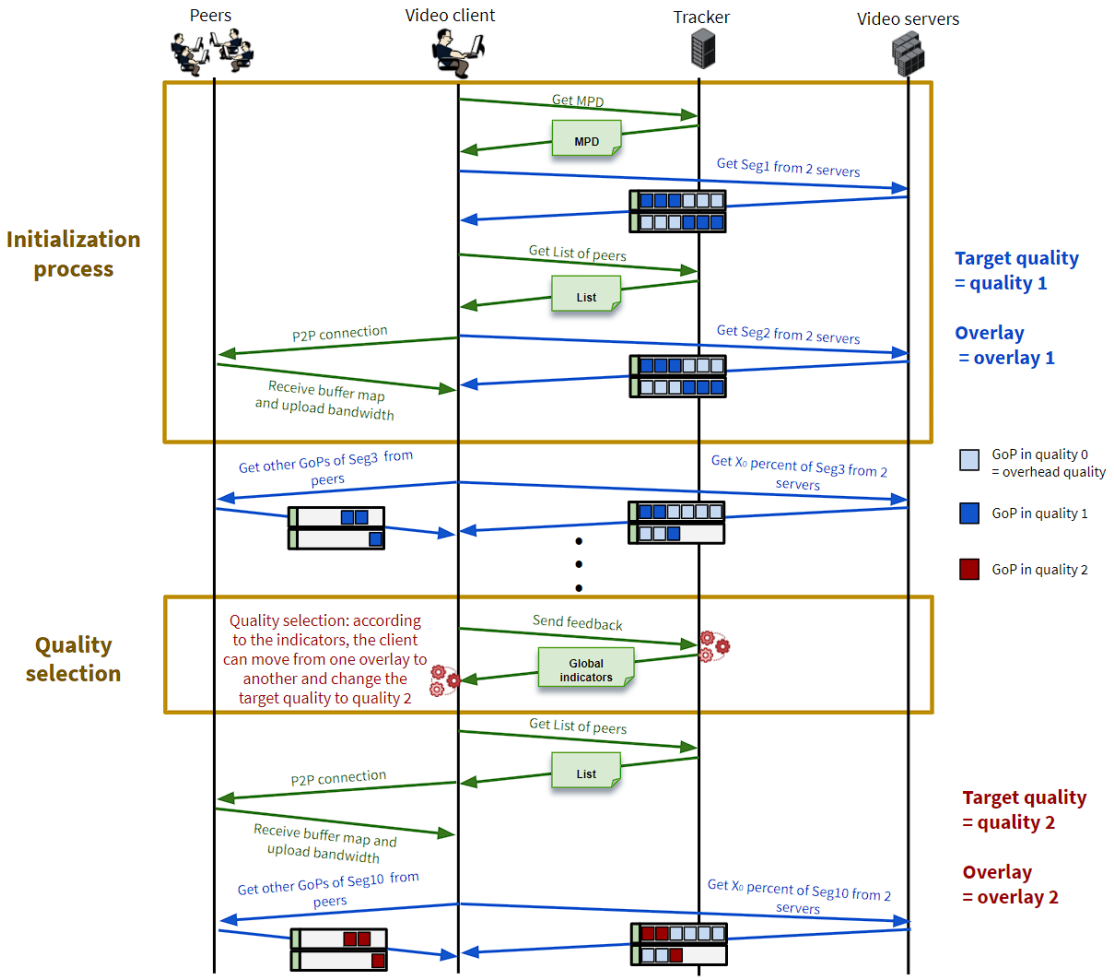

Figure 5.3: Sequence diagram of data exchange in PMS

A sequence diagram describing the main process of PMS is provided in Figure 5.3. The next paragraphs describe the concepts introduced in this diagram.

Streaming signaling. The signaling procedures between a new peer j, the servers, and the remote peers are depicted by the green arrows in Figure 5.1 and Figure 5.3. When a new client pops in, it proactively and periodically requests the Tracker for MPD files listing the nearest servers. The peer registers its ID and notifies itself as an available resource for the overlay it currently belongs to. The peer periodically requests the trackers to obtain a list of K peers in its overlay selected randomly. The peer regularly asks its K neighbors for two key information items used for the peer selection and for the adaptation algorithms controlling the quality and the usage of the server infrastructure: (1) their buffer maps listing the available segments and GoPs cached in the peers' local storage and (2) their estimated upload rates $\alpha_k, \Delta(t) k \in [1..K]$ computed by the peers themselves. Every peer j computes its upload rate $\alpha_j, \Delta(t)$ by taking the highest value between the mean throughput resulting from the number of packets transmitted to remote peers within the last Δ seconds and the highest delivery rate r_j^{max}, $\delta(t)$ observed for one packet within Δ seconds. Moreover, the peer j repeatedly sends keep-alive messages to the trackers so as to maintain its visibility among the current overlay and simultaneously reports several metrics: its upload rate $\alpha_j, \Delta(t)$, the actual amount of data $a_j, \Delta(t)$ it delivered within the last Δ seconds, as well as the amount of data $\Phi_j, \Delta(t)$ it received from the servers. In this way, the tracker follows the evolution of each overlay's population $N_i(t)$ and computes global indicators on each overlay's functioning.

Initialization process and fast start. The sequence diagram of the initialization process is depicted in the first half of in Figure 5.3. In order to minimize the initial video playback delay and enhance the end-user's QoE, without suffering from the delay induced by the communications with the tracker and the neighboring peers, the client relies on servers for the first video segments to be downloaded. At the beginning, a peer is placed in the overlay 1. The client requests two sub-segments with an equal number of GoPs at the bitrate b1 (i.e., the second lowest bitrate listed in the MPD file) and the other GoP at the lowest bitrate from two servers randomly selected in the list of servers listed in the MPD file. At the same time, the video client receives the list of peers from the tracker and initiates P2P connections.

Video data delivery after the initialization Once the P2P communications are established, the initialization process is over. The peer now simultaneously requests the servers to deliver a predetermined percentage X_0 of the GoP composing the video segments to be downloaded at the video bitrate b_i. This video bitrate is the target bitrate and is defined by the overlay in which is the peer. Every peer requests random GoPs at the target bitrate from the servers. For example, a peer 1 may receive GoPs 1, 2, and 3 when a peer 2 may get GoPs 4, 5, and 6. The other GoPs are received at the lowest bitrate. Then, the peers are able to exchange their high-quality GoP in P2P. At the end, both peer 1 and peer 2 have received the GoPs 1, 2, 3, 4, 5, and 6 at the bitrate b_i. The value of X_0 is an essential parameter that determines the system's scalability and its QoE capabilities. Video clients periodically send feedback to the tracker and receive global indicators containing information about the health of the system. With the help of these indicators, video players are able to move (or not) from one overlay to another and modify the requested target quality. More details about the quality selection algorithm are going to be provided in subsection 5.2.2.

Peer selection. Periodically, every peer asks the trackers for a list of K neighbors in order to obtain from them their buffer maps and upload rates. These neighboring peers are then requested to deliver the remaining percentage $1 - X_0$ of GoPs —the one that is not delivered from the content servers. Due to the high heterogeneity of peers' upload capacity, it is essential not to rapidly choke the low upload rate peers or under-utilize the ones with high upload rates. Choking low upload rate peers will lead to late request delivery for the neighboring peers using them, resulting in degraded QoE for many end-users and a handicap for the global video delivery system performance. Oppositely, leaving the high upload rate peers poorly used or inactive will induce a poor utilization of the available resources and greater operating costs supported at the server infrastructure level. Instead, every peer advocates for a fair usage of the available resources and accordingly locally optimizes the usage of neighboring peers' upload rate capacity for each segment delivery. Ideally, the greater the upload throughput of a neighboring peer k, the higher its chances to be selected for the delivery of GoP c. Consequently, the assignment process follows a discrete random variable, and we set the probability $p_{c,k}$ of peer k to be assigned the delivery of GoP c as:

$$p_{c,k} = \frac{\delta_{c,k}.\alpha_{k,\Delta}(t)}{\sum_{l=1}^{K} \delta_{c,l}.\alpha_{l,\Delta}(t)} \tag{5.1}$$

where $\delta_{c,l} = 1$ if peer l has previously cached the chunk c in its local storage and $\delta_{c,l} = 0$ otherwise. As $\sum_{k=1}^{K} p_{c,k} = 1$, the assignment of each chunk c is ensured.

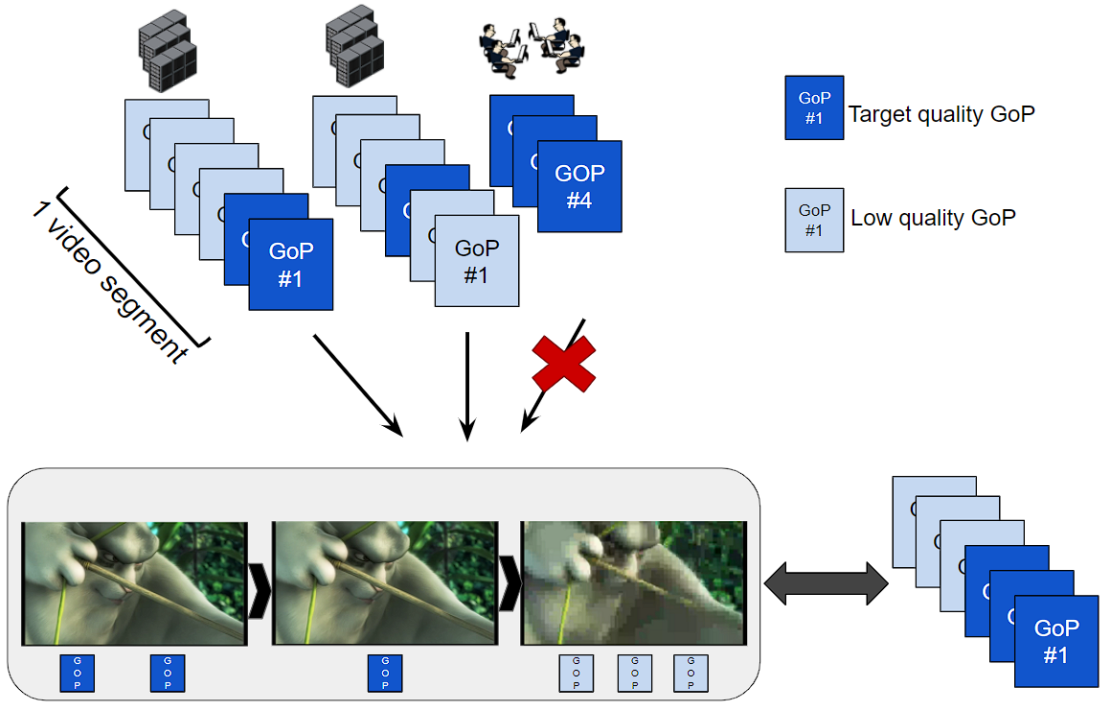

Figure 5.4: Impact of P2P network impairments and peer churn on video quality in PMS

Resiliency to P2P network impairments and peer churn. During the delivery of sub-segments from remote peers, unexpected events may occur, such as suddenly low performing communication channels, as well as remote peers becoming unavailable because they are no longer in the current overlay or because they disconnect from the streaming session. Losing some GoPs may force the video player to display some video frames in a sub-optimal quality, as depicted in Figure 5.4. In PMS, the client consumption protocol incorporates in-segment download adaptation rules to avoid streaming session disruption. To that end, three client-side in-segment download adaptation rules have been designed for P2P data exchange. The first two rules permit to handover the delivery of GoPs to the content servers due to low-performing remote peers, and to avoid a too fast buffer depletion that could eventually lead to video glitches. The third rule allows to display a temporarily low visual quality by canceling the sub-segment from the low-performing peers and relying on the GoPs at the redundant bitrate already received.

The three rules are illustrated in Figure 5.5. They consist in:

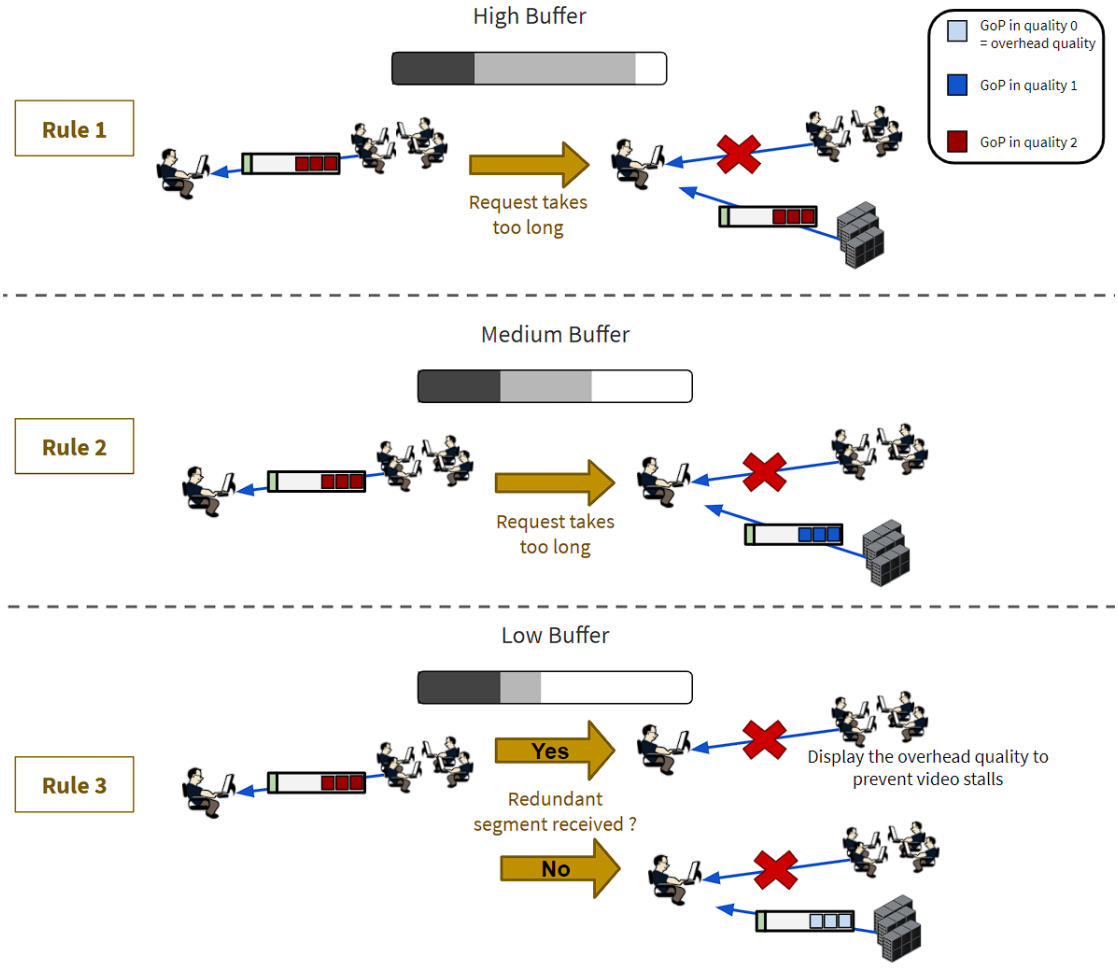

Figure 5.5: In-segment download adaptation rules for P2P communications

1. **Rule 1.** The first rule is activated when the buffer level is high (i.e. above a threshold t1). If a utilized remote peer becomes unavailable or if the video client estimate that the download will take too much time, then the P2P request is stopped. In the meantime, the client re-assigns the not yet delivered GoPs from this peer to the servers at the target video quality.

2. **Rule 2.** The second rule is activated when the buffer level is medium (i.e. between t1 and t2 with $t1 < t2$). If a utilized remote peer becomes unavailable and if the video client estimate that the download will take too much time, then the P2P request is stopped and the client re-assigns the not yet delivered GoPs from this peer to the servers. This time, the GoPs are not requested at the target quality, but at the best quality that can be downloaded in a fixed period of time. By doing so,

the request will be faster and the playback will not be interrupted while displaying a slightly lower quality.

3. **Rule 3.** The third rule is activated when the buffer level is low (i.e. below a threshold t3 with t3 < t2 < t1). If the client has already received some GoPs at the overhead bitrate that can replace the missing GoPs from the P2P sub-segment request at the desired bitrate, then the client can abandon the current P2P download if it takes too long, merges the available sub-segments and displays content to the end-user. This last resort reactive rule attempts to avoid interrupting the video playback by displaying a temporary sub-optimal visual quality to the end-users. If the overhead GoPs are not available, then the client re-assigns the not yet delivered GoPs from this peer to the servers at the lowest quality to get them as fast as possible.

5.2.2 Quality selection algorithm

In most of the current adaptive streaming solutions over HTTP, the client-centric decisions to adjust the desired quality are based on observable local parameters (observed throughput, buffer depletion speed, etc.). Hence, the client selfishly attempts to maximize its QoE by consuming the network resources made available for its streaming session. Considering the heterogeneous and highly unreliable nature of the P2P components in the PMS system, consuming peers can no longer perform quality adaptation based on local parameters only. Indeed, such consumption behaviors could lead to resources starvation (namely the upload throughput) first in the P2P overlays and second in the server infrastructure. We propose a distributed quality adaptation algorithm running at peer site, where each peer has to be fair to all the others by moving upwards or downwards in the overlay system according to both local and global indicators on the system functioning. Table 5.1 references the indicators and variables used in the PMS system for quality adaptation.

Two global indicators concluding on the system's health are computed by the trackers based on the metrics reported by the peers within the last τ seconds: the overlay's capacity index [Wu et al., 2009] $\kappa_{i,\tau}(t)$ to deliver the consumed quality at bitrate b_i and the system delivery efficiency $\eta_{i,\tau}(t)$ for each content quality i. The two global indicators are computed as follows:

$$\kappa_{i,\tau}(t) = \frac{\sum_{j=1}^{N(t)} \gamma_{i,j}(t).\alpha_{j,\tau}(t)}{N_i(t).b_i} \tag{5.2}$$

Table 5.1: Variables used for quality adaptation in PMS

Symbol	Description
$\lambda_{j,n}^{server}$	*Local server throughput:* Network throughput from the server infrastructure estimated by peer j
$\lambda_{j,n}^{P2P}$	*Local P2P throughout:* Network throughput from the neighboring peers estimated by peer j
$\kappa_{i,\tau}(t)$	Overlay capacity index to deliver the consumed quality at bitrate b_i during τ
$\eta_{i,\tau}(t)$	Actual delivery efficiency of quality i during τ
η_{thresh}	Efficiency threshold
$\alpha_{j,\Delta}(t)$	Upload rate computed by peer j during Δ
$rmax_{j,\Delta}(t)$	Highest delivery rate observed from one packet during Δ
$a_{j,\Delta}(t)$	Data delivered by peer j to remote peers during Δ
$\Phi_{j,\Delta}(t)$	Data delivered from the server infrastructure to peer j during Δ
X_0	Percentage of data initially requested by each peer from the server infrastructure for every video segment
O_{max}	Percentage of overhead requested from the server infrastructure for every video segment

$$\eta_{i,\tau}(t) = \frac{\sum_{j=1}^{N(t)} \gamma_{i,j}(t).(a_{j,\tau}(t) + \Phi_{j,\tau}(t))}{N_i(t).b_i.\tau} \tag{5.3}$$

with $\gamma_{i,j}(t) = 1$ if peer j is in overlay i at time t, and $\gamma_{i,j}(t) = 0$ otherwise. The overlay's capacity index is a high level indicator of the overlay's capacity to deliver content at bitrate b_i to its demanding peers. It represents the ratio of the achievable upload throughput of the peers composing the overlay to the global throughput demand of the peers for the quality consumed. When $\kappa_{i,\tau}(t)$ is greater than $1 - X_0$, the overlay i is supposedly capable of providing enough throughput for all its peers. If $\kappa_{i,\tau}(t)$ falls below $1 - X_0$, the overlay attains a critical regime.

The delivery efficiency $\eta_{i,\tau}(t)$ is a more precise indicator of the entire system's functioning when compared to the overlay's capacity. Indeed, it represents the ratio of the delivered data from both remote peers and servers to the demanded data, computed every τ second, for the last τ seconds. The overlay's capacity shows the potential of the overlay to sustain a global throughput to the peers composing it, whereas the delivery

efficiency captures the state of the video quality delivery within the overlay. The closer $\eta_{i,\tau}(t)$ is to 1, the better the delivery of quality i is carried out by the entire system.

Moreover, every peer j monitors respectively the estimated download throughput $\lambda_{j,n}^{P2P}$ and $\lambda_{j,n}^{server}$ by adding up the estimated throughputs on each peer-to-peer and client-server communication channels. These local indicators reflect the P2P streaming system's and the server system's throughput performance according to the peer's point of view and are respectively referred to as local P2P throughput and local server throughput.

The quality adaptation process allows the peers to move from one overlay to an adjacent one only, aiming at minimizing the negative effects of high quality variation amplitudes on QoE, as reported by Yitong et al. [2013a]. During the download of segment n, the peer j located in the overlay i retrieves the global indicators $\eta_{i,\tau}(t)$, $\kappa_{i,\tau}(t)$, $\eta_{i+1,\tau}(t)$ and $\kappa_{i+1,\tau}(t)$ of both the current and the above overlays. At the end of the download of every segment n, the peer runs the quality adaptation process detailed in Algorithm 1.

Algorithm 1 leads the peer's movements among overlays by having it preserve the overlays' capacity and the delivery efficiency of each quality according to the predefined and enforced utilization of the server infrastructure (i.e. X_0). Downward movements in the overlay architecture are aggressively influenced by local indicators. Indeed, such indicators provide the peer with a local view of the P2P system's behavior toward its streaming session, and let it know whether it should keep or downgrade the requested quality in order to avoid buffer starvation and to maintain satisfying QoE. Alternatively, upward-movement decisions are conservatively influenced by both the peer's local indicators and the global indicators of the current and target overlays. The global indicators inform the peer whether an upgrade decision in the requested quality is harmless for the functioning of the current and target overlays, while the local indicators permit the peer to understand whether the actual content servers can sustain the required throughput for the delivery of the quality $i + 1$.

First, the peer verifies whether its local view of P2P system's throughput $\lambda_{j,n}^{P2P}$ and its local view of server system's throughput capacity $\lambda_{j,n}^{server}$ can respectively sustain its demand for P2P throughput in the current overlay and its demand for server throughput (line 1 of Algorithm 1). In this case, the peer may have reached either its maximum download capacity or the maximum throughput capacity of server and neighboring peers. Consequently, the peer is allowed to stay within the current overlay i.

In Algorithm 1, the demand for server throughput is $(X_0.b_i + \delta_{i,max})$, where $\delta_{i,max}$ is the maximum allowed extra throughput that can be utilized for bitrate b_i:

Algorithm 1 PMS content quality adaptation

Inputs: $\lambda_{j,n}^{server}$, $\lambda_{j,n}^{P2P}$, O_{max}, $\alpha_{j,\Delta}(t)$, $\eta_{i+1,\tau}(t)$, η_{thresh}, $\kappa_{i,\tau}(t)$, $\kappa_{i+1,\tau}(t)$, $\delta_{i,max}$, $\delta_{i+1,max}$
Output: overlay migration decision
if $(1 - X_0).b_i < \lambda_{j,n}^{P2P} \leq (1 - X_0).b_{i+1}$ **and** $X_0.b_i.\delta_{i,max} < \lambda_{j,n}^{server} \leq X_0.b_{i+1}.\delta_{i+1,max}$ **then**
  Stay in overlay i
else if $\lambda_{j,n}^{P2P} > (1 - X_0).b_{i+1}$ **or** $\lambda_{j,n}^{server} > X_0.b_{i+1}.\delta_{i+1,max}$ **then**
  if $\kappa_{i,\tau}(t) \leq 1 - X_0$ **and** $\alpha_{j,\Delta}(t) \geq (1 - X_0).b_i$ **then**
    Stay in overlay i
  else if $(\alpha_{j,\Delta}(t) \geq (1 - X_0).b_{i+1})$ **or** $(\kappa_{i+1,\tau}(t) > 1 - X_0$ **and** $\eta_{i+1,\tau}(t) \geq \eta_{thresh})$ **then**
    Upgrade to overlay $i + 1$
  else
    Stay in overlay i
  end if
else
  Downgrade to overlay $i - 1$
end if

$$\delta_{i,max} = \frac{O_{max}.b_i}{1 - O_{max}} \tag{5.4}$$

In the case where the local P2P throughput or the local server throughput are high enough for the peer to reach a better video quality (line 3), the peer first checks whether it can move to the upper overlay safely without harming the overlays' health and their global indicators. Although the peer may have the download capacities to move upward, Algorithm 1 enforces the peer to remain at its current quality i if the current overlay's capacity is critically low (i.e.,$\kappa_{i,\tau}(t) \leq 1 - X_0$) and if the peer's upload capacity contributes significantly to the overlay (i.e., $\alpha_{j,\Delta}(t) \geq (1 - X_0).b_i$ at line 4). If not, in case the peer can significantly contribute to P2P transfer or the overlay's capacity is high enough and the $i + 1$ quality delivery efficiency is greater than a threshold η_{thresh} (line 6), then the peer joins the upper overlay. Finally, when the requested quality can neither be maintained nor upgraded, the peer moves down to overlay $i - 1$ (line 12).

5.3 Second step towards a pragmatic P2P system: PMS+

Despite being innovative and improving current quality adaptation algorithms of P2P systems, several drawbacks have been identified in PMS.

Firstly, the decentralized quality adaptation algorithm adds a lot of complexity into the tracker and requires a lot of computing resources. This can be an obstacle to **scalability** because if many users join the system, the tracker might be easily overloaded and may answer with a delay. This situation is not acceptable because the quality selection mechanism would be delayed too and the peers would be locked in an inappropriate overlay, hence increasing the chance of video stalls.

Secondly, the **quality adaptation** mechanism does not perform fast enough. The client always begins in the lowest quality and needs several seconds to gather metrics and feedback before considering upgrading it. Moreover, this upgrade can only be performed one quality at a time besides not being guaranteed since relying on the health of each overlay. Thus, the peers with a good throughput may be blocked during several minutes in low-quality while they should be able, in principle, to download the best quality —situation unlikely to happen in a CDN-only scenario. As well, since the bitrate downgrade is also performed one quality at a time, a client with an unexpected bandwidth loss may be blocked in a higher overlay and may experience stalling events.

Thirdly, the **P2P efficiency** is not optimal. Every peer in PMS requests **random** GoPs from the servers. For example, a peer 1 can request GOPs 1, 2, 3 and a peer 2 can request GoPs 3, 5, 6. In this situation, peer 1 and peer 2 can exchange GoPs 1, 2, 5, and 6. However, the GoP 4 is not possessed by any peer and will be downloaded from the servers. If a lot of peers are consuming the same stream from the same overlay, the chances to have all the GoPs available to be exchanged in P2P are high. But if only a few peers are watching the same content, some GoPs may not be ready to be exchanged in P2P and the peers may have to request more GoPs from the server, reducing the P2P efficiency.

In order to solve these issues, we propose an improved hybrid CDN/P2P system called PMS+, acting as the evolution of PMS. The goal of PMS+ is to increase the scalability and to speed up the quality selection mechanism while maintaining a high quality of experience and retrieving a maximum of video data from the other peers.

5.3.1 System specifications and architecture

As shown in Figure 5.6, the PMS+ system is composed of the same three major components than PMS: (1) a tracker; (2) content servers; (3) peers. However, one difference is that the peers are not just sorted into overlays but are placed in small groups. The idea behind this group-based approach is to create microcosms in which peers can communi-

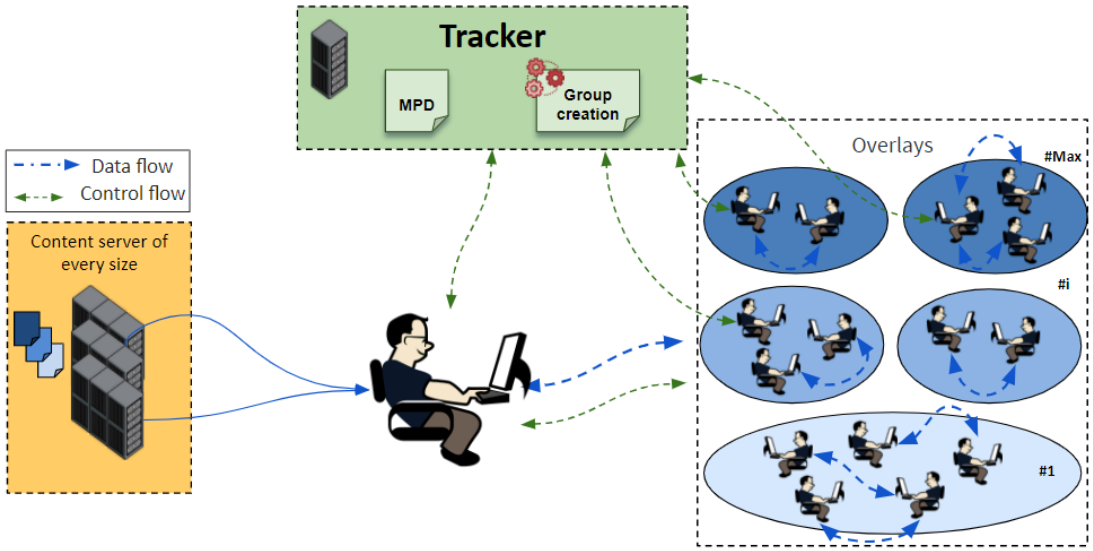

Figure 5.6: Architecture of PMS+

cate with their neighbors and can coordinate their efforts to optimize the P2P delivery. The main goal of this communication and coordination process is to permit a lot of decentralized communication between peers and avoid centralized feedback from the clients to the tracker, hence alleviating tracker's work.

The idea of the group-based P2P approach is not new. The authors of [Chun et al., 2012] propose an architecture involving groups of peers to deliver video content. Their solution relies on the election of a leader on every group. The leaders are able to communicate with each other in a tree-based architecture. Once a leader receives a video segment, the data are exchanged with both the peers composing the group and the leader of another group. Such group-based system has not been implemented and is only evaluated through a theoretical analysis in [Chun et al., 2012]. Despite being interesting, the idea to select a leader and to use basic tree-based architecture might not be optimal in a real video streaming environment because the system can be put in a very difficult position when a leader does not have a good upload bandwidth and leaves the stream.

The groups proposed in PMS+ are independent and does not interact with each other. As illustrated in Figure 5.7, peers inside a group are forming a complete graph by being connected to every other peers. They can then reliably exchange information about their download throughput, upload throughput, and buffer map. Sending all of this information is made possible by the small number of peers in one group that eases the propagation of the control data. With the help of this local coordination, peers do

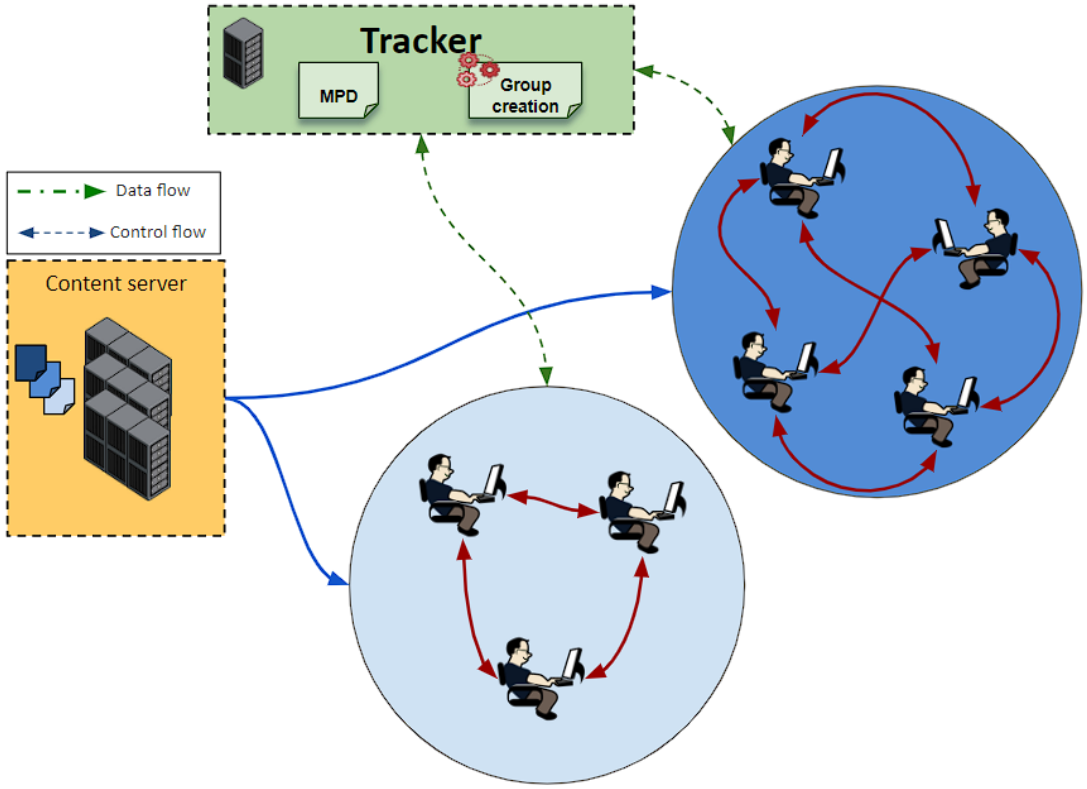

Figure 5.7: Groups forming complete graphs between peers

not need to report metrics to a single tracker every few seconds. The transmission of metrics is now fully decentralized and can scale in case of a massive connection of users at the same time.

5.3.2 Algorithms

This subsection presents the algorithms and mechanisms of PMS+. An overview of the different blocks and their interactions is depicted in Figure 5.8.

Initialization process. In PMS+, the initialization process has two objectives. First, the initial video playback delay should be minimized. Same as in PMS and current HAS systems, the first segment is downloaded at the lowest bitrate from the servers. Second, the initialization process can be viewed as an opportunity to collect information about the download bandwidth capacity of the peer. The first 5 seconds of video are downloaded from the servers. Hence, the client has some time to retrieve enough information to

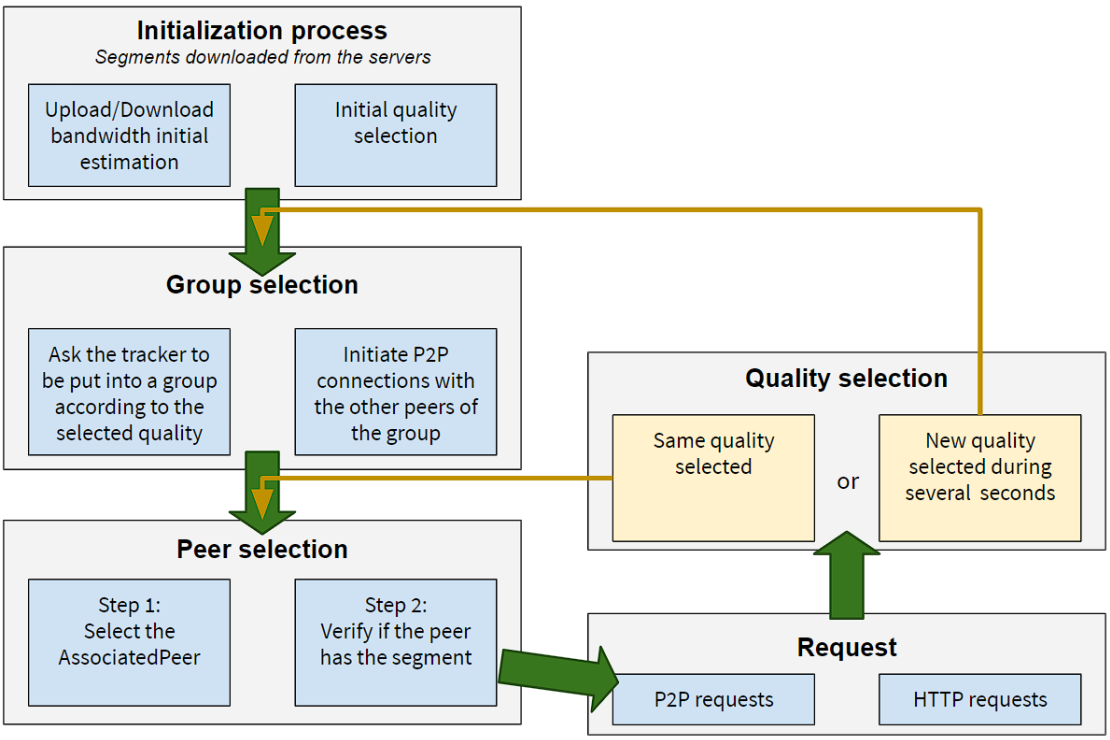

Figure 5.8: Main mechanisms of PMS+

estimate the appropriate bitrate. This time is also used to initiate a connection with the tracker. At the end of the initialization process, the peer sends a message to the tracker containing the quality selected and its current position in the playback. The quality is chosen by using a simple adaptive bitrate method. After downloading a segment, the peer can have a decent estimation of its throughput by measuring both the duration and the number of bytes carried by a request.

