Combining in-network caching, HTTP adaptive streaming and multipath to improve video quality of experience

Vitalii Poliakov

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Amélioration de la Qualité d'Expérience vidéo en combinant Streaming Adaptif, Caching Réseau et Multipath

Vitalii POLIAKOV

Université Côte d'Azur, CNRS, I3S

Présentée en vue de l'obtention du grade de docteur en Informatique d'Université Côte d'Azur
Dirigée par :
Dr. Lucile Sassatelli
Dr. Damien Saucez
Soutenue le : 11 décembre 2018

Devant le jury, composé de :
Prof. César Viho…….Examinateur, Président du jury, Université de Rennes I
Prof. Rachid El-Azouzi..Rapporteur, Université d'Avignon et Pays de Vaucluse
Prof. Stefano Secci……..Rapporteur, CNAM
Dr. Jérémie Leguay……Invité, Huawei Research France
Dr. Paolo Medagliani….Invité, Huawei Research France
Dr. Lucile Sassatelli…....Directrice de thèse, Université Côte d'Azur, CNRS, I3S
Dr. Damien Saucez…….Co-encadrant, Université Côte d'Azur, INRIA Méditerranée

Combining in-network caching, HTTP Adaptive Streaming and Multipath to improve video Quality Of Experience

Vitalii Poliakov

Université Côte d'Azur, CNRS, I3S

This dissertation is submitted for the degree of
Doctor of Philosophy

Université Côte d'Azur December 2018

Abstract

Volume of video traffic has grown considerably in recent years: Cisco has predicted that its share in the Internet will attain 81% by 2021, doubling its net traffic volume as compared to 2016. Such an abrupt increase in video traffic is pushing the capabilities of Internet Service Providers' networks (ISPs) to their limits, which results in their overutilisation and, consequently, decreased users' Quality of Experience (QoE) of video sessions.

This thesis attempts to tackle the problem of improving users' video QoE without relying on ISPs to upgrade their networks. To achieve this, we have chosen to combine such technologies as in-network caching, HTTP Adaptive Streaming (HAS), and multipath data transport. We start with exploration of interaction between HAS and caching. We design an optimal boundary for performance of a HAS quality adaptation algorithm, and compare it to the performance of Rate-Based, Buffer-Based, and state-of-the-art optimisation-based algorithm called Fast MPC. We confirm the benefits of achieving cache-awareness in quality adaptation when operating in a presence of proxy cache, and propose such an extension to Fast MPC.

Concluding on the difficulty of achieving cache-awareness, we take a step back to study a video delivery system on a large scale, where in-network caches are represented by Content Delivery Networks (CDNs). They deploy caches inside ISPs and dispose of their own outside video servers. As a novelty, we consider users to have a simultaneous multipath connectivity to several ISP networks. We anticipate, however, that in-network caches cannot be accessed with multipath as they are not accessible from outside networks. This brings two new operational points to the system: video clients can either access outside multipath servers with aggregate bandwidth (which may increase their QoE, but will also bring more traffic into ISP), or stream their content from a closer cache through only single connectivity (bringing less traffic into ISP). This potential disagreement in ISP and CDN objectives leads to suboptimal system performance, which we identify by modelling an optimal boundary for joint ISP – CP performance. In response to this, we develop two collaboration schemes between two actors, performance of which can approach optimal boundary for our settings, and discuss its practical implementation.

Keywords–Internet video streaming, HTTP Adaptive Streaming, Caching, Multipath, Quality of Experience, Content Delivery Network, Linear optimisation

Résumé

Le volume de trafic vidéo s'est considérablement accru pendant ces dernières années: selon Cisco, sa part dans l'Internet atteindra 81% d'ici 2021, soit le double du volume de son trafic net par rapport à 2016. Cette augmentation aussi importante du trafic vidéo repousse les capacités des réseaux des Fournisseurs de Access Internet (FAI), ce qui entraîne leur surutilisation et, par conséquent, la dégradation de la Qualité d'Expérience (QoE) vidéo perçu par leurs utilisateurs.

Cette thèse tente de résoudre le problème de l'amélioration de la QoE vidéo des utilisateurs sans compter sur les FAI pour mettre à niveau leurs réseaux. Pour ce faire, nous avons choisi de combiner des technologies telles que la mise en cache (Caching), le streaming adaptatif HTTP (HAS) et le transport de données Multipath. Nous commençons par l'exploration de l'interaction entre HAS et le Caching. Nous concevons une limite optimale pour la performance d'un algorithme d'adaptation de la qualité pour la HAS et la comparons à la performance de l'algorithmes basés de débit, de buffer, et à la pointe de l'optimisation appelé Fast MPC. Nous confirmons les avantages de la reconnaissance du cache dans l'adaptation de la qualité lors de l'utilisation d'un cache proxy et proposons une telle extension à Fast MPC.

Concluant sur la difficulté de la reconnaissance de la cache, nous prenons un pas en arrière pour étudier un système de diffusion vidéo à grande échelle, où les caches en réseau sont représentés par des réseaux de distribution de contenu (Content Delivery Networks, CDN). Ils déploient des caches à l'intérieur des réseaux des FAI et disposent de leurs propres serveurs vidéo externes. Comme nouveauté, nous considérons que les utilisateurs disposent d'une connectivité multi-accès simultanée vers plusieurs réseaux de FAI. Nous prévoyons toutefois que l'accès aux caches dans le réseau ne soit pas possible par Multipath, car ils ne sont pas accessibles depuis des réseaux extérieurs. Cela apporte deux nouveaux régimes opérationnels au système: les clients vidéo peuvent accéder aux serveurs Multipath extérieurs avec une bande passante agrégée (ce qui peut augmenter leur QoE, mais apportera également plus de trafic au FAI), ou diffuser leur contenu à partir d'un cache plus étroit via une seule connexion (apportant moins de trafic dans FAI). Ce désaccord potentiel dans les objectifs FAI et CDN conduit à des performances système non optimales, que nous identifions en

modélisant une limite optimale pour les performances FAI – CP conjointes. En réponse à cela, nous développons deux schémas de collaboration entre deux acteurs, dont les performances peuvent s'approcher des limites optimales pour nos paramètres, et nous discutons de sa mise en oeuvre pratique.

Keywords–Diffusion vidéo sur Internet, HTTP Adaptive Streaming, Caching, Multipath, Qualité d'Experience, Réseau de diffusion de contenu, Optimisation linéaire

Acknowledgements

Indeed, the first and foremost acknowledgement goes to my supervisors: Lucile Sassatelli and Damien Saucez. Indeed, I would not be able to finish the presented PhD project without their support and input, but instead I would like to express my gratitude for the efforts, time and devotion they have put to make me a better researcher. The path ahead is long and winding, but their advising have for sure made myself a better thinker (not for me to judge by how much though). Their radically different outlook on the domain had certainly not made my time easier, but in return it granted me with a wider understanding of internetworking research and differences in the ways to tackle it. I will never forget the unending patience of Lucile in the process of my (often slow) general and mathematical training, nor the hours-long discussions with Damien on most different subjects that always carried some useful message. I thank you for this.

Karyna, my love, has been an indissociable part of my time as a doctoral student. I went all this path holding hands with her, receiving her support and growing together with her. I can't thank her enough for helping me keep my mind intact in the finishing months of my thesis, when the nights were the most dark and full of terrors. See you on the other side!

I should thank all the folks at I3S and INRIA, both permanent and not, both graduated and present. I had interesting discussions with many of you, and being a part of these two labs have helped to make my three years bright and amusing.

I'd like also to say thanks to Prof. Eric Rondeau and Dr. Jean-Philippe Georges, whose eventual decision to respond to my several days-late message have ultimately given me an opportunity to behold the incomprehensible beauty of an arbutus tree.

This thesis marks the end of my studentship, which happened to be quite long. Numerous people have taken part in shaping my skills and personality over all these years (for good or bad), but the most important of those are my parents and relatives. They are in most part responsible in letting me see this day alive and become the person I am today, and there will never be a word good enough to express my recognition of this.

Vitalii Poliakov,
10 Dec 2018

Table of contents

List of figures

List of tables

Chapter 1

Introduction and background

Ever since computer networking left comfortable and well-controlled confines of university research, the battle between Internet Service Providers (ISPs) and increasing user demands has been taking place in all its ferociousness. Since the very beginning, the answer to increased bandwidth demand has been to over-provision ISP networks. This race has been one of the main driving force behind advancements in data transport technologies for local, metropolitan, and wide-area networks. However, two main factors make such an approach insufficient to meet users' expectations already today.

First, proliferation of broadband internet access in the 2000s and emergence of Internet video platforms (most notably, Youtube and online Video-on-Demand providers) has brought a manyfold increase in video demands. Being a very bandwidth-consuming kind of application, together with unprecedented demand volumes, video traffic is forecasted by Cisco to represent 82% of all Internet traffic by 2021 up from already observed 60% in 2016 [35]. This is not a limit: a very recent proliferation of devices supporting viewing and creation of Ultra High Definition (4096+ lines) video content with elevated framerates (60+ frames per second) and Virtual Reality applications suggest that video traffic volumes today are far from being on their peak: for instance, the same Cisco report [35] predicts 20-fold increase in VR (virtual reality) and AR (augmented reality) traffic by 2021.

Second, the amount of Internet users does not yet seem to have attained its limit. Only about half of Earth population has continuous access to Internet today (4.2 billion persons, according to Internet World Stats[1]), mostly living in developed or actively developing countries. Internet connectivity becoming a commodity nowadays, we can, perhaps, expect the rest of human population to become connected in coming (though maybe not nearest) years. In addition to that, report [20] suggests that the number of Internet-capable devices

[1] https://www.internetworldstats.com/stats.htm

per capita will increase up to 3.5 by 2021, being a consequence of Internet of Things industry development.

These two factors demonstrate how the future of Internet demand may become too overwhelming for both ISPs and technological advancements in telecoms they rely on. In the shed of this perspective, a part of today's internetworking research is attempting to tackle the coming challenges by improving efficiency of infrastructure utilisation, rather than by increasing its capacity, especially in the domain of Internet video.

One of the basic techniques to decrease the volume of video traffic circulating in the networks is in-network caching [97, 163, 120]. Though mainly developed as a solution to improve client's access latency to web-services, the idea of storing content close to the user has proven to have a potential in the video delivery domain. Quite often, network topologies are designed with larger capacity close to the user [25, 37]. Placing a cache in the closer-to-user, well-provisioned part of the network may help ISP to avoid transporting video (and other) traffic through its entire backhaul network multiple times – but, instead, to carry it only once and then stream from the cache for other interested users. Research in the area of caching has got boosted recently with introduction of Mobile Edge Computing (MEC [17, 56]) and Edge Caching [94]. In the context of video delivery, these ideas stress upon an opportunity to use network edge (mainly wireless) to cache video content and to perform transcoding in order to reduce the amounts of video traffic in the backhaul and above.

Until recently, there existed no specific standard for Internet video transmission and its functionality, in particular, quality adaptation and ability to be cached. Real Time Protocol (RTP [140]) and its family was designed to facilitate streaming of multimedia content in best-effort networks. This includes Multicast delivery, flow monitoring, media payload identification and sequence timestamping. However, being an OSI protocol, RTP did not address content storage format or its quality adaptation. HTTP could also be used to transport media, though treating it as regular web-content. *Range-requests* headers could be used to have a limited control over video playback. Not only this was complicating the task of video caching (with objects being too large or too unique, like in range-request byte ranges that can be different for every individual user), but it also made quality adaptation rather complex. The emergence of the concept of HTTP Adaptive Streaming [122, das] has given a significant advance towards solving these two tasks. With video files being transmitted by HTTP and stored in pre-defined segments (also called chunks) that are the same for everybody within the service, it finally became possible to easily (even transparently) cache video content with high hit-rates and to switch video qualities on-the-fly without complicated calculations of byte ranges. An indissociable part of HAS is its quality adaptation algorithm; having started

with simple downlink bitrate estimations (Rate-Based algorithm), research in this direction made quite a long way to even exploiting machine learning for quality selection [102].

With development of multipath data transport (such as MPTCP [15]) and emerging concept of Multi-RAT [27], a new perspective on serving heavy content to the clients have opened. Originally, multipath transport has been meant mainly for resiliency [90, 137]; our consideration is to use multipath bandwidth aggregation so as to improve clients' connectivity. Today, even with all the recent advancements in mobile radio technologies, average LTE downlink bandwidth is less than 15 Mbps in many developed countries (e.g., 13.4 Mbps in France and 12.3 Mbps in USA [fra]), which gives no room for future video applications as described above [3]. Even the advent of 5G mostly promises improvements in radio-access bandwidth for users, though observing the LTE figures just above can make one wonder if the promise of 1 Gbps downlink rate would hold in real environment with typical consumer devices. Since mobile video consumption becomes more and more prevalent, developing a solution to alleviate slow (on average) mobile connectivity becomes an important endeavour. MPTCP gives an opportunity to use several wireless connectivities (for instance, Wi-Fi and LTE) simultaneously as to obtain a higher aggregate connectivity bandwidth which may help in serving demanding video applications.

Different to technological advancements in Internet video delivery referenced above, the concept of Quality of Experience (QoE [111, 130, 157]) is helping researchers to measure and quantify the improvements in the domain of video delivery (among others). As opposed to the concept of Quality of Service (QoS), low latency or high bandwidth might not always result in good user experience [75]. Instead, subjective assessment, like Mean Opinion Score (MOS [129, 154]), or application-level video metrics, like duration of stalls or video frame Peak Signal-to-Noise-Ratio (PSNR), can be used to assess exact user experience. Direct focus on the latter, therefore, provides a good base to design and test novel video delivery techniques.

This thesis presents a work on the intersection of these technologies (in-network caching, HAS, multipath transport), with a goal of improving Quality of Experience in video applications in the shed of the mentioned future challenges. We consider caching to be the enabler in reducing video traffic volume in ISP networks, while HAS is the key to provide video users with as much QoE as possible given the changing level of QoS in best-effort networks. Multipath transport, in its turn, is expected to allow consumption of very high bitrate video types. Using them altogether to achieve our goal, however, has proven to not be straightforward. The problems and contributions of this thesis are outlined as following:

- It has been reported that Rate-Based quality adaptation algorithm for HAS can be fooled by the presence of a transparent in-network cache; this effect has been given a name of *bitrate oscillations* [84]. This happens due to inability of such an algorithm to predict the provenance of an incoming video segment (close cache or far away server), and therefore, its download bitrate. We conduct a study of this phenomenon with a goal of identifying the magnitude of its effect on video QoE. To achieve this, we ask the following questions. If we would imagine a perfectly cache-aware quality adaptation algorithm that would always know the download bitrate of future segments in any quality – how would it perform in terms of QoE metrics in the case of cache presence, and how far from this perfect boundary would commonly-implemented algorithms perform? Considering that novel Fast MPC quality adaptation algorithm was developed as an optimisation-based algorithm, how would it perform under cache presence? To answer presented questions, this thesis offers the following contributions:

  1. By applying Mixed-Integer Linear Programming (MILP), we define a performance boundary for a quality adaptation algorithm in a presence of cache;
  2. We test Rate-Based and Buffer-Based adaptation algorithms against this boundary in a realistic experimental testbed (based on modified VLC media player), monitoring important operational QoE metrics (video quality, stall duration and quality changes) and observe a room for improvement in their performance – as compared to the boundary;
  3. We test a novel Fast MPC adaptation algorithm and find that it, though being rather close to the boundary, offers suboptimal performance for some clients. We then examine whether cache-awareness (i.e., knowing the provenance of future segments, but not their exact bitrate) can help alleviate this drawback – and conclude that it can effectively decrease the amount of quality switches for those clients and generally improve performance of others in terms of quality switches and stall duration;
  4. We discuss the implementability of cache-awareness.

- We see a potential in using multiple connectivity at the client's device in terms of bandwidth aggregation (that may lead to higher service QoE). On the other hand, CPs and their Content Delivery Networks (CDNs) often deploy their own caches at ISPs in addition to the servers deployed in their own premises. Since caches inside ISPs are not meant to be reachable from outside of their networks, we can either serve clients from in-network caches using singlepath transport (lower traffic strain on ISP but possibly lower client QoE), or give up on caching benefits in order to use the outside

server using multipath transport (higher ISP traffic strain but possibly higher user QoE). Having two parties, namely, ISP and CP with such a potentially contradicting interest may render the operation of a video delivery system suboptimal in multipath scenario. Nevertheless, multipath brings new operational points to a video delivery system by introducing such a contradiction. What benefit do these new operational points bring for the different interest of Content Provider (higher client quality) and Internet Service Provider (need to serve new and more demanding services, quality for all services, low congestion)? Or, in other words, what strategy is better for both of those actors and for which use-cases: serving from higher in the network to potentially achieve higher bandwidth, or closer to the client edge to decrease network congestion? Then, in case if CP and ISP collaborated in order to achieve a mutually-beneficial performance of a video delivery system in terms of their objectives, how would this performance differ from a present-day scenario (where independent ISPs and CPs do not collaborate? To study these new options, this thesis provides the following contributions:

1. Using MILP, we define a theoretical boundary for perfect joint ISP – CP performance in terms of their objectives (network congestion and achieved video bitrate correspondingly) in a multipath-enabled system;
2. In the same way, we model the behaviour of two actors of a video delivery system: an ISP and CP. We conduct a numerical evaluation of the new operational points, that are brought by multipath, with respect to actors' objectives. We find that, according to the perfect boundary, performance of both actors can be significantly improved if we depart from non-collaborative strategies, by as far as up to 50% less network congestion at the same level of video quality in our simulation settings;
3. We design two collaboration schemes that involve a different amount of interaction and test them against the perfect theoretical boundary. Our evaluation proves that these strategies, despite light and distributed (thanks to Lagrangian Relaxation), bring substantial improvement to the performance of both ISP and CP as compared to non-collaborative strategy, as well as in fairness between their objectives;
4. We discuss the real-world implementability of those collaboration schemes.

- Research in video delivery systems involving HAS and multipath often necessitates a focus on large systems. Big research groups with connections in content provider industry can use real infrastructures to conduct their research. Other researchers are left with resorting to simulations or numerical evaluations for this kind of studies.

However, there exist no open-source network simulator that would support HAS and multipath transport out-of-the-box. There exist several isolated community models for MPEG-DASH and MPTCP (representing both technologies) for the NS-3 [51] network simulator, but we found it highly difficult to integrate them together so as to obtain a distribution with both features built-in. Having a network simulator with such features would help the community to continue conducting quality research on large-scale video delivery beyond numerical evaluations. In this thesis, we will briefly present our efforts on building an NS-3 distribution that includes MPEG-DASH and MPTCP, which is our last contribution.

The manuscript is organised in four chapters, apart from this introduction, and an Appendix.

A brief overview of internet video delivery evolution and current state of the art, which is the subject of Chapter 1 right after this introduction.

HAS quality adaptation and caching: in Chapter 2, we confirm that indeed, in certain conditions, Rate-Based quality adaptation of HAS can be fooled by the cache presence resulting in bitrate oscillations. We formulate an optimal boundary for a performance of hypothetical perfectly cache-aware adaptation algorithm for our scenario. Then we test Rate-Based and Buffer-Based algorithms against this issue and compare their QoE performance to the boundary. After that we present a recently-proposed optimisation-based algorithm called Fast MPC and proceed to its evaluation against the optimal boundary. We find that Fast MPC still suffers from cache presence in our realistic settings, we discuss a cache-aware extension to it. We demonstrate that cache-awareness can greatly decrease the amount of quality switches and stall duration for the clients that are among the first to request a popular video from the outside network. We discuss how is it possible to achieve cache-awareness for a quality-adaptation algorithm, and come to a conclusion that caching infrastructure may benefit from being managed by the CP or having explicit or implicit signalling with the CP or its users.

Interplay between multipath and caching: after previous conclusion, in Chapter 3 we take a step back and consider a video delivery system at large. We consider a multipath-enabled system where ISP and CP might potentially have different preference on multipath. We attempt to evaluate the benefit (or lack thereof) of using multipath data transport for such a system. We start by modelling the optimization problems of both actors, ISPs and CDN, and designing a simulation testbed using a mobile operator topology. After that, in order to determine the potential benefit of ISP – CP collaboration, we define a theoretical boundary for their joint performance as to find whether there is a room for improvement (in terms of actors' objectives) – which happens to be quite significant.

Collaboration strategies: in previous chapter we demonstrated the interest of collaboration between ISP and CDN as to better take advantage of both caching and multipath. In Chapter 4, we develop the idea of collaboration by introducing two joint distributed models with different volume of information exchange. After that, we proceed with a numerical evaluation of devised strategies. Finally, we discuss the implementability of our two collaboration strategies considering the real-world political and technological constraints.

Efforts in NS-3: in Appendix A, we briefly present the motivations and proceed to details about the resulting distribution and its limitations.

1.1 Evolution of Internet video streaming

1.1.1 Early days of video streaming

Proliferation of the internet in the beginning of 1990s has motivated academia and industry to think about new applications for it. I would like to quote Little and Venkatesh [93]: "If one believes the claims, we will soon be able to view movies, play video games, browse libraries, order pizza, and participate in office meetings (in our bare feet!) from within the confines of our homes". I think this passage is a great artefact demonstrating the excitement for the new opportunities which I could not (sadly) experience at my age. The one which is related to this thesis is indeed the ability to "view movies", which back then was known as a VoD (Video-on-Demand) service. At that time, neither Internet nor video content providers were capable of supporting online video streaming due to a multitude of limitations; the issues were not even quite formulated, and of course far from being solved.

It's important to establish that in the beginning VoD was considered as an over-the-top service exclusive to content providers, akin to CATV channels. Today, in contrast, video streaming is just another way of surfing the internet, being a highly interactive entertainment that often has rather loose relation between its content provider and the streaming platform.

Perhaps one of the first complete surveys discussing them is the paper I have just quoted. For one, it lays down quite a detailed prospective on different actors of the online video streaming system. According to them, one of the obstacles on the way of implementing such a system are the content producers and their lack of understanding of the financial incentives for it. Visual entertainment industry then was based on television broadcasting with its very particular and defined end-user behaviour. Advertisers feared that introduction of the VoD would have broken the established dayparts, plus users could potentially have a possibility to skip advertisements. In addition to that, digitalizing the content and viewing it on a general-purpose computer device would inevitably give endless opportunities for copyright

infringement and piracy. Already these two problems were driving down the industry's interest in a new internet service. Nevertheless, as we can see today, it wasn't overly complicated to overcome those: advertising experienced a "second birth" with ad targeting systems and isolated advertisement frames that could not be skipped, and copyright security is arguably improved with rather controversial Digital Rights Management (DRM, [47, 155]) system.

The main technological limitation, according to this survey [93], was indeed the bandwidth capabilities of the Internet. Authors note that while ADSL (considered fast enough for VoD) client connectivity was already rather widely deployed back then, Internet Service Providers (ISP) and Content Providers (CP) were expected to have difficulties with managing a finite amount of their resources challenged by a big amount of VoD users. To overcome this issue, they discussed several techniques. First, user traffic characterisation could help the content being distributed closer to the end-users at certain time periods or at occasions of increased popularity; this idea is very notable as, in a nutshell, it describes something very close to a video CDN already in 1994! Another interesting proposition from them was bandwidth reservation, even as far as enforcing the end users to book their viewing time in advance. It can be argued whether this would be a good idea from user experience point of view, but we do know that the idea of resource reservation in internet did not happen to be very popular as of today.

Among other things, this survey gives a glimpse on a problem of relating the network QoS to the viewing quality in a VoD scenario. Authors conclude that, ideally, two quality metrics should be developed and monitored: one for the system (they call it "system QoS"), and another one for the viewer ("delivery quality"). One can immediately notice that what they are talking is the QoE, and it is quite impressive that researchers were already thinking about this, even without having a real commercial internet VoD system in front of their eyes (though some limited deployments already existed back then).

Even though technological and commercial limitations were yet to be defied, competition has inevitably forced video content providers to implement such a service with some degree of interactivity; those degrees are described by Gelman and others in [7]. According to them, a *Pay-Per View (PPV)* scheme is merely a broadcasting of a desired content at a particular timeslot, without giving the end-user any control over the playback. Some implementations of this sort worked rather well - e.g., a user could buy temporary access to a channel[2] broadcasting a very new film 24/7 in sequence, essentially offering a "home cinema"; such an approach, nevertheless, can be hardly justified as a real VoD system. An evolution of a PPV

[2]in an Internet video delivery system, the term "channel" is used in a meaning of an isolated flow incoming from the video server having a bitrate, that is calculated with respect to other flow requirements and link or server capacity.

is a *Quasi-VoD*, which represents a system broadcasting some content over several channels (physical or logical) with a certain time shift, such that a user could access demanded video with a reduced delay. Still, it doesn't provide any playback control but adds a burden of managing the interest thresholds when deciding the showing schedule. In spite of these, Quasi-VoD appeared as a sensible solution and happened to be interesting for the researchers (as we will see later). Finally, a *True VoD* system should represent a video streaming service as we experience it today, or, as authors say, "...equivalent, in terms of user control, to the VCR playback of rental programs".

Quasi-VoD systems has caught attention from research community in the end of 1990s. Multicast enabled video services has happened to be feasible for both CPs and their subscribers, but their drawback initially was in a quite obvious lack of flexibility in time for the user (thus making it not much different from a usual television. However, since the internet channels' bandwidths, available for the CPs, were already rather bigger than the required video bitrate, researchers started to think on how to use this property in an efficient way. Viswanathan and Imielinski in [160] propose a method to dramatically decrease the end-user waiting time while maintaining the broadcasting nature of the service; they call it Pyramidal Broadcasting. Their approach is to divide the content into several segments: the first segment of the video should be very short and thus repeated very often over the channel, but subsequent segments are longer with the last segment being a substantial part of the video. In addition to that, the system would leverage the fact that communication channel's bandwidth is probably much higher than the video's bitrate, so video segments could be transmitted faster than their consumption "rate". Finally, the end-user equipment is supposed to have memory in order to store the quickly transported video segments in order to later show them at normal rate. In this way the end-user could quickly access the beginning of the video, which was delivered to it faster than real-time and so would give the system some time to access next, longer video segments. Juhn and Tseng develop this idea with Harmonic Broadcasting [66], which reduces the amount of channel bandwidth needed to broadcast a video while preserving low start-up delay. The drawback of these two solutions is the requirement for a relatively big buffer space at the end-user device, which could be quite expensive at that time. Hua and Sheu in [58] propose a so-called Skyscraper Broadcasting technique that aims at reducing the lengths of segments and by so also reducing buffer requirements. Several other algorithms (some of them also have architecture-related names) has been developed at the same time; reader could refer to [69] for a more detailed survey on this topic.

Another interesting technique to improve efficiency of a multicast Quasi-VoD is called *patching* [57]. Hua et al. in their article propose to dynamically allocate streaming channels

as a function of users' demand. Basic approach assumes that every channel streams a video (or its part) in its entirety for a fixed amount of time. In contrary to this, patching will try to minimise the channel utilization time by using neighbour channels and client buffering. Imagine a scenario: a channel has recently started a transmission, but a considerable amount of users have requested the same content some time after that. Instead of waiting for the next scheduled channel, and/or using pyramidal techniques, the system will accept those clients and instruct their settopbox to buffer the transmission on that recent channel. Meanwhile, a new channel will be allocated which will broadcast the same video but only a part of it - the one which has already been streamed on the neighbour channel. Once this part has finished streaming to the new clients, the newly-allocated channel will be released, and their settopbox will continue showing the buffered frames from neighbour channel until the end. The advantage is that some channels in the system become busy only for a short amount of time, and that client admission becomes more flexible - compared to basic broadcasting. On the other hand, depending on the delay between the beginning of a scheduled transmission and the admission of new clients, the buffer requirements might be quite heavy; this issue will be discussed shortly.

While the measures I have just described did not yet qualify for a True VoD, they could definitely make Quasi-VoD a temporary measure before necessary technological and financial advancements happened. The next step in this direction was providing a user with control over the playback. While today a playback control functionality is an integral part of any video viewing experience, it was quite a novel service for the online video streaming at that time. Since digital video had only make its way in developed countries just on the brink of millennia, the reference way of controlling was the one of the video cassette recorders' (VCRs). It was providing a possibility to *pause* the playback, set it to *fast-forward* and to *rewind* the video. The *seek* functionality (which is essential nowadays) was not available on consumer VCR devices and thus was not considered at that time.

Such a functionality is usually associated with a True VoD because playback control should (quite obviously) only affect the person who is in fact controlling. Quasi-VoD with its broadcasting nature cannot allow the streaming to be altered by a single user without affecting everyone else, however, it has been proven to be possible to exploit the buffered data from scheduled broadcasting schemes (including pyramidal ones and related) to achieve limited playback control. Almerot and Ammar in [7] explain two ways of implementing VCR functionality in a multicast video broadcasting system with scheduled showing repetition (which is one of Quasi-VoD variants):

- First of them is called *continuous interactivity*. According to it, the user joins the closest by time broadcasting channel; whenever he or she invokes, for example, the

pause action, the playback is paused but the video streaming continues. The client's buffer is supposed to store the arriving video frames, such that when the client resumes they could be used to continue the playback without changing the broadcast channel. In case the resumed video is being shown at normal rate (in real time), the buffer will stay filled with the same amount of stored video until the end of the showing, as the broadcasting channel is now pause-duration seconds ahead of the client and this difference in playback positions will have to be buffered. This means that in order to keep the same broadcasting channel, the total pause duration during the video showing has to be lower than the buffer capacity at the client, otherwise a channel change will be required with possible service interruption. The *fast-forward* and *rewind* functions are naturally operating over the buffered parts of the video in a straightforward way, or otherwise require channel change. The notable drawback of this approach is, indeed, the buffer space requirements. Works of that time refer to video bitrate of 1.5 Mbps (MPEG-1 SIF/PAL resolution) that needs 112.5 MByte of disk space to buffer 10 minutes. On the other hand, changing channels may result in service refusal, and continuous interactivity decreases the need in changing.

- The second approach is *discontinuous interactivity*. It exploits the fact of scheduled repetition of the video broadcasts over multiple channels: every invocation of any control operation requires channel change, such that no buffering is required but the *pause* has a fixed duration of a factor of the time slot length (which is the difference in time between two consecutive broadcasts). The range of *fast-forward* and *rewind* will also be bound to the time slot duration and will happen immediately. While such a scheme needs no buffer, it could be arguably less convenient for the subscribers, especially if the time slot length is big; additionally, service blocking can be an issue when changing channels.

A slightly different way of implementing video broadcasting and VCR functions was demonstrated by Poon and Lo in [127]. Instead of allocating the entire CP's available bandwidth to multicast channels, they consider a case when some of the bandwidth is left for intermittent unicast streaming sessions. Clients start by joining a multicast channel, but are moved to individual multicast channels once they invoke VCR functions. Authors propose to keep unicast channel bandwidth much higher than the required video bitrate, such that the client could quickly replenish his buffer with video frames up until the playback moment of some close multicast channel, then change to it and release the unicast channel. The main focus of this article was on reducing the service blocking, so together with certain server-side logic their solution is in fact quite effective in regards to that focus.

1.1.2 Video streaming becomes viable

The beginning of 2000s gave rise to many internet technologies; some helped to increase network resources, others proposed new ways of being connected to the internet. This inevitably led to an explosion in the number of subscribers, together with many internet applications (like video streaming) becoming more accessible to them. Among those events, the key once in proliferation of research on video streaming are: emergence of peer-to-peer networks, advances in wireless networks, increase in interest for video caching and development of scalable video codecs.

Peer-to-peer video streaming

Research on ad-hoc and peer-to-peer networks attracted great attention in the beginning of 2000s. It has been quickly understood that peer-to-peer technology has a big potential in the domain of video streaming: instead of streaming the content from a far-away server or an in-network cache, subscribers could pull their content from other subscribers in their area. With popular enough service this could help to move the video transmission load to the pre-last-mile network section (which often has more available resources than the usually aggregated backhaul sections). As Liu et al. report in [96], two types of peer-to-peer topologies can be pinpointed: *tree-based* and *mesh-based*.

The first type represents a tree topology overlay constructed by peers; such a clear hierarchy is supposed to keep the maintenance burden low while providing the interesting concept of peering. On of the disadvantages of a tree-based peer-to-peer network is the user churn: in our case not only the actual user leaves the system, but also all the peers below him on the tree would become disconnected from the hierarchy. This disadvantage is quite important for a video streaming system, which requires timely content reception and, thus, benefits from uninterrupted connectivity. This issue has been addressed in several research works that, together with discussing other necessary arrangements, offer their vision on a tree-based VoD system. Jin and Bestavros in [64] propose to cache a sliding window worth of video frames at the clients, such that whenever a client comes later within the duration of this sliding window, the cached content could be relayed and forwarded from the first client to the new one. Since the cache content "slides" forward together with the playback of the clients, the system requires only one stream channel to serve all the clients that have arrived within a sliding window duration between each other. A somewhat different approach is presented by Guo et al. in [48] intend to extend the already-discussed patching technique to the peer-to-peer delivery system. The core of their work is in designing an algorithm to construct a peer-to-peer multicast tree that fits best for employing patched video transmission:

a tree should span across clients of roughly the same arrival time such that they can be served with one multicast stream while using patches to mitigate the difference in their join time. In addition, they define measures to protect streaming continuity from network disruptions (mainly by re-contacting the video server as to recover stream flows)

In contrary to the tree-based approach, the mesh-based peer-to-peer network is not supposed to have a vertical hierarchy. Vlavianos et al. in [161] demonstrate their take on a mesh-based VoD: their idea is to use BitTorrent for this purpose. The video file is broken into small pieces that are retrieved and cached by the peers, such that they could be shared with newer peers. They notice that the fact of BitTorrent protocol prioritizing retrieval of rare file pieces is not suitable for VoD where streaming clients/peers prefer pieces to arrive more or less in the order of playback. To alleviate this issue, they develop a piece selection policy to better respond the VoD requirements in a face of quickly changing peer availability in the network. A more recent mesh-based VoD solution is presented by Magharei and Rejaie in [99]. They identify that a peer in a mesh might suffer from bandwidth bottleneck (the aggregate bandwidth with other peers is not enough to fully exploit its own connectivity) and from content bottleneck (when there is not enough unique content pieces around to fully exploit the connectivity with other peers). Their work, consequently, proposes strategies for network overlay construction and piece retrieval as to more efficiently use network resources and avoid the aforementioned bottlenecks. At the same time, they uncover many other insights about the operation of such a system.

While large content providers (like Youtube) have skipped peer-to-peer networking in their content delivery strategies (judging from the lack of reports one that matter), such an approach became rather popular in live video streaming services that has been successfully introduced to the market between 2005 and 2010 (like [149, 167]). Notwithstanding such a (limited) popularity, later years have seen a business model shift for online video distribution that happened to be incompatible with peer-to-peer approach (due to associated copyright protection considerations), which lead to its decay after 2010s – just to be reborn within context of hybrid Content Delivery Networks [172, 176, 11, 173], which benefit from peer-to-peer connectivity to cache and deliver video content.

Video streaming over wireless networks

The timeframe under discussion was a period of global adoption of high-speed packet-switched wireless networks, such as IMT2000 and IEE 802.11. Theoretical data bitrates provided with these technologies was quite enough to ensure an acceptably good video streaming quality (i.e., at least an uninterrupted service at a resolution adapted to the device screen). On the other hand, radio medium quality changes very fast, so transmission bitrate

and delays during a wireless transmission vary significantly as well. Initial studies have found that then-current generation wireless network do not provide enough inter-layer cooperation [144] as to respond well to the prerequisites of video streaming (e.g., tight arrival deadline requirements). This was, perhaps, the main research direction on the intersection of wireless networks and video streaming.

Advances in channel-adaptive video streaming... gives important video parameters.

Khan et al in [70] propose their variant of a cross-layer optimisation framework for video streaming in wireless networks. They use a centralised mechanism that has access to all network layers in order to observe their state and impose operational parameters. Since the parameter space to optimise on is quite large, they count on reducing them by abstraction. Once the optimisation is done for the abstracted parameters, they are distributed to their respective layers which then translate them back into operational parameters. Some of the described layers and their operational parameters are: application layer (video bitrate, encoding), data-link layer (TDMA slots/CDMA carriers, directional beams), physical layer (channel coding, modulation, power). The objective function for the optimisation is chosen to be the average of all users' (within optimisation realm) Peak Signal-to-Noise Ration, PSNR [169]. Authors discover that the cost of performing such an optimisation increases quickly with growing number of radio clients and available parameter value ranges; on the other hand, they conclude that their approach performs well even after limited degrees of freedom for the parameters.

Slightly more practical approaches were proposed to improve video streaming over wireless, such as modifications of ARQ, application layer scheduling [144] and transport layer congestion control [29].

Overall, considerations of cross-layer protocol collaboration appear to be an important subject of research on ameliorating video streaming over wireless networks - even for subsequent wireless technology generations.

Video caching

The concept of caching has been used for improving QoS in the internet since the early days of Web [97, 163]. The idea is that a storage server is placed close to the end-user such that whenever he requests something from internet, the objects transmitted to him are stored at that server. Stored objects then could serve later users requesting the same content, improving latency and often providing higher bandwidth [163]. These two benefits come from offloading the demand from main servers, and because caches can be deployed in a highly-provisioned parts of the network thus avoiding architectural bottlenecks.

Just as with peer-to-peer networks, potential of caching technologies in video streaming was quickly uncovered by industry and research. Normally, video content is rarely updated so it can be considered as fixed content and thus suits well for caching. Nevertheless, several particularities has to be taken into account:

- Disk space requirements: even at relatively low resolutions, full video files of a film-long duration need a considerable amount of storage space (as was demonstrated in previous section). Taking into account the storage costs in the beginning of 2000s together with the amount of popular video content at a single timeframe, a large-scale video caching solution could become too expensive.
- Bandwidth and I/O requirements: video streaming is a demanding application not only in terms of network bandwidth, but also with respect to the storage bandwidth. Designing a caching solution has no bottleneck in these aspects is a complex task, so developing a cache policy that respects such limitations might be a good investment.
- Management overhead of caching parts of a video: it is often needed to cache videos not in a single piece but in parts or even incomplete (e.g., due to some parts of a video being not popular enough to justify caching). At that time video content was stored at the source as a single video file, and was requested by the client in byte ranges - so it was quite difficult to align cached ranges, original source ranges and the ranges in the client's buffer.

Research community gladly accepted these challenges, resulting in a number of original architectural solutions. A most straightforward way of battling the bandwidth and disk limitations is to reduce the volume of video content cached. Liu and Xu [95] provide a good summary of common solutions for that. They distinguish between homogeneous and heterogeneous client demands, which defines a caching solution.

The first type, homogeneous, defines that video clients have similar hardware configuration and network connectivity, so that they would request the same version/resolution of the video content. Having homogeneous client demand simplifies cache management, since often only a single version of a video content has to be be stored and used later for subsequent requests. This assumption was certainly applicable in the first decade of new millenium, when displays and digital video existed in rather limited range of resolutions. The second type, heterogeneous, means that clients have a wide range of devices with different characteristics, which also makes their requested content different in its version/resolution. This complicates the caching process; we touch upon on of the related issues in Chapter 2.

Several caching solutions for homogeneous clients have been brought to the light at that time, as outlined in [95]. One of them, called *Sliding-Interval Caching* [30], suggests that a cache should not store the entire video transmission passing through it, but only a short window of it – thus, in essence, a storage-limited LRU per each video instance. With the course of the playback, new video frames will arrive to the cache's "stack" which will cause the eviction of the oldest frames in the window. As a result, in case another client will request the same content shortly after (in the same time window), cache could serve him from its storage. Past the specified time window, of course, client will have to request the content from the server. Such a solution is demonstrated by the authors to greatly reduce disk storage requirements with some degree of cache hit ration decrease (for the clients who arrive too late), but can potentially work very well when client demand times are well-defined. Another caching solution described is called *Prefix caching* [142]. It proposes to cache an initial video portion, called *prefix*; whenever a client starts accessing the prefix of a video, cache would start prefetching the remaining part of the video (a *suffix* in order to serve it to the client when he finishes watching the prefix part. Such a scheme can smooth out transmission bitrate peaks, while maintaining high video quality for the client. Finally, in order to achieve higher byte-hit ratio (that can improve disk space usage efficiency), *Segment caching* has been proposed [171]. This solution entails dividing video files into segments of a certain length and "utility", that can help manage their caching.

One of the most important mechanics in web-caching is *caching policies*, which define how to maintain constrained cache storage as to maximise cache hit-rate. In a realistic environment, where the amount of unique requested cacheable objects is much greater than cache storage capabilities, proper selection of caching policy can be of a crucial importance for cache operation. The most basic strategy is called Least-Recently-Used, which implies that when cache storage is full, the oldest object on the disk will be evicted upon an event of a new incoming object. There exist plenty of other, presumably more smart policies; interested reader can address, for example, the work of Ali et al. [6] for more insight.

One could already notice that solutions like prefix or segment caching assume some degree of interaction between client, cache and server. Also one should remember that one of the video caching challenges was a burden of managing the cached video parts, which might also be alleviated using interaction between entities. At that time, however, there was no standard framework for such an interoperation, so one possibility to implement it was using control protocols such as RTCP and RTSP (for QoS guarantee and content streaming control). We can already imagine the complexity of the entire caching architecture that was under consideration of researchers at that time; such a level of interoperation suggested that caching infrastructure could be consolidated into bigger entity and benefit from extended

geographical visibility – which sounds exactly like a video Content Delivery Network (CDN). Being an enormous network of caches tailored to deliver content from a certain provider (like Youtube or Vimeo), distributed all over the world, they make use of aforementioned (and not only) research to greatly improve users' QoS. CDNs and their advanced caching techniques, without any doubts, are the main contributors to today's unprecedented performance of the video streaming.

Early stages in adapting video quality

Even though advancements in caching technologies have certainly contributed to growing of the video streaming, beginning of 2000s have seen another research direction born that happened to be just as important: quality adaptation. Until now, we have been mostly assuming that clients cannot request different video qualities once they requested the content. That is a considerable omission, since the real-life internet connectivity can suffer huge variations, calling for an ability to adapt the requested quality accordingly in order to be able to efficiently use client's achievable bandwidth.

As a first attempt, researchers studied the possibility to make use of layered video encoding schemes. In this way, the video is being encoded into several layers – a base layer and a set of enhancement ones. The base layer provides basic information for the video series to be decoded at a base quality, while supplementing enhancement layers at the moment of decoding will provide additional information to improve the resulting quality of a reconstructed video. While the studies on layered video encoding started even around the beginning of 1990s [153], the approach have only received its standard in 2007 [141]. Theoretical background of layered encoding is out of scope of this document; we are, however, interested in how was it used for video quality adaptation during streaming. A particular attention has been devoted to this subject in the beginning of 2000s, resulting in a number of research pieces. Rejaie et al. in [134] discuss a mechanism that helps to absorb bandwidth fluctuations caused by transport congestion control. According to them, this approach can trade short-term video quality improvement for long-term quality stability and general better viewing experience. A more general look on the variation of available bandwidth is taken by Kim and Ammar [77]. They propose an optimal video quality adaptation algorithm that is focused on minimizing quality fluctuations while maintaining high bandwidth utility. Finally, a rather modern approach is presented by Wang et al. [165]: they develop a machine learning framework for performing quality adaptation for layered codecs. In contrast to other works, they are dealing with all compression domains in a systematic manner; in addition, this paper is one of the rare ones that performs subjective studies to validate their approach.

Continuing research on layered video encoding and its application to internet video streaming has demonstrated the viability and benefits of adapting video quality according to available bandwidth. Together with this, the observations conducted during this time would pave the way to the development of modern streaming systems - that would bring substantial improvement in user perceived quality, would focus on QoE, rather than on QoS, and be built to be adaptive. Quality adaptation is discussed in more detail in Section 1.1.3.