Group selection. Once a peer finishes its initialization phase, the tracker can assign it to a group. The peer requesting to be put into a group is referred as the **asking peer**. A group, like an overlay, is defined by a quality. Every single peer in a group is supposed to receive the same quality. When a peer sends a request to join a group, the tracker executes Algorithm 2. This algorithm introduces a set of variables. The list of groups **Gr**[] is a list saved inside the tracker where all the groups for this video are saved. Two variables are set on tracker side by the system: G_{max}^{size} which defines the maximum number of peers in a single group and Δ_{max}^{PP} which is the maximum deviation between the position in the playback of the peers already in the group and the asking peer. This second variable is

important because the peers cannot save the video segments in memory for a too long period of time since they do not all have the same capacities in terms of memory. If the asking peer is too far from the position of the other peers, video data may not be exchanged. Every **group** GR is linked with a quality level (**group.**Q), to be certain the asking peer is requesting the same quality, and the $GR.PP_{min}^{estim}$ and $Gr.PP_{max}^{estim}$ which are the estimated minimum and maximum playback positions of the peers inside the group. These values are estimated using the playback position declared by the peer and the date of update of this information. The asking peer sends information to the tracker. In the algorithm, input information are referenced as askingPeerInfo (AP). AP embeds data like $AP.Q_{target}$ the video quality selected by the peer and $AP.PP_{current}$ the current playback position of the peer.

Algorithm 2 Group creation, executed in the tracker

Inputs: Gr[], G_{max}^{size}, Δ_{max}^{PP}, askingPeerInfo AP

isGroupSelected = false
for Gr **in** Gr[] **do**
  if $AP.Q_{target} = Gr.Q$ **and** $Gr.size < G_{max}^{size}$ **and** $AP.PP_{current} > Gr.PP_{min}^{estim} - \Delta_{max}^{PP}$ **and** $AP.PP_{current} < Gr.PP_{min}^{estim} + \Delta_{max}^{PP}$ **then**
    groupSelected = Gr (The peer will be inserted inside Gr.)
    isGroupSelected = true
  end if
end for
if isGroupSelected = false **then**
  groupSelected = new Group(AP) (The peer will be inserted inside a new group.)
end if
return groupSelected

Once a group is selected, the tracker sends a message to the asking peer with the list of the other peers inside this group. The peer can then create a WebRTC connection with every one of them.

Peer signaling and ordered list. Once in a group, every peer create a P2P connection with every other. The peers exchange their upload/download bandwidth and a map of the segments they possess called the buffer map. With the help of this buffer map, peers can deduce their current position in the video stream. With this information, every peer creates an ordered list of the peers in the group according to their playback position. If two of them are at the same playback position, they are sorted according to the date on which they joined the group. Every peer is responsible for computing the

ordered list of peers with the information they received from the other peers. Every time a peer sends an updated playback position, every peer should reorder the list. By doing so, every peer should have the same list in the same order at any moment. Figure 5.9 presents a sequence diagram of the signaling message exchange by the peers. In this figure, the red nuts highlight the time when the peers should reorder the list. The sorted list of peers is a keystone of the peer selection mechanism of PMS+.

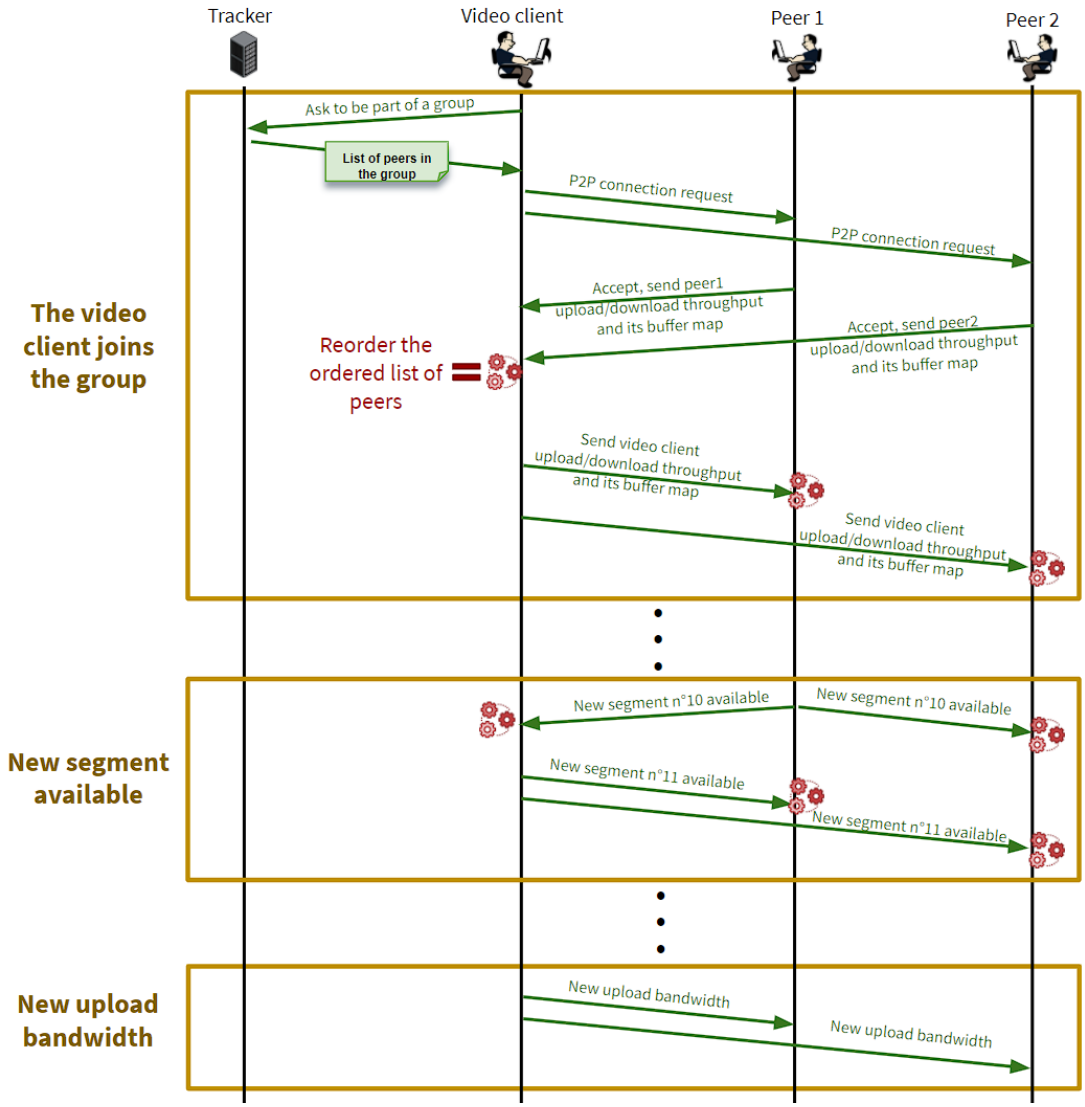

Figure 5.9: Overview of the messages exchanged by the peers of the same group during a video streaming session

Peer selection and data delivery. The goal of this mechanism is to optimize the P2P delivery by selecting the most appropriate peers and avoiding conflicts between two peers downloading video data from a predecessor with insufficient upload bandwidth. The ordered list of peers presented in the last paragraph plays an important role in this mechanism.

Figure 5.10: Step 1: The AssociatedPeer for every peer is selected

The peer download selection algorithm, in its essence, is illustrated in Figure 5.10. The first peer joining the system downloads every video data from the server. Related to its upload capabilities, it has a capacity to deliver two segments at a time. A second peer arrives and retrieves everything from the first one (i.e., the one that arrived just before). An association is thus created: AssociatedPeer(Peer2) = Peer 1. The delivery capacity of Peer 2 is only one segment at a time. Recursively, the third peer joining the group gets the segments from the second (AssociatedPeer(Peer3) = Peer 2) but, since

it is e.g, in a ADSL network, it has no delivery capacity. Following the same idea, the fourth peer tries to select the third one, which is his direct predecessor. However, since the latter does not have a sufficient upload throughput and neither does the second, it turns out to the first (AssociatedPeer(Peer4) = Peer 1). In case no predecessor would have enough delivery capacity, the segments get downloaded from the servers and the related peer would have no AssociatedPeer.

The pre-download decision process consists of two steps. The first step focuses on the association of peers with their 'good delivery-capable' predecessors, as described in the last paragraph and depicted in Figure 5.10. During this step, the video client computes with a deterministic function the peer designed as 'AssociatedPeer' by each member of its group, i.e., the one chosen to deliver the selected segment at the requested quality. This first step is essential in order to have a clear overview of the situation at each moment. Every peer should know the list of all the existing associations (AssociatedPeer(Peer X) = Peer Y) in the group so that it can act accordingly. The objective is to avoid a conflicting situation, where some peers are asked to deliver while they do not have the capacity (anymore) to do it.

The second step consists in the verification of the segment availability at the AssociatedPeer and is described in Algorithm 3. In this algorithm, $PP_{segment}$ is the playback position of the requested segment, $PP_{associatedPeer}$ is the current playback position of the AssociatedPeer, BO is the buffer occupancy of the video client, and BO_{thresh} is the buffer occupancy threshold.

Algorithm 3 Step 2: verification of the availability of the segment in the AssociatedPeer

Inputs: $PP_{segment}$, $PP_{associatedpeer}$, BO, BO_{thresh}

if AssociatedPeer owns the segment **then**
  Download the segment from the AssociatedPeer
else if $PP_{segment} \geq PP_{associatedPeer}$ **and** $BO \geq BO_{thresh}$ **then**
  Wait a few seconds, reorder the list, and retry from step 1.
else if $PP_{segment} < PP_{associatedPeer}$ **and** $BO \geq BO_{thresh}$ **then**
  Reorder the list and retry from step 1.
else if $BO < BO_{thresh}$ **then**
  Download the segment from the server.
end if

The client verifies that his AssociatedPeer has the segment it needs. This assumption should most of the time be true because of the playback position parameter in the group selection algorithm and because the peers are sorted according to their playback position

in the ordered list. However, if the segment is not available within the AssociatedPeer, due to seeking or connectivity problems, we rely on the playback position and on the buffer to react. If the playback position of the desired segment is higher than the position of its AssociatedPeer and its buffer occupancy BO is higher than a given threshold BO_{thresh}, then we give some time to react. The process restarts from Step 1, since the peer selection may have changed. In case the playback position of the desired segment is lower than the playback position of its AssociatedPeer and its buffer occupancy BO is also higher than a given threshold BO_{thresh}, then it means that the list needs to be re-ordered, and we restart the process from Step 1 since we have time to do it. If we don't have much time left, i.e., the client's buffer occupancy BO is lower than a given threshold BO_{thresh}, the download of the segment is performed from the server.

Quality adaptation and group modifications. A peer uses both the throughput estimation and the buffer occupancy to select a quality. After every request to the servers, the video client estimates its current download throughput. If the buffer occupancy is lower than a specific threshold, the throughput estimation is multiplied by a factor lower than one. Then, the video player is able to select the highest quality with a bitrate lower than the throughput estimation as its new target quality.

The quality selection can be done every time it is needed without restriction. However, if the quality is modified, the peer remains in the same group during a few seconds. This behavior is designed to prevent a lot of useless group modifications every time a short and temporary bandwidth loss occurs. Since no other peer in the group should possess the new selected quality, the peer should have to download the segment from the servers. After downloading 15 seconds of video in a different quality that the one of the rest of the group, the peer leaves the group and asks the tracker for a new group by sending updated information.

With the same idea, if a peer notices that its playback position is too far from the playback position of the other peers (e.g., because a user moved further in the video or paused the stream), the peer leaves the group and asks for a new one too.

Upload and download throughput estimation. The estimation of the download and upload throughput is not trivial in P2P systems. Figure 5.11 provides an overview of the upload and download throughput estimation mechanisms implemented in the PMS+ client. Usually, in HAS solutions, the download throughput is estimated at the end of a request by looking at the size of the data received and the time needed to get them.

Nonetheless, this estimation for the download throughput is not accurate when video data are downloaded from another peer because it depends on the upload throughput of this serving peer. Indeed, if the serving peer has less upload capacity than the client peer, the resulting calculated download throughput for the client would not be accurate. To overcome this issue, the client will calculate and update the estimated download throughput only based on data retrieved from servers.

The upload throughput is even harder to estimate. First, it depends on the other peer's download throughput. Second, while all the multiple downloads start at the same time, the multiple uploads may begin at different moments. It is difficult to estimate if two uploads have been performed at the same time or not, and if the observed upload bandwidth is reliable. To solve this issue, we introduce a test during the initialization phase. A few bytes are sent by every new client inside a POST request to our server. By measuring the time needed to make this request, a reliable estimation of the upload bandwidth is computed.

After this initialization, the estimated upload bandwidth of the client can be increased or reduced due to bandwidth variations. Because of the potential multiple requests and the impact of other peers download throughput, the reduction of the estimated upload bandwidth must be done carefully with a specific mechanism. This mechanism is more precisely described in Algorithm 4, with the following terms:

- UR_i is the list of client's i delivery requests performed since the last upload estimation.
- Every upload/delivery request UR is associated with an upload value calculated during the upload phase ($UR.UP_{estimated}$).
- $UR_{max}.peer.DL$ is the download throughput declared by the served peer for the same delivery.
- $UP_{current}$ is the current upload bandwidth officially declared by the client.
- UP_{test} is the upload bandwidth calculated by sending a few bytes in a POST request sent to a server during the initialization process.
- $UP_{min}^{Q_{target}}$ is the lowest upload value needed to send one segment of the target quality to another peer.
- F_a is the factor by which the download throughput of the other peer should be higher than the upload of the client to consider its feedback valid.

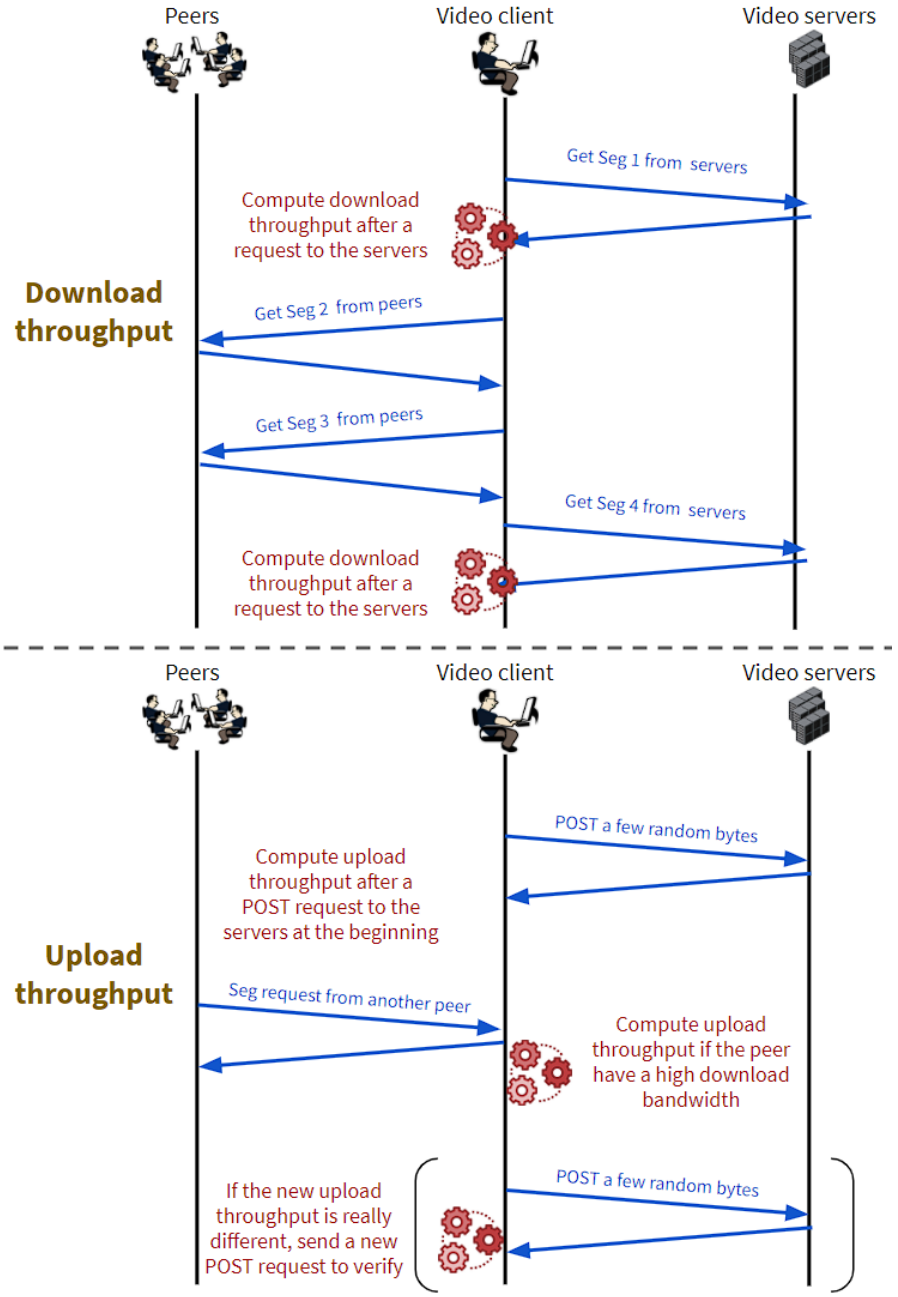

Figure 5.11: Overview of the upload and download throughput estimation in PMS+

When a client calculates —on one of its delivery— a higher upload bandwidth than the currently declared one, it updates the latter with this new value. If a lower upload bandwidth is calculated by the client when it sends data to a peer with a very high download bandwidth capacity, the value of its official upload bandwidth is decreased

Algorithm 4 Upload throughput estimation in multiple-P2P context

Inputs: UR_i, F_a, $UP_{current}$, UP_{test}, $UP_{min}^{Q_{target}}$

$UR_{max} = \max_i\{UR_i.UP_{estimated}\}$
if $UR_{max}.UP_{estimated} > UP_{current}$ **then**
  $UP_{current} = UR_{max}.UP_{estimated}$
else if $UR_{max}.peer.DL > F_a * UP_{current}$ **then**
  if $UR_{max}.UP_{estimated} > UP_{min}^{Q_{target}}$ **then**
    $UP_{current} = UR_{max}.UP_{estimated}$
  else if $UP_{test} > UP_{min}^{Q_{target}}$ **then**
    Retry UP_{test} and $UP_{current} = UP_{test}$
  end if
end if
if $UP_{current}$ has been modified **then**
  Send the new $UP_current$ to every peer of the group
end if

to this value. However, if the latter is lower than $UP_{min}^{Q_{target}}$, then we re-launch the initialization test to a server in order to re-adjust properly the estimations.

5.4 Large-scale evaluation and results

The objective when designing the PMS+ solution was to evaluate the system in a real use case. The player is implemented in a production level large scale platform delivering live streams to real users. PMS+ is deployed in two French leading websites of the public webcam industry. Webcams are focusing famous places in French cities, beaches to keep an eye on the waves, and strategic places in highways to inform about the current traffic. Anonymized metrics are periodically retrieved to evaluate the health of the system.

5.4.1 Large-scale platform description

The large scale platform is designed to receive, save, update, transcode, and deliver video streams from hundreds of live cameras. Transcoding requires a lot of CPU and delivery needs a lot of upload throughput. Memory and storage are not as important as CPU and bandwidth for live streaming from webcams since only a small window of a few minutes is saved per camera on disk. A total of 14 cloud servers have been instantiated to run this platform:

- 9 are used to receive, save the metadata and transcode the content;

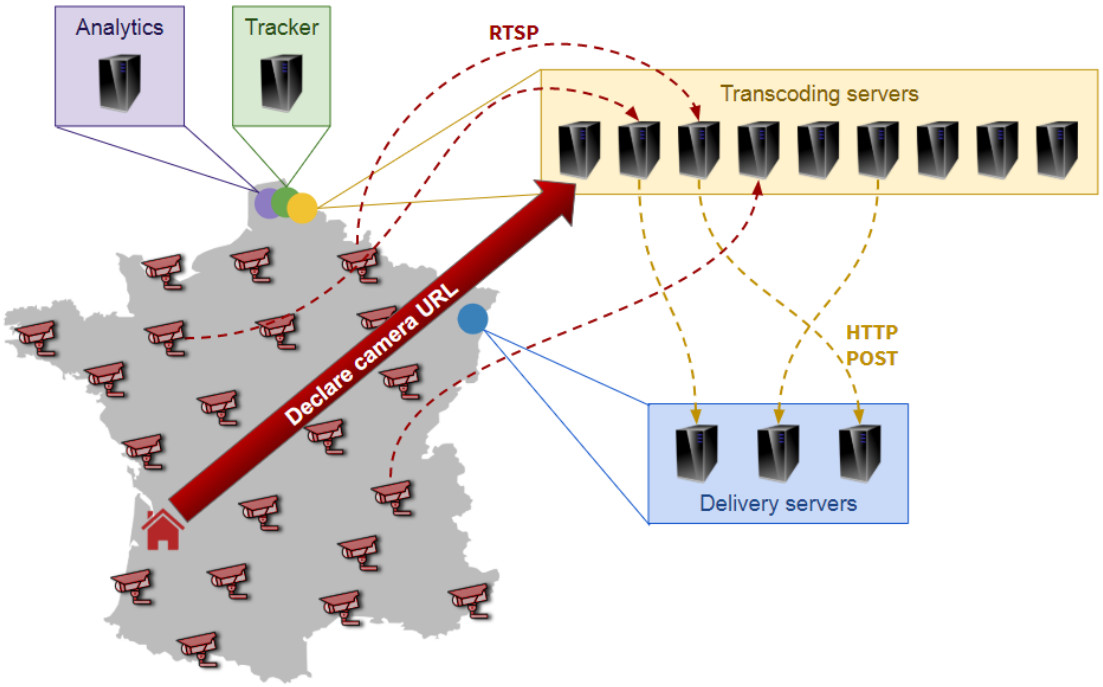

Figure 5.12: Platform to transcode video streams from numbers of live cameras

- 3 have a 1 Gbps upload bandwidth and are used to save and deliver the video segments;
- 1 runs a tracker to connect the peers and create the overlays/groups;
- 1 is used to collect anonymized data from the clients about the streaming session in order to create QoE graphs.

An overview of the platform is provided in Figure 5.12. First, the webcam provider, located in Bordeaux, France, can enter metadata describing the live streams by using a REST API deployed on top of the transcoding servers. One live stream is composed of at least a title, an ID, and the URL of the source stream from the cameras on top of other information. Then, transcoding modules can retrieve source streams from cameras by using the RTSP protocol. The streams are transcoded in several qualities, depending on the bitrate and resolution of the input stream. The quality of input streams are defined by the quality of the camera and by the available bandwidth at their access network. Some cities in France have access to very high bandwidth through optical fiber while other places only have old copper lines. The resolution and bitrate ranges from full-HD at 1 Mbps to 4K at 8 Mbps. Because of this plurality of input quality, the

absolute values of video bitrate displayed to the end-users need to be compared with the bitrate of the input stream to define the P2P solution average quality displayed. Once transcoded, every video segment is uploaded and saved into delivery servers by using HTTP POST requests. With the same idea, old segments are removed with the help of HTTP DELETE requests.

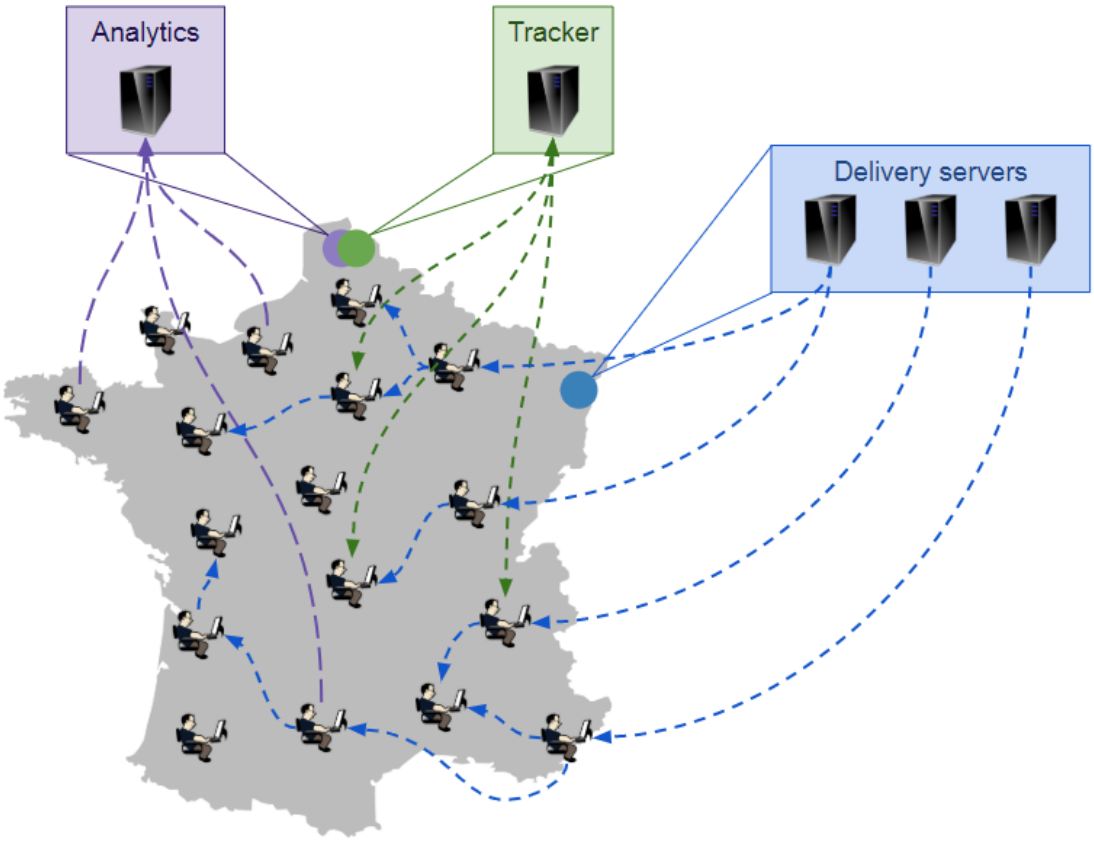

Figure 5.13: Delivering live streams to thousands of users using PMS+ and receiving anonymized metrics from users

Once the video data available in the delivery servers, the live streaming sessions can begin, as illustrated in Figure 5.13. Video clients from anywhere in France are able to get MPD files and retrieve video segments. Video players are also connected with a tracker running on top of another server. With the help of P2P signaling permitted by the tracker, peers can transfer data to one another. Finally, every 10 seconds, every video client sends data to an analytic server. These metrics are anonymized and describes the quantity of video segments retrieved and the number of stalling events experienced by the client during a period of time.

5.4.2 Analytics service

In order to collect data for this evaluation, several modules have been deployed in the analytics server. Firstly, an Elasticsearch server receives the metrics sent by the video clients. Elasticsearch is an open source high performance time series database used to easily store, search and compute data. Secondly, a Grafana server is used for the visualization of the data. Grafana is an open source analytics and monitoring solution used to display data retrieved from various databases, and, in particular, from an Elasticsearch module.

When a video is played, data are collected and aggregated by the client-side JavaScript player and sent to the Elasticsearch server via HTTP POST requests. The data sent are anonymized. They are composed of the following fields:

```
    "MB_server": 0.75,
    "MB_peer": 0.77,
    "client_uuid": "6eeee240-7262-46be-b94b-f38a4408318c",
    "sec_server": 3,
    "sec_peer": 3,
    "qualities": {
        "Q1":0,
        "Q2":0,
        "Q3":1.52
    },
    "timestamp": "2020-06-30T16:11:14.445Z",
    "video_id": "videoid",
    "sec_rebuf": 2,
    "glitches": 1
```

The field **client_uuid** is an Universally Unique IDentifier (UUID) created during the player start up. This **client_uuid** allows to identify the different sessions and to compute an accurate number of views. It is not linked with any personal information and cannot be used to identify a user. It can neither be used to find a link between a user and the video watched because a new UUID is created every time the video player is refreshed. The other fields carry information related with the performances of the video streaming system and are useful to evaluate the global quality of experience of a streaming session. **MB_server** and **MB_peer** are respectively the quantity of data received from the servers and from other peers in megabytes. With the same idea, **sec_server**

and **sec_peer** are the number of seconds of video received from the servers and from other peers. The field **qualities** embeds the quantity of data downloaded for every quality level. The **timestamp** is the date on which the metrics have been generated. Finally, **sec_rebuf** and **glitches** are respectively the total duration of stalling events and their effective number.

Data collected are then accessible in Grafana dashboards. An example taken from a private live stream is provided in Figure 5.14, Figure 5.15 and Figure 5.16. The metrics detailed in the last paragraph are aggregated together in order to provide human-readable quality of experience visualizations. The dashboards provide several graphs to monitor four important information about the video streaming sessions: P2P efficiency, video quality, reliability, and user behavior.

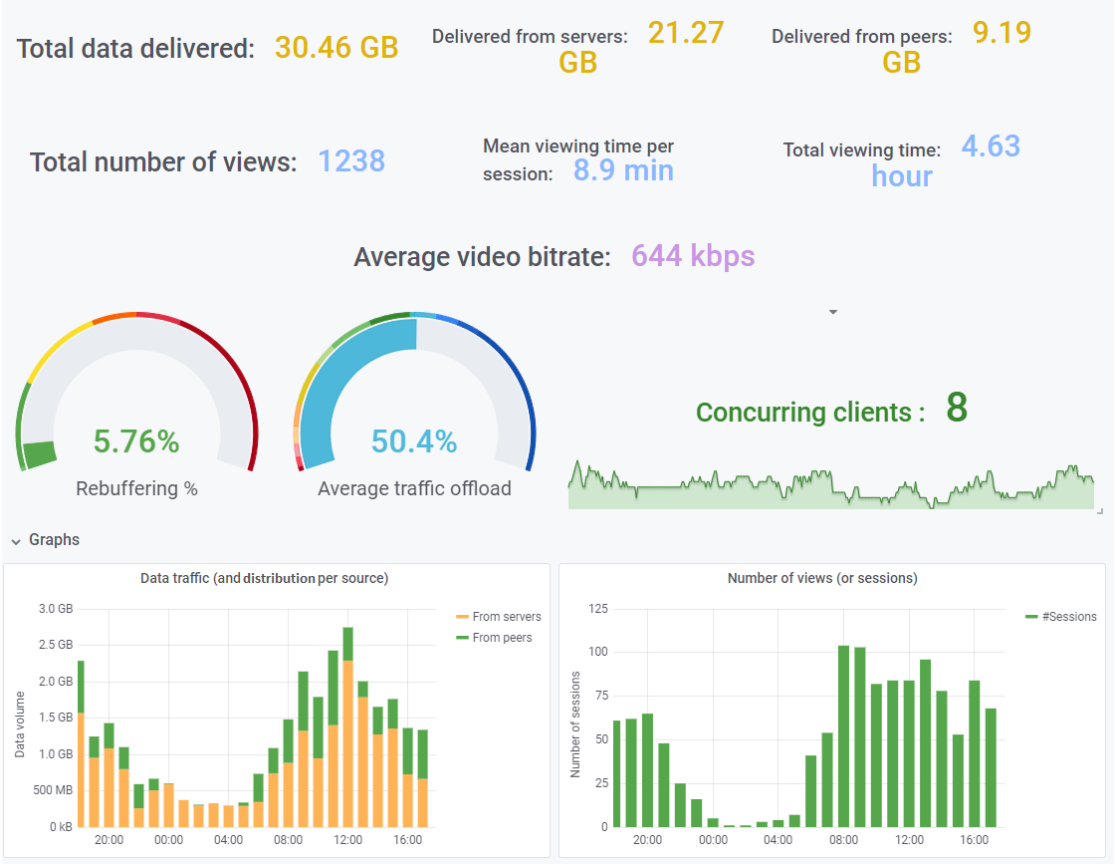

Figure 5.14: Example of metrics received for one live stream during one day - Part 1

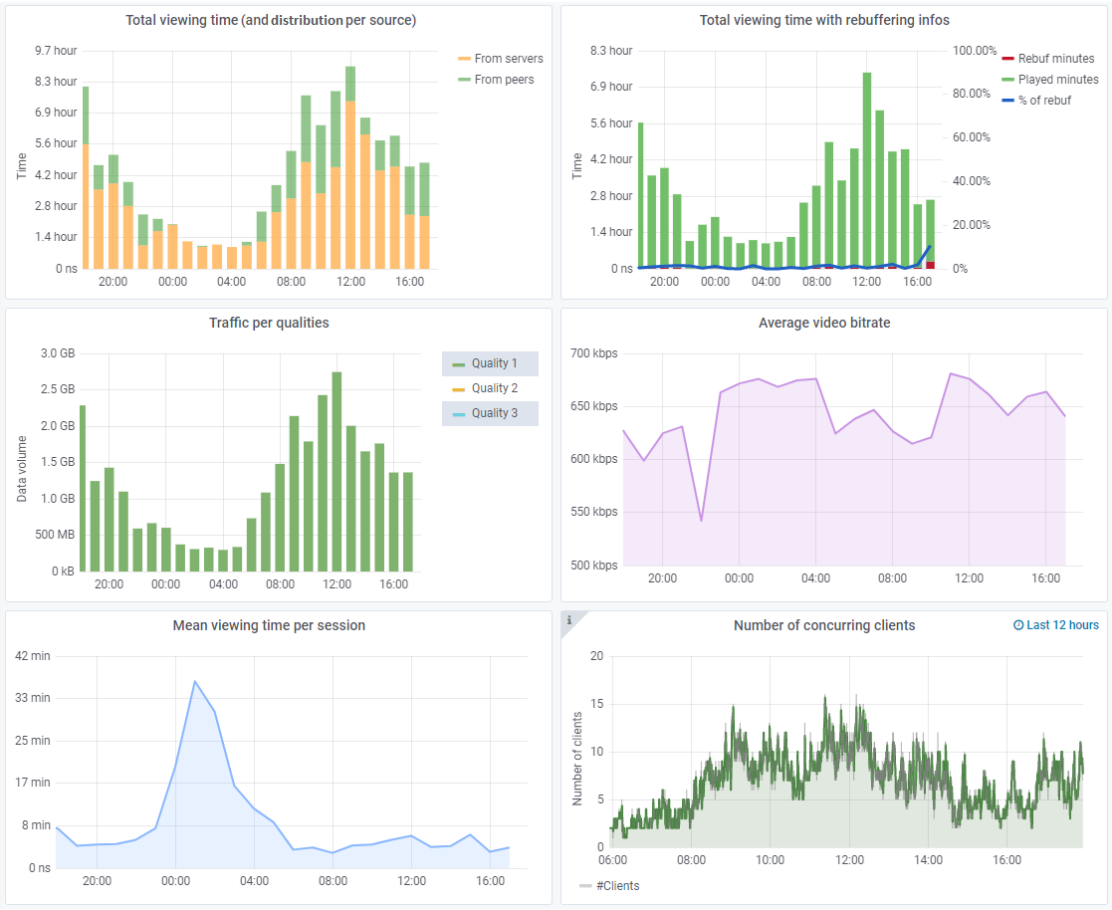

Figure 5.15: Example of metrics received for one live stream during one day - Part 2

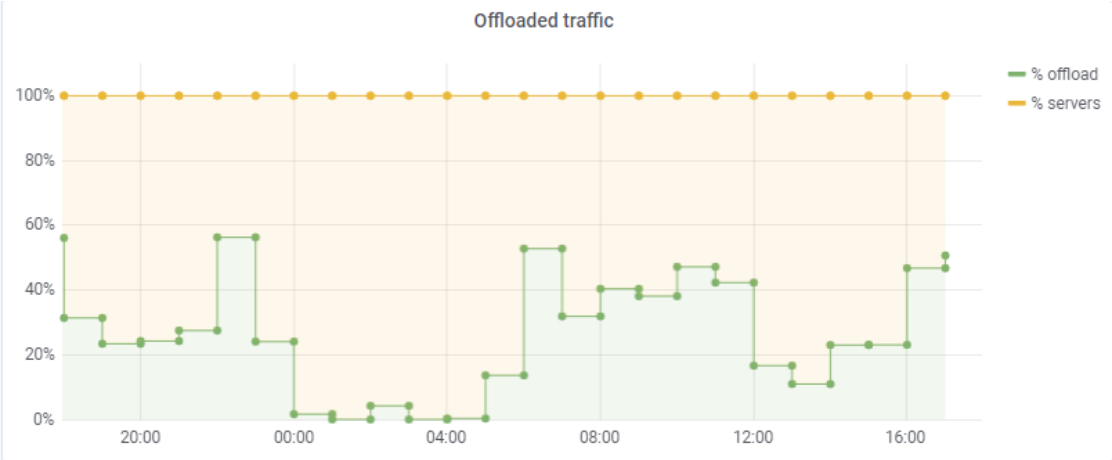

Figure 5.16: Example of metrics received for one live stream during one day -Part 3 - P2P offload

P2P efficiency. Metrics **MB_server**, **MB_peer**, **sec_server**, and **sec_peer** are used to create graphs about the quantity and quality of video downloaded from servers and P2P. "Data traffic (and distribution per source)" (Figure 5.14), "Total viewing time (and distribution per source)" (Figure 5.15, and "Offloaded traffic" (Figure 5.16) respectively provide information about the quantity of data, the number of seconds of video and the distribution of downloads between servers on one hand and via P2P on the other hand. For the same quality of experience, it is usually better to get most of the data from other peers in order to reduce the cost of the servers as well as the risks of a sudden overload.