1.1.3 Modern day video streaming

Until the middle of 2000s Video on Demand over Internet was a rather localised business case. Such providers existed mostly as subsidiaries of bigger cable companies, and rarely scaled even to country-wide deployments. In addition to that, the audience of such services was quite low due to slow development of broadband Internet connectivity and old habits of preferring physical copies of the content. These factors were making the problem of VoD scaling doable, yet challenging.

Nevertheless, penetration of fast Internet connectivity made VoD much more accessible, and content providers' business models started to change as to accommodate the new potential market opportunities, which is turn started to change the peoples' habits. The launch of Vimeo in 2004, Youtube in 2005, Netflix (online streaming branch of it) in 2007, and a myriad of other local and global video streaming services of a VoD type overall was an event of such an immense magnitude that it would shape people culture very shortly. Immediate availability of the content, added social aspect (in case of Youtube/Vimeo/etc), endless interactivity - these are only few reasons why modern video streaming effectively replaced the traditional TV for quite a large slice of the society; in addition to this, global character of the Internet called for consolidation of the market in hands of several large global content providers.

Rapid increase of user base required an equivalent increase in network capabilities as to fully take advantage of the new market. At the same, research and development actions in higher definition video also started to affect network providers. Such an abrupt increase in network traffic did not go well along with ISPs' financially-justified plans on hardware upgrades, so research essentially had to continue the work on improving the qualitative parameters of the video streaming in a constrained network environment - just like it was before.

This section will, therefore, present the state of three important research directions that, in our opinion, shaped the modern Internet video streaming and that helped the Internet to rather successfully deliver the required capabilities for it - thus being a part of its success nowadays. Those research directions are *Content Delivery Networks*, *Adaptive video streaming*, and focus on *Quality of Experience*.

Quality of Experience

Traditionally, the performance of computer networks has been assessed using the Quality of Service metrics, such as transmission bitrate, delays, jitter and packet loss. It is quite evident that those metrics represent the actual network-level behaviour, which in many cases translate well into the application layer performance. Some examples are: higher transmission bitrates result in better performance of file sharing applications; lower delays straightforwardly benefit business applications (like financial and stock handling); high jitter can be harmful for real-time applications. On the other hand, several peculiarities arise when using QoS as a measure of user's satisfaction:

- Users are rarely interested by the QoS metrics themselves. Again, if drawing an example from video streaming, the viewer would instead pay attention to the video quality or the start-up delay. Such parameters might depend on several network QoS metrics at the same time, plus at other parameters such as server capacity and availability.

- Assessing the actual subjective effect as a dependency from the QoS metrics is difficult and has been slightly neglected until recently. An example in the domain of video streaming would be: indeed we know that high transmission bitrate will allow the user to watch the content at high quality, but is it sure that lower video quality would significantly decrease the subjective satisfaction of the user? In case it would not, we could think of deliberate shaping of the user's bandwidth as to optimise the performance of other adjacent clients and applications in the network, without risking to loose the first client.

As a result, the relation between network QoS metrics and actual application performance often becomes rather indirect. Instead, it has been proposed to measure the Quality of Experience (QoE) directly at the application in order to guarantee its performance. It is not very clear, however, how to quantify the QoE; besides, by doing using it one loses the ability to control the network with an objective of direct maximisation of the application's performance the QoS and QoE metrics are not directly related (at least in the video streaming domain, which is the focus of this thesis).

All these considerations bring up three research problems that has to be solved as to make the concept of QoE mature: (i) QoE assessment, which tries to establish a common and underpinned quantification scale for QoE, (ii) QoE modeling, that is aimed at developing relationship models between QoE and objective metrics (preferably network QoS), and (iii) QoE management that uses QoE models to perform (often cross-layer) network management as to improve application's QoE.

QoE assessment

Being a subjective metric by its nature, QoE is difficult to assess, yet even harder to make use of that assessment - due to management processes usually rely on objective metrics. A significant piece of research exists for this purpose; Shatz et al. [138] distinguish between **subjective** (as to quantify the proper subjective *experience*) and **objective** (as to foster integration of QoE into current, QoS-based systems) QoE assessment.

A subjective QoE assessment is naturally conducted by the application audience - e.g., video viewers in our case of video delivery. Subjects, who participate in the assessment, are presented with a set of tests that involve the assessed application under different test conditions. Those conditions affect the application's performance, which is then evaluated by the test participants and expressed in so-called *Mean Opinion Scores*, or MOS. MOS is an arithmetic mean of scores, subjectively given by a participant over a simple pre-defined scale for the application's performance; this concept is defined in ITU-T P.800.1. ITU has also established standards (ITU-R BT.500 and ITU-T P.910) for assessment modalities, which include considerations for a test set, room set-up and others. One could object that such a method leaves too many uncontrollable variables (such as participants' bias, psychological and contextual factors), and this has indeed been a debate throughout the years of research. As [138] points out, certain studies attempted to overcome this limitation by passing to somewhat more objective metrics such as quality and speed of application-related goal completion [78, 24], user engagement [40] or biological factors [168, 101]. Additionally, conducting such kind of assessment is cumbersome and time-consuming, hence objective methods have gained their share of research attention.

Objective QoE assessment, as perhaps already understood, is basing on objective metrics that are known to have a certain relation with the application's performance (which can them be called operational metrics, or application-level metrics). As synthesised by [138], the first step in establishing objective assessment is deriving a model that would map the subjective metrics (like MOS) to objective ones (like video bitrate, frame noise-to-signal ratio, or duration of playback stalls in the case of video streaming). Despite the difficulties in establishing such relations and models that would map both metric types in a principled manner, it is the objective metrics that are widespread in evaluating QoE-related solutions in research – as we will see later in the thesis.

The next step on the way to a successful QoE management system is developing QoE models that map network QoS metrics to the QoE ones, which is discussed next.

QoE modeling

Models for QoE can be classified in many different ways. According to [14, 147], some of the categories include: type of application/service considered (e.g., calls, video, file transfer);

reference information requirements (whether the model needs to compare the content/data before transmission and after); black-box or white-box approach; the degree of incorporation of psychological factors into the QoE assessment. In general, the amount of categories is quite large such that most of the proposed models up to date are unique in some way or another, offering both advantages and disadvantages over others. Some models specific to multimedia or video streaming applications are discussed here.

QoE is not replacing QoS, but rather acts in complement, because general network configuration is done with respect to the former. Due to this, it is interesting to know how do individual QoS metrics affect application's QoE. It has been noticed that certain applications can have *logarithmic*, *exponential*, or *power* relationship between the QoS and MOS [132, 53, 79]. It can be noted how these relationships demonstrate that the impact of QoS degrade is affecting user experience the most when the achieved application quality is high.

Hossfeld et al. in their work [55] have made a step further by attempting to tackle the association of multiple QoS metrics to MOS. They develop additive and multiplicative models and present a example of applying them to the case of video streaming. They conclude, however, that the complexity of QoE assessment and modeling require fine tailoring of each model to each application (perhaps even to its configuration).

It was clear, however, that QoE is a complex multidimensional concept that requires a proper in-depth study. Kilkki in work [74] is among the first to attempt to unwind this multidimensionality. No models or specific QoE or QoS parameters are studied in this work. Instead, author discusses the need to develop a framework for relation of technical parameters to the user experience. As he explains that the magnitude of the fact that the QoE is highly subjective, he concludes about the need of an interdisciplinary research group that would need to cooperate for the development of a proper QoE modeling study. He notices that different research domains use different terminology and understanding of the problem, so a part of the work is devoted to harmonisation of the terminology of QoE versus QoS.

Perkis et al. [126] is another early work on QoE modeling. It does not result in an exact model either, but rather discuss the meaning of QoE and report about an experiment to measure it - by collecting and interpreting the observed experience. They argue that the problem of developing a QoE model is highly complex and multidimensional; it appears that they consider a black-box approach as a suitable one for solving the problem. To overcome this limitation, they designed an experiment for a mobile VoD use-case with 10 volunteers that would complete questionnaires about their experience of using the service. Relying on the collected data, they have uncovered and explained several fundamental (mis-)conceptions about mobile video streaming.

Vold et al. [162], in contrary to the previously mentioned works, propose an actual framework to construct an algorithm for QoE estimation using objective data as the input (thus attempting to apply a white-box approach). They define QoE as a function of human perception components, QoS, and the quality context; the last is a very interesting concept, as it represents a set of objective QoE parameters and mappings between them. They argue that careful selection of those is of crucial importance, and cite several examples on how to approach this task. Authors establish a vertical hierarchy for constructing such an objective QoE model, which is composed of *end-user layer, service layer*, and *transport layer*. They define the required parameters at each layer and design relationships between them, hereby fulfilling their objective. While the evaluation keeps undisclose the exact models of their own, they do provide a glimpse on the resultant QoE predictions for different kind of services in an Next Generation Network environment.

De Moor et al. in their work [38] make a similar contribution with an emphasis on testbed-oriented Living Lab environment. They offer a distributed framework for multidimensional assessment of QoE that comprises user, system and context aspects. They also provide details regarding practical implementation of such a framework for common vendors of mobile operating systems.

One can note that proposed QoE does not result in some sort of a ready-to-be-used analytical expression, but instead in frameworks that try to cope with overwhelming multi-dimensionality of QoE modeling. Recently, a different approach has been undertaken for this task: instead of constructing fundamental relationship between countless parameters and dimensions, it was proposed to hide all the complexity in a black box by, for instance, applying machine learning techniques - which are admittedly well-fitted to such kind of problems. An interested reader can refer to [9, 118, 26, 18] for more details on that subject.

QoE management

The term “management” is quite broad, so with respect to QoE it applies to diverse applications and environments. In line with the main focus of this thesis, which is video streaming, we can isolate several main domains of QoE management research for this application:

- Management in wireless networks: literature on this topic mainly emphasises video-awareness on such issues as radio resource access and scheduling, power control and, of course, backhaul routing;
- Exploitation of centralised network control.

Research on QoE-based network management in wireless networks is extremely active, yielding not only in abundant literature but also in several surveys since its emergence. Ernst

et al. [41] review the articles related to Heterogeneous Wireless Networks (HWN). Qadir et al. [128] review papers related to QoE optimisation mechanisms in general, though it happens that the majority of the considered literature focuses in wireless networks of different kinds. Sousa et al. [150] survey existing literature specifically on QoE-aware wireless resource scheduling. One can address these surveys to gain a broad understanding of existing solutions.

Centralised QoE management has gained a particular momentum due to its very design: a centralised controller can easily control actors of a complex system, and this property can be used to improve their operational efficiency. For instance, Bouten et al. in [23] develop and evaluate a system that orchestrates HTTP Adaptive Streaming clients' decisions as to improve their common optimality. For this they use transparent proxies that limit available quality representations for clients; they do not, however, give any detailed insights on how can such a system be implemented in reality, but they do discuss certain somewhat hypothetical mechanisms for that. Nevertheless, they report that centralised optimisation for HAS can considerably improve video streaming performance in studied objective QoE metrics, such as average video bitrate and quality instability.

Software-Defined Networking [103, 80] has given an opportunity to implement such centralised management designs, owing to an idea of heaving a central controller managing the routing in an SDN network, as well as openness of the OpenFlow protocol [104] to controller implementations.

Nam et al. [113] propose an a solution for SDN that monitors end-to-end performance of video streaming flows in order to optimise their routing. They monitor operational QoE metrics of video applications such as playback buffer occupancy and playback state, and for the routing part they employ MPLS. They report that, due to such a centralised monitoring and routing control, they area able to achieve significant QoE improvement at users' video streams.

Liotou et al. [92] provide a design and detailed analysis of a QoE-focused cooperation between a mobile operator and a video content provider. According to their idea, mobile operator can report to the content provider with capabilities of its infrastructure, which helps the former to proactively assist his clients in video quality selection.

Bentaleb et al. [19] develop an architecture that is using capabilities of an SDN-enabled network in order to control the resource allocation for HAS clients. A particularity of their approach is that the QoE-management application is decoupled from the actual SDN controller, as is offloads the computation burden from the controller itself and, therefore, achieves good scalability in respect to the number of video clients. They report significant improvements in the amount of clients receiving the same QoE provision as in conventional schemes.

Works above declare convincing improvements in various QoE metrics for video applications, achieved by exploiting SDN. Zinner et al. [178], however, suggest the research community that using SDN and OpenFlow for QoE management is not exactly as easy as plug-and-play. They explore the different techniques that are provided by the SDN paradigm in terms of traffic management, and test whether some of those techniques impact the actual operation of video streaming applications. Basing on their experiments, they conclude that demand-based resource allocation, with all its value, can actually affect the TCP performance for short amount of time, thus potentially harming application QoE. Their advice is therefore to evaluate this impact against the expected benefit from using dynamic resource allocation.

Overall, we can observe QoE to be a de-facto standard for performance assessment in applications that are used by humans. With such a perspective in mind, we can expect an increase in human-focused networking systems designs, which will benefit both users and network infrastructures.

Content Delivery Networks

Concept of a Content Delivery Network can be though of as an evolution of caching in the direction of globalisation. In that way, a CDN is a highly distributed network of cache servers (which are called *surrogate servers* in this context), which are strategically placed across multiple ISP networks in multiple regions and countries. Figure 1.1 visualises the CDN as explained. Such a pervasiveness allows CDNs to facilitate the delivery of their clients' content to a very large audience mostly regardless of their geographical position. Depending on the scale of a CDN provider, they can place caches in a large amount of ISPs within a geographical region to improve their reach (albeit on the expense of costs and complexity), or attempt to achieve a reasonable content acceleration (meaning QoW or QoE performance) by only installing their caches in one ISP per geographical area [120]. Examples of the first CDN type could be Akamai (its coverage in over 130 countries with presence in about 1700 networks worldwide[3] makes it practically omnipresent), as well as Google with, most prominently, its Youtube delivery infrastructure. An example of the second type CDN could, perhaps, be Amazon CloudFront, which is claimed to have 136 points-of-presence worldwide.

By most part, CDNs are independent from content and service providers due to accompanying costs and necessity to manage international business. Akamai, Cloudfare, Rackspace are some of examples of such CDNs. On the other hand, there exist content providers that are big enough to need and afford their own CDN services, or at least in part. Google is the first

[3] https://www.akamai.com/uk/en/about/facts-figures.jsp, accessed 10/2018

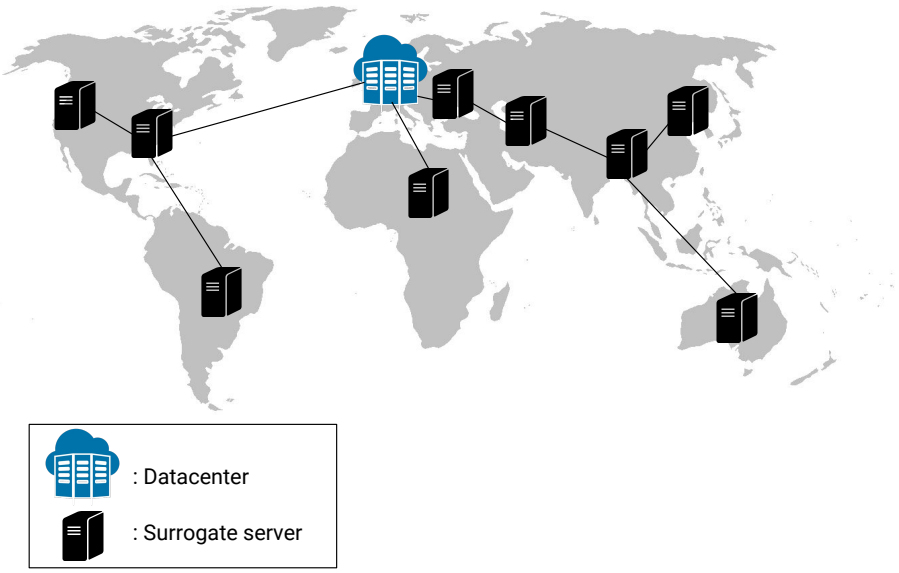


Fig. 1.1 Visual presentation of a CDN

example coming to mind, while Netflix [22] and Facebook [65] have also been attempting to enter this technological space since several years ago.

A problem of request routing is critical in CDNs: the objective is to determine a proper cache to serve the user, so this process defines his acceleration performance. Wang et al. [166] isolate two tasks as parts of request routing: server selection and server redirecting. Server selection mechanism is doing the actual work on finding an appropriate cache. A popular approach to this is to select the closest cache to the user [4], but other considerations can be applied as well, for instance, basic QoS metrics [117], or infrastructural considerations [159]. Once a cache is selected, server redirecting mechanism enforces the selection for the requesting user. A typical way of doing this in today's CDN is by using Domain Name System (DNS) [166]. Such a solution is rather easy and efficient, since users start with resolving the content server's name anyways, and the DNS response can report him back the selected cache. Nevertheless, other approaches exist; an interested reader can consult [124, 32] on this topic. A benefit of having an infrastructure and method for request routing also gives CDNs an opportunity to provide other related services, such as DDoS-protection and load balancing [120].

Another important mechanism in CDNs is content outsourcing, which is dealing with questions of provisioning of content replicas to the surrogate caches. Even when assuming decreases in storage cost, it remains quite clear that it is not possible to replicate all original

content in its entirety at caches due to cost and rackspace constraints. Efficient methods on placing the replicas are, therefore, vital for CDN operation. Pathan and Buyya in [124] talk about three types of content outsourcing: *cooperative push-based*, *cooperative pull-based*, and *non-cooperative pull-based*. According to them, within cooperative push-based method, CDN proactively pushes the content from original servers to surrogates. Being cooperative, it implies that CDN can use a centralised entity to optimise this procedure and prefetch the replicas on surrogates. This is done with a purpose of maximising hit-rates, storage efficiency or other metrics, which has a potential to bring elevated operational efficiency and user QoS/QoE. In its turn, cooperative push-based approach entails similar ideas, though instead of a central entity, logic and cooperation are supposed to be done in a distributed manner by the surrogate caches themselves, which is then reactively pulled from the origin. These two methods sound very promising, but authors report that at their time no commercial implementation has been reported for them. The last one, non-collaborative pull-based, is the easiest to implement as it only involves reactive replica requests upon the event of a cache-miss.

There is plenty of more specific mechanisms defining the operation of CDNs. Since this thesis does not focus on the former, we invite an interested ready to consult the very survey of Pathan and Buyya [124] for the basis of CDN constitution.

Today's success of CDNs (Akamai reports that over 50% percent of Fortune500 companies use their services) validates the timeliness and effectiveness of such a concept. Nevertheless, development in the domain of CDNs never stops. One of the promising research directions is telco CDN development and efforts on their interconnection (mutual and with independent CDNs). This direction is interesting in the context of this thesis, more precisely, Chapter 4 where we explore the cooperation of ISPs and content providers. Spagna et al. [151] offer design considerations for building ISP-owned CDNs with a slight focus on mobile networks. They discuss such questions as cache placement, request routing and content outsourcing. Balachandran et al. [12] present, among other, their analysis of potential benefits from telco CDN federation that augments their inherently limited reach and enables them to compete against independent CDNs. Bang et al. [13] present a protocol for CDN interconnection that is complying with IETF CDNI working group vision of this problem [116]. In addition to this, they perform its field tests involving three major South Korea's ISPs.

HTTP adaptive Streaming

As briefly noted in Section 1.1.2, it is important for a video transmission system to be able to adapt its transmitted quality with respect to changing network conditions, as best-effort networks are inherently not stable in terms of their QoS parameters. We can, perhaps,

compare adaptive and non-adaptive video streaming against their application-level QoE focus. Streaming the content in a single quantity should normally bring stability in video quality for the entire showing duration; in reality, changing connectivity bandwidth can potentially go below the requirements of the video stream for quite extended periods, which may cause playback buffer depletion and, as a consequence, playback stalls. One solution to avoid this would be to stream the video in a quality low enough to not suffer from connectivity bandwidth fluctuations. The by-design QoE focus of a non-adaptive streaming then seems to be concentrated at reducing video instability, on the expense of video quality or playback stalls (or both together).

An adaptive streaming system, on the contrary, exploits changing network conditions to switch qualities on-the-fly. This feature gives the possibility stream the video in a high quality for the periods of high connectivity bandwidth, but fall back to lower quality upon bandwidth drops in order to avoid video stalls. The focus here seems to be on video quality and avoiding stalls, but on the expense of instability.

Considering the subjectiveness of QoE, it can be argued whether it is more important to prevent video instability or permanent quality degradation, but works [52, 40, 109, 46] conclude that playback stalls and their characteristics influence user engagement the most, so we agree on avoiding them at first place. Regardless of the preference, today we are the witnesses of the advent of adaptive streaming systems due to their ability to efficiently utilise all client's available bandwidth while effectively preventing playback stalls.

Today, adaptive video streaming is mainly implemented in a form of HTTP Adaptive Streaming (HAS). The idea is to store video content on the server in several qualities, and then split each quality representation further into multiple segments (also called chunks) in time domain (with segment duration of two seconds or more). Each resulting segment is stored as a separate file, which can be requested individually by the client (or rather his player). Transporting these segments is designated to be done with HTTP, which brings compatibility with Internet infrastructure – most importantly, facilitated caching (which we will consider in Chapter 2). Client's player runs an algorithm that monitors network conditions and selects an appropriate quality to request next, build up a playback buffer that is used to show the video. HAS is a client-driven solution, meaning that content request logic is incorporated into the client's player, making the system scalable. Picture 1.2 gives a visual presentation of HAS's operation.

Since HTTP is used to fetch the segments, client has to know their individual addresses. For that, a manifest file containing this information (and some other metadata) is requested by HAS client in the very beginning of the streaming session.

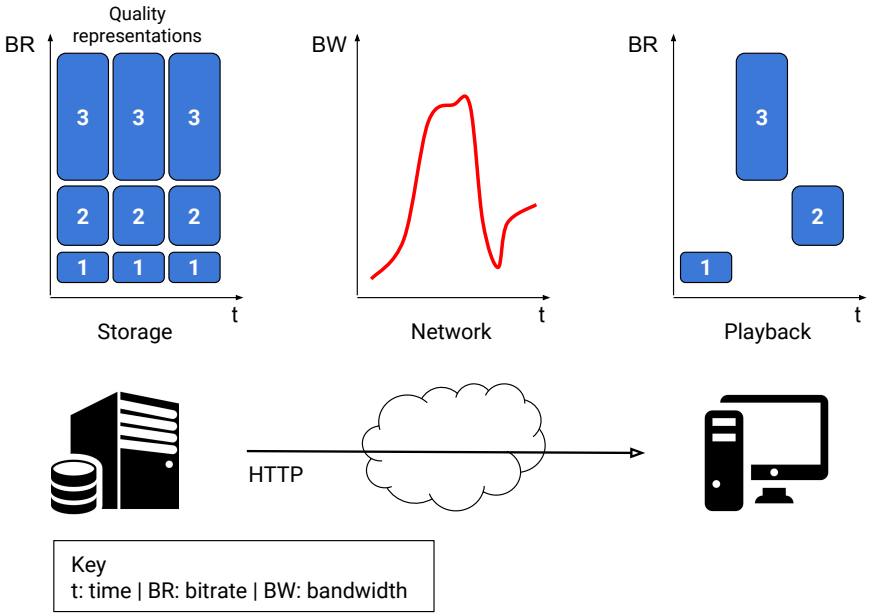


Fig. 1.2 Video delivery in HTTP Adaptive Streaming

There exist four major HAS implementations: Adobe HDS, Apple HLS, Microsoft SS, and ISO/IEC MPEG-DASH. All four of them support the mentioned above basic concept, and differ in exact definitions of manifest file format and in support of extra features like ad insertion[4]. MPEG-DASH, being the only international standard among the four, gains considerably more traction and support.

The central component of a HAS system is its quality adaptation algorithm, as it is defining its performance first and foremost. As quality adaptation is usually operating with respect to changing connectivity bandwidth, the most basic algorithm is assessing the network conditions in exactly that aspect. Such an algorithm is commonly called *Rate-Based* quality adaptation (RBA). It can monitor connectivity bandwidth by calculating last segment's downlink bitrate and then incorporate it into an Exponential Weighted Moving Average (EWMA) in order to define which video quality to request next (according its video bitrate). Being very simple in implementation, it has been found to be somewhat unreliable in its basic form due to inherent on-off behaviour of HAS: the need to transmit individual segments may cause complex effects at the transport congestion control level, affecting throughput estimations and fairness [62, 91]. Soon after the emergence of HAS, plain RBA has been eclipsed in QoE and flexibility by other algorithms, major representatives of which we discuss

[4]https://bitmovin.com/mpeg-dash-vs-apple-hls-vs-microsoft-smooth-streaming-vs-adobe-hds/; accessed 10/2018

next. These presented algorithms are arranged in order of their publishing; every one of them attempts to introduce an algorithm more performant in terms of QoE than the previous one(s).

State-of-the-art HAS quality adaptation algorithms

Jiang et al. [62] develop a quality adaptation algorithm (called FESTIVE) that is destined to address three goals of an adaptation problem: fairness, efficiency, and stability. They perform an in-depth analysis on how to achieve these goals, and develop recommendations for a quality adaptation algorithm. Authors report that: randomized chunk scheduling helps to avoid synchronization biases in sampling the connectivity state; stateful bitrate selection can alleviate the bias in the interplay between requested bitrate and estimated bandwidth; a delayed update approach improves selection stability and efficiency; bandwidth estimator should use the harmonic mean of downlink throughput over recent segments so as to be robust to outliers.

Li et al. [91] in their algorithm PANDA use throughput estimation to drive quality selection, though instead of immediately reacting to measurements, they only slightly step up in quality as to explore the available bandwidth, and falls back in case network conditions worsen. They compare their approach with TCP congestion control in a way that it is interpreting the Additive Increase / Multiplicative Decrease (AIMD) with its decision logic. Such a solution allows to more accurately estimate the available bandwidth and to reduce quality instability.

Huang et al. [59] propose to use buffer occupancy instead of throughput monitoring. Their idea is to define two buffer levels defining algorithms operation. In the beginning, when the buffer is empty, algorithm fetches the minimum quality representation segments. Once it reaches the first defined buffer level, it starts to fetch quality representations that are determined as a (presumably increasing) function of buffer occupancy. At some point, if network conditions allow, buffer occupancy will reach a level that can allow fetching highest-quality representations even if such a connectivity bandwidth is not always available. They also consider simple throughput-based capacity estimation in order to improve algorithm's performance during startup phase, as well as address quality instability.

Yin et al. [174] provide an algorithm that is based on Model Predictive Control. Their algorithms operations consists in predicting downlink bandwidth for a short future window of segments, and then solving an optimisation problem as to maximise an objective function being a weighted sum of application-level QoE metrics. Being the subject of study in Chapter 2, a more detailed explanation is deferred until Section 2.3.

Spiteri et al. [152] introduce an algorithm which they call BOLA. They formulate an adaptive bitrate problem as utility maximisation, and develop a buffer-based online control

algorithm that is focused on minimising playback stalls (or, alternatively, quality instability). As the base framework they use Lyapunov optimisation. An interesting feature of their algorithm is that it provides a provably guaranteed utility. Finally, they made efforts to push their algorithm into a popular DASH player dash.js.[5]

One of the most recent algorithms is proposed by Mao et al. [102], and it is quite different from the rest as it is not, in fact, an algorithm. Instead, it's a reinforcement learning system [156, 108] that, according to the authors, learns a quality adaptation logic on-the-fly without any prior knowledge about the task or its assumptions. As inputs, this system monitors a multitude of metrics and video system characteristics (like bandwidth samples, playback buffer occupancy, video chunk sizes). Their extensive evaluation demonstrates that Pensieve (as authors call their system) outperforms all major HAS adaptation algorithms in a wide variety of network conditions and settings.

HAS against Scalable Video Coding In Section 1.1.2 we were discussing Scalable Video Coding (SVC) that enabled quality adaptation in video streaming. Though being promising as a concept, it could not take hold in the VoD-type video streaming, being replaced by fixed-coding HTTP Adaptive Streaming. While Famaey et al. [42] report that SVC-based quality adaptation provides better performance in conditions with sudden and temporary bandwidth fluctuations, as compared to AVC (Advance Video Coding; they use H.264 encoding for both). SVC also reduces the amount of disk space to store the content in multiple qualities, compared to having to store multiple files with AVC. Nevertheless, they agree on the fact that SVC introduces a significant encoding overhead (up to 25% as compared to AVC at the same visual quality). Kalva et al. [68] compare the cost of maintaining video delivery infrastructure for H.264 SVC- and AVC-HAS system. They differentiate between storage and bandwidth cost. They take Amazon Web Services' tariffs on these resources, and demonstrate how AVC-based system happens to be cheaper than SVC for even smallest content providers, owing this to the cost of storage being fixed per month, but bandwidth being paid for per each client session. In addition to inherent computational complexity for scalable decoding, which did not have a chance (at least yet) to be implemented in off-the-shelf hardware decoders, SVC did not manage to get enough traction for VoD systems.

[5] https://github.com/Dash-Industry-Forum/dash.js/wiki; accessed 10/2018

Chapter 2

HAS quality adaptation and caching

As discussed in the Introduction and Background section, HTTP Adaptive Streaming (HAS, [33, 148]) greatly facilitates video content caching due to every video segment being a unique HTTP object. At the same time, Lee et al. ([85]) have shown that caching video segments happens to be detrimental to HAS operation in case if the quality adaptation algorithm uses observed bandwidth as a reference; they call it a "bitrate oscillation" phenomenon.

Let us take as an example a simplified network depicted at Fig. 2.1. It represents a backhaul network with a cache somewhere close to the client. Quite commonly, caches are connected to the clients with a high-capacity links (often being the last mile connectivity, 10 Mbps in the example), while the network segment from cache to the server can have lower performance (due to the level of congestion in higher parts of the network, by so representing a congested backhaul; 3 Mbps in the example). Imagine a scenario, where several users request the same video one after another and their quality adaptation algorithm is rate-based. This video has two qualities: one at a bitrate of 2 Mbps, and another at 5 Mbps. The first user will have to watch the video at the first quality and download all the video segments from the server. These segments will be stored at the proxy cache (assuming the cache capacity is big enough, of course). The next user then will start receiving segments from the cache (now without backhaul bottleneck; in our case - at 10 Mbps), and falsely realize that there is a room to upgrade in quality. He will then decide to request a higher-quality segment next, which is, however, not cached and so will be transported from the server. Since the server-cache connectivity is not enough to support 5 Mbps video streaming, the upcoming segment will be downloaded slower than real-time. The player then will have to show already-buffered segments to support continuous playback, by so exhausting the buffer, which may lead to its depletion and playback stalls if the cache-server connectivity is too slow. To counteract this, client's player may fall back to lower quality - which will then be streamed again from the

cache thus fooling the client again. Such a behaviour may cause not only playback stalls, but also frequent quality changes, both of which are agreed [110, 54] to severely impact the QoE.

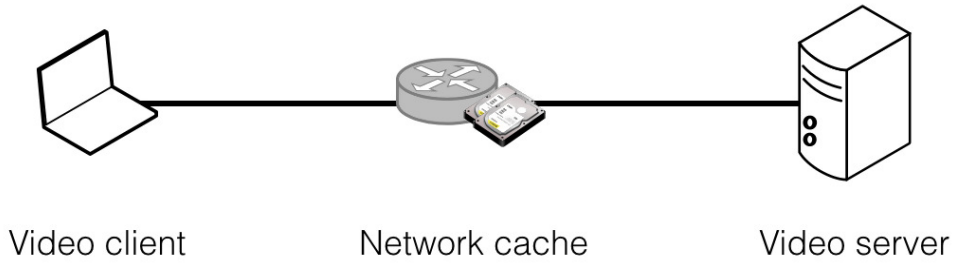


Fig. 2.1 Reference environment

Transparent caching of video segments requires, indeed, unencrypted transmissions. However, a considerable amount of video traffic originates from major content providers (such as Youtube, Netflix, Vimeo, Hulu and others), many of which own CDNs and encrypt their transmissions thus depriving ISPs of any control over caching decisions (unless man-in-the-middle SSL proxies are deployed). Nonetheless, there exist smaller content provider services that use no encryption. Often they are related to distribution of pirated materials, and might not make use of any CDN services for copyright reasons. Such services can still account for a considerable amount of traffic, so ISPs might be interested in employing their own caching infrastructure - which will greatly benefit from eliminating bitrate oscillations.

A number of studies have demonstrated their vision on solving this issue. Kreuzberger et al. [82] propose to employ traffic shaping to influence quality decisions on client in the context of ICN. They develop a model that optimises the Structural Similarity (SSIM) metric of transmitted videos, while taking into account that any node in the network can cache the passing content. With the help of video popularity profiles, they are able to estimate achievable video bitrate for their traffic shaping policy. While the main focus of their work is on fairness between competing video flows (that has been demonstrated to be an issue due to the operation of DASH), they report their technique to be working well against bitrate oscillations too.

Juluri and Medhi in [67] propose to predict client decisions in order to prefetch potentially necessary segments. This is achieved by using a dedicated framework that parses client requests and predicts the quality of future requested video segments using a generalised behaviour pattern. A later work [106] developed this idea and presented evaluations, which confirmed an improvement in prefetching accuracy provided that the client quality adaptation relies on throughput measurements. They do not directly consider the issue of bitrate

oscillations, but such a smart cache management can be an effective tool to alleviate the issue.

There exist other studies that do not rely on network entities collaboration, but instead propose to use playback buffer for quality decisions as it is not directly affected by difference in path capacities. Presented in more detail in Chapter 1.1, works [59, 175] propose quality adaptation algorithms that aim at maintaining a certain level of playback buffer occupancies as to prevent video stalls; if this buffer grows sustainably and surpasses that level, the algorithm may decide to upgrade in quality, otherwise – to downgrade as to quickly refill the buffer. While these algorithms have not been specifically designed to solve the problem of bitrate oscillations due to the presence of an in-network cache, they still manage to provide some degree of protection from variation in client-server end-to-end bandwidth. These solutions, nevertheless, apply heuristic approach which is not guaranteed to maximise users' QoE.

Our approach is to find a quality adaptation algorithm that would work reliably under cache presence. Though, as we have seen, there exist solutions that potentially alleviate bitrate oscillations by means of convoluted traffic or cache management, we would instead be strongly interested in developing a solution that approaches the issue in a principled way, while requiring only client-side modifications. Yin et al. in [174] have (then-)recently proposed a quality adaptation algorithm called FastMPC, developed from ground-up with a goal to be optimal in formally maximizing the QoE of a client during video streaming. FastMPC relies on a short-term bandwidth prediction (within several video chunk requests ahead), which is then used as an input for a problem of maximisation of weighted sum of several operational QoE metrics. QoE presentation as a weighted sum of operational metrics has caught our attention: being not only better-performing in terms of video quality and stall duration than then-current buffer-based and rate-based algorithms (provided that the bandwidth estimations are accurate enough), such an approach also gives a flexibility in tuning the QoE for application needs. Realising the potential of this approach in driving the quality selection in HAS, we attempt to test it against the issue of bitrate oscillations (from presence of in-network caches) and to see if there is a room to improve in regards of QoE metrics.

This chapter, therefore, seeks an answer to the following questions:

- How could a theoretical cache-aware optimal algorithm perform in terms of QoE metrics in the case of cache presence?
- Rate-based (also known as RBA) and buffer-based (also known as BBA) quality adaptation algorithms are commonly implemented in HAS players today; how far are they from that theoretical optimal?

- Considering that FastMPC is developed as an optimisation-based algorithm, how would it perform under cache presence?

This chapter is structured as follows. Section 2.1 will be devoted to the formulation of the theoretical performance boundary for a oracle-like generic quality adaptation algorithm under the presence of a cache. Section 2.2 will present the performance evaluation results of the Rate-Based (RBA) and Buffer-Based (BBA) quality adaptation algorithms in a scenario with an in-network cache. Section 2.3 will explain the operation of FastMPC, an optimisation-based algorithm. Section 2.4 will explain the cache-awareness for FastMPC, as well as provide practical considerations for implementing it in real-life conditions. Section 2.5 will demonstrate the performance of FastMPC and its perfectly cache-aware version in a presence of a cache. Section 2.6 will, finally, conclude the chapter.

2.1 Theoretical performance boundary under cache presence

Issues like bitrate oscillations and, in general, incorrect quality adaptation decisions during video streaming come from a fact that the algorithm simply cannot predict the future. Quite naturally, achieving the maximum theoretical performance at quality adaptation means that the algorithm is supposed to perfectly know network condition changes and client arrivals far ahead. In our work we use integer linear optimisation to implement such a theoretical boundary.

Indeed, the *performance* of a quality adaptation algorithm can be defined in many ways; since this work deals with the user QoE at first and foremost, this is what we will mean when referring to the performance of an algorithm. The objective function of the optimisation problem for the theoretical boundary, therefore, is exactly the QoE, and the problem will be aimed at maximising it.

Representing QoE as an objective function is not easy; as it has been explained in Chapter 1.1, QoE is a complex metric involving subjective experience of the users. Assessing QoE in such a way is rather difficult and perhaps not suitable for inclusion into an optimisation problem, so we pass from pure QoE to operational QoE metrics. Such operational metrics can include: startup time, amount of quality changes, duration of playback stalls. These metrics are objective; nevertheless, their objectiveness represents rather accurately the resulting subjective QoE. Indeed, the threshold between “good QoE” and “bad QoE” relative to, for instance, the duration of playback stalls will be different for every person, but it is sure that the less that duration is, the higher QoE will be reported by the user [54].

Even if the actual QoE is subjective, it can still be expressed as a mixture of objective operational QoE metrics, with different importance assumed for each of them. The already-mentioned work of Yin et al. [174] develops an optimisation problem as a basis for their FastMPC adaptation algorithm, which defined the objective function of QoE in a following way. For their QoE function they consider four objective operational QoE metrics:

- Video quality Q, which is directly related to the video bitrate of the representation.
- Instability of the video quality $\Delta Q = |Q_{k+1} - Q_k|$ - where k is the chunk index.
- Stall time, which is expressed through buffer deficit $dBp = (Size(Q_k)/C_k) - B_k$. Here C_k is the downlink bitrate for the client, and B_k is a client's playback buffer occupation in seconds, both at the moment of receiving the k-th chunk. In case if the difference is negative, the available bandwidth won't allow downloading the chunk before the playback buffer deplites (by this causing a stall).
- Startup delay T_s.

$$QOE_1^K = \sum_{k=1}^{K} Q_k - \lambda \sum_{k=1}^{K-1} \Delta Q_{k+1} - \mu \sum_{k=1}^{K} dBp_k - \mu^s T_r^s) \tag{2.1}$$

This expression is defined for chunks $k = 1 \ldots K$. From these metrics, we can derive more informative ones which can be used for a complex QoE assessment:

- **Average Video Quality**, the ratio of sum of all the qualities selected for segments during playback to the number of segments;
- **Total Stall Time**, the cumulative duration of buffer deficit over K segments (or simply the total time playback was blocked during playback);
- **Startup Delay**, the time elapsed between the request for playback and the playing of the first video frame.
- **Average Quality Instability**, the ratio of sum of all quality differences between two consecutive segments to number of segments minus one.
- **Average Buffer Level**, the ratio of sum of all playback buffer levels over number of segments.

A useful benefit of such formulation of the objective QoE function is its methodic way of accounting for the operational QoE metrics. Instead of keeping focus on a certain metric when

designing an adaptation algorithm, this linear combination (and the associated algorithm) binds all necessary metrics with the help of weights. In such a way the objective function can be tuned for the exact requirements of the client or content provider, thus offering flexibility.

As a measure of reducing complexity of the study, we consider *sequential* requesting of the same video content, such that no transmission takes place at the same time as another one. Request and transmission of all video chunks during one viewing session forms a *run*. During the course of the *run*, every transmitted video chunk is being cached in the network, thus affecting the decisions of the next run. Like this we are eliminating the effects entailed by several instances of adaptation algorithms working in parallel.

The model of a theoretical performance boundary should know the future system parameters for every future chunk request. Referring to the above breakdown of operational QoE metrics, we can conclude that Video Quality, Buffer Deficit and Start-up Time are defined by the downlink bandwidth of the client's connectivity. This is true for both RBA and BBA quality selection algorithms, even if in the former case the quality to request is only indirectly influenced by the connectivity bandwidth. The Quality Instability is, on the other hand, defined by the decision logic of an algorithm. In order to implement a theoretical performance boundary, we need to know the downlink bandwidth at the moment of requesting every future chunk, as well as to define a quality decision logic for that bandwidth.

As we will see in the following section, our experiment of comparing RBA and BBA to the theoretical boundary will be performed on a topology, depicted at Figure 2.1, for clients coming one after another and with no cross-traffic. A cache between the client and the server will be storing every video chunk passing through it, so, overall, the only thing preventing clients' players to reliably estimate connectivity bandwidth is the lack of knowledge about cache state. In our theoretical performance boundary we then assume that the algorithm has a complete visibility over the cache state. Quality decision logic, in its turn, will be rate-based so the algorithm will use its knowledge on downlink bitrate directly for its decisions.

Finally, in order to make performance boundary optimal, it has to assume control for all future decisions, even for future clients. Our optimisation problem for it should, therefore, maximise the Equation 2.1 for all the chunks in the video, as well as for all the subsequent video request by different clients. In this way, the problem can be formulated as

$$\max \sum_{r=1}^{R} \Big(\sum_{k=1}^{K} Q_{r,k} - \lambda \sum_{k=1}^{K-1} \Delta Q_{r,k+1} - \mu \sum_{k=1}^{K} dBp_{r,k} - \mu^{s} T^{s} \Big) \tag{2.2}$$

where the QoE function is now accounted not only for every chunk K, but also for every subsequent client request $r = 1 \dots R$.

The meaning of this model is the following. We assume that, during a video transmission, segments fetched by the quality adaptation algorithm will be stored in the in-network cache. During the next playback of the same video, those cached segment qualities will most likely be fetched again, but now from the cache hence with a greater download rate. In this case, client will increase its network conditions estimation and fetch segments of higher qualities than in the previous playback, which in turn will also be cached. This process will repeat over several consecutive runs until all the segments of the highest quality will become cached, allowing the client to take full advantage of the faster link and thus to provide better QoE. Our study aims at understanding the behaviour of quality adaptation algorithms during this transient phase.

2.2 Performance of an optimal quality selection and comparison with commonly-implemented algorithms

Since the QoE function of the Equation 2.2 is a weighted sum of different operational QoE parameters, the outcome of the optimisation will not necessarily reflect an experience considered as good. It is therefore important to assess how the performance of the theoretical boundary varies with different weight values.

We have programmed the aforementioned model in a Python simulator (that solves the model in the IBM CPLEX) and monitored three operational QoE metrics:

- average video quality for all the clients, displayed for each consecutive run
- quality instability, displayed per each run; this metric shows the share of chunk requests that have been decided a different (from previous request) quality
- rebuffering ratio, displayed for each run; this metric shows the share of total video duration that was spent in a stall (this total video duration is, therefore, a sum of the video length and stall duration)

Figures 2.2 – 2.4 demonstrate the performance of the model with different values of λ (for quality instability) and μ (for stall time), in terms of the mentioned metrics (vertical axis) for each consecutive run (horizontal axis). The video quality term is set constant with a weight of 1 in the Equation 2.1, so the rest of terms are relative to it. As explained by Krishnan and Sitamaran [83], the start-up delay is perhaps better kept minimal. Controlling the exact values of it (or any other metric) is difficult in an optimal model, so while the impact of a several seconds long start-up delay seems to be acceptable in terms of QoE (though with

increasing abandon probability with each second after two-second wait, according to the authors), obtaining more than half-minute wait times will most definitely be equivalent to a reject; for these reasons we decided to set fixed $\mu_s = 100$ to maintain a negligible start-up wait time. For the quality instability and stall time weights, the following combinations are considered:

- $\lambda = 10, \mu = 10$: this combination happens to be the most balanced. We can see that while the achieved average video quality for the first run is among the lowest, performance of the other two metrics are in between the other combinations. In addition to that, all three metrics almost attain the maximum performance already at the second run. Basing on this results, we accept this combination as a **reference** for optimal boundary in our settings, and we will compare the existing quality adaptation algorithms to an optimal, generated with these weights.