Video quality. Metrics **MB_server**, **MB_peer**, **sec_server**, **sec_peer** and **qualities** are also used to provide graphs about video quality. "Traffic per qualities" (Figure 5.15) gives data downloaded for every quality. With this graph, it is possible to see if the best quality is downloaded most of the time or not. "Average video bitrate" (Figure 5.15) precises the average bitrate downloaded by the clients. The closer to the bitrate of the best quality, the better.

Reliability. Metrics **glitches** and **sec_rebuf** are used to create graphs dealing with the quantity and severity of video stalls experienced by the users. The two main Grafana panels in this category are the "Rebuffering %" gauge (Figure 5.14) and the "Total viewing time with rebuffering infos" bar graph (Figure 5.15). The latter compares the seconds of video watched against the seconds of rebuffering. The smaller the rebuffering, the better.

User behavior and popularity. Two data are interesting to study user behaviors: **client_uuid** and **timestamp**. These metrics are used to provide the number of users watching the video at the same time ("Number of concurring clients" - Figure 5.15), the number of new users starting a streaming session during a period of time ("Number of views (or session)" - Figure 5.14) and the average duration of a streaming session ("Mean viewing time per session" - Figure 5.15). These graphs can provide useful information to understand how the users are consuming the live streams and might be important to discuss the other results.

5.4.3 Evaluated players and implementations

The PMS player has been implemented on top of the open source **dash.js** [DASH-IF Player, 2020]. New class and functions have been added inside the player. New adaptive rules have been implemented and the network loading functionalities have been modified to manage multiple requests at the same time, as well as WebRTC messages. External functions have been provided to handle the P2P signaling, the communication with the tracker and the periodic feedback to the analytics.

The PMS+ player has been implemented as a plug-in that can connect to several open-source players like dash.js [DASH-IF Player, 2020], hls.js [hls.js, 2020], and videojs [Video.js, 2020]. This plug-in uses the segment scheduler of the base players but overrides the adaptation decision to apply its own. The PMS+ plug-in embeds every functionality including quality adaptation, download management, P2P signaling, communication with the tracker, and metrics reporting.

Two other players are considered during this experiment. The first one is a standard dash.js player without P2P. The second is a simple **custom P2P-DASH** player based on the P2P mechanisms observed in the open-source P2P HAS plug-in implemented by Novage and called P2P-Media-Loader [Novage, 2020]. This player relies on simple adaptation rules. Peers watching the same streams are put into random "swarms" —similar to our groups in PMS+— and communicate with each other to exchange the segments they need. In a swarm, one random peer downloads a segment in HTTP, and the other ones tries to get this segment in P2P. Unlike PMS+, this solution does not try to spread the downloads between the available peers to avoid conflict, does not come with a clear organization between the peers to download only the minimum data from the server, and does not embed an efficient upload estimation function. To summarize, this player tries to download in P2P when it can, but should be limited theoretically by the random conflicts and the constrained upload bandwidth of real environments.

5.4.4 Results of the experiment

The players have been deployed in two websites delivering video streams from webcams during one month each. The anonymized metrics described previously were periodically reported by the clients in order to get feedback on QoE. During the experiment, thousands of clients —varying according to the date and the hour of the day— have concurrently consumed hundreds of live streams at the same time. At the end, an important quantity of data have been retrieved.

Audience curves. In order to select representative and interesting data to evaluate our player, let us first take a look at the audience curves of the cameras. Three different types of audience have been identified. Figure 5.17 is a typical trace from a set of popular webcams on a working day in the middle of the week. As it could have been expected, the curve grows in the morning to reach a plateau during day time before falling at night. Interestingly, a small peak is often observed around midday. This time frame corresponds to the beginning of the lunch break for most workers. Figure 5.18 is a representative trace from the same cameras on the week-end. The curve go up higher in the morning. The peak is reached during the afternoon. However, this time, a small drop is regularly observed around 1pm. During the week-end or during holidays, the lunch break becomes a lunch time. Last but not least, Figure 5.19 illustrates a typical curve observed during unusual events. Considering the use case of webcam for tourism, this kind of curve is often observed during a particular weather phenomenon. It can be for example a snow day, a storm, or a sunny day after several weeks of rain. But it can also be caused by a newsworthy event. Similar curves have been observed in April 2019 when Notre-Dame cathedral in Paris was on fire and in March 2020 when the lockdown was pronounced during the Covid-19 pandemic. Fortunately, peaks have been observed in happier times when radio hosts encouraged their own audience to open the website for a game they organized in a city.

Representative data selection In the experiment, we gathered data from every type of day. The first two are very common and can be seen almost every week whereas the curves obtained after unusual events can not always be predicted. A large peak took place when the Covid-19 lock-down started in France. At this time, the DASH player was evaluated. The end of the lock-down was pronounced when our custom P2P-DASH was deployed. Similar peak were observed during sunny days when PMS and PMS+ have been evaluated. In addition, we removed the night hours because only a few users were then watching videos. From 20pm to 8am, the system is not stressed and the results are not significant. As a matter of fact, the P2P cannot work if only isolated users are downloading the segments. With the same idea, we chose to ignore the less popular cameras. We ended up compiling data received during 448 hours by the 70 most popular cameras. Between 800 and 3600 users were concurrently consuming video streams from the cameras at the same time. Therefore, a total of 492800 hours of videos (or 20533 days) have been watched throughout the experiment.

Figure 5.17: Typical audience curve during a working day

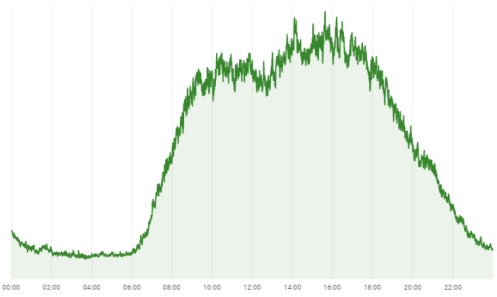

Figure 5.18: Typical audience curve during a week-end or a holiday

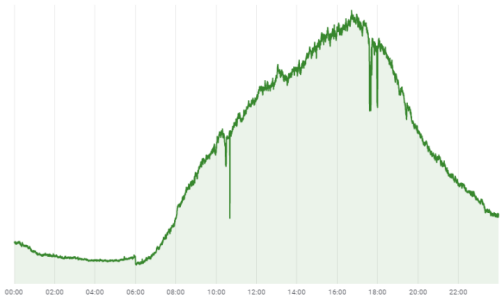

Figure 5.19: Typical audience curve during an unusual event (first days of lock-down during Covid-19 pandemic, snow day, newsworthy event, first sunny day after weeks of rain, etc.)

In this experiment, the two most common QoE metrics are considered, namely the video quality displayed to the user and the duration of video stalls. Box plots of these metrics for every evaluated player are available in Figure 5.20 and 5.21. Moreover, in order to evaluate the efficiency of the P2P solution and the infrastructure costs that could be saved, we gathered the quantity of data received from other peers in Figure 5.22. The next paragraphs detail the results.

Video quality. As mentioned in the last section, every webcam does not have the same video bitrate. It depends on the quality of the camera and the upload throughput of their access network connection. Considering the bitrate of the source, the video streams are available in one, two, three or four qualities. Hence, every video does not have the same bitrate and the same qualities available. In order to evaluate the capacity of a video player to display the best quality, we selected a subset of webcams with source bitrates between 4 and 6 Mbps and encoded in three qualities (480p at 500 kbps, 720p at 1.5 Mbps and the best quality in full-HD or 4K with a bitrate equal to the source bitrate). We computed the time spent in high-quality and compared it with the total video hours retrieved. A percentage is hence obtained. If the percentage is high, the video is received most of the time in high-quality. If the percentage is lower, then the player has not been able to display the best quality to every user most of the time.

The results are available in Figure 5.20. DASH is able to get the best quality 90% of the time. This illustrates the importance HAS streaming have earned in the video streaming community: when the infrastructure is well scaled, the player works well. However, DASH has the lowest minimum compared with other players. When the servers are overloaded or when the network is congested, the client is no longer able to download the best quality. The low quarter of the custom P2P-DASH is higher than DASH, meaning that with the help of P2P, the client is not as impacted as DASH when the servers are stressed. Nonetheless, the median quality is lower than DASH. Because of the limited upload bandwidth of some peers, some requests must be stopped and a lower quality is retrieved from the servers. PMS has a lower minimum value than P2P-DASH, but a higher average video quality. This is a direct consequence of the overlay selection algorithm implemented in the client. In most situations, the overlays and multiple-source aggregations reduce the risk of requesting data to peers with low upload bandwidth. But because the player starts at the lowest quality and increases the quality one by one,

some peers may need some time to receive a better quality. Last but not least, PMS+ outperforms every other player. The best quality is received 90% of the time, like DASH when the resources of the infrastructure are enough, but it does not suffer from congestion with the help of P2P. Moreover, PMS+ has the highest lower quarter thanks to the peer organization maximizing the availability of video segments and avoiding concurrent access to a peer with small upload bandwidth.

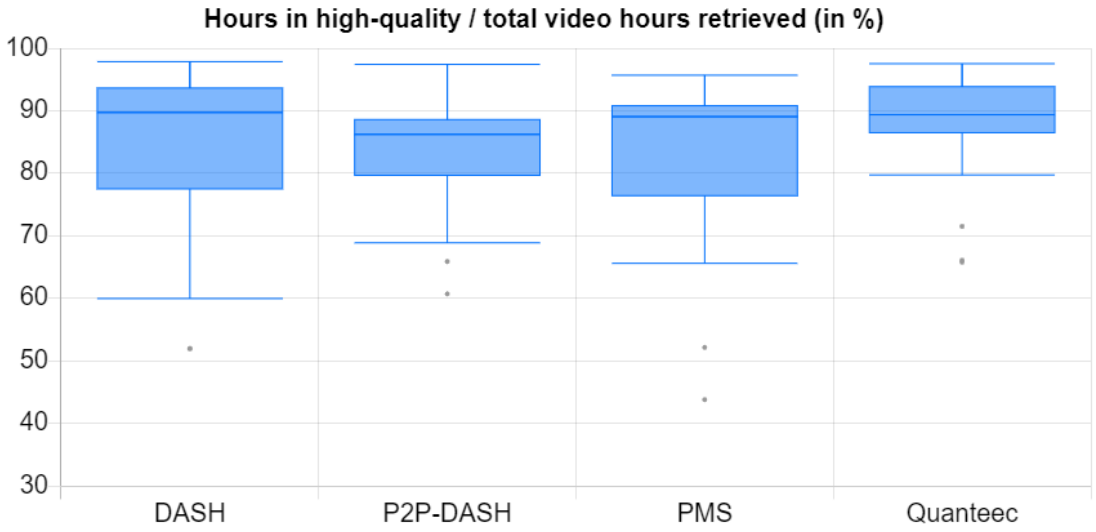

Figure 5.20: Video quality for the four evaluated players

Video stalls. Figure 5.21 shows the duration of video stalls in comparison with the duration of video watched. The lower the percentage, the better. The results and their interpretation are similar to the video quality ones above. The median percentage of video stalls for DASH is 1.8%. Once again, this illustrates that DASH works well in a well-scaled system. However, when the infrastructure is stressed, the rebuffering increases. The higher quarter of DASH is the highest and a number of peaks have been observed up to 33% during unusual events and when the number of concurrent users is rising. This means that a part of the users might have trouble receiving the live streams. P2P-DASH and PMS are not affected by these peaks like DASH. However, the median percentage of video stalls is higher as a consequence of the limited upload bandwidth of the peers and the concurrent access when several clients try to request data from the same neighbor. PMS+ outperforms the other players in terms of reliability. The player is able to maintain a low number of video stalls and its maximum is only very occasionally higher than 5%.

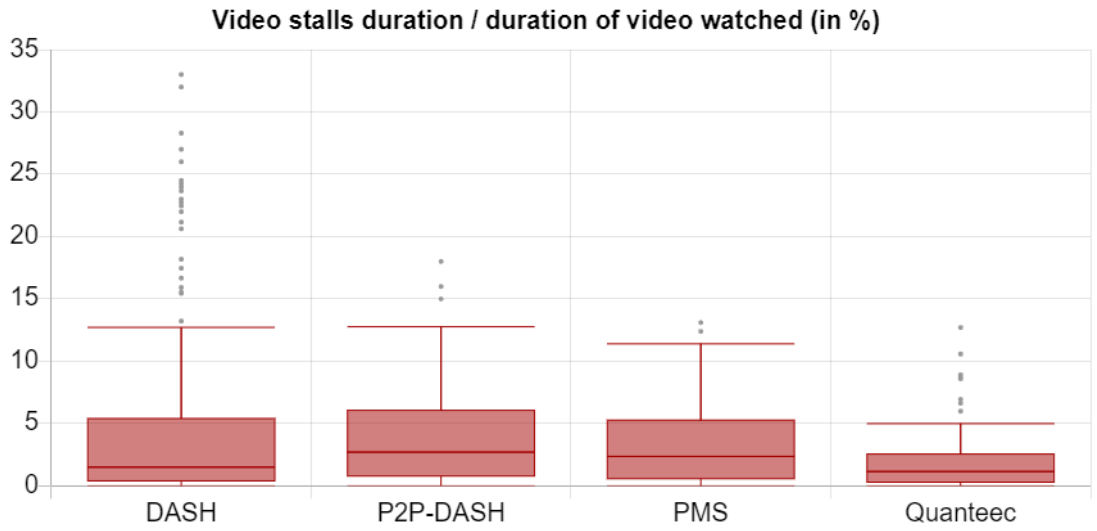

Figure 5.21: Video stalls for the four evaluated players

P2P offload. We define the P2P offload as the duration of video segments received from the peers compared with the duration of video retrieved from the servers. A low percentage means that a lot of data are received from the servers, increasing the risks of server overload, as well as the cost of the infrastructure needed to serve the same number of users. On the other side, a high percentage is the sign of an efficient P2P delivery mechanism. The results are available in Figure 5.22. Obviously, the DASH player does not embed P2P capability and stays at 0%. Between 10% and 40% of P2P offload are observed for P2P-DASH and PMS. The mean value of PMS is slightly higher thanks to the overlays. Because the peers are not allowed to leave an overlay if it would have a bad impact for the QoE of their neighbors, P2P interactions are encouraged. However, in those two video clients, random segments are downloaded from the servers. This does not optimize the quantity of data exchanged between peers because of the redundancy. Moreover, when the upload bandwidth of a peer is not sufficient, the video data are requested from the servers. The median P2P offload of PMS+ is close to 40%. During peak hours, PMS+ is able to be close to 70%. In average, thanks to the P2P organization inside the groups, more segments are exchanged between the peers.

As conclusion, PMS+ outperforms other players in this large scale experiment. Both the QoE metrics and the P2P offload are better. PMS is, in most cases, better than a simple P2P-DASH but it is at the cost of a little fairness when the video quality is concerned since some peers end up blocked in low qualities when they could have watched full-HD content.

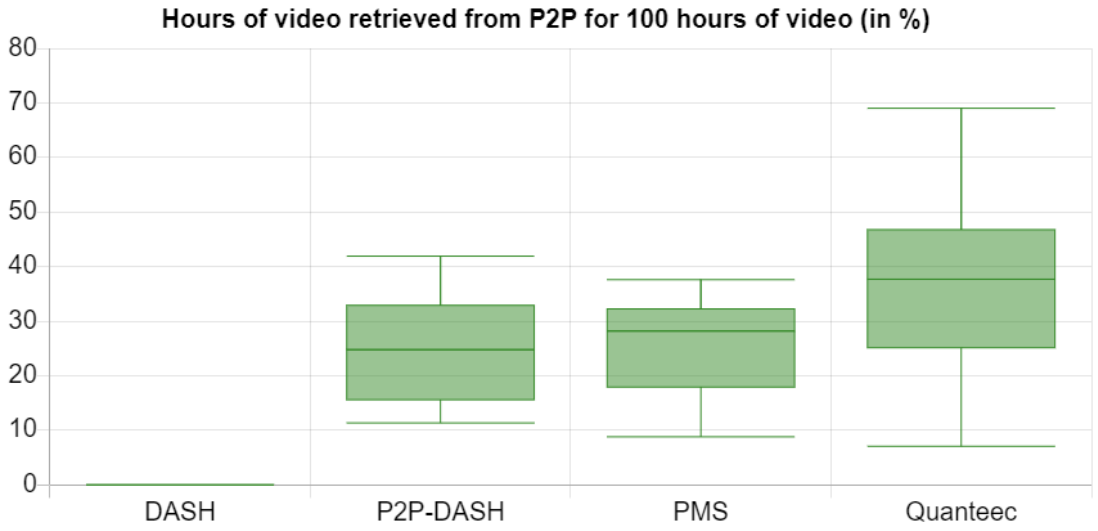

Figure 5.22: P2P offload for the four evaluated players

5.4.5 Discussion and comparison with state-of-the-art results

When compared with experiments from the state of the art of P2P systems, the P2P offload presented by [Anjum et al., 2017] and [Roverso et al., 2012] seems better. The first authors provide the traffic saving of commercial P2P assisted CDNs such as Kankan [Zhang et al., 2014], NetSession [Akamai, 2020] and LiveSky [Yin et al., 2009a] that have been presented in Chapter 4. For Kankan and NetSession, the P2P savings are above 70% and can reach 90%. The second authors present an evaluation of a P2P solution deployed by Hive Streaming. Their P2P savings are around 77%. In this subsection, we discuss the differences between those studies and our experiment.

Service type. Kakan and NetSession are VoD streaming services relying on P2P to deliver their most famous videos. On the opposite, our platform delivers live content from webcams. Live and VoD are not the same type of streaming service. While live videos keep receiving new segments, VoD segments are available since the beginning of the session. Video players can build strong buffers and save more segments to be delivered to future users. Moreover, both systems relies on an application to work. Unlike browser-based solutions, video streaming applications can save more data on the hard drive and might run in background, increasing the P2P exchange rate even when the user is no longer watching the video. The same paper [Anjum et al., 2017] also discusses the results

of LiveSky, a more similar live streaming service. The P2P savings for their solution are around 35%, which is in the same order of magnitude than our results for typical days.

Experimental setup. In [Roverso et al., 2012], the authors evaluate Smooth Cache, a live streaming system running in a Microsoft Smooth Streaming player. Their solution was deployed in a development environment. A few users were invited to watch a single live stream encoded in three different qualities. The best quality was set to 1.4 Mbps. In our experiment, users were not always connected to the same live streams at the same time, and they might not have been watching the same quality. Both the number and the distribution of users between the qualities are important in PMS and PMS+. Moreover, our live streams were composed of full-HD and 4K videos with a maximum bitrate of 8 Mbps, significantly higher than the 1.4 Mbps proposed by the authors of SmoothCache. This has en important impact for peers connected e.g., with ADSL modems. They do not have enough upload bandwidth to deliver high-bitrate content whereas with smaller bitrates, it would be easier for them to contribute by sending small segments.

Specific behaviors related with webcams. We observed very specific behaviors related with video streaming from webcams. The average viewing time per user is around 3 minutes which is very short. Because every camera presents a fixed point of view, users do not spent a lot of time in front of a single stream. Instead, we suspect that they are more likely to go from one webcam to another to compare the weather or the traffic of the highway. Due to the initialization phase of PMS+ and the slow quality upgrades of PMS, the clients sometimes do not have enough time to create reliable connections with their neighbors, reducing the potential P2P offload of the system. During the experiment, we also discovered that a lot of users were not closing the tab in their browsers when they were leaving the websites. The authors of [Dubroy and Balakrishnan, 2010] have studied this behavior and reported that the number of tabs open by a user is often between 5 and 15. This number goes up to 30 for some users. When a video client is running in a non-focused tab, the peer is still technically active and connected with others but it is not downloading segments any longer, hence creating confusion in the whole process. In addition, when the user finally decides to reopen the tab, the interruption might be reported as a video stall.

Considering the elements discussed in this subsection, it appears that the results of our experiment seem consistent with the state of the art. Moreover, this evaluation has helped to better understand the specificities of live video delivery from webcams.

5.5 Conclusion

In this chapter, we proposed a novel solution for video streaming over the Internet by relying on P2P-compatible adaptive mechanisms and multiple-source capabilities. The system has been developed in two steps. The first one, PMS, consisted in creating different overlays for every video quality. By relying on both global and remote metrics, a multi-source adaptive quality algorithm has been introduced, addressing the reliability issue by monitoring the health of every overlay before allowing a contributing peer to move from one overlay to another. Some drawbacks were identified and a second solution, PMS+, took over. PMS+ considers small groups of peers collaborating to retrieve video content. This system proposes faster adaptive mechanisms and a better utilization of P2P connections with the help of a local organization process between the peers.

Our solutions were evaluated in two leading websites of the public webcam industry. The large scale experiment has showed the benefits of PMS+ in terms of QoE and scalability compared with standard HAS systems and simple P2P clients, especially during peak periods. Moreover, the experiment have highlighted some specificities of video delivery from webcams such as the user behaviors and their impact on QoE. Both the data retrieved and the observations made helped to better understand this use case and improve future P2P streaming systems.

Chapter summary

PMS+ is an innovative hybrid CDN/P2P multi-source streaming system. It embeds strong quality selection and data-exchange optimization.
PMS+ has been evaluated in production environment with real clients and results have shown the solution performs better than SotA solutions for live streams from webcams.

Chapter 6

Multiple-Source streaming over Remote Radio Light Head: a pragmatic video streaming system for future light indoor 5G networks

"A man must make of his life a ladder that he never ceases to climb – if you're not rising, you are slipping down the rungs, my friend"

Andrew Ryan

6.1 Introduction

Today's wireless indoor networks rely on Wireless Local Area Networks (WLAN) and often suffer from congestion and interference, whilst modern building materials are restricting the propagation of Radio Frequency (RF) and an important number of user equipment may be in competition for the available bandwidth. The 5G Internet Radio-Light (IoRL) project aims to provide reliable intra-building wireless networks by using visible light and millimeter wave parts of the electromagnetic spectrum (VLC/mmWave)

The research leading to the results of this chapter was supported by the EU Horizon 2020 program towards the Internet of Radio-Light (IoRL) project H2020-ICT 761992.

in addition with existing WLAN. One of the project's goals is to interconnect with outdoor 4G/5G systems and thus satisfy 5G requirements, namely providing an important bandwidth and a low network latency to every end-user. Several indoor use cases which could benefit from such system were identified in museums [Cosmas et al., 2018a], home buildings [Cosmas et al., 2018c], supermarkets [Cosmas et al., 2017] and train stations [Cosmas et al., 2018b]. One recurring theme in every use case is the need for efficient video delivery solutions.

As a matter of fact, more and more users are consuming high bitrate contents like Ultra HD (UHD) video or immersive media from both mobile and desktop devices [Cisco, 2018]. The project intents to achieve the same goals as future 5G networks in terms of video delivery: being able to reliably send UHD live and on-demand videos with a good Quality of Experience (QoE) to the end-user. Considering the plurality and the heterogeneity of the IoRL network architecture, the reliability can be tough to insure using state-of-the-art single-path video streaming system because the video playback can be stalled in case of a signal loss if a specific wireless path (i.e. a light or a mmWave connection) is temporarily occluded.

By taking advantage of the plural network paths available by design in the IoRL architecture, we propose Multiple-Source Streaming over Remote Radio Light Head (MSS/RRLH), a DASH-compliant end-to-end multiple-path streaming system increasing both the reliability and the QoE of streaming sessions. MSS/RRLH is an evolution of Over-the-Top HAS solutions (such as the MPEG-DASH [Sodagar, 2011] and HLS [HLS, 2017]) derived from previous work on MS-Stream (see Chapter 3, section 1). The system simultaneously uses several paths to download the video segments over HTTP. The proposed end-to-end solution is composed of a MSS/RRLH Server, deployed as a VNF inside an intra-building Home IP Gateway (HIPG) and a MSS/RRLH Client running on the user equipment.

In this chapter, a first section introduces the scenarios related with video streaming in IoRL and the architecture of the system designed during the project. In a second section, we presents the MSS/RRLH solution. After a subsection dealing with the architecture of the system, the main concepts and the different algorithms for VoD and live streaming are detailed. The last section is about an evaluation of the solution. After explaining the lab testbed and the condition of the experiment, the results are analyzed to show the benefits of our contribution.

6.2 IORL Video streaming scenarios and overall architecture

6.2.1 Use cases and video streaming scenarios

The IoRL project deals with four different use cases: (1) in museums [Cosmas et al., 2018a], (2) in home buildings [Cosmas et al., 2018c], (3) in supermarkets [Cosmas et al., 2017] and (4) in train stations [Cosmas et al., 2018b]. Every one of them includes video streaming related scenarios, either directly when a specific video delivery service is identified or indirectly when only basic advertising videos are displayed. The various video streaming scenarios for every use case are detailed below.

Home building. It can be difficult to receive high video quality in 4K and 8K to the TV or to a desktop computer nowadays without a wired optical fiber connection. 5G indoor connections may be useful in buildings to deliver high-quality and also immersive media via mobile devices and virtual headsets. Another topic related to video streaming is live content creation. With the rise of live platforms (e.g. Twitch [Twitch, 2020]), more and more people are willing to live stream not only video game sessions, but also musical performances or specific events requiring mobile capture devices. The high bandwidth and low latency of 5G networks might help for all of these scenarios to both upload and deliver very high-quality videos to every user.

Museum. Large museums may need to offer rich media content to visitors in order to provide additional information on specific artifacts or entertainment for children, for instance. Another scenario to be considered is virtual tourism using headsets and immersive videos. The characteristics of 5G networks and reliable video delivery mechanisms are useful in both situations to deliver the content in high-quality to every on-site or distant visitor at peak hours.

Supermarket. The supermarket owners might benefit from high-quality video streaming too. Videos can be used to provide information or advertisement to shopping users according to their localization in the mall, or a distant scenario when ordering online. Moreover, like in museum scenarios, they could be useful to entertain children.

Train Station. In train stations, reliable video delivery is important for both customers and maintainers. Customers may want to have access to their favorite videos

while waiting for train or during travel. Low latency live streaming and high-quality video conferencing is of a clear importance for remote intervention when an expert is not on site to help the technicians.

6.2.2 Overall architecture and requirements

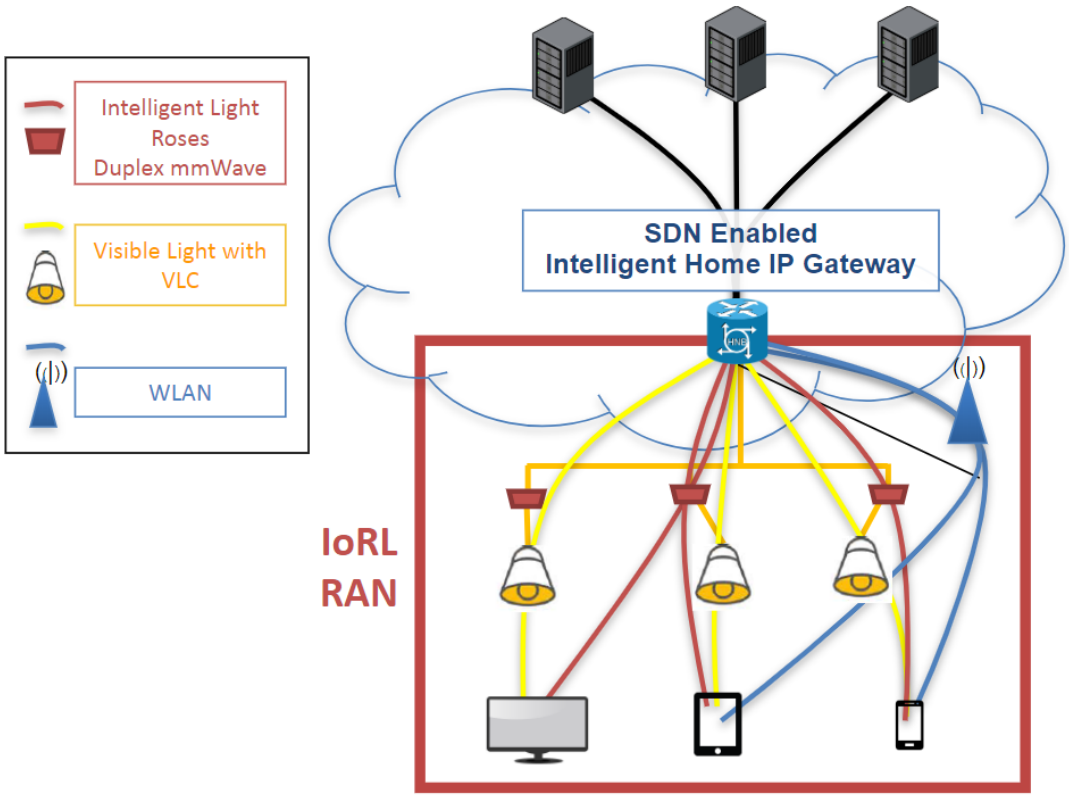

Figure 6.1: IoRL RAN architecture

This subsection deals with the architecture of IoRL. This architecture has been designed by the partners of the project in order to achieve the various use cases mentioned before.

The IoRL network architecture [Cosmas et al., 2018a] [Cosmas et al., 2018c] [Cosmas et al., 2017] [Cosmas et al., 2018b] is depicted in Figure 6.1. The access network is composed of three different transmitters: Visible Light Communications (VLC), mmWave antennas, and WLAN routers. The user equipment can connect to one or several wireless access points to send and receive data packets, considering they are in range. The

heterogeneity of wireless networks calls for innovative methods to quickly and reliably deliver packets through the multiple available paths.

All the wireless transmitters are connected via an Ethernet ring to a Home IP Gateway (HIPG). This HIPG is responsible for running SDN/NFV local modules, i.e. video detection, video transcoding, video delivery, security, and accurate positioning. The HIPG location at the edge of the network is ideal to perform video caching and on-the-fly video transcoding. Figure 6.2 shows the various VNF deployed inside the server. The VNF are stored in a repository and managed by a NFV Orchestrator (NFVO). They are connected to a SDN switch redirecting the packets to one or several virtual functions when necessary. The MSS/RRLH VNF, in yellow, is the contribution presented in this chapter.

The figure also illustrate that, as the name suggests, the HIPG also acts as a gateway to the outside Internet. Video clients send packets through the IoRL radio access network. The data are received by the virtual switch of the HIPG. Inside the server, the data can be forwarded to VNFs if needed. Then, the packets are sent to the Internet either directly or through a 4G/5G antenna. Packet received from the Internet follows the same process in reverse order. Data are received by the SDN switch and can be forwarded to an appropriate VNF, like the security VFN to verify the packets or the transcoder VNF if the data embeds video content. After that, packets are sent to the user equipment by using the high-bandwidth VLC downlink.

6.3 The MSS/RRLH solution

This section presents our contribution to improve video delivery in IoRL: MSS/RRLH.