- $\lambda = 1, \mu = 1$: video quality has the same importance as the other two parameters. Indeed, the numerical ranges of the parameters are too different to have a meaningful interpretation of such a combination. For our settings, the curves with these parameters happen to perform better-than-reference for the video quality, but generally worse for the rebuffering ratio and quality instability.

- $\lambda = 0, \mu = 0$: stall duration and quality instability are effectively excluded from the objective function, so they might be sacrificed in order to achieve the high video quality and low startup delay. This is partially confirmed by the curves. The rebuffering ratio is among the highest for the tested combinations, the quality achieved is absolute (the highest average quality possible for all the runs), but there is no quality instability observed - which is quite logical, since achieving absolute maximum video quality means requesting chunks of only one, highest, quality. It is also interesting to note that while the first run experiences heavy playback stalls, subsequent runs are played smoothly. This is certainly the outcome of only one quality being requested all the time, so it becomes cached after the first run.

- $\lambda = 0, \mu = 10$: stall duration has more importance than video quality and quality instability does not take part in the optimisation. The latter is expected to be sacrificed for the other metrics.

- $\lambda = 10, \mu = 0$: the same as previous case, but now the stall time is disregarded in favour of other metrics. In such a case we might expect to obtain results very close to those of $\lambda = 0, \mu = 0$ combination, since achieving highest possible video quality means no quality changes (to lower representations). This is confirmed by the curves.

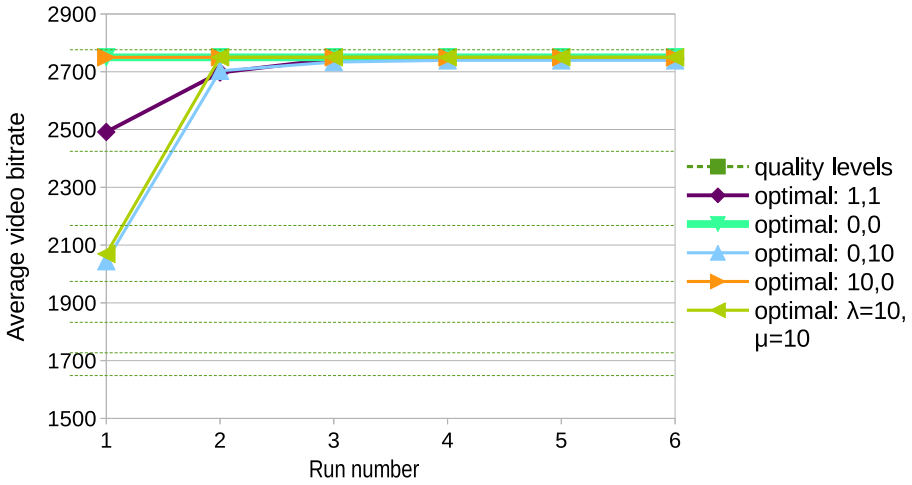


Fig. 2.2 Comparison of RBA and BBAs to the optimal policy for average video bitrate, for all condecutive clients runs.

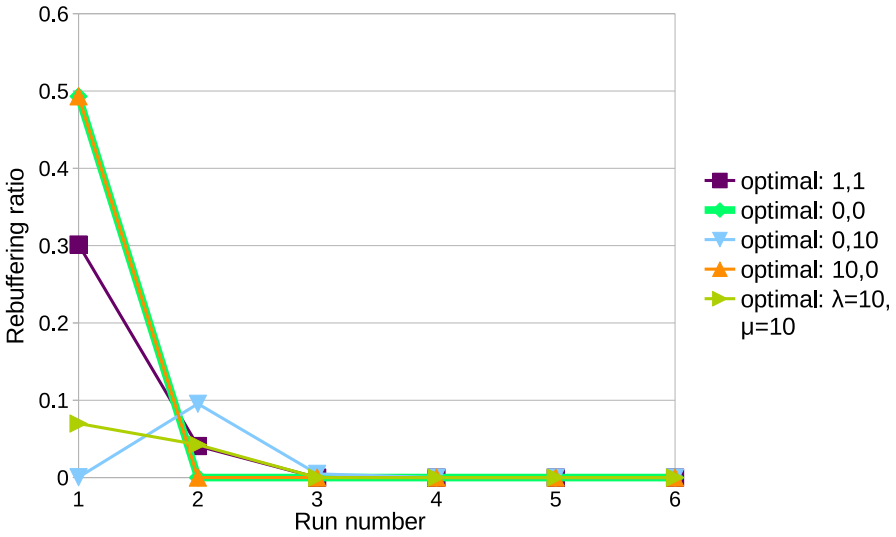


Fig. 2.3 Comparison of RBA and BBAs to the optimal policy for stall ratio, for all condecutive clients runs.

Now having defined the theoretical boundary a cache-aware algorithm's performance, we can proceed to finding out how far is it from the respective performance of current common quality adaptation algorithms. These adaptation logics we first consider as references are:

- Rate-Based (RBA): basic implementation that maintains the EWMA of the observed connectivity bandwidth and makes its next-quality decisions basing on it. This algorithm is available out-of-the-box in various open-source HAS-capable video players, such as VLC or dash.js.

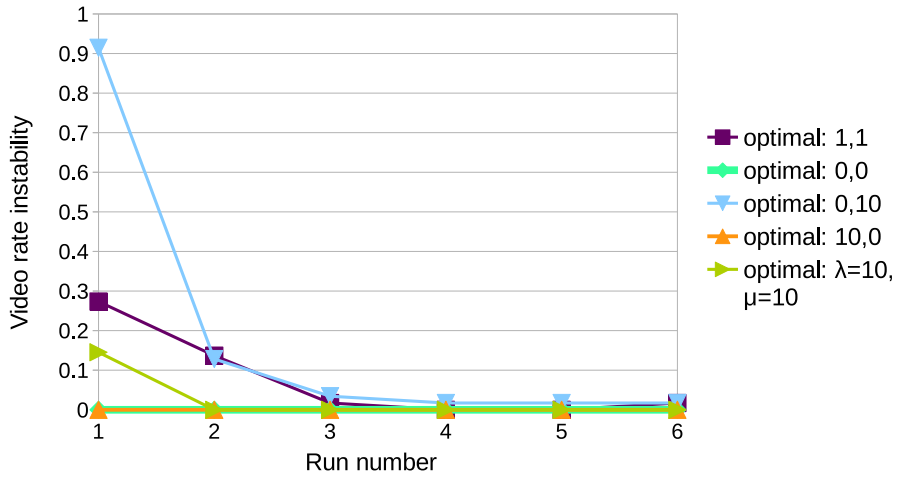


Fig. 2.4 Comparison of RBA and BBAs to the optimal policy for quality instability, for all condecutive clients runs.

- Buffer-Based (BBA): implementations of BBA2 and BBA3 according to [59]. BBA2 is focused on increasing the achieved video bitrate (potentially on the expense of increased number of quality changes during playback), while BBA3 favours video quality stability (potentially on the expense of lower average video bitrate).

The experimental setup is the following. The testbed is a chain topology made of three virtual GNU/Linux machines, as depicted on figure 2.5: an Apache HTTP server, an instrumented video player based on VLC v.3.0.0 and a Squid proxy acting as a transparent cache (with enough storage to cache all our segments). The *tc-netem* Linux tool from the *traffic control* suite is used to emulate a cache-server link of capacity $C_s = 2$ Mbps while the link between the cache-client link is non-constrained with $C_c = 1$ Gbps; such a generalization can be reasonable for modern mobile networks with edge caches. For all the experiments we stream the *Big Buck Bunny* [1] video that we encoded in HLS with the bit rate of the maximum quality representation higher than C_s. This video is made of $K = 300$ chunks of 2s-duration each.

As mentioned before, the optimal boundary is obtained using the Equation 2.2 with weights $\lambda = 10$, $\mu = 10$, $\mu_s = 100$. We observe that these parameters allow to obtain a rather high video rate over all runs (consecutive clients) while keeping the quality instability and stall time comparatively low, and maintain a negligible start-up delay.

Points of the next figures are generated with 15 samples, and the 95%-confidence intervals are shown (except for the results from the optimal model). The random component

[1] Available at https://peach.blender.org

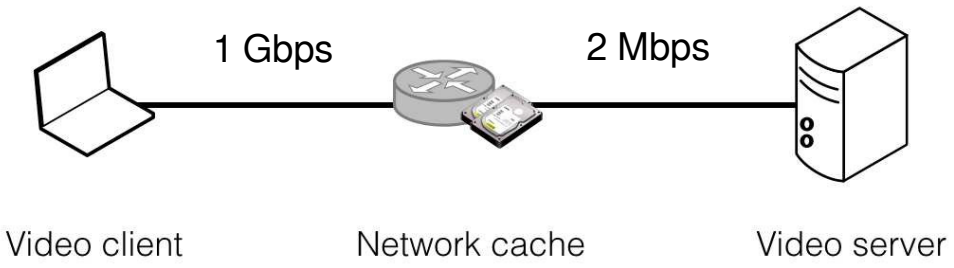


Fig. 2.5 Reference environment

that provides different results between samples is coming from random number generator initialisation for the optimal boundary, and overall software complexity for VLC testbed.

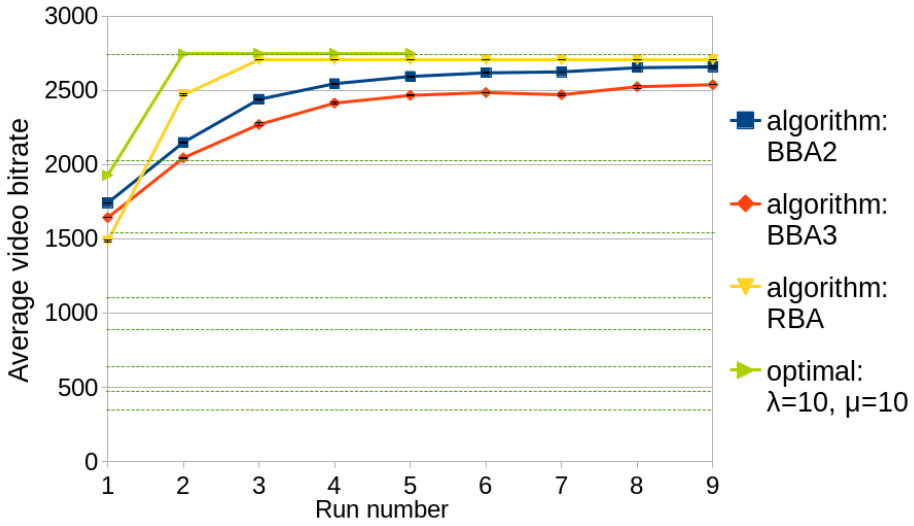


Fig. 2.6 Comparison of RBA and BBAs to the optimal policy for average video bitrate.

Comparing run 1 (when no cached chunks are available) with next runs, Figures 2.6 – 2.8 shows as expected that caching improves video bit rate. By being more aggressive in increasing the requested bit rate, RBA converges faster to the maximum rate, while both BBA1 and BBA2 lag behind. This faster convergence of RBA comes however at the expense of stalls during playback, in particular during the second run. By construction, BBAs strive to avoid stalls primarily. The video bitrate oscillations due to caching are seen in the high instability over the first runs. In agreement to its very design, BBA3 has a lower instability than BBA2 when the downloading bandwidth varies over consecutive chunks (i.e., in the first runs), on the expense of a slightly lower bit rate.

As for the results from the optimal model, interestingly, the instability and rebufferings are moved to the first and second clients: this can be explained as the optimization is performed

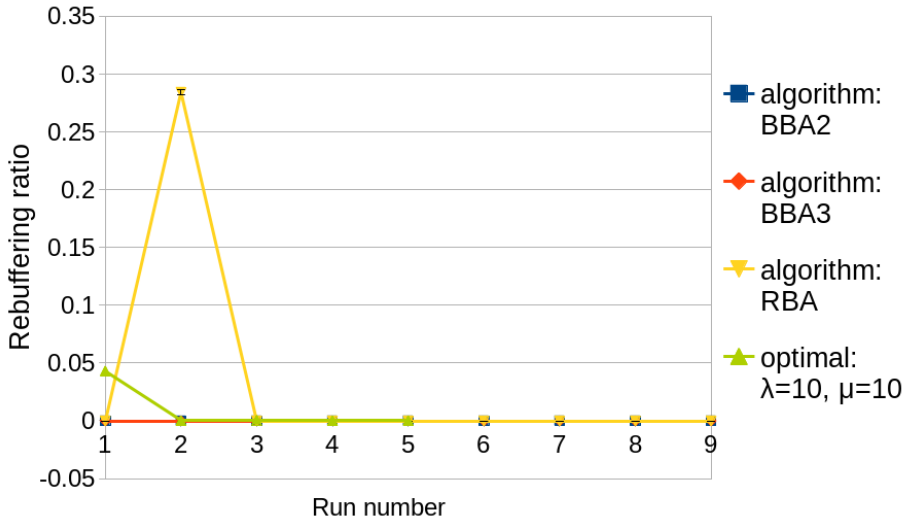


Fig. 2.7 Comparison of RBA and BBAs to the optimal policy for stall ratio.

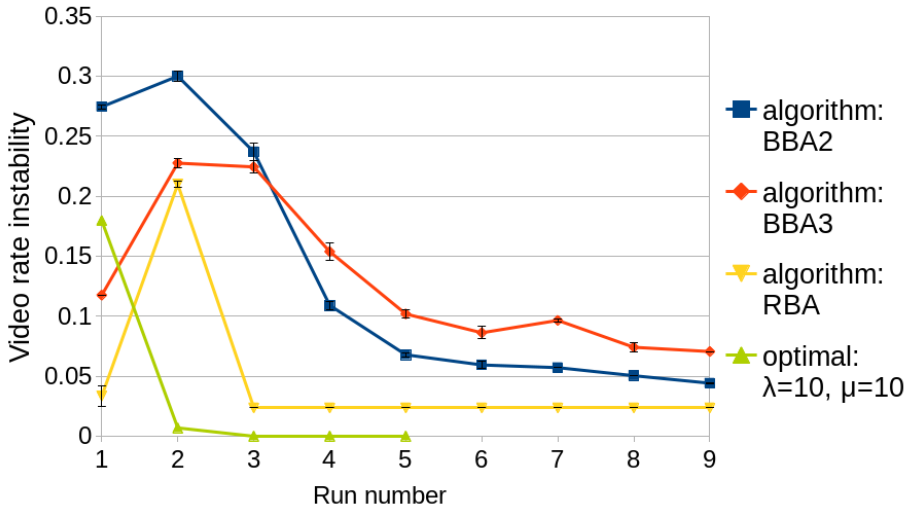


Fig. 2.8 Comparison of RBA and BBAs to the optimal policy for quality instability.

knowing the number of runs/clients. In order to maximize the average QoE over all clients, it turns out to be better to lightly penalize the first clients in anticipation of the next ones. Doing so allows to fetch higher quality chunks in the cache, for the benefit of the next clients. Despite the apparent unfairness of such decision (note that the client fairness is not part of the optimization), it actually benefits the majority, thereby unveiling that the instrumentation of rate decision can be a straight-forward path to control cache state. Note that the chosen example is rather extreme, as stalls are undergone by the first two clients, which can compel

them to abandon viewing. This can be easily avoided by changing the weights of rebufferings in the optimization.

Overall, if judging from the optimal results, we can observe that both RBA and BBA are left with a room for improvement in all three operational QoE metrics, which urges us to consider in the next section an optimisation-based quality adaptation algorithm – Fast MPC – with an expectation of achieving better QoE performance.

2.3 Model Predictive Control as a technique to implement optimal quality adaptation algorithms

The optimal performance boundary, presented in previous section, has been inspired by the work of Yin et al. [174]. In this article they propose to use Model Predictive Control approach in order to implement a quality adaptation algorithm that solves a linear integer optimisation problem to make a decision. Different to their optimisation problem, our performance boundary has a look-ahead horizon spanning over multiple consecutive video viewings as to take into account the in-network caching and evolution of its state over multiple runs. The proposed MPC/FastMPC algorithm has been demonstrated to perform better than the competition (among which are RBA and BBA algorithms), and in this section we will assess how does in-network caching affect performance of an MPC-based quality adaptation algorithm with a common cache-unaware EWMA-based downlink bitrate predictor, and compare its performance to the previously-defined theoretical boundary.

2.3.1 Background on MPC

The analytical expression Eq. 2.1 allows to make a decision about qualities of all video segments at once as a result of optimization. Such an approach, however, can hardly be implemented in reality due to common unavailability of accurate link capacity predictions for the entire video duration, so authors have called upon using a short *look-ahead window* (e.g., 10 seconds long) as it is possible to develop accurate predictor for such short windows.

The MPC is therefore a two-phase algorithm. First phase is Prediction, during which the download bitrates for the next segments in the look-ahead window are anticipated; these predictions are used as input to the optimisation phase. According to the authors, the accuracy of bitrate prediction plays an important role in resulting performance. The second phase is an Optimisation. MPC algorithm forms a control loop where current system parameters are fed to the optimisation problem to decide the quality of future segments. As required by the objective function of this problem (Eq. 2.1), system parameters are the previously selected

quality, current buffer occupancy and the prediction of future chunk download bitrates. To avoid solving a linear integer optimization problem before downloading each segment, Yin et al. [174] propose efficient pre-computation and data representation as a potential solution, which they call FastMPC. Indeed, the full objective function is only evaluated for the first chunk of the video; once the playback has started, there is no need to include the startup time into the optimisation, so it can be eliminated from the equation.

2.3.2 MPC in presence of a cache

To test the behaviour of the Fast MPC algorithm in the presence of cache, we have implemented the reference environment entirely in our Python simulator using segment quality sizes from the same video as in the previous section. The weight coefficients in MPC objective function are slightly different this time, with $\lambda = 1$, $\mu = 10$ and $\mu_s = 100$; this is made to explicitly eliminate the stalls being certainly among the most important video QoE components. As before, the link between the client and the cache has considerably larger bitrate capacity than the one between the cache and the server. The video to be streamed is encoded into several qualities, where the highest quality has an average bitrate greater than cache-server link bitrate, but lower than the other link. In the simulator, we repeated 5 times a sequence of 8 consecutive playback runs of the same video where the cache is flushed before the first run of a sequence. Each run is started after the end of the previous one. It has to be noted also that in the following experiments the video quality is not expressed in video bitrate, but rather in the quality index itself - from 0 (no segments requested) to 8 (all chunks are of the highest representations). With this modification the terms of the objective function will have comparable values, simplifying the choice of the weights. Figures 2.9 – 2.11 validates the progressive caching of segments with repetitions of playbacks to eventually reach the highest video quality after 5 playback repetitions. The x-axis shows the video playback run while the y-axis is the average value of the metric of interest over the 5 repetitions with the 95% confidence interval, namely *average video quality*, *average quality instability*, and *average buffer level*. We omit to depict evolution of the stall time as our weight configuration mostly eliminates those.

With Figures 2.9 – 2.11, one can note that under the presence of a cache MPC improves its performance in terms of average video quality, but suffers from significant increase of quality instability during the second consecutive run, which could potentially decrease the QoE as users are sensitive to quality fluctuations. On the other hand, once the transient phase is finished (after enough quality representations are cached), quality instability performance becomes better for when the cache is present in the network, as compared to a no-cache case.

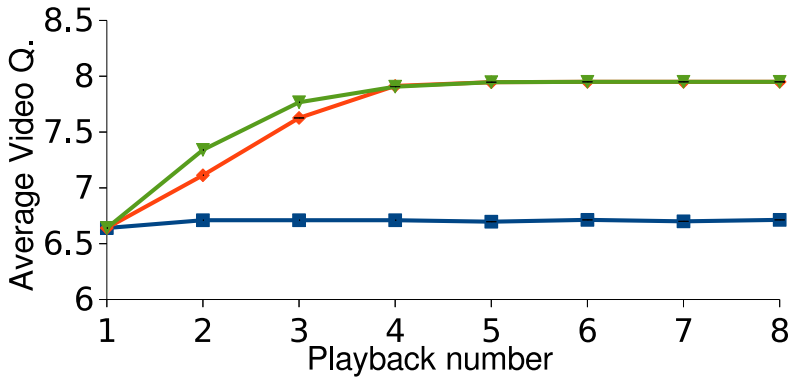


Fig. 2.9 MPC performance depending on cache presence and cache-awareness for average video quality.

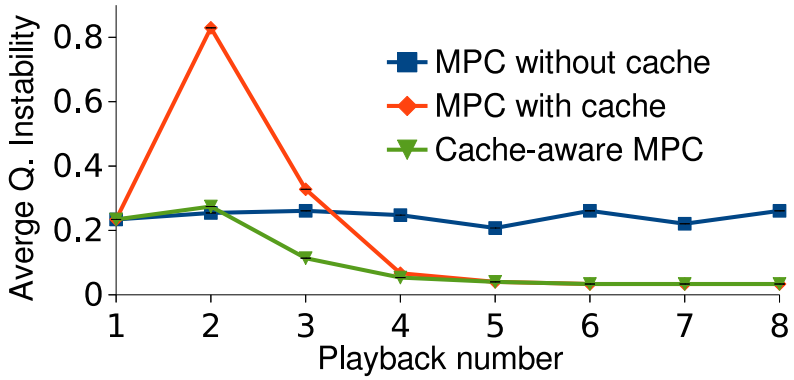


Fig. 2.10 MPC performance depending on cache presence and cache-awareness for stall ratio.