6.3.1 MSS/RRLH system description and integration in the overall architecture

Multiple-Source Streaming over Remote Radio Light Head (MSS/RRLH) is an end-to-end multiple-path streaming system. It aims to increase both the reliability and the QoE of video streaming sessions for 5G use cases involving multiple network paths. MSS/RRLH is an evolution of DASH solutions and is derived from previous work on MS-Stream (see Chapter 3, section 1). The system simultaneously uses several paths to download the video segments over HTTP. Similar to MS-Stream, the bandwidth available on the different path are aggregated to deliver a better video quality. Moreover, if one path is congested or becomes out-of-range, the data received are still decodable and

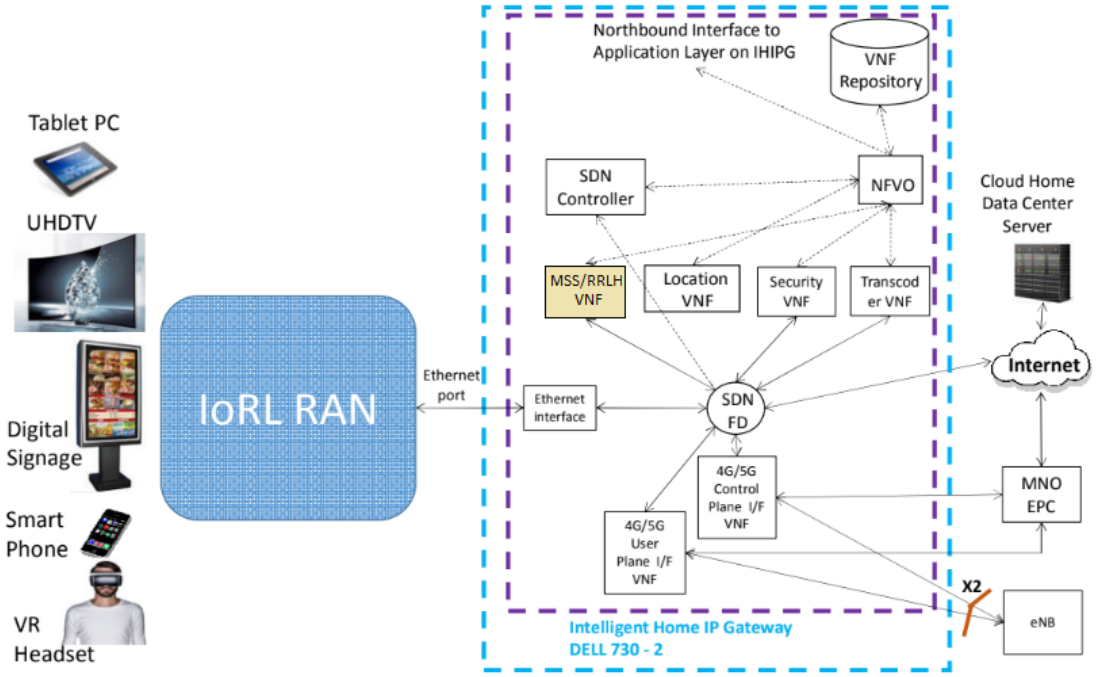

Figure 6.2: IoRL HIPG architecture

the playback is not interrupted. Because VoD and low-latency live streaming are very different use cases, MSS/RRLH embeds specific algorithms to respond to both of them. The proposed end-to-end solution is composed of a MSS/RRLH Server, deployed as a VNF inside the HIPG, and a MSS/RRLH Client running on the user equipment.

The MSS/RRLH Server module is deployed in the SDN/NFV environment of the HIPG. Client devices are connected to the HIPG and the MSS/RRLH Server through VLC/mmWave and WLAN networks, as illustrated in Figure 6.3. This module is responsible for the creation and delivery of video segments in multiple qualities. The input video data are routed through the SDN network to the local MSS VNF where the data are transcoded into MSS/RRLH content in numerous qualities by an embedded MSS Transcoder. The segments contain a very small duration of about 500ms of video in order to reduce the latency induced by the transcoding process. After this operation, the video becomes available for request by the MSS/RRLH Clients through the different network paths available by design in the IoRL project. The input video packets may come from a local camera or an external video stream navigating through the HIPG.

The MSS/RRLH Client is a video player requesting the server through several paths to deliver multiple streams generated from the existing set of content qualities to handle

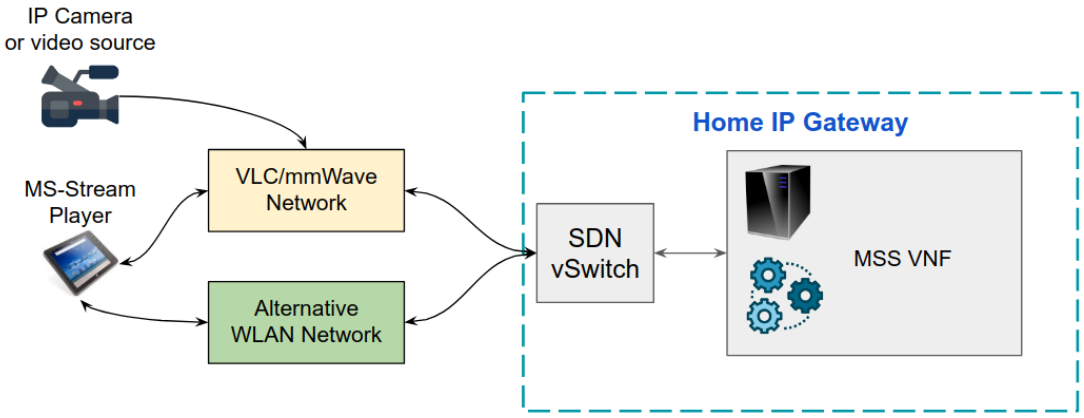

Figure 6.3: Simplified architecture of MSS/RRLH for Live streaming use case

network-path heterogeneity. In the 5G IorL project, the client is used to retrieve video segments in high-quality from multiple-path and simultaneously redundant segments in low-quality to provide reliability in case of an interruption in the data transmission. When retrieved, the requested streams are merged in order to reconstruct and display the original requested content quality.

In the event of a stream loss or outdated delivery, content playback continuity is not affected, only image quality is. Additionally, if the considered network paths experience outages or throughput degradation, the MSS/RRLH client relies on content-adaptation mechanisms to avoid QoE degradation. Thanks to its codec agnosticism and DASH-compliance, this protocol represents an evolving solution that can be applied to many 5G scenarios.

6.3.2 MSS/RRLH in-depth concept

A first version of MS-Stream, a video streaming system with multiple-source capabilities has been described in *Chapter 3*. This system acts as the starting point to design MSS/RRLH.

For every segment, a MS-Stream client assembles individual sub-segment requests, using as many servers as necessary to satisfy its target bitrate. Segments and sub-segments are formed of short (e.g., 0.5 s) video frames sequences gathered into independent units called GoPs. Every video server can serve a GoP in different bitrates, from low-quality to higher-quality versions. A sub-segment assembles GoPs, some in low-quality and others in the high-quality level requested by the client. This allows to obtain a high-quality

version of each GoP from exactly one server, while also requesting this same GoP in low-quality from other servers as a fallback.

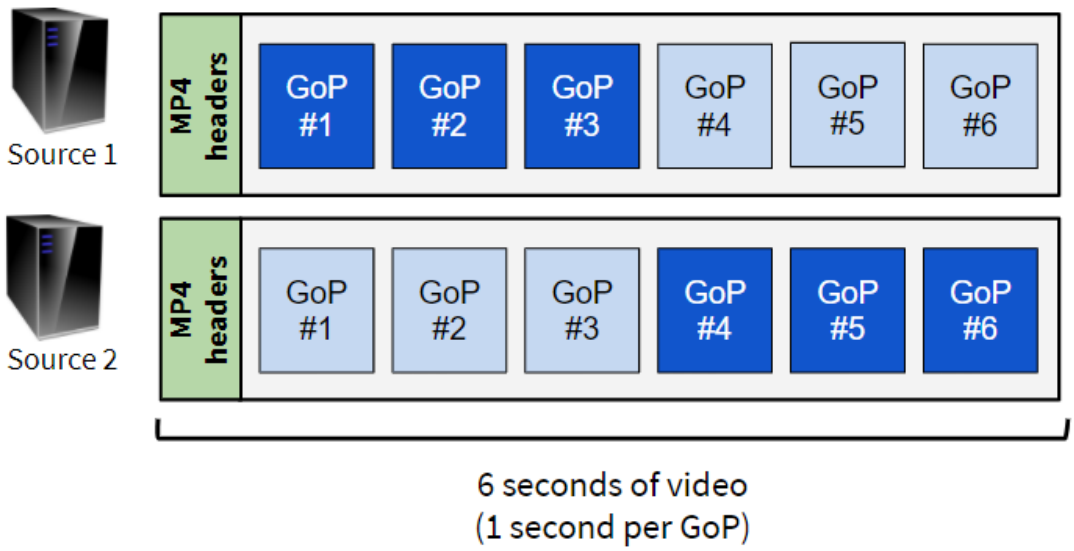

Figure 6.4: Multiple GoPs

Let us take Figure 6.4 as an example and assume that the video segment is formed of 6 GoPs. The client asks the first server, with a bandwidth capacity of 2 Mbps, for GoPs 1, 2 and 3 in high-quality and GoPs 4, 5 and 6 in low-quality. Similarly, the second server with the same bandwidth capacity is being asked by the same client to send GoPs 4, 5 and 6 in high-quality, and GoPs 1, 2 and 3 in low-quality. The client finally assembles a video segment with the highest received bitrate for every GoP and creates a segment with 6 GoPs in high-quality.

Video servers embed a **Description Creator** (Figure 6.5) responsible for the creation of custom subsegments. This specific constraint on the server-side is an obstacle for the adoption of an MSStream-like solution in existing infrastructures that rely on standard HTTP servers. Moreover, the creation of custom subsegments in real time prevents HTTP caching strategies. This is major issue because both current systems and current 5G research topics on video streaming rely on video segment caching to improve high-quality content delivery.

In MSS/RRLH, we propose to move the description creation constraint to the video transcoding process. It is important to point out that every video transcoder on the market has options to configure the size of the segments. The idea of the solution is to

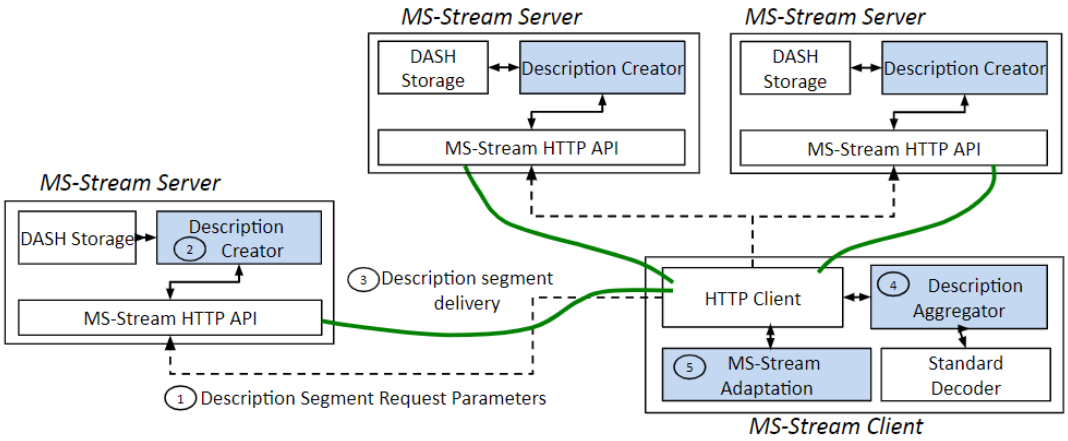

Figure 6.5: MS-Stream logical architecture

set the size of one segment to the size of a single GoP. Hence, the video client no longer selects the GoP to be inserted into a segment but performs the same adaptation logic over several segments at the same time (Figure 6.6). Even better, a new set of rules can be added in the client to select the number of GoPs to be managed during the decision process. MS-Stream has to make decision for a fixed number of GoPs, 6 in our example, because the segments have to keep their original size. On the opposite, MSS/RRLH can adapt the number of GoPs to download for the same decision. For example, it is possible to perform the multiple-source adaptation for 3 GoPs instead of 6 to get them more quickly.

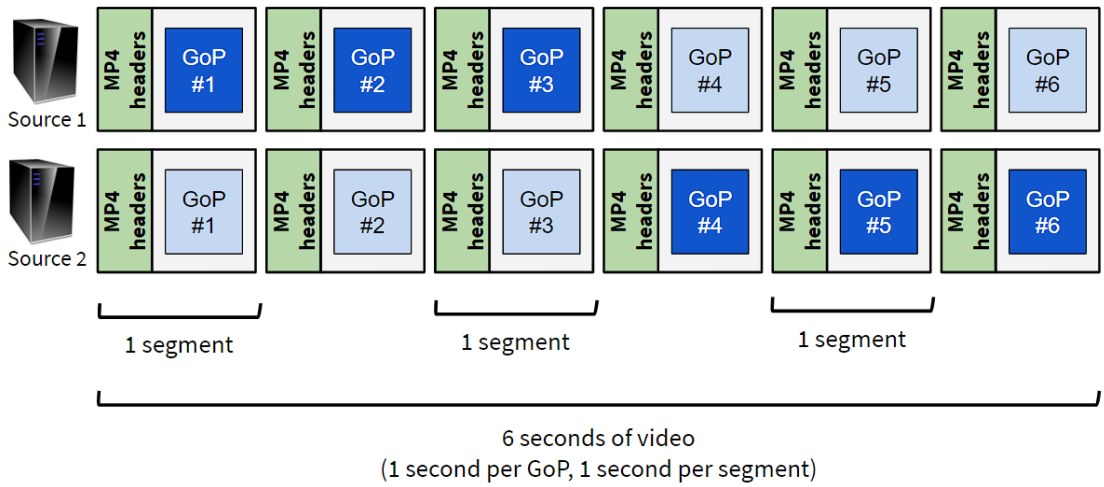

Figure 6.6: One GoP per segment

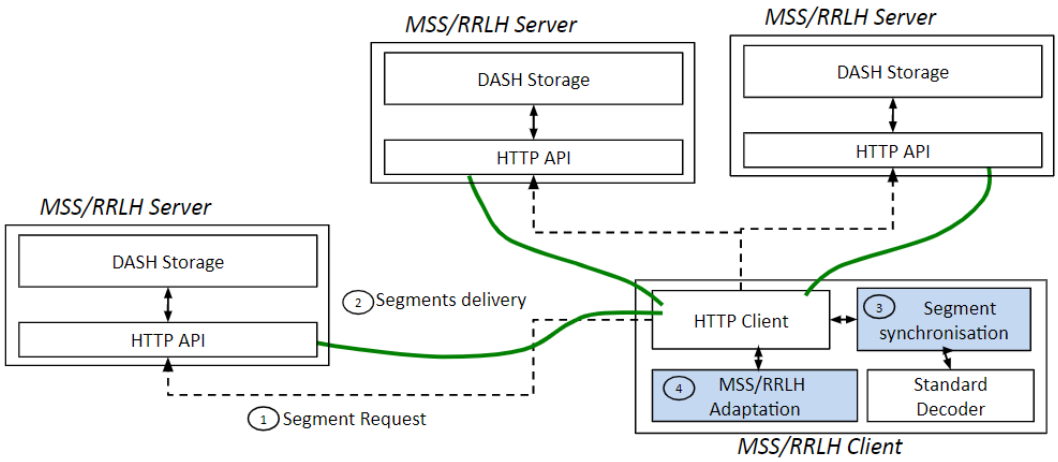

Figure 6.7: MSS/RRLH simplified logical architecture

The new logical architecture is provided in Figure 6.7. The server now can be a classic HTTP server with standard features. This simplification permits to ease the integration of MSS/RRLH into the industry by limiting the constraints on server-side. Because the segments are standard DASH segments, HTTP caching can be effective and deployed.

Nonetheless, the modification described in this section comes with a cost. Comparing Figure 6.4 and Figure 6.6 clearly shows that the number of MP4 headers have been multiplied by **n**, with **n** being the number of GoPs in one segment in the previous MS-Stream system. Even worse, when the segments are downloaded, the number of HTTP requests are multiplied by the same factor, and so is the weight of the HTTP headers.

The cost of the solution is defined as the quantity of data added by both MP4 and HTTP headers in MSS/RRLH. The cost for a quality is defined in Equation 6.1. In this equation, we consider K 6-GoP-long segments and M 1-GoP-long segments for the same number of video frames. $H_{http,k}$ is the size in bytes of the HTTP header and $H_{mp4,k}$ is the size in bytes of the MP4 header for segment k. Finally, $S_{i,k}^{6gop}$ is the size of the video frames for segment k.

$$Cost_i = 1 - \frac{\sum_{k=1}^{K(6gop)} H_{http,k} + H_{mp4,k} + S_{i,k}^{6gop}}{\sum_{m=1}^{M(1gop)} H_{http,m} + H_{mp4,m} + S_{i,m}^{1gop}} \tag{6.1}$$

To evaluate the cost, a simple experiment is conducted. The goal of this experiment is to compare the quantity of data downloaded by the client with MS-Stream and with MSS/RRLH. Several videos are transcoded in four different qualities: (Q1) 854x480p at

0.7Mbps, (Q2) 1280x720p at 2Mbps, (Q3) 1920x1080p at 5Mbps, and (Q4) 3840x2160p at 10Mbps. Every video is segmented in two different versions. In the first version, 6-GoP-long segments are created. In the second version, the output is composed of 1-GoP-long segments. Every segment of the videos is downloaded in a browser for every quality.

An average value of the cost for every quality is given in Table 6.1. In the lowest quality, GoPs themselves are lighter because every frame is encoded using fewer bytes. Hence, the impact of MP4 and HTTP headers is bigger. In a full-HD or 4K video, the cost is lower than 1%. When every quality is fully downloaded, the average cost is around 1%.

Table 6.1: Average cost when transcoding with 1 GoP per segment instead of 6 GoPs per segment for every quality

Qualities downloaded	Q1 480p/0.7Mbps	Q2 720p/2Mbps	Q3 1080p/5Mbps	Q4 2160p/10Mbps	Every quality
Average cost (in %)	6.41	2.3	0.92	0.48	1.07

To conclude, the transition from using multiple GoPs per segment in MS-Stream to using smaller segments with a single GoP per segment in MSS/RRLH offers new opportunities in terms of compatibility with current systems, video caching and quality adaptation possibilities. The only drawback is an additional 1% overhead of downloaded data which is quite low for an UHD video streaming use case, especially in regards with what is offered by the solution. It is still worth to note that the cost is higher for very low qualities.

6.3.3 MSS/RRLH mechanisms for VoD

This subsection presents the algorithms developed for Video-on-Demand streaming use case. In this use case, videos are transcoded offline and all the segments are available before the beginning of the streaming session. Hence, the video player is able to maintain a large buffer of video data.

The MSS/RRLH client embeds several mechanisms to select the quality and distribute the data among multiple network paths. The adaptation capabilities is performed in a two-phase protocol. The first phase consists of prior-download adaptation decisions for the upcoming streams. The goal of this step is to create the requests to be sent to the different sources. This first phase includes overhead reduction and GoP number management mechanisms. The second phase consists in performing in-segment download

adaptation so as to ensure smooth video playback. Figure 6.8 shows the main components of the mechanisms described in this section.

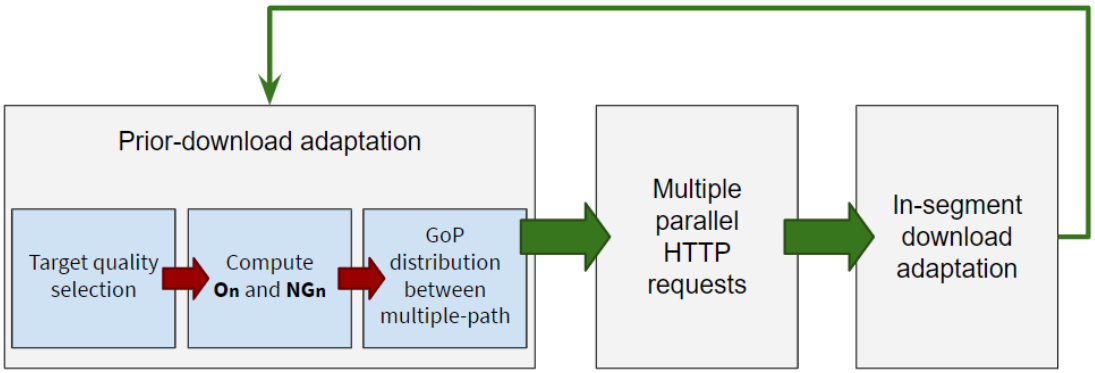

Figure 6.8: Overview of MSS/RRLH mechanisms for VoD

Prior-download adaptation. The prior-download adaptation algorithm has been designed to create the video segment requests that will be sent to the HTTP server(s). First, the bandwidth observed on every path during the download of the last few segments is analyzed and summed in order to select the most appropriate video quality through the most promising path. This quality is called the target quality and is defined as the one that should be displayed to the user by the video streaming system. Second, the player is willing to select the streams to request for every other network path.

The objective of this second step in the 5G IoRL wireless network is to distribute the target quality GoP and the redundant ones to be downloaded between the different paths, according to their bandwidth. Examples of GoP distribution are provided in Figure 6.9. In most of the case, a majority of target quality segments are expected to be requested from the VLC/mmWave network because of its important bandwidth. In this scenario, most of the redundant segments are downloaded from the WLAN. But in specific situations, for example if the end-user is positioned at the limit of the VLC/mmWave range where the throughput is lower, the WLAN network could have a better bandwidth and the target quality GoP distribution could be 50-50. Once the downloads are started and if the connection with the VLC/mmWave network is lost, the player should be able to use the segments received from the WLAN to display a reliable video streaming experience at the cost of visual quality. However, to reduce the amount of useless data downloaded and to optimize the multi-path adaptation, this second step also embeds overhead reduction and GoP number management functionalities.

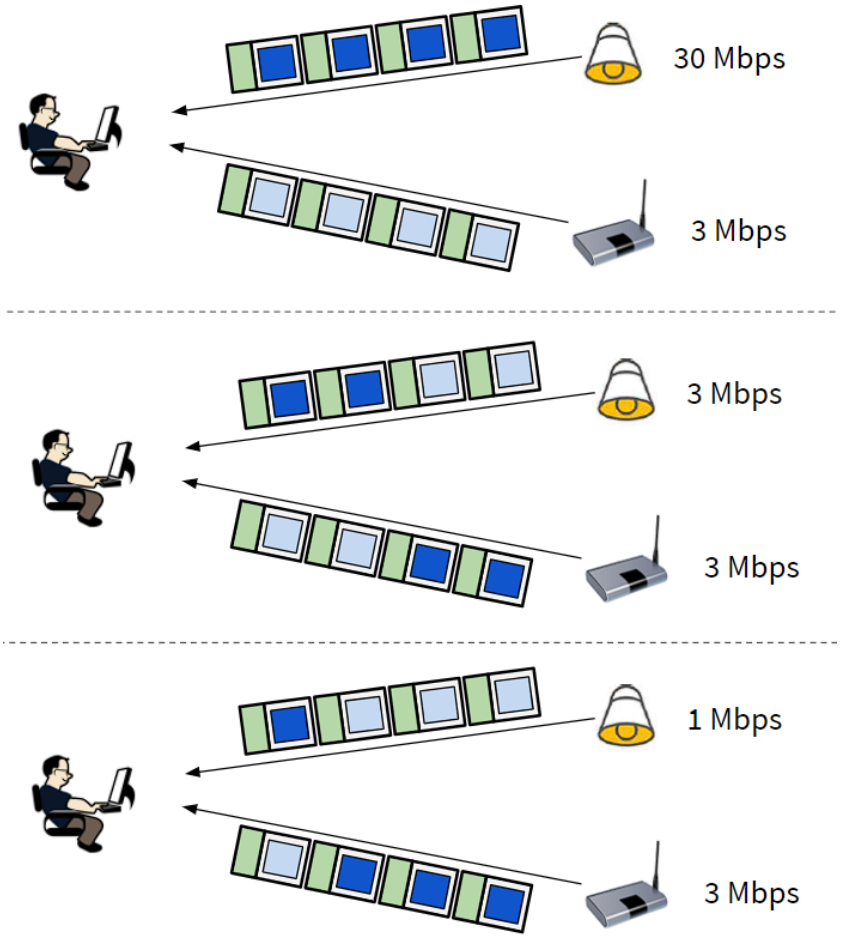

Figure 6.9: GoP distribution according to the bandwidth of the network path

Overhead reduction (OR). Downloading a low-quality GoP and a high-quality GoP at the same time can ensure the reliability of the streaming session but comes with a drawback. As a matter of fact, a part of the video data downloaded is not used. The streaming session hence comes with an *overhead*. The overhead O is defined in (Equation 6.2) where D is the quantity of data in bytes.

$$O = \frac{D_{unused}}{D_{used}} \tag{6.2}$$

MSS/RRLH for VoD tackles this bandwidth consumption overhead problem by limiting and minimizing the number of low-quality GoPs requested. The approach is based on the fact that providing resiliency to the streaming session is not always profitable

in improving the end-users' QoE, especially when the buffered content allows sufficient time to react to impairments that can affect the streaming session's continuity or the displayed video quality.

We propose to limit the overhead by lowering the amount of low-quality GoPs according to the buffer occupancy. This solution is called overhead reduction (OR). It is made possible by the modifications presented in the previous section. For instance, with 6 GoPs per segment, it is not possible to remove some of them while maintaining independently decodable segments. But with segments containing only 1 GoP and decisions made over several segments, it becomes possible to not request some low-quality segments when they are not necessary, as illustrated in Figure 6.10. When the buffer is low, receiving segments is critical to avoid video stalls. Hence, redundant segments are requested. However, when the video buffer gets bigger, the video player have the time to retry a failed request and redundant segments are not necessary. This behavior is illustrated in Figure 6.11.

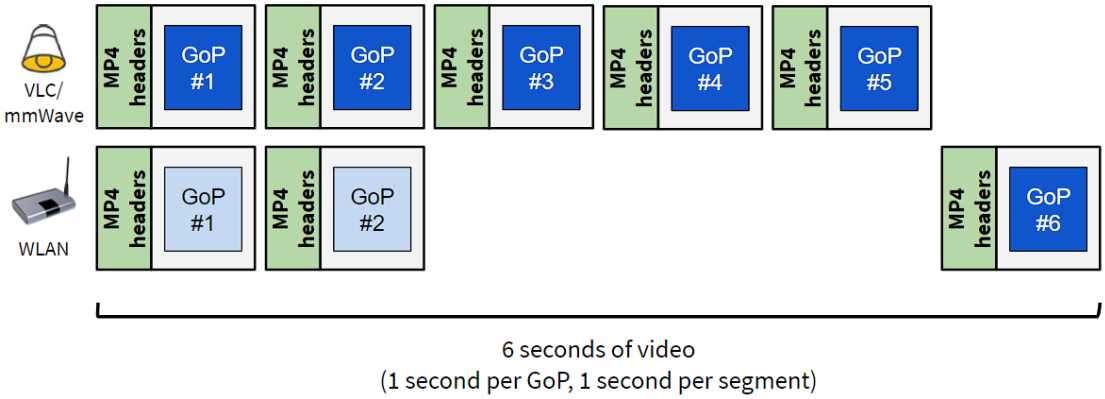

Figure 6.10: Multi-source decision with a lowered amount of low-quality GoPs

For every segment n to be retrieved, the MSS/RRLH client adjusts its bandwidth overhead percentage $O_n\%$ according to the buffer occupancy *bufLeveln* before the download of the segment. Figure 6.11 and Equation 6.3 expose the proposed relation between the buffered content and the low-quality data percentage selection. A maximum level of resiliency is ensured by requiring a maximum percentage O_{max} of low-quality GoPs when the buffer level is below a predefined lower bound ϵ. Similarly, a minimum percentage O_{min} of low-quality GoPs is set when the buffer level exceeds a given upper bound σ. This value of O_{min} could be 0. In this situation, the client does not download low-quality GoPs at all when the stream is healthy. Finally, when the buffered content duration is between σ and ϵ, the value of O_n is a decreasing linear function of *bufLeveln*.

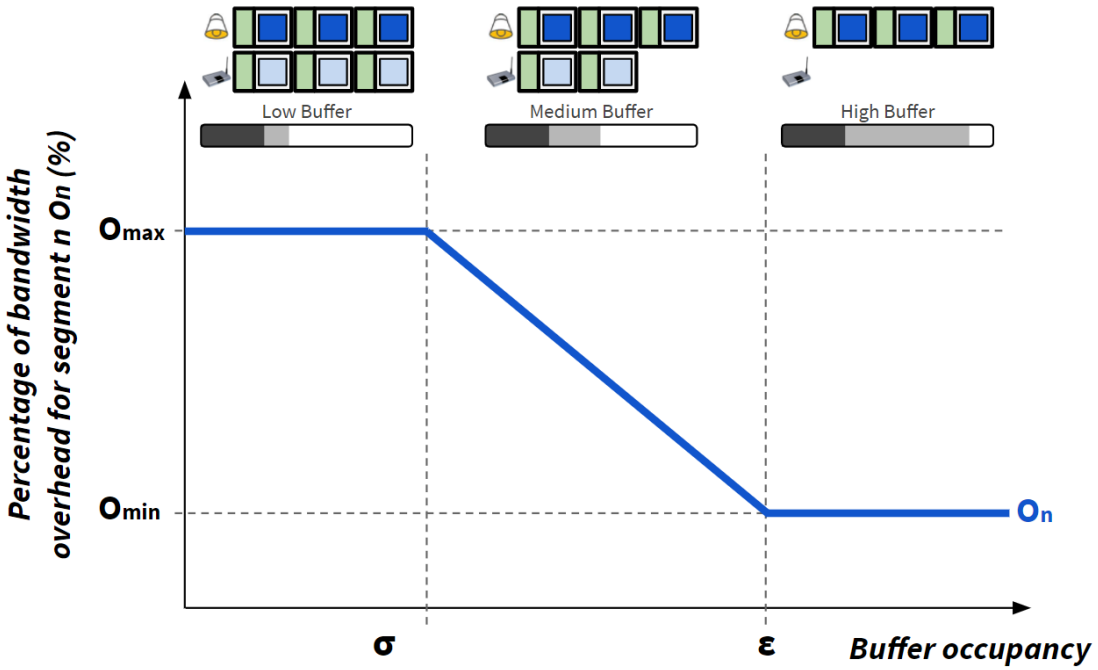

Figure 6.11: Relation between overhead selection and buffer occupancy level

$$O_n = \begin{cases} O_{max} & \text{if } bufLeveln \leq \sigma \\ \frac{O_{min}-O_{max}}{\epsilon-\sigma}.bufLeveln + \sigma.O_{max} - \epsilon.O_{min} & \text{if } \sigma < bufLeveln \leq \epsilon \\ O_{min} & \text{if } \epsilon \leq bufLeveln \end{cases} \quad (6.3)$$

In Figure 6.11, we can see that when the buffer is low, a redundant low-quality version of all the GoPs is retrieved to ensure the reliability of the video stream. Once enough buffer have been received, the overhead percentage becomes lower and only some low-quality GoPs are downloaded. Last but not least, when the video buffer is high, the overhead percentage is minimal and redundant GoPs are not needed anymore.

Once the overhead percentage is obtained, the number of low-quality GoPs to be requested is computed. The low-quality GoPs are positioned in priority to duplicate the first GoP which is the most critical to ensure a smooth playback. An example of GoP repartition is proposed in Figure 6.10.

GoP number management. In MSS/RRLH, the client selects the number of GoPs to be downloaded at the same time NG_n. MS-Stream has to make decision for a fixed

number of GoPs because the segments have to keep their original size. If a segment is composed of 6 GoPs, MS-Stream must perform a multi-source adaptation mechanism using 6 GoPs. On the opposite, MSS/RRLH can adapt the number of GoPs to be considered by the multi-source adaptation mechanism. As illustrated in Figure 6.12, the multi-source adaptation process can be done using 1, 2, 3 or more GoPs. A higher number of GoPs provides a better adaptation mechanism because the data retrieved from every source can be adjusted more precisely. On the opposite, a lower number of GoPs is less precise but allows to quickly get the segments and proceed with the playback.

Similarly to the overhead, we propose to adapt the number of GoPs according to the buffer occupancy. Hence, the video player is able to make more accurate decisions when the buffer is high and to download video data faster when the buffer is low. Figure 6.13 and Equation 6.4 expose the proposed relation between the buffered content and the number of GoPs used in the multi-source adaptation process. A maximum level of resiliency is ensured by requiring a minimum of GoPs NG_{min} when the buffer level is below a predefined lower bound ϕ. Similarly, a maximum number NG_{max} is set when the buffer level exceeds a given upper bound η. Finally, when the buffered content duration is between ϕ and η, the value of NG_n is a increasing and floored linear function of $bufLeveln$.

$$NG_n = \begin{cases} NG_{min} & \text{if } bufLeveln \leq \phi \\ \lfloor \frac{NG_{max}-NG_{min}}{\eta-\phi}.bufLeveln & \\ +\phi.NG_{min} - \eta.NG_{max} \rfloor & \text{if } \phi < bufLeveln \leq \eta \\ NG_{max} & \text{if } \eta \leq bufLeveln \end{cases} \tag{6.4}$$

In Figure 6.13, we can see that when the buffer is low, only one GoP is retrieved on each path. Therefore, the download is faster and the buffer is filled quickly. Once enough buffer have been received, the number of GoPs NG_n becomes higher and more GoPs are downloaded at the same time. Last but not least, when the video buffer is high, the number of GoPs requested in parallel is maximal and GoP distribution between the network path is more precised.

In-segment download adaptation. In addition to prior-download adaptation established in the first phase, the second phase introduces in-segment download adaptation. Its goal is to ensure the reliability of the video delivery so that no pause during the playback occurs. By simultaneously retrieving streams from several network paths, the probability to receive at least one stream is increased. Nonetheless, streams synchro-

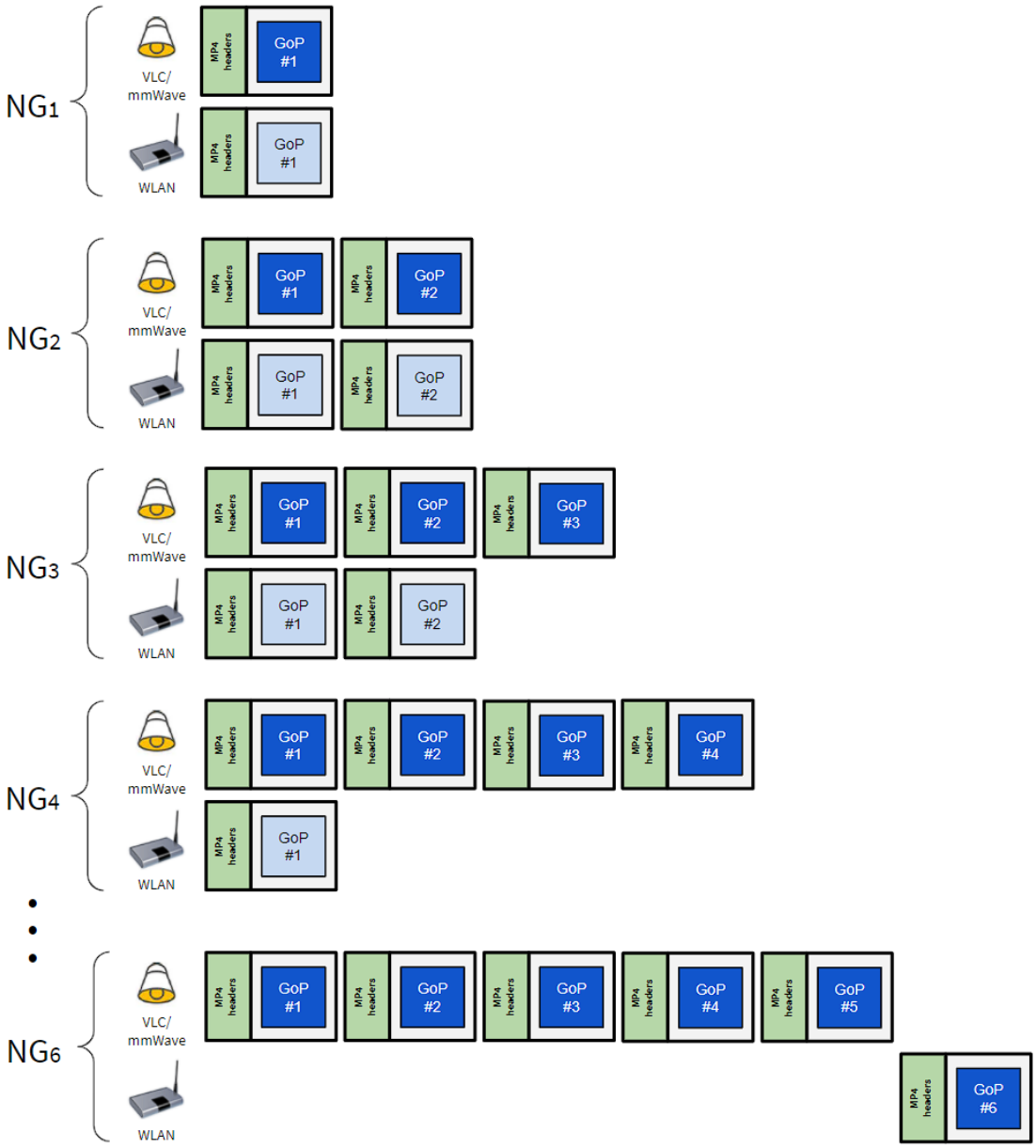

Figure 6.12: Example of GoP number adaptation

nization needs to be resilient to network heterogeneity and to avoid blocking events. A set of rules has been designed for this synchronization at client side: (a) If at least one segment is retrieved, then other downloads of the same segment could be abandoned (depending on the buffer); this reactive rule ensures the delivery of at least one segment before moving on to the next one.