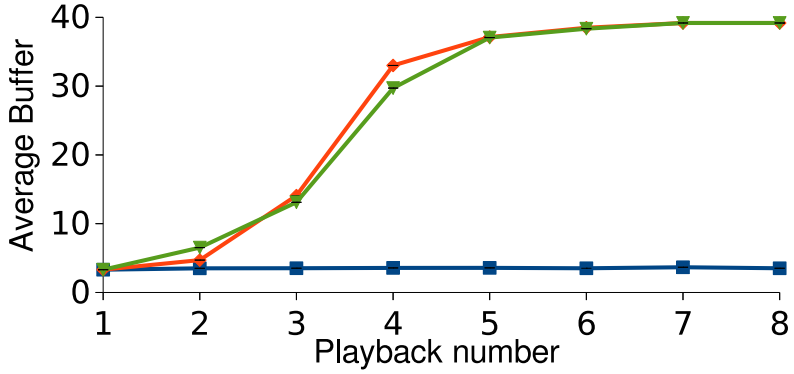


Fig. 2.11 MPC performance depending on cache presence and cache-awareness for average buffer level.

Fig. 2.11 also points out large average buffer levels differences with and without caches. Playback buffer level is not a subject for optimization, therefore put out of focus in favour of

the optimized QoE metrics. Due to this fact, we can see that average buffer levels during playback do not exceed 4 seconds without cache. This means that playback is very likely to stall in case of a major perturbation in the network, which is a major limitation of the objective function shown at Eq. (2.1). Presence of a cache improves this figures already from the second consecutive run.

It must be noted that, while Fig. 2.4 shows high instability of MPC with cache only for clients 2 and 3, this actually means that a critical number of clients can be affected. Indeed, depending on the cache size, this video popularity, the intensity of the other HTTP requests occupying the cache, a regime where clients consistently experiences the cache with a state equivalent to that of clients 2 and 3 shown here can be very frequent.

2.4 The need for cache-awareness in MPC-based algorithm

In case of weight configuration focusing on minimizing stalls and startup time, MPC with cache-unaware predictor manages to provide overall high quality playback under cache presence. However, we have observed the increased average quality instability at early stages of the transient phase when the cache is being populated.

The reason for such an instability in presence of network caches lies in the fact that MPC relies on an accurate future bandwidth prediction. As shown in Sec. 2.3, MPC indirectly needs to know the download bit-rate of the next segment in order to optimize the maximal average segment quality that can be requested to the server to optimally balance quality selection and stall protection with buffer occupancy. In the previous section, our evaluation of MPC was constructing the entire download rate prediction vector (of length equal to lookahead horizon) of just one EWMA (Exponentially Weighted Moving Average) value. This approach yields nearly optimal results as long as all the segments share the same network bottleneck (i.e., downloaded with about the same download rate). However, this assumption doesn't hold true when a cache is put between the client and the server. In this situation, as the cache-unaware MPC is unable to determine which segments will be retrieved from the network cache and which ones will be retrieved from the server, it ends up over-estimating the download bit-rate to retrieve segments. It will thus request segments for a quality higher than what can really be retrieved as in the transient phase high quality segments are not all in the cache yet. MPC control loop will then reduce the download rate prediction for the following segments as it measures the actual rate while retrieving segments. However, the lower quality segments requested by MPC might then be retrieved from the cache, which will fool MPC that will again overestimate the download rate and thus request high bit-rate segments that cannot be retrieved as they are not cached yet. This download rate prediction

error is at the origin of the video quality instability observed for MPC in case of the presence of a network cache.

To study the *cache-awareness* of MPC we therefore consider a download rate predictor having an exact knowledge of whether future video chunks in its viewing session will be coming from cache or not, thus giving an upper bound in algorithm performance. In contrast to predictor in our cache-unaware MPC model, our cache-aware version tracks two different EWMAs. One is for segments coming from the cache, and the other one is for segments coming from the server. Based on its knowledge of cache state, our idealized cache-aware predictor is therefore able to predict the expected download rate for each future segment based on its cache or server origin and optimize its segment quality selection according to its objective function. In this way, the cache-aware predictor takes advantage of the cache presence while leaving a room for fluctuating download rates, which it is not aware of.

2.5 Evaluation of cache-aware MPC-based algorithm

To obtain an upper bound of our cache-aware MPC, we simulated a perfect cache-awareness. For this, the simulator records which segments/qualities have been requested during each consecutive playback run and keeps this information between runs, and the perfect cache-aware predictor is using this information to know complete cache state at any moment in time. As mentioned just above, the difference to the actual optimal boundary becomes the inability of cache-aware predictor to have a 100% correct estimation of exact bandwidths of client-cache and cache-server connectivities.

To evaluate our cache-aware predictor we implemented a trace-driven simulator in Python. It uses segment sizes from the same HLS-ready video as before (Big Buck Bunny) at the input of the optimization in order to compute its runtime parameters such as playback buffer occupancy and video quality decisions.

2.5.1 Simulation settings

The cache is simulated according to Fig. 2.1. In the simulator it is operating by assigning download bitrates of Client-Cache or Cache-Server links to particular qualities of particular segments. These download bitrates are, in turn, used for calculating the actual playback buffer after reception of each segment as a function of previous playback buffer occupancy, selected segment quality size, and the mentioned download bitrate.

We simulate link fluctuations with a mobile dataset [135], which has a collection of download bitrate traces from Telenor HSPA network experienced in a number of scenarios

(e.g., city bus commute, intercity car trip). To make it suitable for our simulations, we have joined the download bitrate traces from several dataset scenarios having close environment, with the resulting tracefile having 1213 samples. We have only selected bitrate traces larger than 60000 Bytes per second. This is done to alleviate inherent drawback of cache behaviour modeling in our simulation: as download bitrate is assigned to the particular quality of a segment, accepting low bitrate traces may cause unrealistic situation when large segment, e.g., of 500 kB, will be assigned a very low download bitrate, e.g., of 10 kBps. In this case the download time will be simulated as 50 seconds, during which the connectivity bitrate could have in reality become larger so segment might have downloaded faster. In order to achieve the required average bitrate, each trace sample is multiplied by a factor in the simulation process. This factor is specifically calculated to set the average of all trace samples to a certain value. In the simulation, we apply this trace by assigning its samples one by one to each quality of each segment, so that each of the latter is associated with a unique download rate sample, with which it will be downloaded in simulation. Three different connectivity configurations are considered in our simulator:

- ***f*** (fixed), where both links have fixed capacity: Client-Cache is 4 Mbps, Cache-Server is 2 Mbps. Note that the Cache-Server link capacity is lower than average bitrate of the maximum video quality encoding.
- ***cv*** (client variation), where Client-Cache link is fluctuating with an average rate of 4Mbps while the Cache-Server link is stable at 2 Mbps.
- ***sv*** (server variation), where Cache-Server link is fluctuating with an average rate of 2 Mbps while the Client-Cache link is stable at 4 Mbps.

For the reference simulation, the following parameters are selected: maximum playback buffer is 40 seconds; look-ahead window is 5; video quality instability weight is 1; buffer deficit weight is 10; startup delay weight is 100. this configuration is equivalent to the one recommended by Yin et al. [174] with only exception of large startup delay weight used for fixing the latter to the same value (0) across all the simulations. Each algorithm is being run eight consecutive times (runs), within each of them cache state is kept. This experiment of eight runs is repeated five times for statistical analysis, with cache being flushed before first run of a new repetition. The random component comes from re-initialisation of the random number generator, that impacts the selection of bandwidth sample from the trace (as discussed above); this, in turn, impacts quality selection and hence cache state. This parameters resemble ones used in Sec. 2.3.2.

2.5.2 Observed metrics

In line with the components of the objective function Eq.2.1, we will focus on the following operational QoE metrics in the experiments that follow:

- Average video quality, for all video chunks during a viewing session
- Quality instability, which is the share of video chunks that have been decided a different quality as compared to that of the previous (to it) chunk
- Total stall time, for the entire duration of viewing session
- Average buffer level, calculated out of buffer levels at each moment of making a quality decision

In order ease the analysis and make it more compact, we present not the bare values of the mentioned metrics but a so-called *relative gain* of those metrics achieved by using a cache-aware predictor as compared to a regular one for an MPC quality decision algorithm, **in total over the transient phase viewing sessions**, which we define as consecutive run one to five. The formula for this relative gain calculation is presented at Equation 2.3. In this equation M stands for "metric" – out of the four presented above.

$$\frac{\sum_{r=1:5}(M^r_{cache\_aware} - M^r_{MPC\_with\_cache})}{\sum_{r=1:5} M^r_{MPC\_with\_cache}} \tag{2.3a}$$

This relative gain shows how much, in percents, the value of some metric is **larger** in case of the cache-aware predictor. For example, if cache-aware predictor has a total stall time of 40 seconds (over five transient phase runs), while regular cache-unaware predictor had 60 seconds of stall over the same runs, then the relative gain for the stall time will be -33.(3)% (since the difference between the two stall time measurements – 20 seconds – is 33.(3)% of the regular predictor measurement). Other metrics are summed up in the same way over first five consecutive runs, and their relative gain is calculated. Please note that obtaining, for instance, a negative value of the relative gain for Stall time is a good result, as it means that the cache-aware predictor's result is smaller than that one of regular predictor (and we, generally, do want the stall time to be smaller), while a "good" relative gain for the Average video quality would perhaps be positive (such that cache-aware result is higher than the cache-unaware one). In the same way, we consider relative gain values to be "good" if they are negative for Quality Instability, and positive for Average buffer level.

2.5.3 Cache-aware predictor evaluation

First of all, we have simulated perfectly cache-aware predictor with the reference parameters in order to directly compare its results with cache-unaware MPC implementation. Simulations (presented by the curves labeled "Cache-aware MPC" in Figs. 2.9 – 2.11) have shown that cache-awareness can significantly alleviate average quality instability compared to the cache-unaware MPC model, while the rest of the shown metrics are not experiencing any noticeable improvement. In order to more deeply observe the effect of cache-awareness,

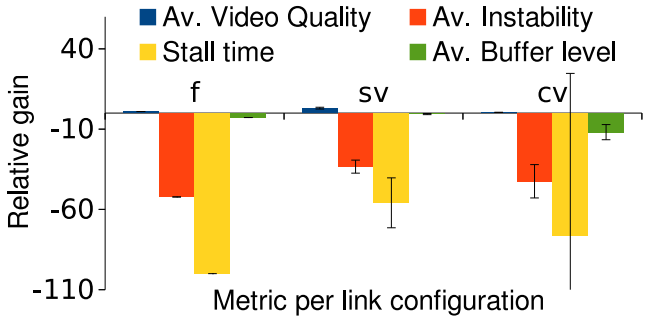


Fig. 2.12 Relative advantage of cache-aware MPC to cache-unaware MPC in a presence of a cache, for reference parameters

we have simulated communication link capacity fluctuations. Fig. 2.12 depicts relative gain in percents in total over transient phase of QoE metrics of perfect cache-awareness compared to cache-unaware MPC with a confidence interval of 95%. It can be seen from the curves that, alike to the configuration with fixed links, cache-aware model improves noticeably the average quality instability performance (i.e., decreasing its value) but does not introduce any significant advantage for average video quality and average buffer level. Additionally, Fig. 2.12 shows an important difference of *total stall time*; though relative advantage of cache-aware predictor in total stall time is considerable, the absolute values are often negligible, especially in case of *cv* and *f* connectivity configurations. The problem of low playback buffer, mentioned in Sec. 2.3.2, becomes noticeable when network connectivity is configured as *sv*, which makes Cache-Server capacity prediction very difficult. Fig. 2.13 shows the stall times over consecutive runs in the mentioned case. As can be seen, stalls are not completely suppressed with cache-awareness; rather, cache-awareness eliminates stalls after less consecutive runs – as compared to the cache-unaware algorithm.

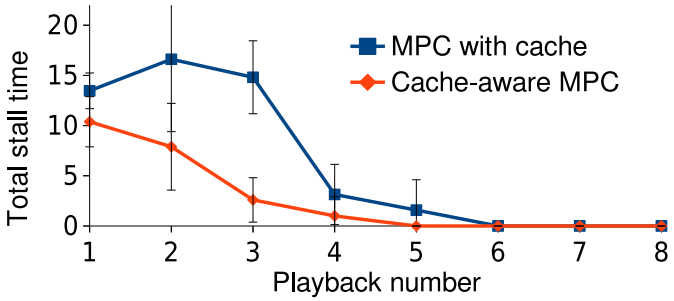
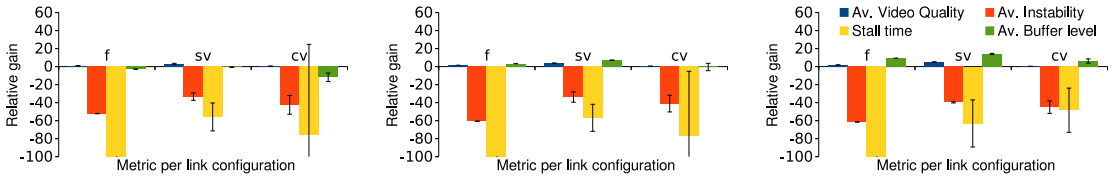
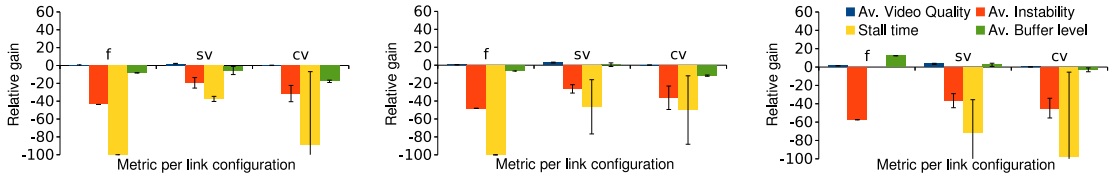


Fig. 2.13 Total stall time over runs for *sv* connectivity configuration and reference maximum buffer, look-ahead window and weights



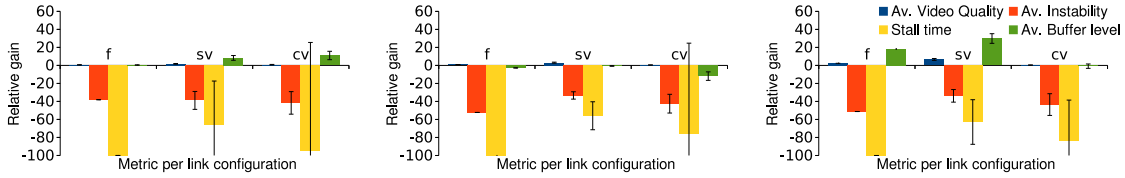
(a) Max buffer 40, window 5 (b) Max buffer 20, window 5 (c) Max buffer 10, window 5

Fig. 2.14 Sensitivity to maximum buffer capacity



(a) Max buffer 40, window 2 (b) Max buffer 40, window 3 (c) Max buffer 40, window 10

Fig. 2.15 Sensitivity to look-ahead window length



(a) Max buffer 40, window 5, 4 qualities (b) Max buffer 40, window 5, 8 qualities (c) Max buffer 40, window 5, 16 qualities

Fig. 2.16 Sensitivity to number of video qualities

2.5.4 Sensitivity analysis

In order to study how does cache-awareness compared to the cache-unaware MPC under different setup parameters, we perform a sensitivity analysis. The following varying parameters are examined:

- Maximum playback buffer: 10, 20, 40 sec;
- Look-ahead window: 2, 3, 5, 10 segments;
- Number of qualities the video is encoded into: 4, 8, 16.

Each combination of these parameters has been explored. The results are represented in the same manner as in Fig. 2.12.

Maximum buffer

Small playback buffers help the cache-aware predictor to reveal its strengths. As can be seen from Fig. 2.14, not all the parameters are clearly improved, but the relative gain in average buffer level shows a trend of increasing with lower maximum buffer. On the other hand, the absolute performance over the transient phase is better with larger buffers. This can be underpinned by the fact that very small buffers give less margins to manoeuvre for the cache-unaware algorithm when facing severe link capacity fluctuations due to caching. On the contrary, large enough buffers (able to keep 10 – 20 video segments) make the cache-unaware MPC more robust to these fluctuations, thus lowering numerical advantage of a cache-aware predictor. Finally, results from large buffers ($>$ 30 segments) are not significantly different from the previous ones.

Look-ahead window

Fig. 2.15 gives an insight regarding sensitivity of cache-aware advantage to different look-ahead windows. It can be noted that larger windows do allow cache-aware predictor to yield even better results compared to cache-unaware MPC. Average video instability and average buffer level are noticeably better with window of 10 than with window of 2, while average video quality and total stall time remain statistically the same. In absolute values, however, the performance of the algorithms are being improved with growing look-ahead window, especially comparing the shortest ones (2 to 3 segments long).

Number of qualities in video encoding

As the original video has been encoded into 8 qualities and because relation of video file sizes to their qualities is non-linear, we used logarithmic interpolation to make synthetic segment sizes for 4 and 16 qualities levels.

Generally, the advantage of cache-awareness grows with growing number of qualities, in configuration with reference parameters (Fig. 2.16). All the metrics are being improved with increasing number of qualities, but effect on average quality instability and total buffer level shows particular improvement. In absolute values, increasing the number of qualities is observed to induce more average instability and larger total stall time. The reason for this is that larger number of qualities provides a capability of a more fine-grained quality selection which might decrease video quality instability with cache-awareness, whereas the cache-unaware MPC would need to jump between more quality levels as to handle the fluctuating segment download rate.

2.5.5 Practical issues of achieving cache-awareness

Accurately predicting the cache state (i.e., which segment qualities were cached before) is a complicated task without directly collaborating with cache. To achieve this without the latter, we propose to use *intra-video popularity profiles* (where each segment is assigned a probability of being watched) leveraging the user retention distribution, and to estimate quality decisions during previous video playback. There exist studies employing such a technique; Siekkinen et al. [146] use this information to better know which video segments sizes to fetch when viewing a video, with an objective to minimise energy consumption of the process; another study, done by Maggi et al. [98] develop a video chunk caching strategy basing on this knowledge.

For a problem of cache-awareness of the client's quality adaptation algorithm, our considerations are built upon the fact that average video quality grows gradually over playback runs, thus suggesting that segment quality decisions are usually close between two consecutive runs. If the download bitrate of segment K is noticeably larger than that of previous segments', it is likely that it comes from a cache. Using the intra-video popularity profile, we can estimate the probability of next segment to be cached. If the profile suggests (comparing it with thresholds) that segment $K+1$ will be cached, but its download bitrate is too low (i.e., likely that it came from video server), then we might presume that previous client's playback buffer could have depleted hence it could have backed off to a lower quality. In this case we can try to estimate which quality it was by performing optimization with the same runtime parameters except for playback buffer, which we believe was zero for the previous client.

Such an approach can potentially yield good results, in case if the intra-video popularity profile is accurate and algorithm's threshold for cached/not cached decisions are adjusted upon each wrong decision. Improvement notwithstanding, discussed ideas will significantly increase algorithm complexity.

A less speculative solution (yet a very effective one) to the problem of cache-awareness would, indeed, be collaborating with the cache infrastructure. In fact, this task is one of the use-cases for the recently proposed (and now actively-developed) MPEG-SAND (Server and Network Assisted DASH) framework [158]. This framework defines architecture and message formats for information exchange between video system actors, and such capabilities can be used to explicitly signal the client about cache state. A recent work of Samain et al. [136] has demonstrated the viability and benefits of such an approach.

2.6 Conclusion

In-network caching is one of the effective methods to tackle the problem of quickly growing video traffic but various studies show that caches may have a negative impact on Quality of Experience when HTTP Adaptive Streaming is used. In this chapter we attempted to quantify this negative impact in terms of different operational QoE metrics for then-common quality adaptation algorithms – Rate-Based and Buffer-Based. To do that, we formulated a theoretical performance boundary of a generic quality adaptation algorithm, that knows in an oracle way the state of the in-network cache (together with the connectivity bandwidth values, though those being fixed to a certain value). We have found that simplistic (yet efficient) design of RBA and BBA does not allow them to properly react to the presence of in-network cache, and by so end up being rather inferior in performance to the theoretical optimal boundary. We then went on to assess the impact of network caches on the cache-unaware MPC quality selection algorithm, as its tunable design gives a potential for better handling of caches. We demonstrate that, in cases when the MPC objective function is focused on eliminating stalls, it takes advantage of caching to increase video quality at the expense of much increased quality instability. The reason for the latter is that cache-unaware MPC (which is based on commonly-used bitrate EWMA predictors) is not able to accurately predict download bitrate when cache is present, just as for BBA and RBA. In order to identify whether cache-awareness can confront this drawback, we propose an upper bound for cache-aware predictor to MPC that knows the exact cache state at any point of time. Such a predictor has shown to bring noticeable improvement by reducing video quality instability; however, cache-awareness is difficult to achieve in real world as it entails, as we discussed, either significantly increasing algorithm complexity in a practical implementation,

or using client-cache collaboration techniques such as those, offered by the MPEG-SAND framework [158, 136].

The perspective of establishing such a collaboration has motivated us to take a look at the problem of video delivery at a bigger scale. In the following chapters, we devote our work to studying the interoperation options between the client (or its content provider) and video caching infrastructure.

Chapter 3

The interplay of multipath and caching

Previous chapter was focused on the interplay between caching and HAS quality decision algorithms. In this chapter we take a step back to see how caching operates on a large scale video streaming system being embodied into a Content Delivery Network. We take the concept of multipath data transmission as a novelty factor in this study.

Today's computers and smartphones often dispose of several separate network interfaces, which makes them capable of multiple connectivity – thus being a component of a novel Multi-RAT network planning approach. This promises better downlink bandwidth and, consequently, potentially better QoE. Multipath capability is especially useful in the perspective of the near-future abrupt increase in content volume. For instance, consumption of the ultra-high definition and Virtual Reality (VR) video content will raise the per-user demands to well above 30 Mbps [3]; at the same time, current state of mobile networks around the world is such that average LTE downlink bandwidth is less than 15 Mbps in many developed countries (e.g., 13.4 Mbps in France and 12.3 Mbps in USA [fra]), which certainly gives no room for future video applications. The idea behind using multipath for tackling this issue is, therefore, to group multiple Internet connections (e.g., Wi-Fi and LTE) together to add up their bandwidth.

Aggregation is commonly used in servers and Ethernet networks to group links and ports to increase their bandwidth and resiliency [137]. It is however restricted to one hop only. For multi-hop, Multi-Path (MP) routing is used in the Internet to balance load across different paths to the same node. Traffic splitting is however often made on transport protocol ports, thereby preventing a given connection to benefit from bandwidth aggregation. For an end device to benefit from it, it must be able to use at least two radio interfaces at the same time, which modern devices are now capable of: MultiPath TCP (MPTCP) adds the ability to use both interfaces for the same service ([15, 43]), and it is already supported in Apple devices. It is initially meant to improve reliability, with one access taking over in case the other comes

down (a notable example is in [90]), but it also offers new perspectives in terms of bandwidth aggregation at the Internet level if one application can send or receive simultaneously a fraction of its bytes on both access networks. This can therefore result in higher quality of experience without the ISPs having to invest as much in re-provisioning their network as the traffic peaks can be smoothened over multiple networks. Initial deployments of this solution show its feasibility [143] and call for studying the case of Internet-wide bandwidth aggregation with MPTCP more carefully. In 2016, a proposal for transparent link bonding for hybrid access networks based on MPTCP has been released [125]. It suggests to have special equipment beyond the borders of the two access networks, so as to enable MPTCP regardless of the capabilities of end server and end client.

In this chapter, we take a look at the interaction of two main actors of a video delivery system: Content Provider (CP) and Internet Service Provider (ISP). CP is an entity that owns the video content; we assume that it is deeply associated with a Content Delivery Network (CDN), either its own (like Youtube) or outsourced (Akamai), so as to increase its serving capacity and improve QoS/QoE of his clients. The task of CP is, therefore, to accept clients' requests and to assign them a server or cache such that their Quality of Experience would be as good as possible. ISP provides network connectivity for the users to access the Internet, including the services of the CP; Content Provider's CDN has caches deployed at the ISP in order to achieve its purpose. The task of ISPs is to accept the traffic from all of their clients/services (not only the CP), for which they make sure all the flows going through their network cause as little congestion as possible (by means of routing optimisation, for instance).

The possibility of bandwidth aggregation can make multipath a key in serving more demanding applications. MP is however not compatible with having a cache inside one of the access networks to serve the client. In case if caching infrastructure is placed close to the clients, it may not be accessible outside of its ISP due to associated network strain; cache connectivity is then limited to only single-path (unless one considers client connected to the same ISP via several access nodes, which is rarely possible). It means that, in order to benefit from multipath bandwidth aggregation, a client needs to request the content from main CP's servers, which are outside of his ISPs and hence multipath-capable. This incompatibility therefore menaces the main principle of the Internet delivery strategy: putting caches close to the users so as to improve accessibility of services to the clients and, more importantly, decrease the load on upper levels of the ISP networks. (see Fig. 3.1).

We, therefore, address the following questions in this chapter:

- What benefit do these new operational points bring for the different (and potentially contradicting) objectives of Content Provider (higher client quality) and Internet

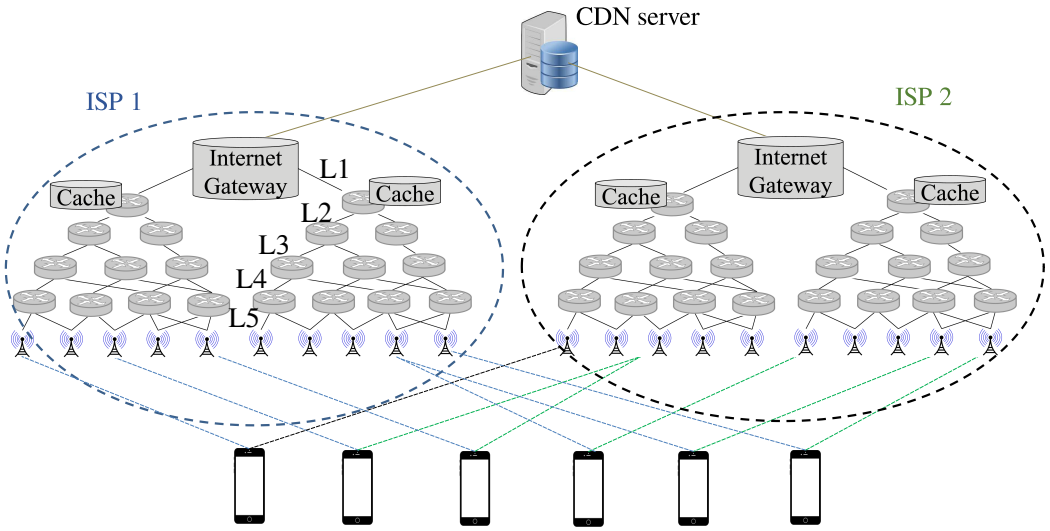


Fig. 3.1 Clients connected to two access networks, with in-network caches and the external CDN server. “L1”, “L2” and so on indicate the ISP topology level.

Service Provider (need to serve new and more demanding services, quality for all services, low congestion)? Or, in other words, what strategy is better for both of those actors and for which use-cases: serving from higher in the network to potentially achieve higher bandwidth, or closer to the client edge to decrease network congestion?

- In case if CP and ISP collaborated in order to achieve a mutually-beneficial performance of a video delivery system in terms of their objectives, how would this performance differ from a present-day scenario (where independent ISPs and CPs do not collaborate?

To answer these questions, we implement a numerical evaluation of the following models of interoperation modes for the pair CP – ISP: (i) *no collaboration / no visibility*, when parties do not cooperate with each other and the CP can only know the ISP’s network situation by end-to-end measurement; (ii) *no collaboration / full visibility*, when parties still do not collaborate but the CP has perfect knowledge of the network conditions at the ISP’s; (iii) *full collaboration* when both parties are essentially merged into a single entity thus making their decisions jointly. In this evaluation we assess such metrics as achieved video bitrate, client acceptance rate and ISP network congestion.

Multipath data transport itself is a well studied area in theory [49, 164]; moreover, a transport protocol called SCTP (RFC 4960) was developed in 2000 that was capable of multipath transport. In [8], Apostolopoulos and Trott were among the first to discuss the use of path- and source-diversity options in relation with the media encoding, in particular

underlying the benefit of Multiple- Description Coding to seamlessly serve a client from different CDN servers. They concluded that using multipath and multisource techniques in video streaming has a potential of improvement in such areas as connectivity fault tolerance of streaming sessions and their resource aggregation. This work, however, mostly explored the potential of the technologies which were not even available at that time (MPTCP, HAS), so a more practical view on it was yet to come.

Nevertheless, it was the development of a dedicated protocol – MPTCP [15] – that gave traction to the whole concept being finally implemented. Chen et al. [31] performed an in-depth study of real world MPTCP performance over mobile and local Wi-Fi networks. They concluded that in basic scenarios (like arbitrary data download from a remote server) MPTCP proves to be a resilient and efficient transport protocol. On the other hand, Arzani et al. [10] studied scheduling policies in MPTCP and provided a slightly less optimistic point of view on its performance. They performed a controlled simulation with the official Linux MPTCP implementation to study the impact of different path characteristics According to them, performance of MPTCP is highly dependent on QoS characteristics of the path which is selected for the main subflow. After initial flow establishment, a good choice of a subflow scheduling policy may also affect MPTCP's performance. Basing on these findings, and in order to avoid the impact of MPTCP implementation design, we decided instead to focus on high-level evaluation to study how does the concept of MP only behaves in our scenario.

Nam et al. [112] address the issue of MPTCP packet reordering in a an environment with highly asymmetrical paths. MPTCP schedules packets of a single data object to multiple subflows, so when bandwidths and latencies of those individual subflow paths are too different, transmission time of the data object can increase significantly because MPTCP out-of-order management has to wait for packet from a "bad" path. They demonstrate that when relative throughput difference between two paths is more than 70%, in certain scenarios singlepath TCP can download objects faster than MPTCP. Basing on these findings, authors propose an SDN-controlled traffic engineering strategy where controller can cut the second MPTCP subflow in case it estimates the path symmetry to be affecting MPTCP performance.

Kim et al. [76] proposed a different design of a path-diverse protocol for data transfer in the Internet. As opposed to MPTCP and SCTP, they move multipath management from transport layer to application layer, and base their design on HTTP protocol. mHTTP, as authors call it, is using range-request HTTP header to request different parts of a file from multiple servers or through multiple connectivities. Being identified as usual HTTP flows by the servers, client's mHTTP process is managing the assembly of received data. They demonstrate that such an approach only requires modifications at the client, and using a vanilla HTTP for individual flows makes it entirely compatible with every today's (HTTP-

oriented) Internet infrastructure, even when using multiple sources (which is not possible with MPTCP due to protocol design). Authors discuss also a multiDNS, a design consideration that enables mHTTP to discover additional source addresses. Being a very interesting concept by itself, to our knowledge it did not leave research confinements thus making MPTCP a more realistic alternative (albeit not as functional).

A problem of the interplay between multipath and CDN server selection has been uncovered by Nikravesh et al. in [115]. They identify how the DNS-based CDN server selection, which is not aware of the MP capability of the client, along with the min-RTT scheduler of MPTCP, may lead to sub-optimal server selection depending on the RTT heterogeneity of the paths and the type of desired transfer (short or long file). While heir focus in this paper is on general MPTCP performance in mobile environment, they do give high-level recommendations on how to improve CDN server selection for MPTCP-capable devices: according to them, for short flows priority should be given to low-latency servers (thus, perhaps, closer ones), whereas for long-lived flows a server with more available bandwidth should be favoured. This work does not, however, touch upon the burden of the ISP in transporting the MPTCP traffic, which is one of our focuses.

Finally, Jiang et al. in [63] provide a mathematical framework for cooperation between CP and ISP. Their idea is to model the behaviour of ISP and CP as optimisation problems with potentially contradicting objectives (those being congestion for ISP and latency for CP). They find that providing CP with complete visibility over ISP's network utilisation, surprisingly, results in worse performance for both of the parties (for their objectives and assumptions), in comparison to only providing end-to-end information to CP. They define a theoretical boundary for idealised joint ISP – CP performance, and conclude that non-collaborative strategies have a significant room for improvement in terms of both parties' objectives, as well as in fairness between them. Finally, they proceed to designing a realistic collaboration for such a scenario that bases itself on the concept of Nash bargaining [114, 21] so as to provide both optimality and fairness properties for their distributed collaboration strategy. In this and the following chapter, we inspire from them in questions of modelling and collaboration design between ISP and CP in a multiple-ISP connectivity environment with a different CP's objective being its video bitrate. In addition to that, we consider the realistic constraint of server unicity (in line with TCP/MPTCP limitations), while Jiang et al. considers a continuous fraction of data can be fetched in parallel from all the servers.

This chapter is organised as follows. Section 3.1 tells about the relationships between different actors of the system and explains their models. After that, Section 3.2 conducts a numerical evaluation of the effect of enabling multipath in our system. We observe that the benefits of multipath happen to be somewhat limited for some metrics in certain cases

- which could be due to the lack of collaboration between CP and ISP. We then construct an upper bound for their joint performance in Section 3.3, and numerically evaluate the theoretical benefit for both CP and ISP from the use of a perfect collaboration in Section 3.4. The chapter closes with concluding remarks in Section 3.5.

3.1 Decision modelling for system's actors

This section models the decision-making process of the following actors: one CP, two independent ISPs and a large group of clients (see Fig. 3.1). Let us consider them in detail.

3.1.1 Content Provider

CP's CDN datacenters are connected to virtually all their clients' ISPs ([170]) either by private circuits (e.g., leased lines), via Internet eXchange Points (IXPs), or via transit providers [139] and can thus leverage the possibility for MP video delivery to their clients. Video streaming servers are located in these datacenters but CDNs can optionally deploy caches directly in ISPs premises (like Netflix OpenConnect[1]) to reduce latency and path length.

We model a scenario where CP's CDN has one datacenter with a pool of servers deployed behind an MP-capable load balancer (i.e., clients see the datacenter as one single server reachable from both ISPs and hosting the entire video catalog). The datacenter is directly connected to both ISPs with a private circuit. As we do not want the content placement to impact the outcome of the interplay, in this study we consider each cache is a replica of the entire CP video catalog. To avoid cross-ISP traffic, caches can only serve clients in the customer cone of their ISP, hence are not MP-capable. The video catalog is constant and all videos are pre-loaded on every cache/server. The rate selection problem of HAS is abstracted as well and we consider the obtained video rate to be the (possibly aggregated) bandwidth. The CP's server selection therefore aims at maximizing the video rate the client will get. We model it with the following integer linear optimization problem.

Let our CP host a server, a set of caches deployed at two ISPs, and a set of clients (each of them is connected to both ISPs). This can be represented as a graph $\mathscr{O} = (\mathscr{V}_{cp}, \mathscr{E}_{cp})$ where $\mathscr{V}_{cp}$ are network nodes and $\mathscr{E}_{cp}$ are directed links connecting them. Nodes in $\mathscr{V}_{cp}$ are classified into two groups - sources and clients. Let us denote the root server and caches as $s \in \mathscr{S}$, and clients as $t \in \mathscr{T}$. Clients are connected to access nodes $d \in \mathscr{D}$ which can only be inferred by the CP. In this way, the traffic between a cache or server and a client is x_{st}, while the path it follows has a capacity C_{st}. It has to be noted that CP's outside server (that belongs to s)

[1] https://openconnect.netflix.com

can support multipath, so the traffic between such servers and their clients would be in fact a sum of two subflows $x_{st} = x_{sd_1} + x_{sd_2}$, where d_1 and d_2 are the access nodes that connect the client to both ISP networks (as we consider all clients connected to a pair of access nodes to have relatively the same downlink bandwidth). This problem does not use the network topology to determine which servers among s are multipath (hence capable of two subflows), and which are only single-path caches. Instead, every traffic flow x_{st} should be formulated explicitly when programming the problem for solving, depending on whether it can have two subflows or not. We assume that ISPs' network topologies are not exposed to the CP, hence any t is only one hop away from any accessible s and the CP estimates the available bandwidth C_{st} of each overlay path. Since this problem is hard to solve for big amount of requests, and considering that it is hard to know the future requests, we can transform our problem into an online decision for each incoming request, as presented in Problem 3.1.

Name	Description
$max\_rate \in \mathbb{R}_{\geq 0}$	maximum bitrate of the requested video
$min\_rate \in \mathbb{R}_{\geq 0}$	minimum bitrate of the requested video
$bw\_est_{sd} \in \mathbb{R}_{\geq 0}$	CP's estimation of the available bandwidth between server $s \in \mathscr{S}$ and access node $d \in \mathscr{D}$

Table 3.1 CP problem Input Parameters

Name	Description
$x_{sd} \in \mathbb{R}_{\geq 0}$	video traffic flowing from server $s \in \mathscr{S}$ to client's access node $d \in \mathscr{D}$
$p_s \in [0;1]$	binary variable indicating whether server/cache $s \in \mathscr{S}$ is selected

Table 3.2 CP problem Decision variables

$$\max \sum_{\substack{s\in\mathscr{S},\\ d\in\mathscr{D}}} x_{sd} \tag{3.1a}$$

s.t. (3.1b)

$$x_{sd} \leq bw\_est_{sd}, \qquad \forall s \in \mathscr{S}, d \in \mathscr{D} \tag{3.1c}$$

$$\sum_{d} x_{sd} \geq min\_rate \cdot p_s \qquad \forall s \in \mathscr{S} \tag{3.1d}$$

$$\sum_{d} x_{sd} \leq max\_rate \cdot p_s \qquad \forall s \in \mathscr{S} \tag{3.1e}$$

$$\sum_{s} p_s = 1 \tag{3.1f}$$

In Eq. 3.1, constraints are presented as following.

- Eq. 3.1c guarantees that no bitrate x_{sd} surpasses the bw_est_{sd} as the result of selection process.
- Eq. 3.1d and Eq. 3.1e guarantees that bitrates x_{st} between any server s and client t (which is the sum of x_{sd}) respect the bitrates of minimum- and maximum-quality representations of a requested video, while also ensuring that $x_{st} = 0$ for all s, p_s of which are zeros.
- Eq. 3.1f guarantees that only one server can be selected.

The bw_est_{sd} parameter in this problem is a CP's estimation of bandwidth between its source and client's access node. While an accurate algorithm for this purpose is of paramount importance for the system's performance, developing one is not the purpose of this chapter. In our study we assume that CP collects such measurements during client service, alike to [45] (clients are grouped by their attachment to ISP's access nodes), and then manages them using EWMA. Our system assumes a fair amount of variability (different video characteristics, network conditions, and so on; this will be demonstrated later), so EWMA has to be smoothened out by a large weight for the previous measurement. At the same time, we have observed that EWMA values in our scenario can decrease until such small values that CP would never choose that path again, thus also ceasing to update this path's EWMA of bandwidth. It is important therefore to re-initialise these values periodically so as to ensure continued and accurate update of bandwidth estimations.

As one can see, we do not allow multiple sources (contrary to [63]) and consider the realistic case of single CP server selection, hence needing an integer linear problem formula-

tion. If the output of the optimization is a rate lower than the minimum bitrate, the request is rejected.

3.1.2 Internet Service Providers

Our considerations on realistic backhaul network topologies is based on [25], i.e., the topology we study is a partially-meshed tree. We consider here that each ISP has the same topology to preserve symmetry and ease the interpretation. Clients connect with the CP's CDN through two ISPs having identical topologies (as depicted on Fig 3.1). Each node of the topology can be a switching node, or a user access node (e.g., an eNodeB); it can also be (connected to) a CDN cache at the same time. We consider that access nodes are located at the bottom of the topology figure, and caches are located at the second from top level. Such design is representative of common mobile backhaul architectures where caches are deployed only at Packet Gateways due to limitations of GTP tunnelling, which hides the backhaul network from users' IP traffic [177]. A cache from one access ISP cannot be used to serve a client from another access ISP, whereby the incompatibility of MP from heterogeneous accesses and in-network caching.

Each ISP wants to provide high video quality to its clients, but also mitigate congestion so as to maintain a reasonable service quality for the services other than video (e.g., web-browsing). To do so, considering an input traffic matrix (impacted by previous CDN server selections by CP), each ISP optimizes routing in its own network by solving a single-path multi-commodity network flow problem.[2] A problem very close to this one is well presented by Jiang et al. ([63], Eq. 1) and fits nicely in our environment, so we inspire from it in this work.

Let each ISP have a network $G = (\mathscr{V}, \mathscr{E})$, containing nodes $\mathscr{V}$ connected with directed links $\mathscr{E}$. Let us denote the flow between any pair of nodes as x_{ij}, where $i, j \in \mathscr{V}$. Each link $l \in \mathscr{E}$ has a capacity C_l. The proportion of a flow x_{ij} over a given link l will be denoted as $r_{ij}^{l} \in [0;1]$

Name	Description
$C_l \in \mathbb{R}_{\geq 0}$	capacity of link $l \in \mathscr{E}$
$x_{ij} \in \mathbb{R}_{\geq 0}$	traffic demand between nodes $i, j \in \mathscr{V}$
Topology information	

Table 3.3 ISP problem Input Parameters

[2]Note that single-path here applies to flows *inside* the ISP, so not being related to the multipath-ness of the overall video traffic at the CP level

Name	Description
$r_l^{ij} \in [0;1]$	a portion of traffic flow between nodes $i, j \in \mathscr{V}$ at link $l \in \mathscr{E}$
$q_l \in \mathbb{R}_{\geq 0}$	congestion metric for link $l \in \mathscr{E}$
$f_l \in \mathbb{R}_{\geq 0}$	traffic flow intensity at link $l \in \mathscr{E}$

Table 3.4 ISP problem Decision variables

$$\min \sum_{l \in \mathscr{E}} q_l \tag{3.2a}$$

s.t. (3.2b)

$$f_l \leq C_l, \qquad \forall l \in \mathscr{E} \tag{3.2c}$$

$$f_l - \sum_{i,j \in \mathscr{V}} x_{ij} \cdot r_l^{ij} = 0, \qquad \forall l \in \mathscr{E} \tag{3.2d}$$

$$f_l \leq q_l, \qquad \forall l \in \mathscr{E} \tag{3.2e}$$

$$3 \cdot f_l - q_l \leq 2/3 \cdot C_l, \qquad \forall l \in \mathscr{E} \tag{3.2f}$$

$$10 \cdot f_l - q_l \leq 16/3 \cdot C_l, \qquad \forall l \in \mathscr{E} \tag{3.2g}$$

$$70 \cdot f_l - q_l \leq 178/3 \cdot C_l, \qquad \forall l \in \mathscr{E} \tag{3.2h}$$

$$500 \cdot f_l - q_l \leq 1468/3 \cdot C_l, \qquad \forall l \in \mathscr{E} \tag{3.2i}$$

$$5000 \cdot f_l - q_l \leq 16318/3 \cdot C_l, \qquad \forall l \in \mathscr{E} \tag{3.2j}$$

$$\sum_{l \in In(v)} r_l^{ij} - \sum_{l \in Out(v)} r_l^{ij} = \begin{cases} +1 \cdot 1_{x_{:v} \geq 0} & \text{if } v = j \\ -1 \cdot 1_{x_{v:} \geq 0} & \text{if } v = i \\ 0 \text{ otherwise} \end{cases}, \quad \forall v, i, j \in \mathscr{V} \tag{3.2k}$$

In Eq. 3.2, constraints are presented as following.

- Eq. 3.2c ensures no flow f_l surpasses the capacity C_l of the link it goes through.
- Eq. 3.2d defines flows f_l through the sum of portions of all flows between every pair of nodes that go over link l.
- Eqs. 3.2e – 3.2j present the linearisation of our congestion function [44, 63].
- Eq. 3.2k is a flow conservation constraint, where 1 is an Indicator function.