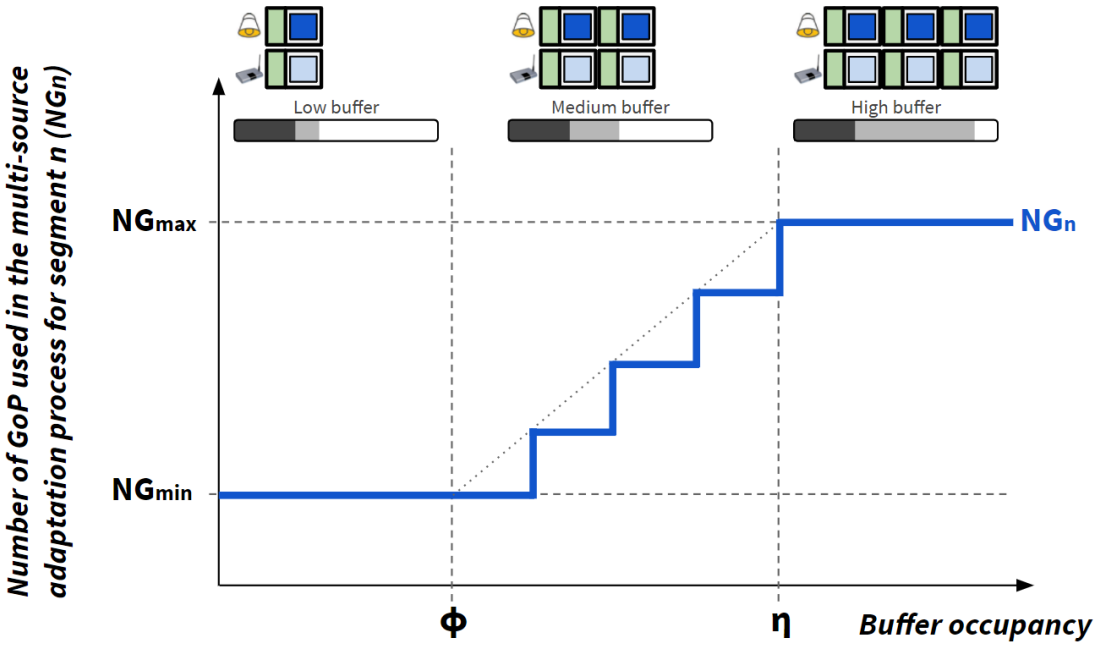

Figure 6.13: Relation between the number of GoPs used and the buffer occupancy level

(b) If the retrieved segment is at the target quality, then other downloads of the same segment are aborted. This rule helps reducing the overhead by aborting useless requests.

(c) If the retrieved segment is at the lowest quality and if the buffered content playout reaches a given lower threshold, then the target quality request is aborted. In doing so, uninterrupted video streaming experience is ensured to the end-users, temporarily providing a sub-optimal visual quality to the end-users.

(d) If the content playout duration available in the client's buffer exceeds twice the average video duration of the NG_{max} GoPs used by the adaptation mechanism, then a timeout value is set on HTTP segment requests. The timeout value reflects a consumption behavior (aggressive, conservative, etc.) and is tuned during the streaming session, according to the available buffered content. Once the timeout has elapsed, segment requests are canceled while satisfying rules (a), (b) or (c). This proactive rule enables the usage of the buffer to compensate for network characteristic fluctuations on the different paths used. The threshold value of twice the average segment duration is arbitrarily chosen in order and allows MSS/RRLH to maintain a high buffer occupancy level in case the deliveries of some segments are delayed.

6.3.4 MSS/RRLH mechanisms for low-latency live streaming

This subsection presents the algorithms developed for low-latency live streams. Unlike what has been presented for VoD, video segments are created on the fly during live streaming sessions. Hence, the video buffer is lower and specific adaptation algorithm have to be developed.

The video delay is an important concern with live streaming. In an ideal world, the user would like to have no delay and watch the stream in real time. However, in practice, due to video transcoding, as well as number of processes and factors at every step of the video delivery process, this low delay constraint is a challenging objective to accomplish.

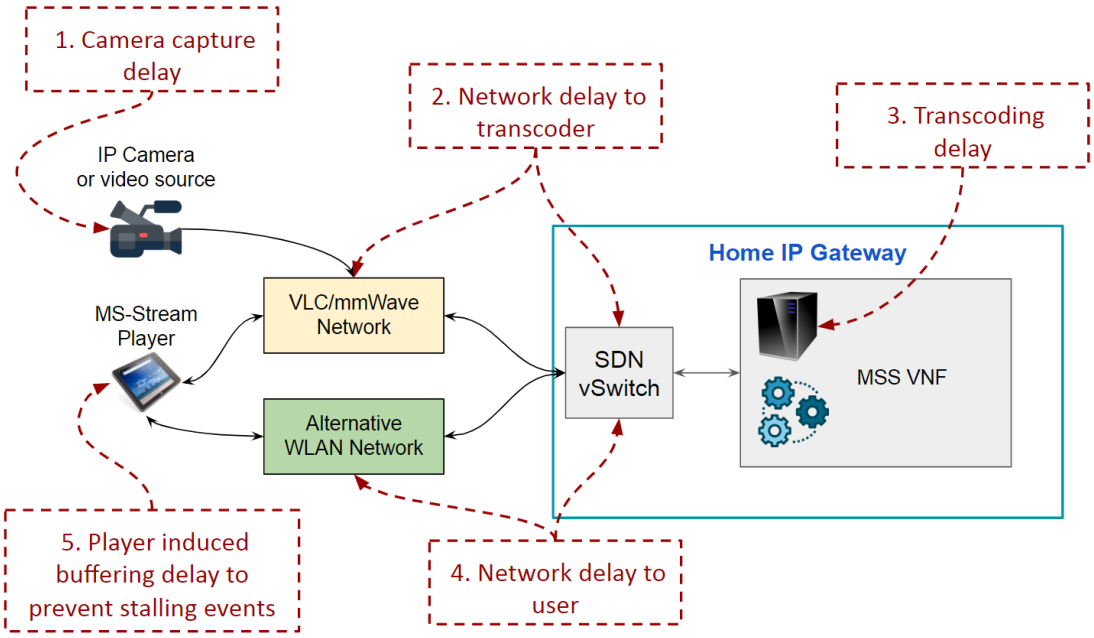

Figure 6.14: Different components of the end-to-end live latency

Figure 6.14 explains where the video delay, called end-to-end video delay, will come from in the overall architecture. The end-to-end delay can be defined as the delay between the time an action happens in reality and the time the same action is displayed to the user terminal. The **camera capture delay** is the delay needed by the camera to capture the picture and to send it to the network. Usually, this first delay is reduced to the minimum by the manufacturers. The **network delay to transcoder** is the delay induced by the transport through the network in order to send the data to the transcoder equipment itself. This delay is often similar with the network delay to the user equipment and both network delays are supposed to be reduced to the minimum by design in the IoRL

RRLH/mmWave network. The **transcoding delay** is the time needed to transcode the stream and create video segments in several qualities. On top of the computation time, this delay can be significant if long segments are used. For example, if the segment is 3 seconds long, then the transcoder is going to buffer the video data during 3 seconds and to introduce a 3 seconds delay minimum as a consequence. Finally, the buffering delay is the delay introduced by the user equipment depending on the video streaming protocol. The goal of the buffering delay is to prevent pauses in the playback by waiting for more data to increase the reliability in case of an unexpected loss of bandwidth. Popular HAS streaming solutions, like DASH and HLS, are known to save a lot of video buffer by default, hence they display the video frames with a 30 to 60 seconds delay. As discussed in the previous section, the above mentioned video buffer is of a clear importance when it comes to insure QoE in case of network bandwidth variations. Video buffer can help to prevent stalling events and reduce the number of quality changes.

Low-latency live streaming video delivery differs from video-on-demand delivery because the time constraint is more important and video players cannot build a video buffer composed of several video segments. However, the adaptation capabilities of MSS/RRLH for live streaming are similar to VoD ones. In a prior-download adaptation mechanism, the player computes the target quality, applies overhead reduction, and distribute the requests among the multiple paths. During the downloads, a in-segment download adaptation mechanism ensures the synchronization of the requests and the reliability of the streaming session.

Prior-download adaptation. In a first step, the bandwidth observed on every path during the download of the last few segments is analyzed in order to select the target quality. In a second step, the GoP requests are distributed between the different paths. The target quality requested from the most promising path and a redundant GoP from the other path. In order to maintain low latency, the prior-download adaptation process of live streams does not embed any GoP number management and makes decisions for 1 segment at a time.

Overhead reduction (OR) Similarly to VoD use cases, downloading the same segment in various qualities from several paths produces an overhead. However, this overhead cannot be diminished using the same buffer-based solution because of the low latency constraint. In order to reduce this overhead, we propose an algorithm deciding for every prior-download decision if a low-quality segment should be downloaded in par-

allel of the target quality segment. The Overhead Reduction (OR) algorithm for live streaming is detailed in **Algorithm 5**. The main idea is to estimate the next RRLH bandwidth using the last measured ones. If this bandwidth estimation is lower than the second lowest quality or if the bandwidth seems to decrease rapidly then it is safer to download an additional low-quality segment from the WLAN network.

Algorithm 5 Overhead Reduction (OR) for live streaming: is a low-quality request necessary to guarantee the reliability?

Input: B_Qi the bitrate of quality i, with i=0 the lowest quality.
Input: RBW_{T-n} the bandwidth of the RRLH path mesured n segments before.

$addLowQuality = false$
if $RBW_{T-1} \leq B_Q1$ **then**
  $addLowQuality = true$
else
  $\Delta RBW = RBW_{T-1} - RBW_{T-2}$
  $nextRBW = RBW_{T-1} + \Delta RBW$
  if $nextRBW \leq B_Q1$ **or** $nextRBW < \frac{2}{3}RBW_{T-1}$ **then**
    $addLowQuality = true$
  end if
end if
return $addLowQuality$

In-segment download adaptation. During low latency live streaming sessions, the video player does not have a lot of time to synchronize multiple requests and compute an advanced set of rules. Hence, the rules for live sessions are simpler than VoD ones:

(a) For a given segment, if the target quality is retrieved, then the other requests are abandoned;

(b) If the request duration reaches a given threshold, the target quality stream download is canceled in favor of the low-quality one in order to ensure uninterrupted video experience.

6.4 Evaluations

The proposed MSS/RRLH system was evaluated in order to study its impact in terms of QoE for mobile users. The QoE refers to the subjectively perceived quality by the end-users. The most important QoE criteria for high-quality and low-delay live streaming sessions are: (a) the end-to-end live delay, (b) the number and duration of video stalling

events due to rebuffering, and (c) the average displayed video bitrate. The objective is to deliver a stream in the highest possible bitrate/quality with the lowest number of stalling events. In live streaming sessions, another objective is to preserve a low end-to-end delay between what is captured by the camera and what is displayed to the end-user terminal.

For the evaluation, we considered both the use case of a live stream acquired from a camera and the use case of a VoD already transcoded and stored on the server. The video data are sent from the camera to the MSS/RRLH system and delivered through multiple paths to the end-user terminal. The user expects to watch the live content with a small delay and a high QoE while he is moving from one VLC/mmWave transmitter to another.

6.4.1 Experimental setup

The video source is acquired from a 4K camera. The video data from the camera are sent to a Virtual Network Function (VNF) running on a server in a local area network. For the VoD use case, a video file is uploaded in advance directly into the same VNF. The server is fitted with a 7th generation Intel Core I7 processor, 16GB of RAM, and an Ubuntu 18.04 OS. The VNF is deployed as a Virtual Machine (VM), embedding several docker containers. Those containers are running the two main modules defined in the contribution: (a) a video transcoder using **ffmpeg** to transcode the input stream in four several qualities and (b) a HTTP server delivering the video segments to the video players. The resolutions and bitrates of the video qualities are: (Q1) 854x480p at 0.7Mbps, (Q2) 1280x720p at 2Mbps, (Q3) 1920x1080p at 5Mbps, and (Q4) 3840x2160p at 10Mbps.

The MSS/RRLH video player has been implemented on top of the open-source dash.js player [DASH-IF Player, 2020]. This player is connected to the VNF through two virtual network paths (Figure 6.15). The first path is simulating the IoRL VLC/mmWave network with variable bandwidth. This bandwidth is modified periodically using a traffic shaper according to different network traces of the VLC/mmWave system and different scenarios. The second path is simulating a WLAN network with a constant 1 Mbps.

6.4.2 Evaluation scenarios

The MSS/RRLH video player is evaluated against two players and according to two pragmatic scenarios.

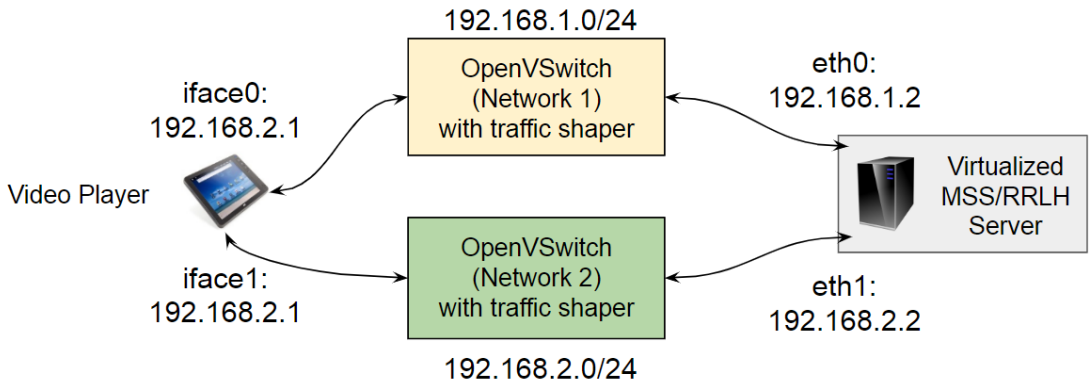

Figure 6.15: Experimental setup: the client is connected to the virtual server through two virtual network paths limited by traffic shapers

Concurrent video players. The first concurrent player is an improved state-of-the-art DASH player. This player is not limited to a one-to-one connection with a server like a standard HAS player but is able to switch between two network paths by periodically sending a request through the unused network. The second concurrent player is a version of MSS/RRLH player but without the overhead reduction mechanism. Therefore, this video player can be described as the greediest multiple-path player possible using the full capacity of parallel networks to reliably retrieve the best quality at the cost of an important overhead.

First scenario. In the first pragmatic scenario, a mobile user is slowly moving under the lights (Figure 6.16). At the beginning, the user is not under a light and cannot use the high capabilities of the VLC/mmWave network. Then, the user walks under the range of multiple VLC/mmWave transmitters. The VLC/mmWave network spatial bandwidth representation can be simulated using Gaussian functions with a peak at 30 Mbps.

Second scenario. In the second pragmatic scenario, an immobile user is consuming multimedia content when the light is suddenly occluded by an obstacle (Figure 6.17). During the first seconds, the user uses the full bandwidth potential of the VLC/mmWave network while, suddenly, an obstacle occludes the signal coming from the VLC/mmWave network. After a period of time, the obstacle is removed and the VLC/mmWave connection is restored.

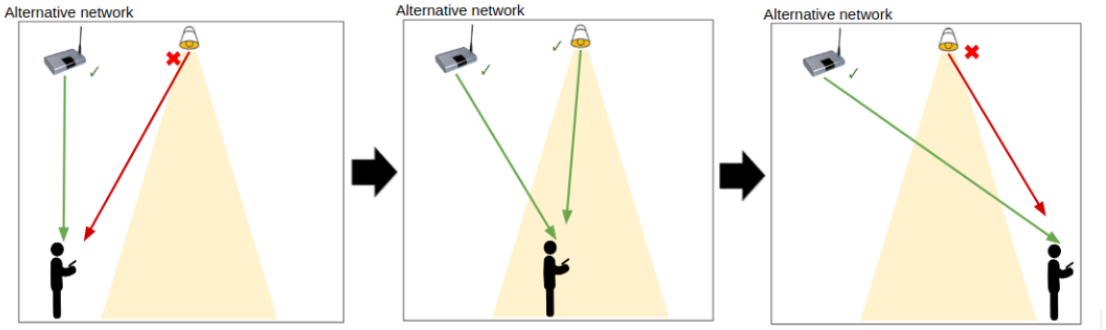

Figure 6.16: Scenario 1: a user is slowly moving behind the lights.

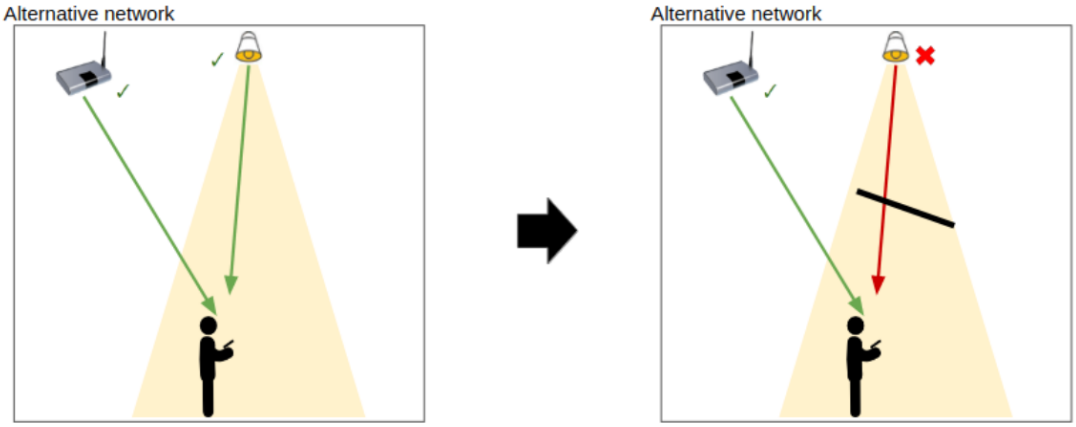

Figure 6.17: Scenario 2: the light is suddenly occluded by an obstacle.

6.4.3 Evaluation results

The results are summarized in Table 6.2. We review the performance of every player with the metrics identified previously, namely the live delay, the reliability, the video bitrate, and the overhead of data downloaded.

Latency. The latency is specific to the live streaming use case. This value is obtained by measuring the time needed to display a video frame after it has been captured. It is important to point out that when the playback is stalled for a few seconds, the live delay is increased by the same duration. Some state-of-the-art players embed mechanisms to maintain a specific latency by periodically seeking to the live point or increasing the playback speed to recover from the interruption. Those mechanisms are not considered in this experiment because they can be inconvenient for specific live streams where

Table 6.2: Summary of results

Video streaming systems	***Average reliable latency (in s)***	***Average duration of stalling events (in s)***	***Average video bitrate variation (in Mbps) MSS/RRLH being the reference***	***Overhead (in %)***
Improved DASH	6	3	- 0.1	0.3
MSS/RRLH without OR	3	<1	+ 0.02	8.1
MSS/RRLH	3	<1	0	2

the end-user expects to watch every second. Instead, the average latency allowing a smooth playback of the live stream is measured. When this latency is reached, the player has enough buffer to avoid the main stalling events due to quick bandwidth decrease. With this experimental setup, the MSS/RRLH reliable latency is around 3 seconds with or without OR, while the improved DASH reliable latency is around 6 seconds. Figure 6.18 compares the live latency of MSS/RRLH with other state-of-the-art player. Our contribution outperforms standard HAS systems. However, low delay solutions using CMAF or HTTP/2 seems to achieve a better latency. The authors of [Bentaleb et al., 2019b] and [Yahia et al., 2019] claims to reliably maintain a latency of 2 seconds during live streaming sessions. But as explain in Chapter 4, these solutions are currently not designed to support multiple heterogeneous paths. WebRTC live video communications have the lowest live delay and similarly, the solution is not compatible with the idea of multiple networks. Moreover, WebRTC live streams have been designed to minimize the latency at the cost of video bitrate and scalability. Video conferencing streams are in low-quality and are challenging to deliver to a lot of users.

Reliability. Representative traces of the video bitrate for every player are available in Figure 6.19 for the 1st scenario and in Figure 6.20 for the 2nd scenario. During the first minutes of every experiment, both MSS/RRLH and MSS/RRLH without OR are downloading the best quality according to the cumulative network bandwidth. The improved DASH player usually suffers from stalling events when the VLC/mmWave network bandwidth is rapidly decreasing. The same behavior is observed during long VoD sessions despite the higher buffer occupancy because the HAS player can not adapt fast enough to some bandwidth variations.

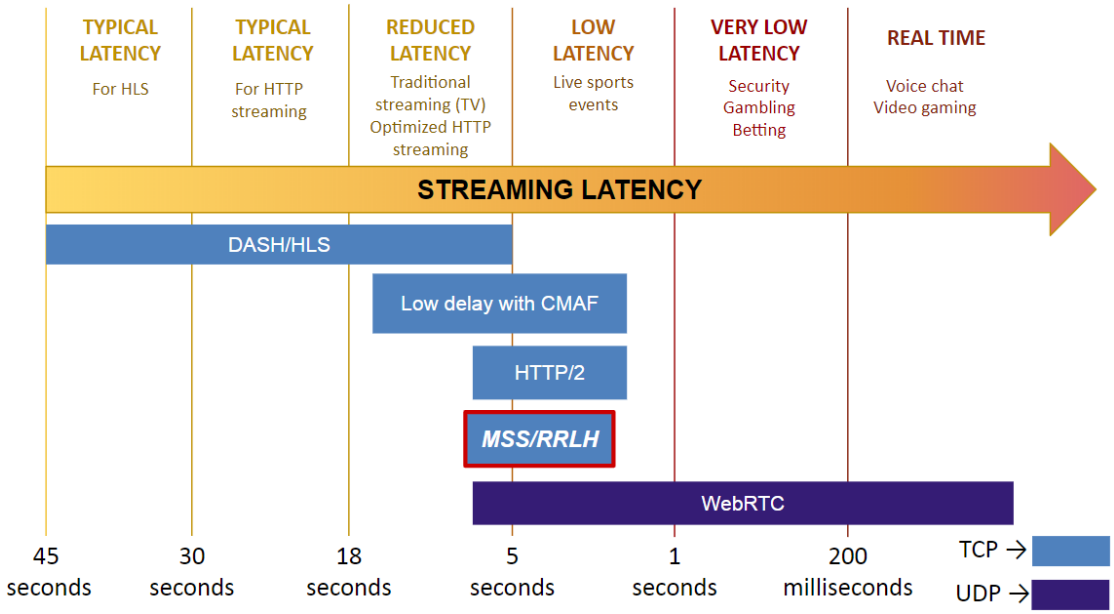

Figure 6.18: Latency range of MSS/RRLH compared with SotA solutions

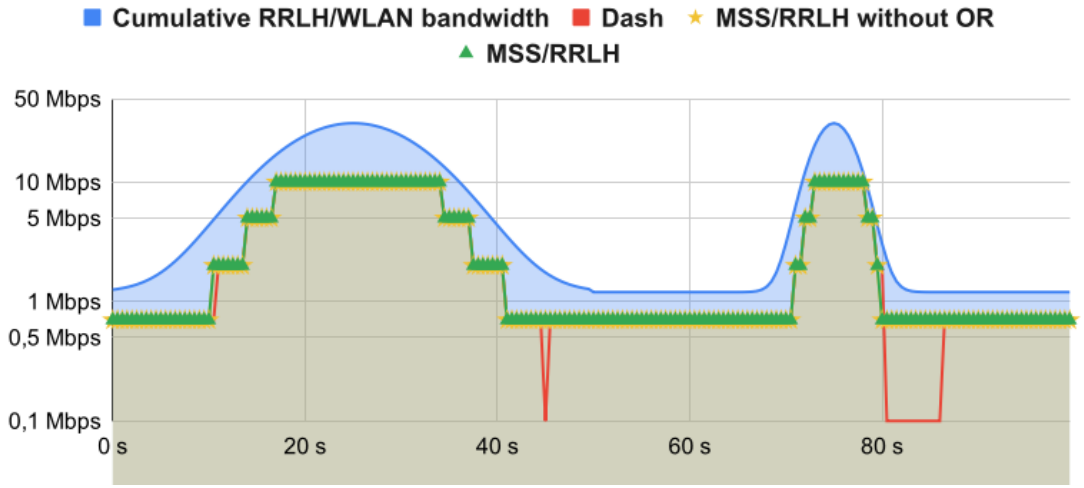

Figure 6.19: Representative traces of the video bitrate received for scenario 1

Video bitrate. In the two scenarios of the experiment, the network bandwidths are intentionally modified and limited. Therefore, the absolute video bitrate is not a relevant value and only the relative bitrate variation between the players is reported. By design, every player is trying to retrieve the best quality according to the VLC/mmWave bandwidth. Hence, the average video bitrates are in the same order of magnitude. The

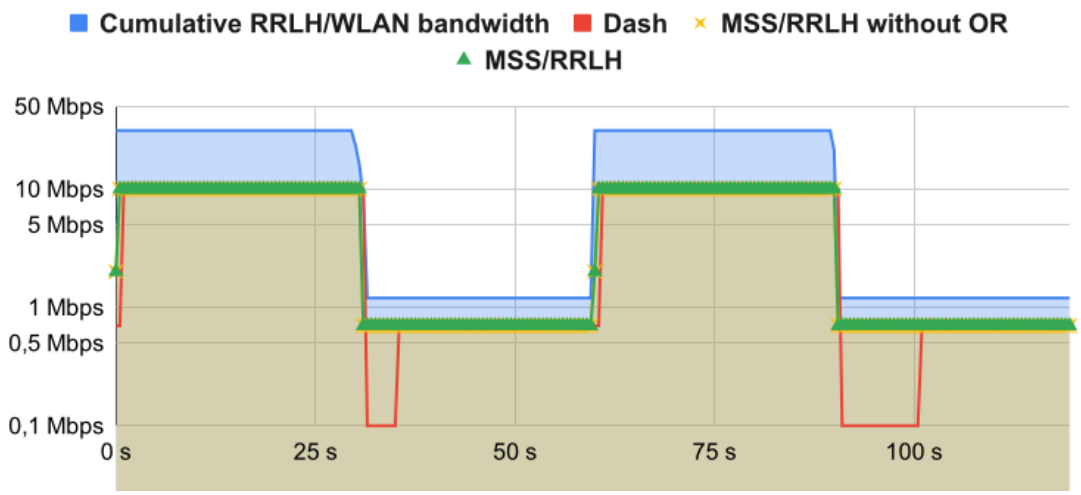

Figure 6.20: Representative traces of the video bitrate received for scenario 2

value is slightly lower for the improved DASH player because it needs one or two more seconds to detect the increasing VLC/mmWave bandwidth.

Overhead. The overhead measurement shows the actual interest of the proposed OR algorithm for both VoD and live use cases. While the greediest multi-path player is downloading 8% more data by consistently getting low-quality segments through the WLAN, the MSS/RRLH solution can reduce this cost to 2% by using the proposed bandwidth prediction mechanism.

To conclude this experiment, MSS/RRLH outperforms state-of-streaming solutions in terms of QoE and reliability. The cost of the solution is a small percentage of data overhead limited by the overhead reduction mechanism. The measured live latency is lower than current HAS solutions, but it is possible to achieve a lower latency by using CMAF or HTTP/2 single path systems. The adaptation of the latter technology inside MSS/RRLH could be an interesting perspective for future works.

6.5 Conclusion

The fifth generation of cellular network is coming with a large variety of new use cases. Considering the important bandwidth and the low network latency promised by 5G, the solution will benefit both outdoor mobile users and indoor devices. The IoRL project focus on indoor networks and proposed to improve current WLAN networks with the

help of visible light and millimeter wave. An intelligent home gateway deployed inside the building will be in charge of routing data traffic through the different path to serve end-users.

Considering the plurality and heterogeneity of networks in the architecture of IoRL, we proposed MSS/RRLH, a video streaming system to reliably deliver high-quality and low delay UHD videos. Our contribution extends and improves previous works on multiple-source streaming. It embeds specific quality and path selection algorithms for VoD and live streaming uses cases. The requests are distributed between the different path according to their observed bandwidth in order to maximize the video quality received while improving the reliability of the system. By design, MSS/RRLH is well suited to be integrated within the architecture of the 5G Internet of Radio Light project, but could be leveraged to any generic use case involving several heterogeneous networks.

The proposed solution has been evaluated in a lab testbed and integrated inside the official demonstrator of the european project. The results outline the impact of MSS/RRLH against state-of-the-art solution. The player is able to maintain an better average video quality in chaotic situation while reducing the number of video stalls. The cost of the solution is an overhead of data downloaded. However, this overhead is minimized by overhead reduction algorithms. In terms of latency, the contribution outperforms HAS video clients by maintaining 3 seconds end-to-end delay. Therefore, some single-path systems proposed in the literature claims to reduce this latency up to 2 seconds.

The results of the experiment opens new perspectives on the subject. Firstly, the solution could be improved by merging the concepts of MSS/RRLH with current CMAF and HTTP/2 systems in order to design an even better multi-source ultra low latency solution. Secondly, the next step would be to continue the development of the solution. Reaching an appropriate technological readiness level would be the first step towards a real product entering the market of indoor video streaming in 5G networks. In particular, this product should be merged with PMS+ to propose a complete video streaming system for current and future networks.

Chapter summary

By extending the multiple-source capabilities of MS-Stream, MSS/RRLH improves the QoE of video streaming sessions in the context of heterogeneous multiple 5G networks. MSS/RRLH has been deployed and evaluated in a lab demonstrator. It offers strong reliability guarantees for a minimal overhead cost while being close to SotA players in terms of live latency.

Chapter 7

Conclusions and Perspectives

"We all make choices, but in the end our choices make us."

Andrew Ryan

This chapter concludes the thesis, gives an overview of the contributions and provides research perspectives.

7.1 Contributions

End-users' quality of experience determines the success and adoption of current and future video streaming services. Video traffic over the Internet will experience a tremendous growth of 118%, and is expected to reach at least 82% of the total Internet traffic by 2022. Given the global trend of increasing video traffic and the issues pertaining to the underlying network architectures, Internet video streaming has gathered considerable attention from both research and industry. Various OTT video streaming systems were deployed over the last few years. These deployments are based on various architectures, mostly cloud platforms and Content Distribution Networks based on HTTP Adaptive Streaming, but also P2P networks and hybrid solutions combining P2P and CDN.

The contributions presented in this thesis can be classified into two main proposals: (1) enhancing QoE and scalability of video streaming solutions by leveraging content quality adaptation and self-scalability of hybrid P2P/multiple-servers streaming systems, and (2) improving reliability and QoE of video streaming sessions in future multi-path indoor 5G networks. In a nutshell, we have made four contributions:

7.1.1 MS-Stream and `Muslin`

As a preliminary work, we introduced **MS-Stream**, an HAS-evolving streaming framework advocating for a client-centric utilization of multiple servers simultaneously. MS-Stream incorporates a client and a server solution. MS-Stream clients simultaneously request several servers to deliver independently decodable sub-segments, created from the existing set of content qualities. Subsegments composition requires a very light preprocessing operation from the servers before sub-segment delivery. When retrieved, the video sub-segments are merged to reconstruct and to display the requested content quality. In the event of sub-segment loss or late delivery, content playback continuity is not affected, only content quality is. MS-Stream was also extended with **`Muslin`** as an answer to CDN-side network and server overload. `Muslin` adjusts the service infrastructure scale and better assigns content servers to clients thanks to periodic feedback.

7.1.2 PMS+

Following on MS-Stream and scalability issues in CDN, we proposed to combine the studied approaches with the P2P paradigm. We came out with **PMS+**, a hybrid P2P/CDN solution for live streaming with scalability and quality adaptation capabilities. PMS+ was developed incrementally through two versions. In the first one, originally called PMS, we put forward a distributed and decentralized quality adaptation algorithm relying upon local and global indicators of the functioning of the P2P data transfers. This new strategy aims at enhancing the end-users QoE while concurrently guaranteeing the successful functioning of the content delivery system. In the second, we improved the data exchange mechanisms by relying on self-organized small group of active peers. The resulting system was totally developed, deployed in a real production environment available publicly to anyone —in partnership with a business client— and evaluated at very large scale. PMS+ was evaluated in a large scale experiment with thousands of users every day during several months. The solution outperformed current HAS systems as well as web-based state of the art P2P streaming solutions.

7.1.3 MSS/RRLH

Considering the plurality and heterogeneity of future intra-building 5G networks, we proposed **MSS/RRLH**, a system to reliably deliver high quality and low delay videos. It relies on edge computing resources and efficient multiple-path quality selection algorithms. MSS/RRLH includes specific mechanisms for VoD and live streaming sessions

such as path selection, buffer-based adaptation and low-latency reliability estimation. The solution has been developed and integrated within the demonstrators of the 5G Internet of Radio Light project. The public demonstration of the full system has been delayed to November 2020 due to 2020 Covid-19 lockdown. However, the system was evaluated in a lab testbed and outperformed current state-of-the-art solutions in terms of video quality and reliability at the cost of a very small overhead of data downloaded.

The research contributions of this thesis have been presented in several international conferences and journals. Additionally, several demonstrators have been developed and awarded several prizes for their scientific and technical excellence. Two final products have also been developed and deployed. The first one has been deployed in production and made available online in two leading websites in the public webcam industry. The second one has been integrated as a proof-of-concept in the official IoRL project demonstrator.

7.2 Perspectives

This thesis opens new perspectives for further work in a short and mid-term line of action.

Users behavior. During the large-scale evaluation of the proposed P2P system, many observations were made concerning users behavior in the context of video delivery from webcams. For instance, a significant number of users do not close their browser tab when leaving the website, and the median playback duration is around three minutes. PMS+ can thus be further improved by speeding up groups creation and managing idle users, both in peer and quality selection algorithms.

Security. One of the main collaborative systems issues in terms of security is freeriding. A freerider is a node that benefits from the system without contributing its fair share to it. Assisting peers could freeride by ignoring or blocking the P2P requests sent by their neighbours. If a large portion of assisting peers behave as such, there might be an impact on the overall QoE. Another security issue if the privacy of streaming sessions. Users generate a history of watched contents when using online video streaming services. The platform provider can then use this data for personalized recommendations for new videos, or for targeted advertising. In P2P systems, malicious peers could have access

to a list of peers watching the same video. This can lead to major threats to privacy as it becomes possible to infer private information about the user, such as his gender, and origin, political, religious or sexual orientation. Protecting users' privacy in a video streaming system requires hiding their access histories from servers and other users.

Both freeriding and privacy issues have been studied and addressed by a fellow PhD candidate. PrivaTube [Da Silva et al., 2019a], a privacy-preserving multiple-source video streaming solution, was proposed to tackle these challenges.

Internet-of-Things. Recently, we have witnessed the emergence of Internet-of-Things (IoT) applications where resources are sparse, highly volatile and distributed. The streaming approach presented in this thesis could easily make use of IoT devices/platforms for the purpose of enhancing QoE. We have started to explore the field of IoT for novel multimedia consumption applications. IoT devices could be viewed as small servers with limited resources. The multi-source capabilities of PMS+ could be adapted to include this specific equipments. Because of their small resources, they could be considered like "super-peers", working similar to standard peers but with a better availability. Adding an IoT device to a group of peers could help them to improve the reliability of P2P exchanges while reducing the quantity of data requested from CDN servers.

User-generated content. Moreover, as social media networks advocate for user-generated content, new challenges rise related to content preparation and production (especially for omnidirectional, virtual reality and augmented reality streaming). New business models emerge, such as rewarding end-users for sharing content or resources to the distributed social media network platform. As an example, users could deliberately choose to share some memory and bandwidth to be part of a P2P group and deliver video segments they are not watching to other peers. Similar to IoT systems described in the last paragraph, they could be integrated in PMS+ as reliable "super-peers" with more availability and more tolerance regarding the delay when they are not watching the stream. Every time their resources are used, a signed feedback could be saved in a dedicated blockchain. Participating users then could be rewarded with cryptocurrencies.

Live latency. The proposed MMS/RRLH is a promising solution to deliver high-quality videos through multiple paths. Both the QoE and the reliability of the system are better than single-path streaming services. However, in terms of live delay, solutions like CMAF [CMAF, 2018, Bentaleb et al., 2019b, Bentaleb et al., 2020a] and HTTP/2 [Yahia

et al., 2019, Wu et al., 2017] seems to achieve a slightly lower latency. Reducing the live latency of MSS/RRLH even further by merging our system with HTTP/2 and CMAF solutions would be an interesting perspective to produce a reliable and very low delay streaming service.

Artificial intelligence. A longer-term perspective to improve video streaming systems could be the use of artificial intelligence. The extensive evaluation presented in this document could help us to extract user behavior traces and available network bandwidth for future experimentation. The same large-scale platform could be used for different applications (e.g. video-on-demand, live event delivery, social network activities, etc.) to create an efficient training dataset for machine learning-based adaptation. However, video players and video streaming applications are very versatile and dynamic environments, with a high churn rate and little reproducibility. To solve these issues, tools such as Reinforcement Learning could be used. Reinforcement Learning allows an agent to discover the right action to take within a specific context based on feedback from its environment. To do so, an adaptation module interacts with its environment by sensing the factors that are expected in-advance to influence its decision. To the best of our knowledge, machine learning and reinforcement learning have never been used in multiple-server and P2P context. Incorporating such a tool into the multiple-source and distributed adaptation logic could be an interesting perspective.

With all the above mentioned perspectives, video streaming in general and our contributions in particular are likely to be improved and extended in future research and industrial works. PMS+ and MSS/RRLH are still at their beginning in terms of innovations.

Appendix A

Publications

1 Published papers in international peer-reviewed conferences

J. Bruneau-Queyreix, M. Lacaud, P. Anapliotis and D. Négru, *"On providing multiple-server support to Dynamic Adaptive Streaming Applications for enhanced QoE,"* in IEEE International Conference on Telecommunications and Multimedia (TEMU), 2016

J. Bruneau-Queyreix, M. Lacaud, D. Négru, J. Batalla, and E. Borcoci, *"QoE Enhancement Through Cost-Effective Adaptation Decision Process for Multiple-Server Streaming over HTTP,"* in IEEE International Conference on Multimedia and Expo (ICME), 2017

J. Bruneau-Queyreix, M. Lacaud, D. Négru, J. Batalla, and E. Borcoci, *"MS-Stream: A Multiple-Source Adaptive Streaming Solution Enhancing Consumers Perceived Quality,"* in IEEE Consumer Communications and Networking Conference (CCNC), 2017

J. Cosmas, M. Lacaud et al., *"A Scaleable and License Free 5G Internet of Radio Light Architecture for Services in Train Stations,"* in European Wireless 2018; 24th European Wireless Conference, Catania, Italy, 2018, pp. 1-6.

J. Cosmas, M. Lacaud et al., *"5G Internet of radio light services for supermarkets,"* in 14th China International Forum on Solid State Lighting: International Forum on Wide Bandgap Semiconductors China (SSLChina: IFWS), pp. 69-73, Beijing, 2017.

S. Da Silva, J. Bruneau-Queyreix, M. Lacaud, D. Négru, L. Réveillère. *"MUSLIN: Achieving High, Fairly Shared QoE Through Multi-Source Live Streaming."* Packet Video Workshop (PV '18).

M. Lacaud, D. Negru, *"Multiple-Source Streaming over Remote Radio Light Head: a pragmatic, efficient and reliable video streaming system for 5G intra-building use cases"*. In proceedings of IEEE Symposium on Broadband Multimedia Systems and Broadcasting, Paris, 2020.

J. Cosmas, N. Jawad, K. Ali, B. Meunier, Y. Zhang, W. Li, M. Gregorczyk, W. Mazurczyk, K. Cabaj, M. Lacaud, D. Negru, S. Cuerva Navas, I. Losas Davila, C. Zarakovitis, H.s Koumaras, M. Kourtis,*"Network and Application Layer Services for High Performance Communications in Buildings"*. In proceedings of IEEE Symposium on Broadband Multimedia Systems and Broadcasting, Paris, 2020.

2 Published articles in international peer-reviewed journals

J. Bruneau-Queyreix, J. Batalla, M. Lacaud and D. Négru, *"PMS: A Novel Scale-Adaptive and Quality-Adaptive Hybrid P2P/Multi-Source Solution for Live Streaming,"* in ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 14, No. 2s, Article 35. 2018.

J. Bruneau-Queyreix, M. Lacaud, D. Négru, J. M. Batalla and E. Borcoci, *"Adding a New Dimension to HTTP Adaptive Streaming Through Multiple-Source Capabilities,"* in IEEE MultiMedia, vol. 25, no. 3, pp. 65-78, July-Sept. 2018, doi: 10.1109/MMUL.2018.112142627.

S. Da Silva, J. Bruneau-Queyreix, M. Lacaud, D. Négru, L. Réveillère. *"MUSLIN: A QoE-Aware CDN Resources Provisioning and Advertising System for Cost-Efficient Multi-Source Live Streaming"*. International Journal of Network Management (IJNM '19).

3 Published demonstrations and posters in international peer-reviewed conferences

J. Bruneau-Queyreix, M. Lacaud and D. Négru, *"A Hybrid P2P/Multi-Server Quality Adaptive Live-Streaming Solution for High End-User's QoE,"* in ACM MultiMedia Conference (MM), demonstration track, 2017

J. Bruneau-Queyreix, M. Lacaud and D. Négru, *"A Multiple-Source Adaptive Streaming Solution Enhancing Consumers Perceived Quality,"* in IEEE Consumer Communications and Networking Conference (CCNC), demonstation track, 2017

S. Da Silva, J. Bruneau-Queyreix, M. Lacaud, L. Réveillère, D. Négru *"MUSLIN demo: High QoE Fair Multi-Source Live Streaming,"* in ACM Multimedia Systems (MMSys), demonstration track, 2018

4 Awards

ICME 2017 Grand Challenge Winner: J. Bruneau-Queyreix, M. Lacaud, D. Negru, *"A Hybrid P2P/Multi-Server Quality Adaptive Live- Streaming Solution for High End-User's QoE,"* in IEEE International Conference on Multimedia and Expo (ICME), DASH-IF Grand Challenge, 2017.

CCNC 2017 Best demonstration award: J. Bruneau-Queyreix, M.Lacaud, D. Negru, *"A Multiple-Source Adaptive Streaming Solution Enhancing Consumers Perceived Quality,"* in IEEE Consumer Communications and Networking Conference (CCNC) – Demonstation track- 2017.

DASH-IF Excellence in DASH Award, third place: S. Da Silva, J. Bruneau-Queyreix, M. Lacaud, D. Négru, L. Réveillère. *"MUSLIN: Achieving High, Fairly Shared QoE Through Multi-Source Live Streaming."* Packet Video Workshop (PV '18).

5 Internet of Radio Light (IoRL) deliverables

Moshe Ran; Adam Kapovits; John Cosmas; Ben Meunier; Kareem Ali; Hongying Meng; Yue Zhang; Li-Ke Huang; Xun Zhang; Chuanxi Huang; Moshe Ran; Einat Ran; Dror Malka; Matteo Satta; Eric Legale; Pascaline Jay; Martin Ganley; Atanas Savov; James Gbadamosi; Rudolf Zetik; Tasos Kourtis; Charilaos Koumaras; Christos Sakkas; Michael-Alexandros Kourtis; Wojciech Mazurczyk; Krzysztof Cabaj; Sibel Malkos; Emre Cakan; Daniel Negru; Mathias Lacaud; Marios Negru; Zion Haddad; Baruch Globen; Eliron Yamina Salomon; Gil Sheffi; Yoav Avinoam; Javier Royo; Jorge Garcia; Eric Legale; Pascaline Jay; Jian Song; Jintao Wang; Min Tong; Xiaohong Cao; Xiao Li; David Sánchez; Pablo Fernandez. 2017. *"D2.1 Definition and Description of the Iorl Use Cases and Derivation of User Requirements"*.

Moshe Ran; Adam Kapovits; John Cosmas; Ben Meunier; Kareem Ali; Nawar Jawad; Mukhald Salih; Hongying Meng; Wei Li; Yue Zhang; Li-Ke Huang; Xun Zhang; Chuanxi Huang; Moshe Ran; Einat Ran; Dror Malka; Eitan Omiyi; Matteo Satta; Eric Legale; Pascaline Jay; Martin Ganley; Atanas Savov; James Gbadamosi; Rudolf Zetik; Tasos Kourtis; Charilaos Koumaras; Christos Sakkas; Michael-Alexandros Kourtis; Wojciech Mazurczyk; Krzysztof Cabaj; Haluk Gökmen; Sibel Malkos; Emre Cakan; Daniel Negru; Mathias Lacaud; Marios Negru; Zion Hadad; Baruch Globen; Rafael Barkan; Eliron Yamina Salomon; Gil Sheff; Yoav Avinoam; Javier Royo; Jorge Garcia; Eric Legale; Pascaline Jay; Jian Song; Jintao Wang; Min Tong; Xiaohong Cao; Xiao Li; David Sánchez; Pablo Fernandez. 2018. *"D2.2 System Functional Requirements and Architecture"*.

Moshe Ran; Adam Kapovits; John Cosmas; Ben Meunier; Kareem Ali; Hongying Meng; Yue Zhang; Li-Ke Huang; Xun Zhang; Chuanxi Huang; Moshe Ran; Einat Ran; Dror Malka; Matteo Satta; Eric Legale; Pascaline Jay; Martin Ganley; Atanas Savov; James Gbadamosi; Rudolf Zetik; Tasos Kourtis; Charilaos Koumaras; Christos Sakkas; Michael-Alexandros Kourtis; Wojciech Mazurczyk; Krzysztof Cabaj; Sibel Malkos; Emre Cakan; Daniel Negru; Mathias Lacaud; Marios Negru; Zion Haddad; Baruch Globen; Eliron Yamina Salomon; Gil Sheffi; Yoav Avinoam; Javier Royo; Jorge Garcia; Eric Legale; Pascaline Jay; Jian Song; Jintao Wang; Min Tong; Xiaohong Cao; Xiao Li; David Sánchez; Pablo Fernandez. 2018. *"D2.3 Functional and Technical Requirements Key Building Blocks"*.

Anastasios Kourtis; Andreas Foteas; Charilaos Koumaras; Charilaos Zarakovitis; Michail-Alexandros Kourtis; Themistocles Anagnostopoulos; Adam Kapovits; John Cosmas; Mukhald Salih; Nawar Jawad; Mathias Lacaud; Krzystof Cabaj; Rudolf Zetik; Yue Zhang; Yong Li; Jintao Wang. 2018. *"D3.1 SDN/NFV Environment in the Home Network Definition, Description and Preliminary Implementation of the Iorl Home Network"*.

Nawar Jawad; Mukhald Salih; Benjamin Meunier;John Cosmas; Charilaos Zarakovitis; Themistocles Anagnostopoulos; Anastasios Kourtis; Mathias Lacaud; Krzysztof Cabaj; Marcin Gregorczyk, Wojciech Mazurczyk; Piotr Nowakowski; Piotr Żórawski; Rudolf Zetik; Ali Eltohamy; Hequen Zhang; *"D3.2 Building Network Services - Intermediate"*.

Xun Zhang; Nawar Jawad; Mukhald Salih; Benjamin Meunier;John Cosmas; Charilaos Zarakovitis; Themistocles Anagnostopoulos; Anastasios Kourtis; Mathias Lacaud; Krzysztof Cabaj; Marcin Gregorczyk, Wojciech Mazurczyk; Piotr Nowakowski; Piotr Żórawski; Rudolf Zetik; Ali Eltohamy; Hequen Zhang; *"D3.3 Building Network Services - Final"*.

Adam Kapovits; John Cosmas; Ben Meunier; Kareem Ali; Hongying Meng; Wei Li; Yue Zhang; Li-Ke Huang; Xun Zhang,; Chuanxi Huang; Martin Ganley; Atanas Savov; James Gbadamosi; Rudolf Zetik,; Robert Müller; Tasos Kourtis; Charilaos Koumaras; Christos Sakkas; Michael-Alexandros Kourtis; Wojciech Mazurczyk; Krzysztof Cabaj; Sibel Malkoş; Memduh Emre Cakan; Alper Özel; Gökhan Özoğur; Siray Ocak; Daniel Negru; Mathias Lacaud; Marios Negru; Israel Koffman; Zion Haddad; Baruch Globen; Rafael Barkan; Eliron Yamina Salomon; Gil Sheffi; Yoav Avinoam; Jorge Garcia-Marquez; Huetzin Pérez Olivas; Yue Zhang; Jintao Wang. 2018. *"D5.1 Technical Specification of Radiolight Receiver"*.

John Cosmas, Ben Meunier, Nawar Jawad, Kareem Ali, Mathias Lacaud, Charilaos Zarakovitis, Sibel Malkos, Silvia de la Orden Rodriguez, Yoav Avinoam, and Moshe Ran. 2019. *"D6.1 Scenario Implementation"*.

John Cosmas; Nawar Jawad; Kareem Ali; Ben Meunier; Wei Li; Yue Zhang; Hequn Zhang; Xun Zhang; Robert Meuller; Charilaos Zarakovitis; Mathias Lacaud; Israel Koffman; Wojciech Mazurczyk; Krzysztof Cabaj; Piotr Nowakowski; Piotr Żórawski; Marcin Gregorczyk; Furkan Comert; Bastien Béchadergue; Jintao Wang; Ignacio Losas Davila. 2020. *"D6.2 Laboratory Testbeds"*

John Cosmas; Nawar Jawad; Muhlad Salih; Jian Song,; Jintao Wang; Wei Li; Hequn Zhang; Yue Zhang; Rudolf Zetik; Atanas Savov; James Gbadamosi; Mathias Lacaud. 2019. *"D7.7 Report on Standardisation and Regulation Activities – Year 2"*.

Appendix B

Résumé en Français

1 Introduction

La qualité d'expérience (*Quality of Experience* - QoE) des utilisateurs est devenu un facteur crucial pour évaluer le succès d'un système de diffusion de vidéos. Selon Cisco [Cisco, 2018], le trafic vidéo représente 75% des données circulant sur Internet en 2020, et cette quantité de données ne va faire qu'augmenter avec l'arrivée des formats en très hautes résolutions (4K, 8K, ...). Pour faire face à une telle croissance, l'amélioration des serveurs de contenu en cœur de réseau est est une opération nécessaire. Cependant, cette opération est souvent onéreuse. Ces coûts élevés, pourtant indispensables pour garantir une bonne QoE et une bonne stabilité, posent un problème majeur pour le design des futures solutions de diffusion de vidéos. Les *Content Delivery Network* (CDNs) - des serveurs puissants et coûteux placés dans les axes réseaux stratégiques pour être proche des utilisateurs - sont actuellement massivement utilisés pour permettre des diffusions à large échelle. La plupart du temps, des techniques d'adaptation de la qualité sur HTTP (*HTTP Adaptive Streaming* - HAS), tels que les standards DASH et HLS sont utilisés pour améliorer la QoE en ajustant la qualité vidéo aux débits observés en temps réel sur le réseau. Grâce à cela, le HAS essaie d'éviter les arrêts sur image involontaires dans les vidéos (appelé "freezes"), causés la plupart du temps par une faible bande passante disponible au niveau du client ou du serveur. Même si les CDNs peuvent servir un très un grand nombre de requêtes, ils restent néanmoins contraints par la taille de leurs infrastructures physiques. À la lumière des prévisions sur l'évolution du trafic vidéo, les dépenses d'exploitation et d'investissement des distributeurs de contenus pour

déployer des CDNs risquent de d'augmenter rapidement, ce qui pourrait rendre ces services coûteux pour les utilisateurs finaux en quête d'une très bonne qualité.

2 Travaux préliminaires: Multiple-Source Streaming

MS-Stream (Multiple-Source Streaming), développé en partenariat avec d'autres chercheurs, est un protocole de streaming vidéo adaptatif compatible avec DASH. Il permet d'utiliser plusieurs serveurs simultanément pour assurer une meilleure *qualité d'expérience* au public, à la fois en réduisant le nombre de coupures et en améliorant la qualité vidéo affichée (grâce à l'agrégation des bandes passantes).

Tout comme pour DASH, la vidéo est d'abord encodée en différentes qualités puis découpée en segments contenant plusieurs groupes d'images. Le contenu est ensuite copié sur plusieurs serveurs différents. Lorsque l'utilisateur souhaite regarder un segment d'un contenu vidéo, le lecteur vidéo fait des requêtes auprès des différents serveurs disponibles. Chaque serveur va alors proposer un sous-segment composé de groupes d'images en bonne qualité et d'autres en qualité basse, en fonction de la bande passante disponible et de sa capacité. De cette manière, le client peut rassembler les différents groupes d'images reçus pour reformer un segment en bonne qualité. Si certains groupes d'images en bonne qualité ne sont pas reçus, il est possible d'utiliser ceux de basse qualité fournis par les autres serveurs afin de compléter le segment et continuer la lecture sans interruption.

Dans l'exemple de la Figure B.1, l'utilisateur a une bande passante de 2 Mb/s avec un serveur, 1 Mb/s avec le deuxième, et 1 Mb/s avec le troisième, une qualité visuelle allant jusqu'à 4 Mb/s pourra être obtenue. Cette qualité est alors supérieure à la qualité que DASH aurait pu fournir (ici au maximum 2 Mb/s avec le premier serveur). Si maintenant le deuxième serveur devient surchargé ou indisponible, l'utilisateur pourra toujours obtenir 2+1 = 3 Mb/s depuis les deux autres serveurs, sans que cela n'interrompe la lecture.

Dans MS-Stream, le client utilise donc les groupes d'images redondants de basse qualité dans l'éventualité où ceux en bonne qualité ne sont pas reçus a temps. La surcharge subie par le réseau en bande passante dépend alors de la qualité des vidéos. En moyenne, nous observons moins de 10% d'augmentation de la bande passante utilisée lors de nos évaluations. De plus, la génération et agrégation des sous-segments a une empreinte minime puisqu'il suffit d'assembler des groupes d'images déjà encodés en différentes qualités par ailleurs.

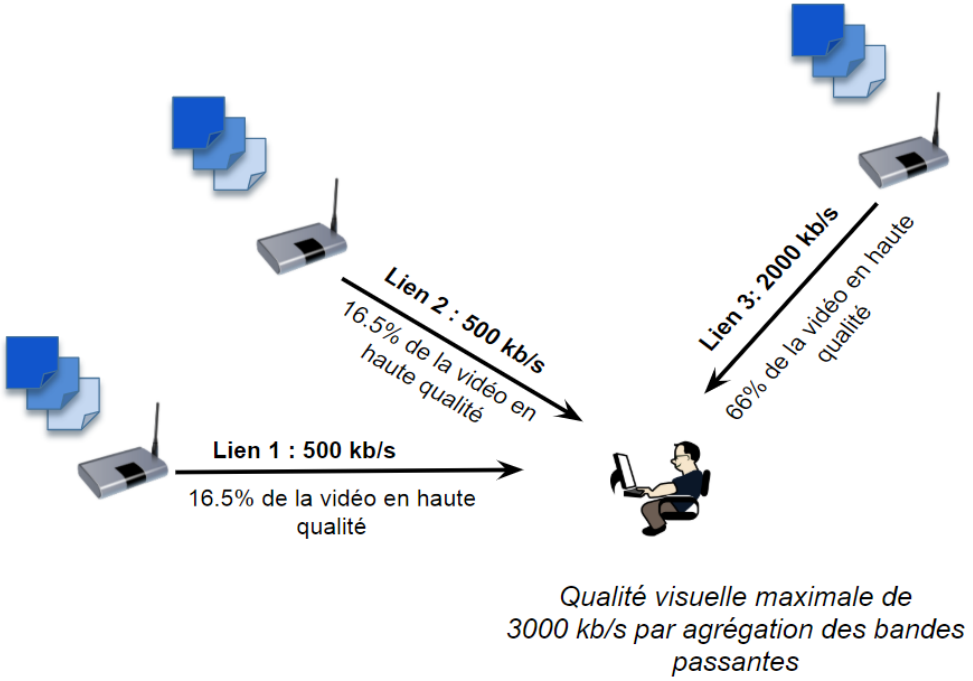

Figure B.1: Aggrégation de bande passante avec MS-Stream

3 PMS+ : un système de diffusion de vidéos pair-à-pair pragmatique pour résoudre les problèmes de passage à l'échelle et de qualité d'expérience dans les réseaux actuels

Dans le but de résoudre les problèmes de faible QoE des systèmes P2P, de manque de débit des serveurs en cœur de réseau et des importants coûts de mise à l'échelle, nous proposons d'exploiter les capacités d'équipements distribués au sein d'un système hybride P2P/multi-sources pragmatique. Notre système met à profit des algorithmes de sélection et de synchronisation multi-sources permettant de télécharger en simultané des segments vidéo depuis plusieurs sources et agréger ainsi les débits de plusieurs pairs, serveurs et équipements connectés pour obtenir une meilleure qualité à moindre coût.

3.1 Description du système

Une première version nommée PMS (P2P Multiple-source Streaming) a d'abord été développée. Comme présenté dans la Figure B.2, le système est composé de trois éléments : (1) des serveurs de contrôle composés d'un serveur de management qui surveille la santé du système à partir de métriques remontées pas les pairs et d'un tracker qui indique aux pairs une liste de K voisins avec qui communiquer; (2) des serveurs de toute taille, y compris des équipements distribués avec peu de ressources provisionnées avec les différentes qualités vidéo; (3) des overlays de niveau applicatif composés de N pairs consommant la même qualité et s'échangeant leurs données vidéo.

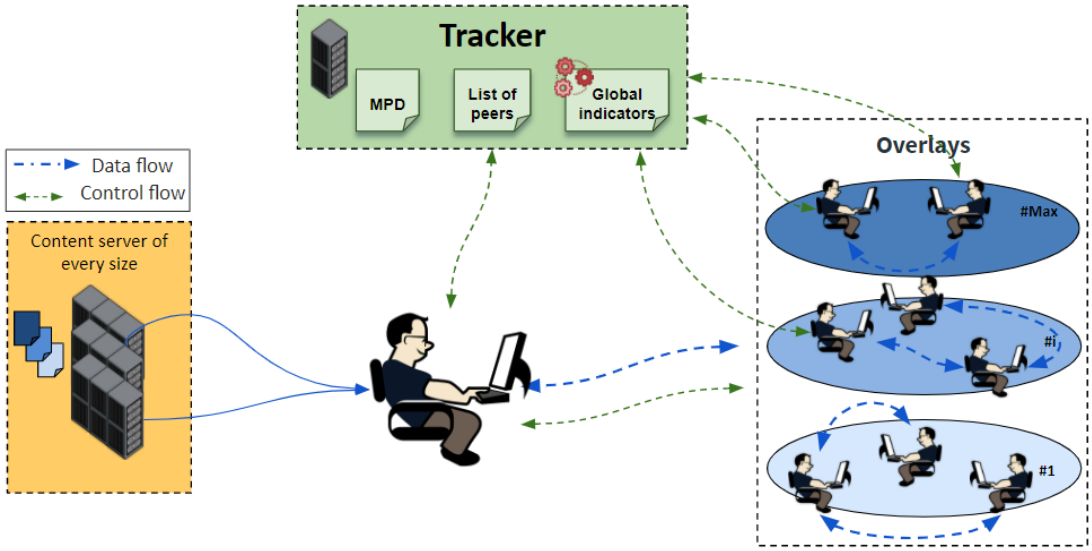

Figure B.2: Architecture de PMS, première version du système

En arrivant dans le système, les pairs se placent dans un overlay, correspondant à une qualité vidéo qu'ils sont capables d'obtenir. Ils téléchargent alors une partie des données vidéo depuis les serveurs et s'échangent le reste des données vidéo entre eux. À chaque instant, les pairs remontent des informations sur le bon fonctionnement des téléchargements et des échanges au serveur de management. Ce serveur leur retourne alors des informations sur la santé globale du système. En fonction de ces informations, les pairs peuvent alors changer d'overlay pour obtenir une meilleure qualité ou changer leur comportement afin d'aider les pairs en difficulté. Tout l'enjeu est d'arriver à mettre en place des algorithmes pragmatiques et multi-critères pour que tous les pairs du système puisse avoir la meilleure qualité possible.

Une seconde version nommée PMS+ (Figure B.3) a ensuite été proposée, afin d'améliorer PMS en cherchant à optimiser l'échange des données entre les pairs et accélérer les changements de qualité pour éviter à un pair de se retrouver bloqué dans une qualité trop haute ou trop basse par rapport à sa bande passante. Les pairs sont maintenant placés dans des petits groupes appartenant aux overlays. À l'intérieur d'un groupe, les pairs s'organisent entre eux et coopèrent pour s'échanger un maximum de données vidéo et ne télécharger que ce qui est absolument nécessaire depuis les serveurs. Le premier pair arrivé dans un groupe télécharge les segments vidéo depuis les serveurs tandis que les suivants essaient de récupérer les segments depuis leurs prédécesseurs. Grâce à un protocole d'échange de données plus efficace, mesurer la santé des overlays n'est plus autant nécessaire et les méchanismes d'adaptation de la qualité peuvent être améliorés pour diminuer le temps nécessaire à la stabilisation du système et améliorer la qualité d'expérience offerte par la solution.

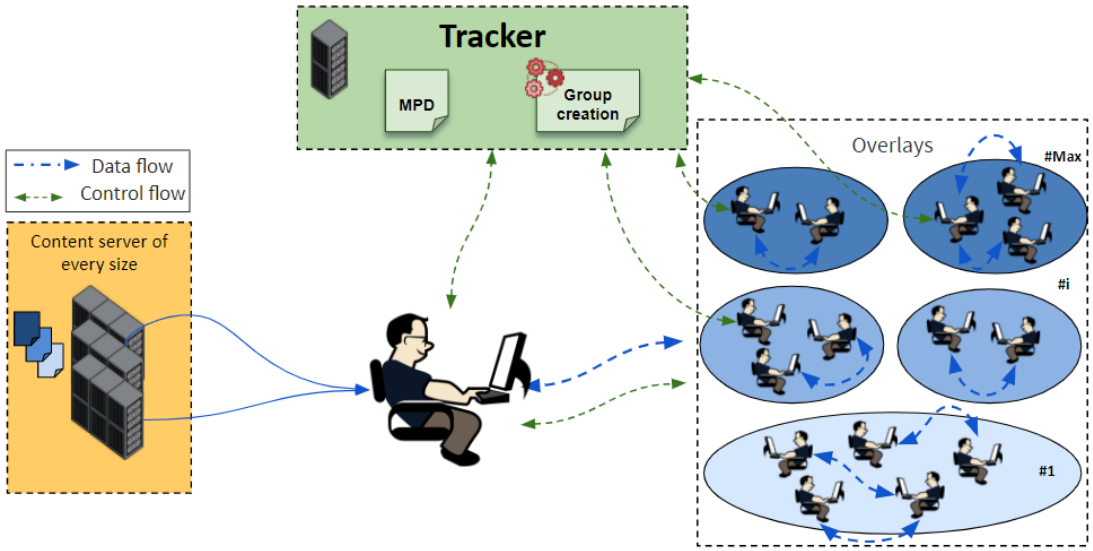

Figure B.3: Architecture de PMS+, seconde version du système

3.2 Evaluation

Notre système de diffusion de vidéos hybride P2P/multi-sources a été implémenté en se conformant au standard DASH. Le lecteur a été développé en se servant du code de base du lecteur DASH-IF (http://dashif.org). Nous avons mené une évaluation large échelle pendant plusieurs mois avec en moyenne plus de 10 000 utilisateurs par jour, équipés de différents appareils (téléphones, ordinateurs avec plusieurs systèmes d'exploitation).

Cette expérience a été réalisée en partenariat avec un acteur majeur du marché français de la diffusion de vidéos provenant de webcams touristiques

Nous avons étudié de multiples critères considérés comme essentiels pour la mesure de la qualité d'expérience, et comparé PMS+ et PMS avec un lecteur HAS standard et un lecteur P2P de l'état de l'art. Les résultats obtenus sont résumés dans les Figures B.4, B.5 et B.6.

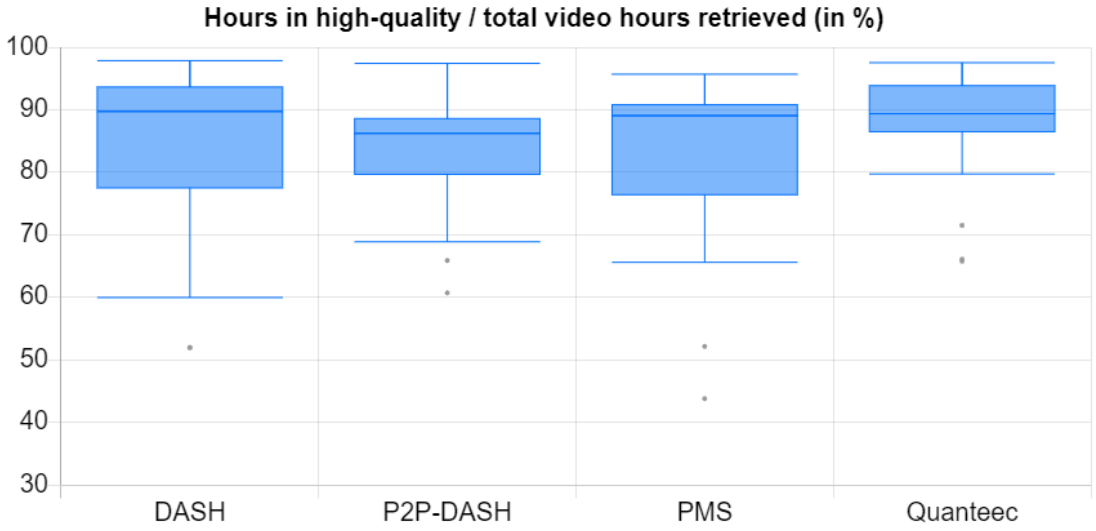

Figure B.4: Pourcentage de temps de vidéo reçu en haute qualité pour les 4 lecteurs. Un haut pourcentage signifie une meilleure qualité vidéo reçue.

Nous pouvons observer que PMS+ permet en moyenne d'obtenir une meilleure qualité vidéo tout en diminuant le nombre d'interruptions (ou freezes) en cours de lecture. En le comparant avec un système P2P de l'état de l'art, nous pouvons constater que notre solution permet d'économiser en moyenne une plus grande proportion des capacités du serveur. Jusqu'à 70% des données ont pu être délivrées en pair-à-pair lorsque l'audience était à son maximum. Cela permet par exemple de servir du contenu en meilleure qualité à un plus grand nombre d'utilisateurs.

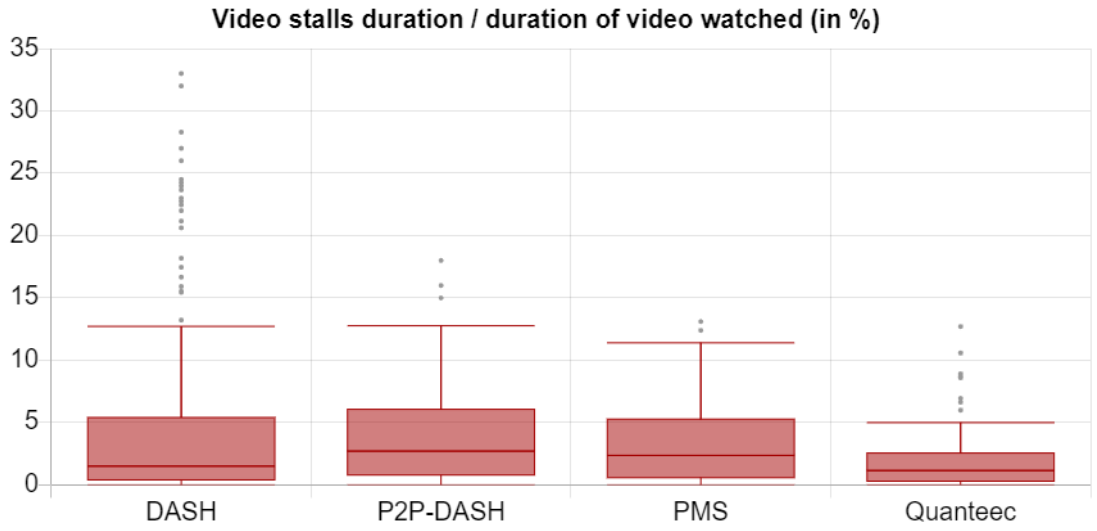

Figure B.5: Pourcentage de temps passé en freeze par rapport au temps de vidéo reçu. Un bas pourcentage signifie que peu d'utilisateurs ont subit des interruptions.

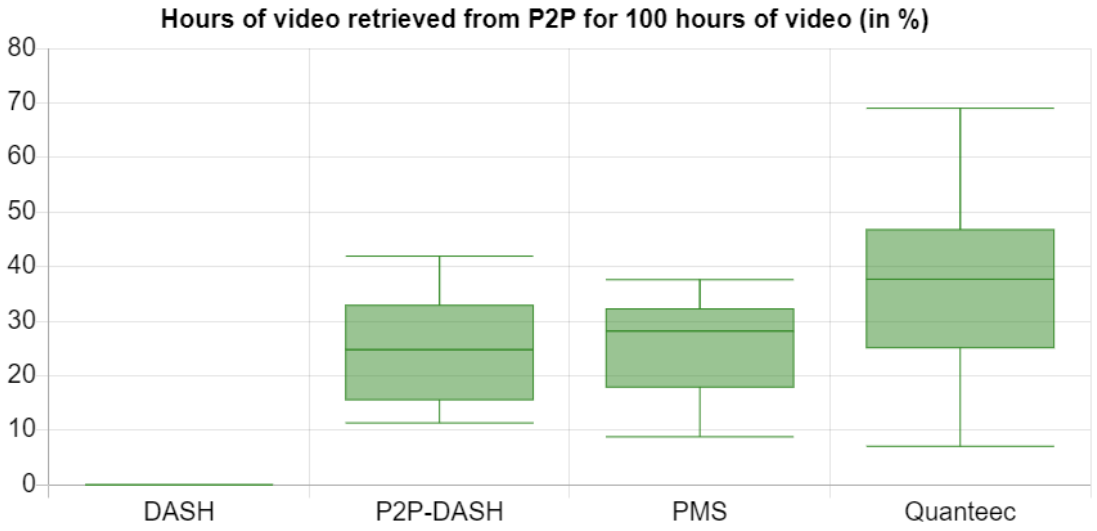

Figure B.6: Pourcentage de temps de vidéo reçu des pairs par rapport au temps reçu des serveurs. Un haut pourcentage signifie que les serveurs sont moins sollicités et que le système peut servir plus d'utilisateurs en haute qualité.

4 Multiple-Source Streaming over Remote Radio Light Head : un système de diffusion de vidéos pragmatique et efficace pour les réseaux 5G du futur

Les réseaux sans-fils d'intérieur d'aujourd'hui s'appuient essentiellement sur la technologie Wi-Fi et souffrent souvent des congestions et des interférences, étant donné que les matériaux des bâtiments modernes peuvent restreindre la propagation des ondes dans les fréquences radio et que de nombreux équipements utilisateurs entrent en compétition pour la bande-passante disponible. Le projet européen 5G *Internet of Radio Light* (IoRL) vise à résoudre ces problèmes en proposant de nouveaux réseaux sans-fils d'intérieur utilisant la lumière visible et les ondes millimétriques (VLC/mmWave) en plus des ondes Wi-Fi déjà existantes.

Un des objectifs de ce projet est de s'interconnecter avec les réseaux 4G et 5G extérieurs et donc de satisfaire les contraintes de la 5G, à savoir offrir des bandes passantes élevées et une latence réseau très faible à tous les utilisateurs. Plusieurs cas d'étude d'intérieur pouvant bénéficier de cette technologie ont été identifié dans les musées [Cosmas et al., 2018a], les habitations [Cosmas et al., 2018c], les supermarchés [Cosmas et al., 2017] et les stations de métros/trains [Cosmas et al., 2018b]. Dans tous ces cas d'études, le thème de la diffusion de vidéo en haute qualité revient toujours, que ce soit pour délivrer du contenu en très haute qualité, envoyer de la publicité ou des documentaires, faire des visites virtuelles à 360 degrés ou maintenir une session en direct entre un réparateur et un expert à distance pour effectuer une opération de maintenance délicate.

De plus en plus d'utilisateurs consomment des contenus en Ultra HD (UHD) ou encore des contenus immersifs aussi bien depuis des terminaux mobiles que des ordinateurs ou des télévisions compatibles. Le projet souhaite pouvoir répondre aux mêmes objectifs en terme de vidéo que les autres projets 5G : être capable de diffuser des vidéos à la demande et des sessions en direct en très haute qualité et de façon fiable aux utilisateurs. Au regard de la pluralité et de l'hétérogénéité de l'architecture réseau d'IoRL, la fiabilité peut être difficile à garantir en utilisant les lecteurs vidéo de l'état de l'art car la lecture peut être coupée temporairement en cas de perte de signal via la lumière visible, par exemple si une lampe est occultée.

En tirant un avantage de la pluralité des chemins réseaux disponible dans l'architecture d'IoRL, nous proposons Multiple-Source Streaming over Remote Radio Light Head (MSS/RRLH), un système de diffusion multi-chemin améliorant à la fois la fiabilité

et la qualité d'expérience des sessions vidéo. MSS/RRLH est une évolution des solutions *Over-the-Top* (i.e. au niveau applicatif) telles que MPEG-DASH [Sodagar, 2011] et HLS [HLS, 2017] et est dérivé de précédents travaux de recherche sur MS-Stream. Le système utilise simultanément plusieurs chemins pour télécharger les segments vidéo. La solution proposée est composée d'un serveur spécifique déployé dans la *Home IP Gateway* (HIPG) de IoRL et d'un lecteur de vidéo utilisé par les terminaux utilisateurs.

4.1 Description du système

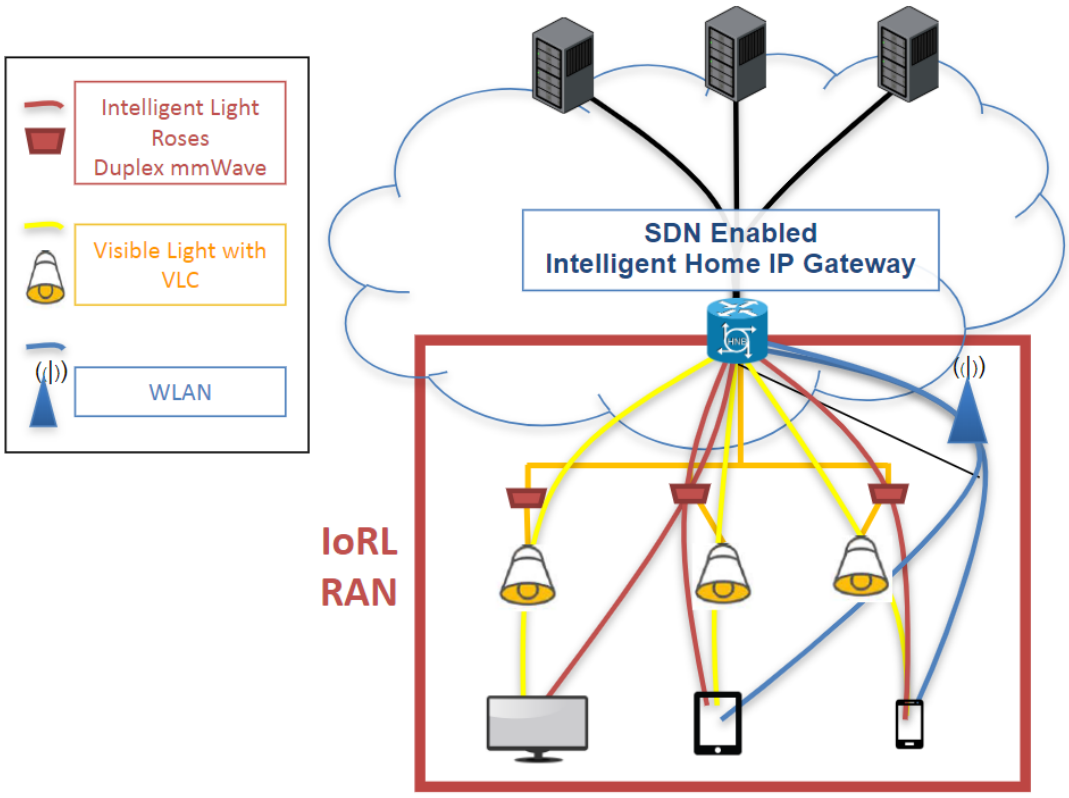

Figure B.7: Architecture réseau du projet IoRL

L'architecture réseau d'IoRL est présentée dans la Figure B.7. L'accès au réseau sans fil peut se faire en utilisant trois types d'émetteurs : les modulateurs de lumière visible (VLC - *Visible Light Communication*), les antennes d'ondes millimétriques (mmWave) et les routeurs Wi-Fi. Un terminal utilisateur peut se connecter à un ou plusieurs des ces émetteurs, à condition d'être à portée. Tous ces accès sont connectés à une *Passerelle réseau Interne sur IP* (HIPG - *Home IP Gateway*) via un réseau Ethernet en anneau.

Cette HIPG est un serveur qui sert à la fois à faire tourner des fonctions réseaux telles que du routage et de l'encodage des paquets vidéo et à la fois à fournir un accès vers l'Internet comme n'importe quelle passerelle. Le déploiement de ces fonctionnalités est facilité grâce à la technologie SDN/NFV embarquée dans le serveur, permettant de développer et de modifier simplement des fonctions réseaux.

Du côté du terminal utilisateur, MSS/RRLH utilise un lecteur vidéo bien particulier. Ce lecteur embarque des algorithmes permettant de télécharger des segments simultanément depuis plusieurs chemins. L'idée principale est d'encoder une vidéo en deux qualités, une qualité haute définition et une qualité basse définition, dans la HIPG. À partir de là, le lecteur va essayer de télécharger des segments en haute qualité depuis le réseaux VLC/mmWave proposant des débits très élevés et des segments en basse qualité depuis le réseau Wi-Fi.

Télécharger des segments en plusieurs qualités mais n'afficher que ceux en haute qualité peut être problématique. La solution décrite jusqu'à présent récupère plus de données qu'elles n'en utilise et peut potentiellement surcharger le réseau. Pour éviter cela, le lecteur vidéo MSS/RRLH contient des mécanismes pour diminuer cette surcharge en évitant de télécharger des images en basse qualité lorsque ce n'est pas utile. Pour une vidéo à la demande, les prochains segments sont téléchargés à l'avance et enregistré dans une mémoire tampon. On regardant l'évolution de la taille de cette mémoire tampon, il est possible de savoir si la lecture va être interrompue prochainement ou si il reste assez de temps pour attendre que le réseau revienne. Dans le cas d'une diffusion en direct, le délai prend une toute autre importance et ne permet pas de jouer avec cette mémoire tampon. En effet, pour maintenir un délai au plus bas (i.e. que le temps entre le moment où une image est capturée par une caméra et le moment où cette même image est affichée sur le terminal de l'utilisateur), il n'est pas possible de travailler avec les prochains segments car ils n'existent pas encore. À la place, le lecteur essaie de faire des prédictions sur l'évolution du réseau pour savoir s'il est utile de récupérer les segments en basse qualité pour assurer la fiabilité.

4.2 Evaluation

Pour l'évaluation, le système MSS/RRLH a été déployé en laboratoire. Le lecteur à été connecté au serveur de diffusion via deux chemin réseaux virtuels. Les caractéristiques de ces chemins réseaux virtuels sont contrôlées et peuvent varier au cours du temps afin de simuler le comportement des réseaux Wi-Fi et VLC/mmWave en suivant des

traces fournies par les partenaires du projet européen et des scénarios bien spécifiques. Le lecteur est évalué pour des vidéos à la demande et des vidéos en direct récupérées depuis une caméra 4K. Il est comparé à un lecteur de l'état de l'art capable de d'observer la bande-passant des réseaux auquel il n'est pas connecté à l'aide d'un ping périodique. Il est également comparé à un lecteur au comportement glouton cherchant à utiliser en permanence les deux réseaux disponibles sans se soucier de risquer une surcharge du réseau à cause de son égoïsme. Des métriques de qualité d'expérience ont été récoltées sur ces players au cours de deux scénarios répétées pendant plusieurs heures. Le premier scénario (Figure B.8) étudié est celui d'un utilisateur passant lentement sous les lampes. Le second scénario (Figure B.8) étudie le cas où la lumière est soudainement occultée par un obstacle.

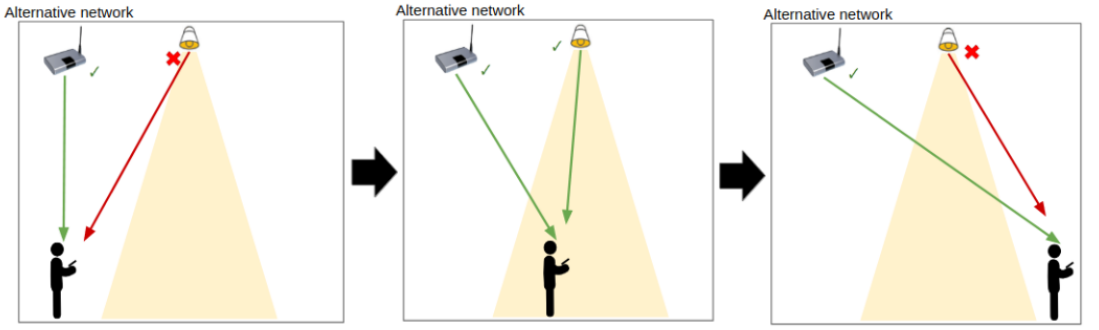

Figure B.8: Scenario 1: un utilisateur se déplace lentement sous les lampes.

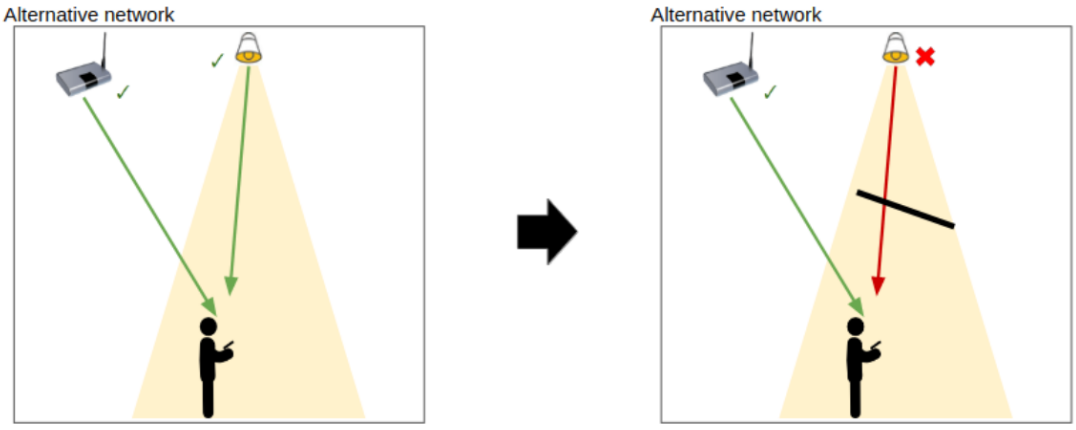

Figure B.9: Scenario 2: la lumière est soudainement occultée par un obstacle

Les résultats sont résumés dans le Tableau B.1. Cette évaluation montrer que la solution MSS/RRLH permet d'éviter les interruptions et de réduire la latence lors des diffusions

en direct par rapport à l'état de l'art pour atteindre 3 secondes de latence stable (i.e la latence qui ne cause pas elle même d'interruptions dans la vidéo). De plus, les mécanismes de contrôles des données inutiles est efficace puisque seulement 2% de ces données sont téléchargées avec MSS/RRLH contre 9% pour un lecteur multi-source égoïste et glouton. Cela permet d'éviter les surcharges dans le réseau et de servir un plus grand nombre d'utilisateurs.

Table B.1: Résumé des résultats obtenus

Lecteur de vidéos	***Latence moyenne observée par rapport à la caméra (en secondes)***	***Durée moyenne des interruptions dans la vidéo (en secondes)***	***Ecart observé en moyenne de la qualité (en Mbit/s, MSS/RRLH étant la référence)***	***Données supplémentaires inutilisées (en % par rapport aux données utiles)***
Etat de l'art amélioré	6	3	- 0.1	0.3
MSS/RRLH glouton	3	<1	+ 0.02	9.1
MSS/RRLH	3	<1	0	2

Bibliography

[3GP-DASH, 2012] 3GP-DASH (2012). Transparent end-to-end packet-switched streaming service (pss); progressive download and dynamic adaptive streaming over http (3gp-dash).

[3GPP, 2020] 3GPP (2020). https://www.3gpp.org/.

[5G-PPP, 2020] 5G-PPP (2020). https://5g-ppp.eu/.

[Abboud et al., 2011] Abboud, O., Zinner, T., Pussep, K., Al-Sabea, S., and Steinmetz, R. (2011). On the impact of quality adaptation in SVC-based P2P video-on-demand systems. In *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11*, page 223, San Jose, CA, USA. ACM Press.

[Abdelwahab et al., 2016] Abdelwahab, S., Hamdaoui, B., Guizani, M., and Znati, T. (2016). Network function virtualization in 5g. *IEEE Communications Magazine*, 54(4):84–91.

[Abdelzaher et al., 2008] Abdelzaher, T., Diao, Y., Hellerstein, J. L., Lu, C., and Zhu, X. (2008). Introduction to Control Theory And Its Application to Computing Systems. In Liu, Z. and Xia, C. H., editors, *Performance Modeling and Engineering*, pages 185–215. Springer US, Boston, MA.

[Adhikari et al., 2012a] Adhikari, V. K., Guo, Y., Hao, F., Hilt, V., and Zhang, Z.-L. (2012a). A tale of three CDNs: An active measurement study of Hulu and its CDNs. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 7–12.

[Adhikari et al., 2012b] Adhikari, V. K., Guo, Y., Hao, F., Varvello, M., Hilt, V., Steiner, M., and Zhang, Z.-L. (2012b). Unreeling netflix: Understanding and improving multi-CDN movie delivery. In *2012 Proceedings IEEE INFOCOM*, pages 1620–1628. IEEE.

[Adhikari et al., 2012c] Adhikari, V. K., Guo, Y., Hao, F., Varvello, M., Hilt, V., Steiner, M., and Zhang, Z.-L. (2012c). Unreeling netflix: Understanding and improving multi-CDN movie delivery. In *2012 Proceedings IEEE INFOCOM*, pages 1620–1628. ISSN: 0743-166X.

[Adhikari et al., 2012d] Adhikari, V. K., Jain, S., Chen, Y., and Zhang, Z.-L. (2012d). Vivisecting youtube: An active measurement study. In *2012 Proceedings IEEE INFOCOM*, pages 2521–2525. IEEE.

[Adhikari et al., 2012e] Adhikari, V. K., Jain, S., Chen, Y., and Zhang, Z.-L. (2012e). Vivisecting YouTube: An active measurement study. In *2012 Proceedings IEEE INFOCOM*, pages 2521–2525. ISSN: 0743-166X.

[Adhikari et al., 2010] Adhikari, V. K., Jain, S., and Zhang, Z.-L. (2010). YouTube traffic dynamics and its interplay with a tier-1 ISP: an ISP perspective. In *Proceedings of the 10th annual conference on Internet measurement - IMC '10*, page 431, Melbourne, Australia. ACM Press.

[Adobe HDS, 2010] Adobe HDS (2010). Adobe http dynamic streaming (hds).

[Akamai, 2020] Akamai (2020). Netsession. https://www.akamai.com/fr/fr/products/media-delivery/netsession-interface-overview.jsp.

[Akhshabi et al., 2012a] Akhshabi, S., Anantakrishnan, L., Begen, A. C., and Dovrolis, C. (2012a). What happens when HTTP adaptive streaming players compete for bandwidth? In *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video - NOSSDAV '12*, page 9, Toronto, Ontario, Canada. ACM Press.

[Akhshabi et al., 2011] Akhshabi, S., Begen, A. C., and Dovrolis, C. (2011). An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP. In *Proceedings of the second annual ACM conference on Multimedia systems*, MMSys '11, pages 157–168, San Jose, CA, USA. Association for Computing Machinery.

[Akhshabi et al., 2012b] Akhshabi, S., Narayanaswamy, S., Begen, A. C., and Dovrolis, C. (2012b). An experimental evaluation of rate-adaptive video players over HTTP. *Signal Processing: Image Communication*, 27(4):271–287.

[Anjum et al., 2017] Anjum, N., Karamshuk, D., Shikh-Bahaei, M., and Sastry, N. (2017). Survey on peer-assisted content delivery networks. *Computer Networks*, 116:79–95.

[Applegate et al., 2010] Applegate, D., Archer, A., Gopalakrishnan, V., Lee, S., and Ramakrishnan, K. K. (2010). Optimal content placement for a large-scale VoD system. In *Proceedings of the 6th International COnference on - Co-NEXT '10*, page 1, Philadelphia, Pennsylvania. ACM Press.

[Baccelli et al., 2013] Baccelli, F., Mathieu, F., Norros, I., and Varloot, R. (2013). Can P2P networks be super-scalable? In *2013 Proceedings IEEE INFOCOM*, pages 1753–1761. ISSN: 0743-166X.

[Balachandran et al., 2013] Balachandran, A., Sekar, V., Akella, A., and Seshan, S. (2013). Analyzing the potential benefits of CDN augmentation strategies for internet video workloads. In *Proceedings of the 2013 conference on Internet measurement conference - IMC '13*, pages 43–56, Barcelona, Spain. ACM Press.

[Basso et al., 2014] Basso, S., Servetti, A., Masala, E., and De Martin, J. C. (2014). Measuring DASH streaming performance from the end users perspective using neubot. In *Proceedings of the 5th ACM Multimedia Systems Conference on - MMSys '14*, pages 1–6, Singapore, Singapore. ACM Press.

[Bentaleb et al., 2019a] Bentaleb, A., Taani, B., Begen, A. C., Timmerer, C., and Zimmermann, R. (2019a). A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP. *IEEE Communications Surveys Tutorials*, 21(1):562–585. Conference Name: IEEE Communications Surveys Tutorials.

[Bentaleb et al., 2019b] Bentaleb, A., Timmerer, C., Begen, A. C., and Zimmermann, R. (2019b). Bandwidth prediction in low-latency chunked streaming. In *Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '19, pages 7–13, Amherst, Massachusetts. Association for Computing Machinery.

[Bentaleb et al., 2020a] Bentaleb, A., Timmerer, C., Begen, A. C., and Zimmermann, R. (2020a). Performance Analysis of ACTE: a Bandwidth Prediction Method for Low-Latency Chunked Streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 0(ja).

[Bentaleb et al., 2020b] Bentaleb, A., Yadav, P. K., Ooi, W. T., and Zimmermann, R. (2020b). DQ-DASH: A Queuing Theory Approach to Distributed Adaptive Video Streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(1):1–24.

[Bobarshad et al., 2012] Bobarshad, H., van der Schaar, M., Aghvami, A. H., Dilmaghani, R. S., and Shikh-Bahaei, M. R. (2012). Analytical Modeling for Delay-Sensitive Video Over WLAN. *IEEE Transactions on Multimedia*, 14(2):401–414.

[Bobarshad et al., 2010] Bobarshad, H., van der Schaar, M., and Shikh-Bahaei, M. R. (2010). A Low-Complexity Analytical Modeling for Cross-Layer Adaptive Error Protection in Video Over WLAN. *IEEE Transactions on Multimedia*, 12(5):427–438.

[Boros et al., 2019] Boros, T., Zuraniewski, P., Hindriks, R., Adrichem, N. v., Thomas, E., and D'Acunto, L. (2019). Enabling Superior and Controllable Video Streaming QoE with 5G Network Orchestration. In *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 124–129. ISSN: 2472-8144.

[Botta et al., 2018] Botta, A., Avallone, A., Garofalo, M., and Ventre, G. (2018). A user-oriented performance comparison of video hosting services. *Computer Communications*, 116:118–131.

[Böttger et al., 2018] Böttger, T., Cuadrado, F., Tyson, G., Castro, I., and Uhlig, S. (2018). 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.

[Bouras et al., 2017] Bouras, C., Kollia, A., and Papazois, A. (2017). Sdn nfv in 5g: Advancements and challenges. In *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pages 107–111.

[Bouten et al., 2012] Bouten, N., Famaey, J., Latré, S., Huysegems, R., De Vleeschauwer, B., Van Leekwijck, W., and De Turck, F. (2012). QoE optimization through in-network quality adaptation for HTTP adaptive streaming. In *Proceedings of the 8th International Conference on Network and Service Management*, CNSM '12, pages 336–342, Las Vegas, Nevada. International Federation for Information Processing.

[Bouten et al., 2014] Bouten, N., Latre, S., Famaey, J., Van Leekwijck, W., and De Turck, F. (2014). In-Network Quality Optimization for Adaptive Video Streaming Services. *IEEE Transactions on Multimedia*, 16(8):2281–2293.

[Bruneau-Queyreix, 2017] Bruneau-Queyreix, J. (2017). *Multi-Criteria Optimization of Content Delivery within the Future Media Internet*. PhD thesis, University of Bordeaux.

[Bruneau-Queyreix et al., 2016] Bruneau-Queyreix, J., Lacaud, M., Anapliotis, P., and Négru, D. (2016). On providing multiple-server support to dynamic adaptive streaming applications for enhanced QoE. In *2016 International Conference on Telecommunications and Multimedia (TEMU)*, pages 1–6.

[Bruneau-Queyreix et al., 2017a] Bruneau-Queyreix, J., Lacaud, M., Negru, D., Batalla, J. M., and Borcoci, E. (2017a). MS-Stream: A multiple-source adaptive streaming solution enhancing consumer's perceived quality. In *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 427–434. ISSN: 2331-9860.

[Bruneau-Queyreix et al., 2017b] Bruneau-Queyreix, J., Lacaud, M., Negru, D., Batalla, J. M., and Borcoci, E. (2017b). QOE enhancement through cost-effective adaptation decision process for multiple-server streaming over HTTP. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. ISSN: 1945-788X.

[Bruneau-Queyreix et al., 2018] Bruneau-Queyreix, J., Lacaud, M., Négru, D., Batalla, J. M., and Borcoci, E. (2018). Adding a New Dimension to HTTP Adaptive Streaming Through Multiple-Source Capabilities. *IEEE MultiMedia*, 25(3):65–78. Conference Name: IEEE MultiMedia.

[Cable Television Laboratories, 2016] Cable Television Laboratories (2016). Ip multicast adaptive bit rate. https://community.cablelabs.com/wiki/plugins/servlet/cablelabs/alfresco/download?id=3edb1609-17ff-4844-87ed-124314a73e7c.

[Capovilla et al., 2010] Capovilla, N., Eberhard, M., Mignanti, S., Petrocco, R., and Vehkaperä, J. (2010). An architecture for distributing scalable content over peer-to-peer networks. In *2010 Second International Conferences on Advances in Multimedia*, pages 1–6.

[Chellouche et al., 2012] Chellouche, S. A., Négru, D., Chen, Y., and Sidibe, M. (2012). Home-Box-assisted content delivery network for Internet Video-on-Demand services. In *2012 IEEE Symposium on Computers and Communications (ISCC)*, pages 000544–000550. ISSN: 1530-1346.

[Chen et al., 2015] Chen, L., Zhou, Y., Jing, M., and Ma, R. T. B. (2015). Thunder crystal: a novel crowdsourcing-based content distribution platform. In *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video - NOSSDAV '15*, pages 43–48, Portland, Oregon. ACM Press.

[Chun et al., 2012] Chun, S., Woo, Y., Shen, Y., and Park, J. (2012). P2p-based group service management for live video streaming communication. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, volume 02, pages 812–816.

[Cisco, 2018] Cisco (2018). Cisco annual internet report (2018–2023) white paper.

[CMAF, 2018] CMAF (2018). Iso/iec 23000-19:2018(en). information technology — multimedia application format (mpeg-a) — part 19: Common media application format (cmaf) for segmented media.

[Cofano et al., 2014] Cofano, G., De Cicco, L., and Mascolo, S. (2014). A Control Architecture for Massive Adaptive Video Streaming Delivery. In *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming - VideoNext '14*, pages 7–12, Sydney, Australia. ACM Press.

[Cosmas et al., 2018a] Cosmas, J., Meunier, B., Ali, K., Jawad, N., Meng, H.-Y., Goutagneux, F., Legale, E., Satta, M., Jay, P., Zhang, X., Huang, C., Garcia, J., Negru, M., Zhang, Y., Kourtis, T., Koumaras, C., Sakkas, C., Huang, L.-K., Zetik, R., Cabaj, K., Mazurczyk, W., Cakan, M. E., and Kapovits, A. (2018a). 5G Internet of radio light services for Musée de la Carte à Jouer. In *2018 Global LIFI Congress (GLC)*, pages 1–6.

[Cosmas et al., 2018b] Cosmas, J., Meunier, B., Ali, K., Jawad, N., Salih, M., Meng, H.-Y., Royo, J., Fernandez, P., Hadad, Z., Gokmen, H., Cakan, M. E., Kourtis, M.-A., Koumaras, H., Sakkas, C., Salomon, E., Avinoam, Y., Negru, D., Lacaud, M., Zhang, Y., Huang, L.-K., Zetik, R., Cabaj, K., Mazurczyk, W., Zhang, X., and Kapovits, A. (2018b). A Scaleable and License Free 5G Internet of Radio Light Architecture for Services in Train Stations. In *European Wireless 2018; 24th European Wireless Conference*, pages 1–6.

[Cosmas et al., 2017] Cosmas, J., Meunier, B., Ali, K., Jawad, N., Salih, M., Meng, H.-Y., Song, J., Wang, J., Tong, M., Cao, X., Li, X., Zhang, X., Huang, C., Zhang, Y., Ran, M., Ran, E., Salomon, E., Avinoam, Y., Hadad, Z., Globen, B., Negru, D., Lacaud, M., Kourtis, M.-A., Koumaras, H., Sakkas, C., Huang, L.-K., Zetik, R., Cabaj, K., Mazurczyk, W., and Kapovits, A. (2017). 5G Internet of radio light services for supermarkets. In *2017 14th China International Forum on Solid State Lighting: International Forum on Wide Bandgap Semiconductors China (SSLChina: IFWS)*, pages 69–73.

[Cosmas et al., 2018c] Cosmas, J., Meunier, B., Ali, K., Jawad, N., Salih, M., Zhang, Y., Hadad, Z., Globen, B., Gokmen, H., Malkos, S., Cakan, M., Koumaras, H., Kourtis, A., Sakkas, C., Negru, D., Lacaud, M., Ran, M., Ran, E., Garcia, J., Li, W., Huang, L.-K., Zetik, R., Cabaj, K., Mazurczyk, W., Zhang, X., and Kapovits, A. (2018c). A 5G Radio-Light SDN Architecture for Wireless and Mobile Network Access in Buildings. In *2018 IEEE 5G World Forum (5GWF)*, pages 135–140.

[Crowcroft et al., 2005] Crowcroft, J., Pias, M., Sharma, R., and Lim, S. (2005). A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, 7(2):72–93.

[Da Silva et al., 2019a] Da Silva, S., Ben Mokhtar, S., Contiu, S., Négru, D., Réveillère, L., and Rivière, E. (2019a). Privatube: Privacy-preserving edge-assisted video streaming. In *Proceedings of the 20th International Middleware Conference*, pages 189–201.

[Da Silva et al., 2018a] Da Silva, S., Bruneau-Queyreix, J., Lacaud, M., Négru, D., and Réveillère, L. (2018a). MUSLIN: Achieving high, fairly shared QoE through multi-source live streaming. In *Proceedings of the 23rd Packet Video Workshop*, pages 54–59.

[Da Silva et al., 2018b] Da Silva, S., Bruneau-Queyreix, J., Lacaud, M., Négru, D., and Réveillère, L. (2018b). MUSLIN demo: high QoE fair multi-source live streaming. In *Proceedings of the 9th ACM Multimedia Systems Conference*, pages 529–532.

[Da Silva et al., 2019b] Da Silva, S., Bruneau-Queyreix, J., Lacaud, M., Negru, D., and Réveillère, L. (2019b). MUSLIN: A QoE-aware CDN resources provisioning and advertising system for cost-efficient multisource live streaming. *International Journal of Network Management*, page e2081.

[Dailymotion, 2020] Dailymotion (2020). The home for videos that matter. `https://www.dailymotion.com`.

[DASH-IF, 2018] DASH-IF (2018). Dash-if position paper: Server and network assisted DASH (sand). `https://dashif.org/docs/SAND-Whitepaper-Dec13-final.pdf`.

[DASH-IF - DASH Industry Forum, 2020] DASH-IF - DASH Industry Forum (2020). https://dashif.org.

[DASH-IF Cases and Vectors, 2014] DASH-IF Cases and Vectors (2014). “http adaptive streaming in practice. netflix tech.

[DASH-IF Cases and Vectors, 2020] DASH-IF Cases and Vectors (2020). Dash-industry forum, 2014. guidelines for implementation: Dash avc/264 test cases and vectors. http://dashif.org/wp-content/uploads/2015/04/dash-avc-264-test-vectors-v09-communityreview.pdf.

[DASH-IF Guidelines, 2020] DASH-IF Guidelines (2020). Guidelines - completed dash-if interoperability documents. http://dashif.org/guidelines/.

[DASH-IF Player, 2020] DASH-IF Player (2020). Industry-forum/dash.js adaptive video player. https://github.com/dash-industry-forum/dash.js.

[DASH-IF Software, 2020] DASH-IF Software (2020). Dash industry forum - software . http://dashif.org/software/.

[DASH-IF Tests, 2020] DASH-IF Tests (2020). Test assets database. http://testassets.dashif.org/.

[De Cicco and Mascolo, 2010] De Cicco, L. and Mascolo, S. (2010). An Experimental Investigation of the Akamai Adaptive Video Streaming. In Leitner, G., Hitz, M., and Holzinger, A., editors, *HCI in Work and Learning, Life and Leisure*, volume 6389, pages 447–464. Springer Berlin Heidelberg, Berlin, Heidelberg.

[De Cicco et al., 2011] De Cicco, L., Mascolo, S., and Palmisano, V. (2011). Feedback control for adaptive live video streaming. In *Proceedings of the second annual ACM conference on Multimedia systems*, MMSys '11, pages 145–156, San Jose, CA, USA. Association for Computing Machinery.

[De Vriendt et al., 2014] De Vriendt, J., De Vleeschauwer, D., and Robinson, D. C. (2014). QoE model for video delivered over an LTE network using HTTP adaptive streaming. *Bell Labs Technical Journal*, 18(4):45–62.

[Deering and Cheriton, 1990] Deering, S. E. and Cheriton, D. R. (1990). Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computer Systems (TOCS)*, 8(2):85–110.

[Deng and Xu, 2013] Deng, H. and Xu, J. (2013). CorePeer: A P2P Mechanism for Hybrid CDN-P2P Architecture. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Gao, Y., Shim, K., Ding, Z., Jin, P., Ren, Z., Xiao, Y., Liu, A., and Qiao, S., editors, *Web-Age*

Information Management, volume 7901, pages 278–286. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Deng et al., 2017] Deng, J., Tyson, G., Cuadrado, F., and Uhlig, S. (2017). Internet scale user-generated live video streaming: The twitch case. In *International Conference on Passive and Active Network Measurement*, pages 60–71. Springer.

[Diallo et al., 2013] Diallo, M. T., Moustafa, H., Afifi, H., and Marechal, N. (2013). Adaptation of audiovisual contents and their delivery means. *Communications of the ACM*, 56(11):86–93.

[Diffserv, 2000] Diffserv (2000). Rfc 2998, a framework for integrated services operation over diffserv networks.

[Diot et al., 2000] Diot, C., Levine, B., Lyles, B., Kassem, H., and Balensiefen, D. (2000). Deployment issues for the IP multicast service and architecture. *IEEE Network*, 14(1):78–88.

[Dubroy and Balakrishnan, 2010] Dubroy, P. and Balakrishnan, R. (2010). A study of tabbed browsing among mozilla firefox users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 673–682, New York, NY, USA. Association for Computing Machinery.

[DVB-DASH, 2020] DVB-DASH (2020). Dvb mpeg-dash profile for transport of iso bmff based dvb services over ip based networks.

[Eberhard et al., 2010] Eberhard, M., Szkaliczki, T., Hellwagner, H., Szobonya, L., and Timmerer, C. (2010). Knapsack problem-based piece-picking algorithms for layered content in peer-to-peer networks. In *Proceedings of the 2010 ACM workshop on Advanced video streaming techniques for peer-to-peer networks and social networking - AVSTP2P '10*, page 71, Firenze, Italy. ACM Press.

[Essaili et al., 2013] Essaili, A. E., Schroeder, D., Staehle, D., Shehada, M., Kellerer, W., and Steinbach, E. (2013). Quality-of-experience driven adaptive HTTP media delivery. In *2013 IEEE International Conference on Communications (ICC)*, pages 2480–2485. ISSN: 1938-1883.

[Famaey et al., 2013] Famaey, J., Latré, S., Bouten, N., Van de Meerssche, W., De Vleeschauwer, B., Van Leekwijck, W., and De Turck, F. (2013). On the merits of svc-based http adaptive streaming. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 419–426.

[Frank et al., 2013] Frank, B., Poese, I., Lin, Y., Smaragdakis, G., Feldmann, A., Maggs, B., Rake, J., Uhlig, S., and Weber, R. (2013). Pushing CDN-ISP collaboration to the limit. *ACM SIGCOMM Computer Communication Review*, 43(3):34–44.

[Gao et al., 2015] Gao, G., Wen, Y., Zhang, W., and Hu, H. (2015). Cost-efficient and QoS-aware content management in media cloud: Implementation and evaluation. In *2015 IEEE International Conference on Communications (ICC)*, pages 6880–6886, London. IEEE.

[Garcia et al., 2014] Garcia, M.-N., De Simone, F., Tavakoli, S., Staelens, N., Egger, S., Brunnstrom, K., and Raake, A. (2014). Quality of experience and HTTP adaptive streaming: A review of subjective studies. In *2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 141–146, Singapore, Singapore. IEEE.

[Ge et al., 2016] Ge, C., Wang, N., Skillman, S., Foster, G., and Cao, Y. (2016). QoE-Driven DASH Video Caching and Adaptation at 5G Mobile Edge. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ACM-ICN '16, pages 237–242, Kyoto, Japan. Association for Computing Machinery.

[Gho et al., 2013] Gho, C., Yeo, H., Lim, H., Hoong, P., and Tan, I. (2013). A comparative study of tree-based and mesh-based overlay p2p media streaming. *International Journal of Multimedia and Ubiquitous Engineering*, 8(4).

[Grafl et al., 2013] Grafl, M., Timmerer, C., Hellwagner, H., Cherif, W., and Ksentini, A. (2013). Evaluation of hybrid Scalable Video Coding for HTTP-based adaptive media streaming with high-definition content. In *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–7.

[Gramatikov and Jaureguizar, 2016] Gramatikov, S. and Jaureguizar, F. (2016). Modelling and analysis of non-cooperative peer-assisted VoD streaming in managed networks. *Multimedia Tools and Applications*, 75(8):4321–4348.

[Ha et al., 2011] Ha, D., Silverton, T., and Fourmeaux, . (2011). A novel Hybrid CDN-P2P mechanism For effective real-time media streaming.

[Hakiri and Berthou, 2015] Hakiri, A. and Berthou, P. (2015). Leveraging sdn for the 5g networks: Trends, prospects and challenges.

[Hive Streaming, 2020] Hive Streaming (2020). Cdn/p2p media delivery solution. https://www.hivestreaming.com/.

[HLS, 2017] HLS (2017). Rfc 8216, http live streaming.

[hls.js, 2020] hls.js (2020). a javascript library which implements an http live streaming client. https://github.com/video-dev/hls.js/.

[Hossfeld et al., 2012] Hossfeld, T., Egger, S., Schatz, R., Fiedler, M., Masuch, K., and Lorentzen, C. (2012). 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.

[Hoßfeld et al., 2011] Hoßfeld, T., Seufert, M., Hirth, M., Zinner, T., Tran-Gia, P., and Schatz, R. (2011). Quantification of youtube QoE via crowdsourcing. In *2011 IEEE International Symposium on Multimedia*, pages 494–499. IEEE.

[Hoßfeld et al., 2013] Hoßfeld, T., Schatz, R., Biersack, E., and Plissonneau, L. (2013). Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience. In Biersack, E., Callegari, C., and Matijasevic, M., editors, *Data Traffic Monitoring and Analysis: From Measurement, Classification, and Anomaly Detection to Quality of Experience*, Lecture Notes in Computer Science, pages 264–301. Springer, Berlin, Heidelberg.

[Hoßfeld et al., 2011] Hoßfeld, T., Seufert, M., Hirth, M., Zinner, T., Tran-Gia, P., and Schatz, R. (2011). Quantification of YouTube QoE via Crowdsourcing. In *2011 IEEE International Symposium on Multimedia*, pages 494–499.

[Hoßfeld et al., 2014] Hoßfeld, T., Seufert, M., Sieber, C., and Zinner, T. (2014). Assessing effect sizes of influence factors towards a qoe model for http adaptive streaming. In *2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 111–116.

[Huang et al., 2007a] Huang, C., Li, J., and Ross, K. W. (2007a). Can internet video-on-demand be profitable? *ACM SIGCOMM Computer Communication Review*, 37(4):133.

[Huang et al., 2008] Huang, C., Wang, A., Li, J., and Ross, K. W. (2008). Understanding hybrid CDN-P2P: why limelight needs its own Red Swoosh. In *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '08, pages 75–80, Braunschweig, Germany. Association for Computing Machinery.

[Huang et al., 2007b] Huang, P., Yao, J., and Chen, H. (2007b). Design and Evaluation of a P2P IPTV System for Heterogeneous Networks. *IEEE Transactions on Multimedia*, 9(8):1568–1579.

[Huang et al., 2012] Huang, T.-Y., Handigol, N., Heller, B., McKeown, N., and Johari, R. (2012). Confused, timid, and unstable: picking a video streaming rate is hard. In *Proceedings of the 2012 ACM conference on Internet measurement conference - IMC '12*, page 225, Boston, Massachusetts, USA. ACM Press.

[Huang et al., 2013] Huang, T.-Y., Johari, R., and McKeown, N. (2013). Downton abbey without the hiccups: buffer-based rate adaptation for HTTP video streaming. In *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking - FhMN '13*, page 9, Hong Kong, China. ACM Press.

[Huang et al., 2014] Huang, T.-Y., Johari, R., McKeown, N., Trunnell, M., and Watson, M. (2014). A buffer-based approach to rate adaptation: evidence from a large video streaming service. In *Proceedings of the 2014 ACM conference on SIGCOMM - SIGCOMM '14*, pages 187–198, Chicago, Illinois, USA. ACM Press.

[Huysegems et al., 2012] Huysegems, R., De Vleeschauwer, B., Wu, T., and Van Leekwijck, W. (2012). SVC-Based HTTP Adaptive Streaming. *Bell Labs Technical Journal*, 16(4):25–41.

[IoRL, 2020] IoRL (2020). Internet of radio light project website. https://iorl.5g-ppp.eu/.

[ISO MPEG-TS, 2019] ISO MPEG-TS (2019). Iso/iec 13818-1:2019 information technology — generic coding of moving pictures and associated audio information — part 1: Systems.

[ISOBMFF, 2015] ISOBMFF (2015). Iso/iec 14496-12:2015 information technology — coding of audio-visual objects — part 12: Iso base media file format.

[ITU-T, 2017] ITU-T (2017). P.1203 : Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport. `https://www.itu.int/rec/T-REC-P.1203`.

[J. Ghoshal and Wang, 2007] J. Ghoshal, B. Ramamurthy, L. X. and Wang, M. (2007). Network architectures for live peer-to-peer media streaming. Technical report, Technical Report TR-UL-CSE-2007-020.

[Jain and Dovrolis, 2004] Jain, M. and Dovrolis, C. (2004). Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement - IMC '04*, page 272, Taormina, Sicily, Italy. ACM Press.

[Jiang et al., 2012] Jiang, J., Sekar, V., and Zhang, H. (2012). Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies - CoNEXT '12*, page 97, Nice, France. ACM Press.

[Jiang et al., 2014] Jiang, J., Sekar, V., and Zhang, H. (2014). Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive. *IEEE/ACM Transactions on Networking*, 22(1):326–340.

[Jin et al., 2016] Jin, Y., Wen, Y., and Guan, K. (2016). Toward Cost-Efficient Content Placement in Media Cloud: Modeling and Analysis. *IEEE Transactions on Multimedia*, 18(5):807–819.

[Jurca et al., 2007] Jurca, D., Chakareski, J., Wagner, J.-P., and Frossard, P. (2007). Enabling adaptive video streaming in P2P systems [Peer-to-Peer Multimedia Streaming]. *IEEE Communications Magazine*, 45(6):108–114.

[Kalva et al., 2012] Kalva, H., Adzic, V., and Furht, B. (2012). Comparing MPEG AVC and SVC for adaptive HTTP streaming. In *2012 IEEE International Conference on Consumer Electronics (ICCE)*, pages 158–159. ISSN: 2158-4001.

[Karamshuk et al., 2015] Karamshuk, D., Sastry, N., Secker, A., and Chandaria, J. (2015). ISP-friendly peer-assisted on-demand streaming of long duration content in BBC iPlayer. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 289–297, Kowloon, Hong Kong. IEEE.

[Kaune et al., 2010] Kaune, S., Rumín, R. C., Tyson, G., Mauthe, A., Guerrero, C., and Steinmetz, R. (2010). Unraveling BitTorrent's File Unavailability: Measurements and Analysis. In *2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*, pages 1–9. ISSN: 2161-3567.

[Keung, 2007] Keung, G. (2007). Coolstreaming: Design, Theory, and Practice. *IEEE Transactions on Multimedia*, 9(8):1661–1671.

[Kim et al., 2011] Kim, T. N., Jeon, S., and Kim, Y. (2011). A CDN-P2P hybrid architecture with content/location awareness for live streaming service networks. In

2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), pages 438–441. ISSN: 2159-1423.

[Kim et al., 2015] Kim, Y., Kim, Y., Yoon, H., and Yeom, I. (2015). Peer-assisted multimedia delivery using periodic multicast. *Information Sciences*, 298:425–446.

[Kua et al., 2017] Kua, J., Armitage, G., and Branch, P. (2017). A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP. *IEEE Communications Surveys & Tutorials*, 19(3):1842–1866.

[Kuhn et al., 2014] Kuhn, N., Lochin, E., Mifdaoui, A., Sarwar, G., Mehani, O., and Boreli, R. (2014). DAPS: Intelligent delay-aware packet scheduling for multipath transport. In *2014 IEEE International Conference on Communications (ICC)*, pages 1222–1227, Sydney, NSW. IEEE.

[Kurosaka and Bandai, 2015] Kurosaka, T. and Bandai, M. (2015). Multipath TCP with multiple ACKs for heterogeneous communication links. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 613–614, Las Vegas, NV, USA. IEEE.

[Lederer et al., 2012] Lederer, S., Müller, C., and Timmerer, C. (2012). Towards peer-assisted dynamic adaptive streaming over HTTP. In *2012 19th International Packet Video Workshop (PV)*, pages 161–166. ISSN: 2167-969X.

[Lei et al., 2010] Lei, J., Shi, L., and Fu, X. (2010). An experimental analysis of Joost peer-to-peer VoD service. *Peer-to-Peer Networking and Applications*, 3(4):351–362.

[Li et al., 2007] Li, J., Cui, Y., and Chang, B. (2007). PeerStreaming: design and implementation of an on-demand distributed streaming system with digital rights management capabilities. *Multimedia Systems*, 13(3):173–190.

[Li et al., 2014] Li, Z., Zhu, X., Gahm, J., Pan, R., Hu, H., Begen, A. C., and Oran, D. (2014). Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE Journal on Selected Areas in Communications*, 32(4):719–733.

[Limelight, 2020] Limelight (2020). Content Delivery & Distribution Network (CDN) | Limelight Networks. Library Catalog: www.limelight.com.

[Lin and Hwang, 2011] Lin, J.-L. and Hwang, W.-L. (2011). Efficient Scalable Video Coding Based on Matching Pursuits. In Radhakrishnan, S., editor, *Effective Video Coding for Multimedia Applications*. InTech.

[Liu et al., 2012a] Liu, H. H., Wang, Y., Yang, Y. R., Wang, H., and Tian, C. (2012a). Optimizing cost and performance for content multihoming. *ACM SIGCOMM Computer Communication Review*, 42(4):371–382.

[Liu et al., 2012b] Liu, X., Dobrian, F., Milner, H., Jiang, J., Sekar, V., Stoica, I., and Zhang, H. (2012b). A case for a coordinated internet video control plane. *ACM SIGCOMM Computer Communication Review*, 42(4):359–370.

[Liu et al., 2008] Liu, Y., Guo, Y., and Liang, C. (2008). A survey on peer-to-peer video streaming systems. *Peer-to-Peer Networking and Applications*, 1(1):18–28.

[Lu et al., 2011] Lu, Z. H., Gao, X. H., Huang, S. J., and Huang, Y. (2011). Scalable and Reliable Live Streaming Service through Coordinating CDN and P2P. In *2011 IEEE 17th International Conference on Parallel and Distributed Systems*, pages 581–588. ISSN: 1521-9097.

[Lv et al., 2012] Lv, Z., Wang, Y., and Yang, Y. R. (2012). An Analysis and Comparison of CDN-P2P-hybrid Content Delivery System and Model. *JCM*.

[Ma et al., 2016] Ma, M., Wang, Z., Su, K., and Sun, L. (2016). Understanding the Power of Smartrouter-Based Peer CDN for Video Streaming. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9, Waikoloa, HI, USA. IEEE.

[Magharei et al., 2007] Magharei, N., Rejaie, R., and Guo, Y. (2007). Mesh or multiple-tree: A comparative study of live p2p streaming approaches. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 1424–1432.

[Markakis et al., 2016] Markakis, E., Negru, D., Bruneau-Queyreix, J., Pallis, E., Mastorakis, G., and Mavromoustakis, C. (2016). *A P2P Home-Box Overlay for Efficient Content Distribution*, pages 199–220. IGI-GLOBAL.

[Medjiah et al., 2014] Medjiah, S., Ahmed, T., and Boutaba, R. (2014). Avoiding Quality Bottlenecks in P2P Adaptive Streaming. *IEEE Journal on Selected Areas in Communications*, 32(4):734–745. Conference Name: IEEE Journal on Selected Areas in Communications.

[Menasche et al., 2009] Menasche, D. S., Rocha, A. A., Li, B., Towsley, D., and Venkataramani, A. (2009). Content availability and bundling in swarming systems. In

Proceedings of the 5th international conference on Emerging networking experiments and technologies - CoNEXT '09, page 121, Rome, Italy. ACM Press.

[Merani and Natali, 2016] Merani, M. L. and Natali, L. (2016). Adaptive Streaming in P2P Live Video Systems: A Distributed Rate Control Approach. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 12(3):46:1–46:23.

[Miller et al., 2012] Miller, K., Quacchio, E., Gennari, G., and Wolisz, A. (2012). Adaptation algorithm for adaptive streaming over HTTP. In *2012 19th International Packet Video Workshop (PV)*, pages 173–178. ISSN: 2167-969X.

[Mok et al., 2018] Mok, R. K., Bajpai, V., Dhamdhere, A., and Claffy, K. (2018). Revealing the load-balancing behavior of youtube traffic on interdomain links. In *International Conference on Passive and Active Network Measurement*, pages 228–240. Springer.

[Mok et al., 2012] Mok, R. K. P., Luo, X., Chan, E. W. W., and Chang, R. K. C. (2012). QDASH: a QoE-aware DASH system. In *Proceedings of the 3rd Multimedia Systems Conference*, MMSys '12, pages 11–22, Chapel Hill, North Carolina. Association for Computing Machinery.

[Moon et al., 2013] Moon, Y.-H., Kim, J.-N., and Youn, C.-H. (2013). Churn-aware optimal layer scheduling scheme for scalable video distribution in super-peer overlay networks. *The Journal of Supercomputing*, 66(2):700–720.

[MPEG-DASH, 2012] MPEG-DASH (2012). Iso/iec 23009-1:2012. information technology — dynamic adaptive streaming over http (dash) — part 1: Media presentation description and segment formats.

[Munir et al., 2019] Munir, S., Shah, M. J., Umer, M., Shah, M. A., and Javed, M. A. (2019). A Novel Model for HD video calling in 5G Networks. In *2019 25th International Conference on Automation and Computing (ICAC)*, pages 1–6.

[Müller et al., 2012a] Müller, C., Lederer, S., and Timmerer, C. (2012a). An evaluation of dynamic adaptive streaming over HTTP in vehicular environments. In *Proceedings of the 4th Workshop on Mobile Video - MoVid '12*, page 37, Chapel Hill, North Carolina. ACM Press.

[Müller et al., 2012b] Müller, C., Renzi, D., Lederer, S., Battista, S., and Timmerer, C. (2012b). Using Scalable Video Coding for Dynamic Adaptive Streaming over HTTP

in mobile environments. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 2208–2212. ISSN: 2076-1465.

[Netflix, 2020] Netflix (2020). Watch tv shows online, watch movies online. `https://www.netflix.com`.

[Ni et al., 2011] Ni, P., Eg, R., Eichhorn, A., Griwodz, C., and Halvorsen, P. (2011). Flicker effects in adaptive video streaming to handheld devices. In *Proceedings of the 19th ACM international conference on Multimedia*, MM '11, pages 463–472, Scottsdale, Arizona, USA. Association for Computing Machinery.

[Nicolas et al., 2013] Nicolas, Y., Wolff, D., Rossi, D., and Finamore, A. (2013). I Tube, YouTube, P2PTube: Assessing ISP benefits of peer-assisted caching of YouTube content. In *IEEE P2P 2013 Proceedings*, pages 1–2. ISSN: 2161-3567.

[Nightingale et al., 2018] Nightingale, J., Salva-Garcia, P., Calero, J. M. A., and Wang, Q. (2018). 5G-QoE: QoE Modelling for Ultra-HD Video Streaming in 5G Networks. *IEEE Transactions on Broadcasting*, 64(2):621–634. Conference Name: IEEE Transactions on Broadcasting.

[Novage, 2020] Novage (2020). P2p-media-loader. open source project. https://github.com/novage/p2p-media-loader.

[Nunes et al., 2014] Nunes, B. A. A., Mendonca, M., Nguyen, X., Obraczka, K., and Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys Tutorials*, 16(3):1617–1634.

[OIPF, 2014] OIPF (2014). OIPF - Release 2 Specification - Volume 2a, HTTP Adaptive Streaming, V2.3.

[Ordonez-Lucena et al., 2017] Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J. J., Lorca, J., and Folgueira, J. (2017). Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges. *IEEE Communications Magazine*, 55(5):80–87.

[Oyman and Singh, 2012] Oyman, O. and Singh, S. (2012). Quality of experience for HTTP adaptive streaming services. *IEEE Communications Magazine*, 50(4):20–27.

[P2P Architecture, 2009] P2P Architecture (2009). Rfc 5694, peer-to-peer (p2p) architecture: Definition, taxonomies, examples, and applicability.

[Passarella, 2012] Passarella, A. (2012). A survey on content-centric technologies for the current Internet: CDN and P2P solutions. *Computer Communications*, 35(1):1–32.

[Patent US8868772, 2014] Patent US8868772 (2014). Apparatus, system, and method for adaptive-rate shifting of streaming content.

[Pathan and Buyya, 2008] Pathan, M. and Buyya, R. (2008). A Taxonomy of CDNs. In Buyya, R., Pathan, M., and Vakali, A., editors, *Content Delivery Networks*, Lecture Notes Electrical Engineering, pages 33–77. Springer, Berlin, Heidelberg.

[Peer5, 2020] Peer5 (2020). Last mile delivery: Ensure coverage in underserved regions and internal networks with peer-assisted delivery. `https://www.peer5.com/p2p`.

[PeerTube, 2020] PeerTube (2020). A decentralized video hosting network, based on free/libre software. `https://joinpeertube.org/en/`.

[Peterson and Sirer, 2009] Peterson, R. and Sirer, E. (2009). Antfarm: Efficient content distribution with managed swarms. *NSDI*, pages 107–122.

[Popa et al., 2010] Popa, L., Ghodsi, A., and Stoica, I. (2010). HTTP as the narrow waist of the future internet. In *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks - Hotnets '10*, pages 1–6, Monterey, California. ACM Press.

[Pu et al., 2011] Pu, W., Zou, Z., and Chen, C. W. (2011). Dynamic adaptive streaming over http from multiple content distribution servers. In *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pages 1–5.

[Qiu and Srikant, 2004] Qiu, D. and Srikant, R. (2004). Modeling and performance analysis of BitTorrent-like peer-to-peer networks. *ACM SIGCOMM Computer Communication Review*, 34(4):367–378.

[Qiu et al., 2010] Qiu, X., Liu, H., Li, D., Zhang, S., Ghosal, D., and Mukherjee, B. (2010). Optimizing HTTP-based Adaptive Video Streaming for wireless access networks. In *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, pages 838–845.

[Quinn and Almeroth, 2001] Quinn, B. and Almeroth, K. (2001). IP Multicast Applications: Challenges and Solutions. Technical Report RFC3170, RFC Editor.

[Raiciu et al., 2012] Raiciu, C., Paasch, C., Barre, S., Ford, A., Honda, M., Duchene, F., Bonaventure, O., and Handley, M. (2012). How hard can it be? designing and implementing a deployable multipath TCP. In *Proceedings of the 9th USENIX conference*

on Networked Systems Design and Implementation, NSDI'12, page 29, San Jose, CA. USENIX Association.

[Ratnasamy et al., 2001] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Schenker, S. (2001). A scalable content-addressable network. *ACM SIGCOMM Computer Communication Review*, 31(4):161–172.

[Roverso et al., 2012] Roverso, R., El-Ansary, S., and Haridi, S. (2012). SmoothCache: HTTP-Live Streaming Goes Peer-to-Peer. In Bestak, R., Kencl, L., Li, L. E., Widmer, J., and Yin, H., editors, *NETWORKING 2012*, Lecture Notes in Computer Science, pages 29–43, Berlin, Heidelberg. Springer.

[Roverso et al., 2011] Roverso, R., Naiem, A., Reda, M., El-Beltagy, M., El-Ansary, S., Franzen, N., and Haridi, S. (2011). On the feasibility of centrally-coordinated Peer-to-Peer live streaming. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 1061–1065. ISSN: 2331-9860.

[RSVP, 1997] RSVP (1997). Rfc 2205, resource reservation protocol (rsvp).

[RTCP, 2003] RTCP (2003). Rfc 3611, rtp control protocol extended reports (rtcp xr).

[RTP, 2003] RTP (2003). Rfc 3550, rtp: A transport protocol for real-time applications.

[RTSP, 1998] RTSP (1998). Rfc 2326, real time streaming protocol (rtsp).

[Sahasrabuddhe and Mukherjee, 2000] Sahasrabuddhe, L. and Mukherjee, B. (2000). Multicast routing algorithms and protocols: a tutorial. *IEEE Network*, 14(1):90–102.

[Sandvine, 2019] Sandvine (2019). Global internet phenomena report. Technical report, Sandvine.

[Sandvine, 2020] Sandvine (2020). Mobile internet phenomena report. Technical report, Sandvine.

[Sani et al., 2017] Sani, Y., Mauthe, A., and Edwards, C. (2017). Adaptive Bitrate Selection: A Survey. *IEEE Communications Surveys & Tutorials*, 19(4):2985–3014.

[Scellato et al., 2011] Scellato, S., Mascolo, C., Musolesi, M., and Crowcroft, J. (2011). Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades. In *Proceedings of the 20th international conference on World wide web - WWW '11*, page 457, Hyderabad, India. ACM Press.

[Schwarz et al., 2007] Schwarz, H., Marpe, D., and Wiegand, T. (2007). Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120.

[Schwarz and Wien, 2008] Schwarz, H. and Wien, M. (2008). The Scalable Video Coding Extension of the H.264/AVC Standard [Standards in a Nutshell]. *IEEE Signal Processing Magazine*, 25(2):135–141.

[SDP, 1998] SDP (1998). Rfc 2327, sdp: Session description protocol.

[Seufert et al., 2014] Seufert, M., Egger, S., Slanina, M., Zinner, T., Hoßfeld, T., and Tran-Gia, P. (2014). A survey on quality of experience of HTTP adaptive streaming. *IEEE Communications Surveys & Tutorials*, 17(1):469–492.

[Seufert et al., 2015] Seufert, M., Egger, S., Slanina, M., Zinner, T., Hoßfeld, T., and Tran-Gia, P. (2015). A Survey on Quality of Experience of HTTP Adaptive Streaming. *IEEE Communications Surveys Tutorials*, 17(1):469–492. Conference Name: IEEE Communications Surveys Tutorials.

[Seyyedi and Akbari, 2011] Seyyedi, S. M. Y. and Akbari, B. (2011). Hybrid cdn-p2p architectures for live video streaming: Comparative study of connected and unconnected meshes. In *2011 International Symposium on Computer Networks and Distributed Systems (CNDS)*, pages 175–180.

[Sieber et al., 2013] Sieber, C., Hoßfeld, T., Zinner, T., Tran-Gia, P., and Timmerer, C. (2013). Implementation and user-centric comparison of a novel adaptation logic for DASH with SVC. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 1318–1323. ISSN: 1573-0077.

[Silverlight, 2017] Silverlight (2017). Silverlight 4 Launch Home | Microsoft Silverlight.

[Singh et al., 2012] Singh, A., Goerg, C., Timm-Giel, A., Scharf, M., and Banniza, T.-R. (2012). Performance comparison of scheduling algorithms for multipath transfer. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 2653–2658. ISSN: 1930-529X.

[Singh et al., 2013] Singh, V., Ahsan, S., and Ott, J. (2013). Mprtp: Multipath considerations for real-time media. In *Proceedings of the 4th ACM Multimedia Systems Conference*, MMSys '13, page 190–201, New York, NY, USA. Association for Computing Machinery.

[Sodagar, 2011] Sodagar, I. (2011). The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia*, 18(4):62–67.

[Sodagar, 2011] Sodagar, I. (2011). The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia*, 18(4):62–67. Conference Name: IEEE MultiMedia.

[Spiteri et al., 2016] Spiteri, K., Urgaonkar, R., and Sitaraman, R. K. (2016). BOLA: Near-optimal bitrate adaptation for online videos. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, San Francisco, CA, USA. IEEE.

[Spotify, 2020] Spotify (2020). https://www.spotify.com/.

[Stockhammer, 2011] Stockhammer, T. (2011). Dynamic adaptive streaming over HTTP –: standards and design principles. In *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11*, page 133, San Jose, CA, USA. ACM Press.

[Streamroot, 2020] Streamroot (2020). Powering the next generation of video delivery. `https://streamroot.io`.

[Stutzbach and Rejaie, 2006] Stutzbach, D. and Rejaie, R. (2006). Understanding churn in peer-to-peer networks. In *Proceedings of the 6th ACM SIGCOMM on Internet measurement - IMC '06*, page 189, Rio de Janeriro, Brazil. ACM Press.

[Sánchez de la Fuente et al., 2011] Sánchez de la Fuente, Y., Schierl, T., Hellge, C., Wiegand, T., Hong, D., De Vleeschauwer, D., Van Leekwijck, W., and Le Louédec, Y. (2011). iDASH: improved dynamic adaptive streaming over HTTP using scalable video coding. In *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11*, page 257, San Jose, CA, USA. ACM Press.

[Tappayuthpijarn et al., 2011] Tappayuthpijarn, K., Stockhammer, T., and Steinbach, E. (2011). HTTP-based scalable video streaming over mobile networks. In *2011 18th IEEE International Conference on Image Processing*, pages 2193–2196. ISSN: 2381-8549.

[Thang et al., 2012] Thang, T., Ho, Q.-D., Kang, J., and Pham, A. (2012). Adaptive streaming of audiovisual content using MPEG DASH. *IEEE Transactions on Consumer Electronics*, 58(1):78–85.

[Tian and Liu, 2012] Tian, G. and Liu, Y. (2012). Towards agile and smooth video adaptation in dynamic HTTP streaming. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, CoNEXT '12, pages 109–120, Nice, France. Association for Computing Machinery.

[Tian and Liu, 2013] Tian, G. and Liu, Y. (2013). On Adaptive HTTP Streaming to Mobile Devices. In *2013 20th International Packet Video Workshop*, pages 1–8, San Jose, CA. IEEE.

[Twitch, 2020] Twitch (2020). Twitch is the world's leading video platform and community for gamers. `https://www.twitch.tv`.

[Unanue et al., 2011] Unanue, I., Urteaga, I., Husemann, R., Del, J., Roesler, V., Rodriguez, A., and Sanchez, P. (2011). A Tutorial on H.264/SVC Scalable Video Coding and its Tradeoff between Quality, Coding Efficiency and Performance. In Del Ser Lorente, J., editor, *Recent Advances on Video Coding*. InTech.

[Varma, 2015] Varma, S. (2015). Flow Control for Video Applications. In *Internet Congestion Control*, pages 173–203. Elsevier.

[Video.js, 2020] Video.js (2020). A web video player built from the ground up for an html5 world. https://videojs.com/.

[Vimeo, 2020] Vimeo (2020). Vod and live streaming platform. https://www.vimeo.com/.

[WebRTC, 2015] WebRTC (2015). Rfc7478, web real-time communication use cases and requirements.

[Wendell and Freedman, 2011] Wendell, P. and Freedman, M. J. (2011). Going viral: flash crowds in an open CDN. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference - IMC '11*, page 549, Berlin, Germany. ACM Press.

[Wu et al., 2009] Wu, D., Liang, C., Liu, Y., and Ross, K. (2009). View-Upload Decoupling: A Redesign of Multi-Channel P2P Video Systems. In *IEEE INFOCOM 2009*, pages 2726–2730. ISSN: 0743-166X.

[Wu et al., 2016] Wu, J., Yuen, C., Cheng, B., Wang, M., and Chen, J. (2016). Streaming High-Quality Mobile Video with Multipath TCP in Heterogeneous Wireless Networks. *IEEE Transactions on Mobile Computing*, 15(9):2345–2361.

[Wu et al., 2017] Wu, X., Zhao, C., Xie, R., and Song, L. (2017). Low Latency MPEG-DASH System Over HTTP 2.0 and WebSocket. In *IFTC*.

[Xiang et al., 2012] Xiang, S., Cai, L., and Pan, J. (2012). Adaptive scalable video streaming in wireless networks. In *Proceedings of the 3rd Multimedia Systems Conference on - MMSys '12*, page 167, Chapel Hill, North Carolina. ACM Press.

[Xiao et al., 2009] Xiao, X., Shi, Y., Gao, Y., and Zhang, Q. (2009). LayerP2P: A New Data Scheduling Approach for Layered Streaming in Heterogeneous Networks. In *IEEE INFOCOM 2009*, pages 603–611. ISSN: 0743-166X.

[Xu et al., 2006] Xu, D., Kulkarni, S. S., Rosenberg, C., and Chai, H.-K. (2006). Analysis of a CDN–P2P hybrid architecture for cost-effective streaming media distribution. *Multimedia Systems*, 11(4):383–399.

[Yahia et al., 2019] Yahia, M. B., Louedec, Y. L., Simon, G., Nuaymi, L., and Corbillon, X. (2019). HTTP/2-based Frame Discarding for Low-Latency Adaptive Video Streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 15(1):18:1–18:23.

[Yin et al., 2009a] Yin, H., Liu, X., Zhan, T., Sekar, V., Qiu, F., Lin, C., Zhang, H., and Li, B. (2009a). Design and deployment of a hybrid CDN-P2P system for live video streaming: experiences with livesky. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 25–34.

[Yin et al., 2009b] Yin, H., Liu, X., Zhan, T., Sekar, V., Qiu, F., Lin, C., Zhang, H., and Li, B. (2009b). Design and deployment of a hybrid CDN-P2P system for live video streaming: experiences with LiveSky. In *Proceedings of the 17th ACM international conference on Multimedia*, MM '09, pages 25–34, Beijing, China. Association for Computing Machinery.

[Yin et al., 2010] Yin, H., Liu, X., Zhan, T., Sekar, V., Qiu, F., Lin, C., Zhang, H., and Li, B. (2010). LiveSky: Enhancing CDN with P2P. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 6(3):16:1–16:19.

[Yin et al., 2015] Yin, X., Jindal, A., Sekar, V., and Sinopoli, B. (2015). A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. *ACM SIGCOMM Computer Communication Review*, 45(4):325–338.

[Yitong et al., 2013] Yitong, L., Yun, S., Yinian, M., Jing, L., Qi, L., and Dacheng, Y. (2013). A study on Quality of Experience for adaptive streaming service. In *2013*

IEEE International Conference on Communications Workshops (ICC), pages 682–686. ISSN: 2164-7038.

[YouTube, 2020] YouTube (2020). Share your videos with friends, family, and the world. `https://www.youtube.com`.

[Youtube, 2020] Youtube (2020). Vod and live streaming platform. https://www.youtube.com/.

[Zhang et al., 2014] Zhang, G., Liu, W., Hei, X., and Cheng, W. (2014). Unreeling xunlei kankan: Understanding hybrid CDN-P2P video-on-demand streaming. *IEEE Transactions on Multimedia*, 17(2):229–242.

[Zhang et al., 2015a] Zhang, G., Liu, W., Hei, X., and Cheng, W. (2015a). Unreeling Xunlei Kankan: Understanding Hybrid CDN-P2P Video-on-Demand Streaming. *IEEE Transactions on Multimedia*, 17(2):229–242. Conference Name: IEEE Transactions on Multimedia.

[Zhang et al., 2015b] Zhang, S., Li, B., and Li, B. (2015b). Presto: Towards fair and efficient HTTP adaptive streaming from multiple servers. In *2015 IEEE International Conference on Communications (ICC)*, pages 6849–6854. ISSN: 1938-1883.

[Zhang et al., 2005a] Zhang, X., chuan Lin, J., Li, B., Tak-Shing, and Yum, P. (2005a). Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, page 2102–2111. IEEE.

[Zhang et al., 2005b] Zhang, X., Liu, J., Li, B., and Yum, Y.-S. (2005b). CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 2102–2111 vol. 3. ISSN: 0743-166X.

[Zhao et al., 2013] Zhao, M., Aditya, P., Chen, A., Lin, Y., Haeberlen, A., Druschel, P., Maggs, B., Wishon, B., and Ponec, M. (2013). Peer-assisted content distribution in Akamai netsession. In *Proceedings of the 2013 conference on Internet measurement conference - IMC '13*, pages 31–42, Barcelona, Spain. ACM Press.

[Zhao et al., 2016] Zhao, S., Li, Z., and Medhi, D. (2016). Low delay MPEG DASH streaming over the WebRTC data channel. *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*.

[Zhou et al., 2013a] Zhou, C., Lin, C.-W., Zhang, X., and Guo, Z. (2013a). Buffer-based smooth rate adaptation for dynamic HTTP streaming. In *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1–9, Kaohsiung, Taiwan. IEEE.

[Zhou et al., 2013b] Zhou, C., Zhang, X., and Guo, Z. (2013b). A control theory based rate adaption scheme for dash over multiple servers. In *2013 Visual Communications and Image Processing (VCIP)*, pages 1–6.

[Zink et al., 2009] Zink, M., Suh, K., Gu, Y., and Kurose, J. (2009). Characteristics of YouTube network traffic at a campus network – Measurements, models, and implications. *Computer Networks*, 53(4):501–514.