We use the cost function from Fortz and Thorup [44], expressing the link congestion as a convex function of the load-to-capacity ratio. This function implements the nature of

congestion in computer networks: it does not show itself until some certain value of link utilisation (commonly considered as 80% to 90%); once this level is surpassed, performance of TCP flows starts to decay very quickly, and augmenting link utilisation over 95% is known to very significantly affect network delays and TCP transmission rates[3].

To mimic the MPLS-based traffic management often implemented by ISP to optimally spread the predicted traffic over pre-established virtual circuits [123], in our study the optimization is performed periodically: once every 30 minutes, all active requests are rerouted so as to decrease total network congestion.

Presented above behaviour models for ISP and CP constitute a relation strategy which we refer to as *non-collaborative no-visibility* starting from this chapter.

The resultant operational behaviour of ISP and CP over the course of serving K video requests can then be summarised in the following algorithm:

3.1.3 Clients

The interest of a client is to maintain the highest possible QoE given the CP's server selection. A client can only stream one video at a time. Clients in this system do not have any direct means of influencing the decisions of the CP and ISPs. Therefore, the role of the client is solely to issue a request to the CP and, by this, bring the load onto ISPs' networks.

3.2 Multipath and caching: balancing between video bitrate and ISP congestion

This section presents the results of a numerical evaluation of how does the MP delivery capability affect the system described in Sec. 3.1. To quantify the effect, we assess the total congestion on all links of both ISPs, average achieved bitrate of client requests (together with their distribution). Even though not explicitly considered neither by ISP nor by CP problems, we also assess the acceptance rate of the those requests by CP. Two demand load scenarios are evaluated: *low load* and *high load* when the demand is lower or higher than the ISP networks can handle, respectively.

Due to a large amount of tunable parameters in the system, some of them had to be set fixed in order to allow us to interpret experimental results. The parameters we fix in this study are:

- Link capacity in the ISP topologies, which is set to 1 Gbps to all links;

[3]https://www.cisco.com/c/en/us/products/collateral/routers/wan-automation-engine/white_paper_c11-728551.html

Algorithm 1

Input: K requests across a certain amount of time; CP bandwidth measurement periodicity $time_{est}^{CP}$; CP bandwidth measurement re-initialisation timeout $time_{reinit}^{CP}$; ISP routing optimisation periodicity $time_{rr}^{ISP}$

for $k \leftarrow$ request out of K until K is empty **do**

CP actions upon receiving the request:

(i) Receives the request, sets its max_rate, min_rate, and determines its destinations d_1 and d_2

(ii) Computes (3.1) and obtains p_s^{CP} for all its sources s

(iii) Establishes multipath connection with the client if selected server is MP-capable

(iv) Starts streaming video content to the client from the source whose $p_s^{CP} = 1$

General CP actions:

if $time_{est}^{CP}$ has elapsed since last bandwidth measurement **then**

for all ongoing video transmissions **do**

(i) Determine its destinations d_1, d_2 and transmission rates x^{s-d_1}, x^{s-d_2}

(ii) $bw\_est_{sd_1} \leftarrow$ EWMA($bw\_est_{sd_1}$, x^{s-d_1})

(iii) $bw\_est_{sd_2} \leftarrow$ EWMA($bw\_est_{sd_2}$, x^{s-d_2})

end for

end if

if $time_{reinit}^{CP}$ has elapsed since last bandwidth measurement re-initialisation **then**

for all $d \in \mathscr{D}$ **do**

(i) $bw\_est_{sd_2} \leftarrow$ Default bandwidth estimation value

end for

end if

General ISP actions:

if $time_{rr}^{ISP}$ has elapsed since last routing optimisation **then**

(i) Computes (3.2) and obtains routing information r_l^{ij}

(ii) Reroutes flows in its network according to r_l^{ij}

end if

end for

- Bandwidth of the client-to-ISP connectivity, set to 20 Mbps for each connection (out of two) of each incoming client
- Amount of different video types, available for request; we limit the study to only one video type, highest-quality representation of which has a bitrate of 50 Mbps. Its lowest quality representation having 5 Mbps bitrate; no service for a request is possible if the CP estimates that the available bandwidth is inferior to the minimum possible rate.
- ISP, according to his behaviour, performs re-routing periodically; we set this interval to 30 minutes.
- We set the CP to prefer selecting a cache instead of a server in case the bandwidth estimation from cache to client is equal or higher than that of server to client.

Other parameters are set varying as following.

- Similarly to *Li et al.* [89], video duration distribution includes considerations regarding users who abandon videos before it ends as well as video popularity, and is presented on Fig. 3.2.
- The video catalog is made of $10,000$ videos following a $Zipf(0.8)$ popularity distribution.
- Clients request videos according to a Poisson process with two arrival rate modes: $\lambda = 0.02$ (requests per second for a single access node, out of 10) for *low load*, and $\lambda = 0.2$ for *high load*. With the simulation time being fixed, the specified load values result in 720 requests for the entire simulation time of about two and a half hours with low load, and 7200 requests overall for high load.

Figure 3.3 shows the average request bitrate and the acceptance rate for low and high load scenarios, while Figure 3.4 demonstrates the total network congestion values for the same scenarios. As expected, running the experiment without MP at low load provides us with near-100% acceptance (due to well-selected load intensity) with an average video bitrate of 20 Mbps - which is exactly equal to the capacity of client-to-ISP connectivity. Enabling the MP for such a case gives the clients an option to use both of their connections, by this raising the achieved average bitrate to roughly 35 Mbps. Such an increase in client demand with unchanged arrival intensity pushes the capabilities of the network to its limit, which results in slightly reduced acceptance rate. Enabling MP also increases the congestion in the network due to the highest-level links being useful with MP.

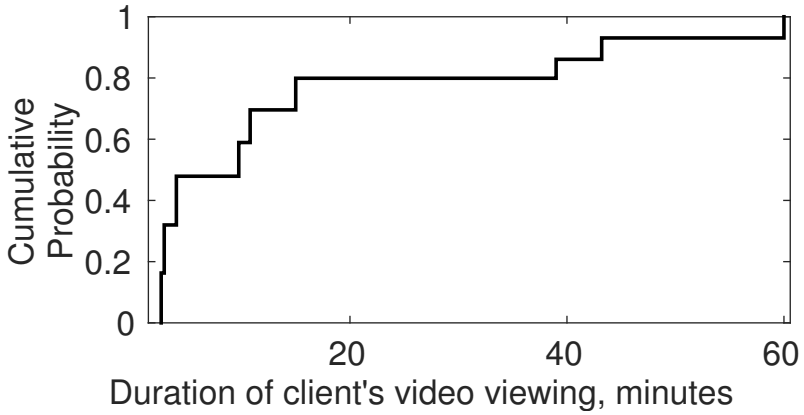


Fig. 3.2 Users' duration of viewing in the incoming workload

In the case of high load the initial client demand is already too large for the network to handle, which results in expectedly low acceptance rate. The achieved video bitrate is still equal to the client-to-ISP connectivity bandwidth: this is due to the ISP link bandwidth being divisible by that value, and since we only have one video type, there is no incentive for the client to receive a different video representation. In contrary to the low load scenario, enabling MP can only bring marginal improvement in terms of video bitrate - which is logical, since the network is already overloaded. It does, however, manage to improve the acceptance rate: the double-connectivity provided by MP essentially allows the client to request another video representation (with the bitrate of 40 Mbps), which incidentally brings more variability to the CP's bandwidth estimations. This allows CP to accept more clients provided that some of them will have lower video bitrate. As for the congestion values, the situation is the same as with low load scenario: MP gives an incentive to use the highest-level links, which consequently results in higher total congestion. It is notable, however, that enabling MP does not incur any additional congestion at the lower topology levels.

Finally, Figures 3.5 and 3.6 demonstrate the statistical properties of video bitrates for accepted requests for both of our load scenarios. The 'plus' points depicted are considered as outliers, but only because their value lie beyond the a certain range from the 25th and 75th percentiles. For the low load case, Figure 3.5 confirms the obtained improvement in the video bitrate once MP is enabled: even though the considered outlier samples are rather numerous (also considering the total amount of requests for low load), the absolute majority of the requests achieve bitrates close to 40 Mbps. Meanwhile, Figure 3.6 also confirms the lack of solid improvement in this regards for the high load scenario; moreover, increased request acceptance results in some isolated requests being served with a much reduced video bitrate, as seen from the outliers on that figure. It should also be noted that, while a no-MP

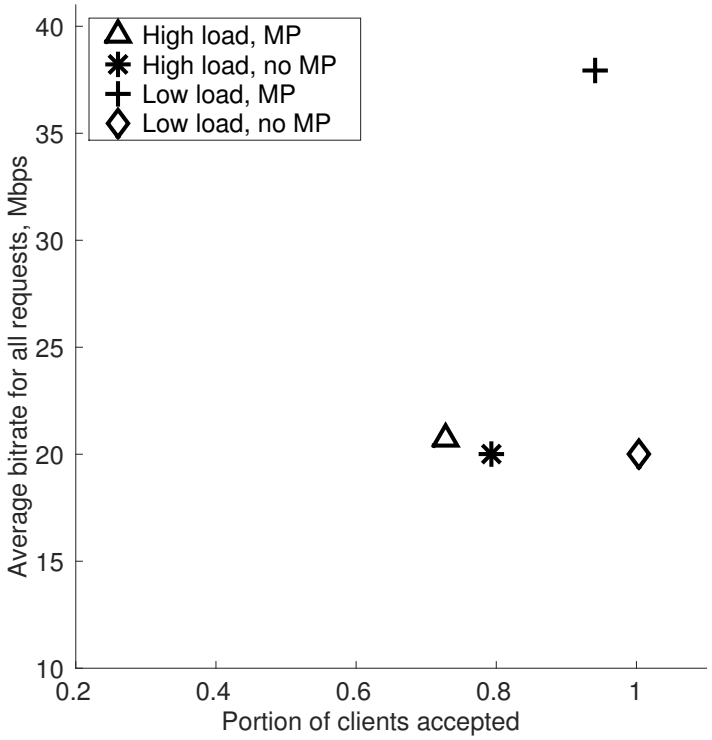
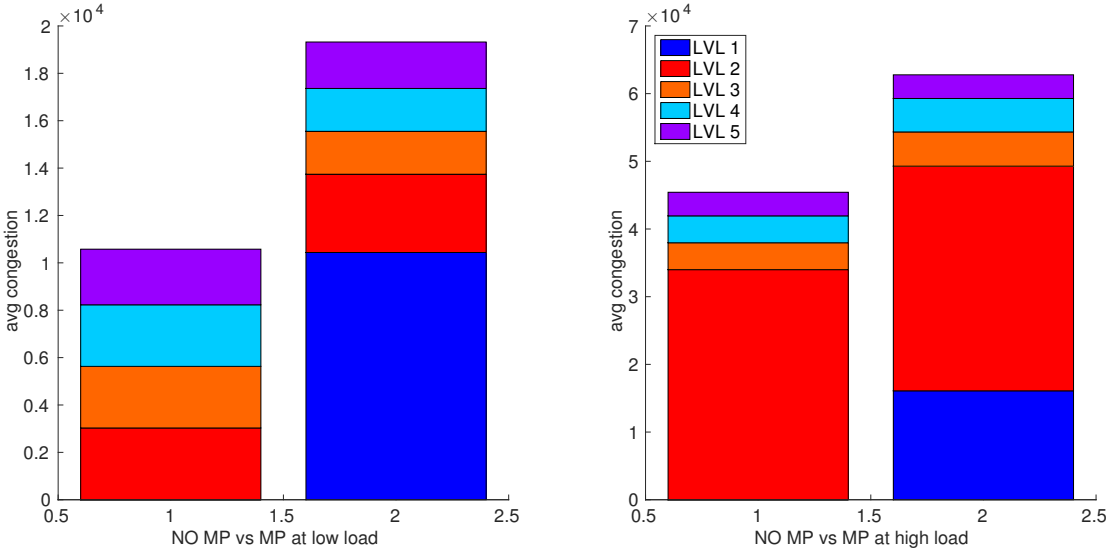


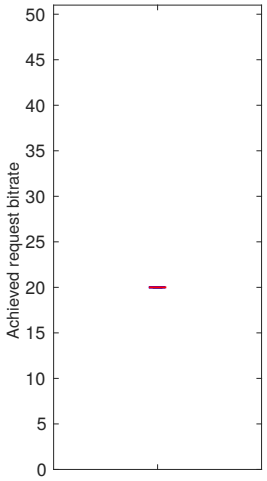
Fig. 3.3 Average request bitrate vs request acceptance rate.



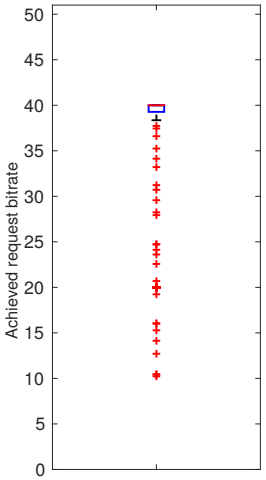
(a) Low load scenario

(b) High load scenario

Fig. 3.4 ISP congestion. Different components of the bars represent the share of total congestion taking place on an indicated topology level (as designated on Fig. 3.1).

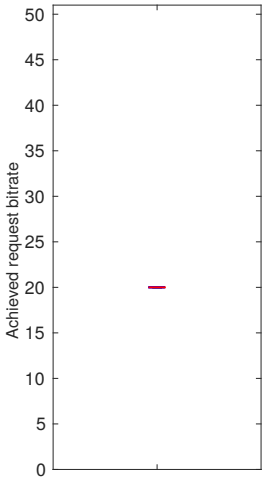


(a) Without MP

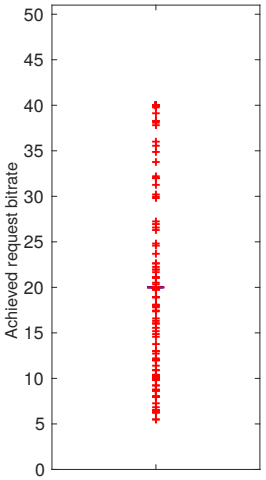


(b) With MP

Fig. 3.5 Box plots for achieved request bitrate at low load with caches.



(a) Without MP



(b) With MP

Fig. 3.6 Box plots for achieved request bitrate at high load with caches.

case at high load demonstrates all (accepted) clients to achieve a bitrate of 20 Mbps (just like the no-MP case of low load), it happens not because high load profile is unable to saturate the network (acceptance figures clearly say it does), but because link capacities in our network (1 Gbps) is divisible by the access link capacity for each client (20 Mbps), so clients gradually fill the ISP network with only slots of 20 Mbps flows.

Overall, we can notice that enabling the multipath for transporting video data in our settings benefits the most for the low-load scenarios. For the high load case, improvement only comes in terms of request acceptance. Regardless of the incoming load profile (that are studied here), MP brings a significant increase in the total network congestion - which is, however, justified by our topology.

3.3 Theoretical limit for collaboration between ISP and CP

We have found that improvements brought by the multipath are also accompanied by a big increase in network congestion. It would be interesting to find whether there is a room to improve the performance of both actors (in terms of their objectives or metrics of interest) with a smarter design of the video distribution system. One could have already noticed

several weak points of the system we have described before; it builds upon several important assumptions.

First, server selection done by CP follows a greedy strategy. Every server allocation happens for the exact incoming request and exact network conditions, which are taken into account by the CP in order to make the best immediate decision. This is quite natural since little is known about the future incoming requests. There exists research on predicting the future traffic characteristics (for instance, [107, 61] but it is out of scope of this thesis.

Next, in order to keep the environment realistic, we have given the CP no visibility over exact traffic situation at the ISPs. The CP is supposed to work around this issues by maintaining estimates of available bandwidths until client locations, which possibly limits the performance compared to having accurate path information. A simple form of collaboration between CP and ISP would be by granting complete visibility of ISP link bandwidth information to the CP (hereby lifting the discussed limitation).

In order to identify whether such extended visibility yields improvements and for which metrics, we have formulated a *Full-visibility* case, presented at Tables 3.5, 3.6 and Equation 3.3. It is equivalent to the no-visibility formulation (Equation 3.1), the only difference being the use of effective link instantaneous bandwidth (obtained from the ISP) instead of CP's own estimation of it. We *do not* consider this model our optimal boundary.

Name	Description
$max\_rate \in \mathbb{R}_{\geq 0}$	maximum bitrate of the requested video
$min\_rate \in \mathbb{R}_{\geq 0}$	minimum bitrate of the requested video
$bw\_eff_{sd} \in \mathbb{R}_{\geq 0}$	Effective value of the available bandwidth between server $s \in \mathscr{S}$ and client $d \in \mathscr{D}$ obtained from the ISP

Table 3.5 Full-Visibility CP problem Input Parameters

Name	Description
$x_{sd} \in \mathbb{R}_{\geq 0}$	video traffic flowing from server $s \in \mathscr{S}$ to client $d \in \mathscr{D}$
$p_s \in \{0;1\}$	binary variable indicating whether server/cache $s \in \mathscr{S}$ is selected

Table 3.6 Full-Visibility CP problem Decision Variables

Full-Visibility model of the CP (3.3a)

$$\max \sum_{\substack{s\in\mathscr{S},\\ d\in\mathscr{D}}} x_{sd} \tag{3.3b}$$

s.t. (3.3c)

$$x_{sd} \leq bw\_eff_{sd} \qquad \forall s \in \mathscr{S}, t \in \mathscr{D} \tag{3.3d}$$

$$\sum_{d} x_{sd} \geq min\_rate \cdot p_s \qquad \forall s \in \mathscr{S} \tag{3.3e}$$

$$\sum_{d} x_{sd} \leq max\_rate \cdot p_s \qquad \forall s \in \mathscr{S} \tag{3.3f}$$

$$\sum_{s} p_s = 1 \tag{3.3g}$$

The final assumption is that we preclude any information exchange between the two parties. Their decisions are made in disregard of the interests and current states of the other actor, which makes them take uninformed decisions that could be suboptimal. In order to assess the potential of joint decision-making to improve actors' metrics of interest when using multipath and in-network caching, we have introduced a perfect joint server and routing selection optimisation problem.

The idea behind joint optimisation problem consists in considering both CP and ISP as a single entity: in such a case, both parties could have exact and complete information about each other in real time, and would be able to decide on both server selections and routing. We can safely consider this scenario as a perfect decision, since interests of both parties are taken into account and are optimised over. Nevertheless, even in such a complete collaboration, the objectives of CP and ISP stay unaligned - i.e., favouring one of them might come at the expense of another. To handle such cases, *Pareto optimality* is commonly considered. In the considered joint decision problem, the objective function would become a weighted sum of both parties' objectives. The weight between the individual objectives would signify the importance of each; in case they are not aligned, solving the optimisation for different values of the weight would result in a so-called *Pareto-optimal solution*, at which no improvement in one individual objective can come without sacrificing another. As an outcome, no particular solution would be strictly better than another, so a Pareto curve - composed of solutions for different weights - is used to visualise the tradeoff between the individual objectives.

Such perfect joint optimisation for our scenario is presented at Equation 3.4. The basic mathematical notations for it are as follows. We consider a network composed of nodes $\mathscr{N}$ that are connected between each other with the help of links/edges $\mathscr{L}$. Nodes in $\mathscr{N}$ are

classified into two groups - sources and clients. Let us denote the root server and caches as $\mathscr{I}$ The input parameters and optimisation variables for the joint problem Eq. 3.4 are specified in Tables 3.7 and 3.8. Here and later in the thesis we refer to this model as *Full-Collaboration*.

Full-Collaboration model: (3.4a)

$$\max(\sum_{d\in\mathscr{D}} x_d - \gamma\cdot\sum_{l\in\mathscr{L}} q_l) \tag{3.4b}$$

s.t. (3.4c)

$$x_{d_1} + x_{d_2} \geq min\_rate \qquad \mathscr{D} = d_1, d_2 \tag{3.4d}$$

$$x_{d_1} + x_{d_2} \leq max\_rate \qquad \mathscr{D} = d_1, d_2 \tag{3.4e}$$

$$x_s \geq min\_rate \cdot p_s \qquad \forall s \in \mathscr{N} \setminus \mathscr{D} \tag{3.4f}$$

$$x_s \leq max\_rate \cdot p_s \qquad \forall s \in \mathscr{N} \setminus \mathscr{D} \tag{3.4g}$$

$$\sum_{s\in\mathscr{N}} p_s = 1 \tag{3.4h}$$

$$p_s = 0 \qquad \forall s \in \mathscr{N} \setminus \mathscr{I} \tag{3.4i}$$

$$f_l \leq C_l, \qquad \forall l \in \mathscr{L} \tag{3.4j}$$

$$f_l - \sum_{s\in\mathscr{N}} x_l^s = flowinuse_l, \qquad \forall l \in \mathscr{L} \tag{3.4k}$$

$$f_l \leq q_l, \qquad \forall l \in \mathscr{L} \tag{3.4l}$$

$$3\cdot f_l - q_l \leq 2/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{3.4m}$$

$$10\cdot f_l - q_l \leq 16/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{3.4n}$$

$$70\cdot f_l - q_l \leq 178/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{3.4o}$$

$$500\cdot f_l - q_l \leq 1468/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{3.4p}$$

$$5000\cdot f_l - q_l \leq 16318/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{3.4q}$$

$$\sum_{l\in In(i)} x_l^s - \sum_{l\in Out(i)} x_l^s = \begin{cases} +x_i & \text{if } i \in \mathscr{D} \\ -x_i & \text{otherwise} \end{cases}, \qquad \forall s, i \in \mathscr{N} \tag{3.4r}$$

As established before, the formulation at Equation 3.3 is a combination of two separate problems from previous section, with a slight change in the variables definition. The main change is the introduction of a joint decision variable x_l^s that aims to incorporate the decision process of both objective functions. An important detail one could note from the formulation is that the problem is still greedy, as it optimises only for the next incoming request. This is done in order to keep the execution time reasonably short, while the gains of jointly solving

Name	Description
$max\_rate \in \mathbb{R}_{\geq 0}$	maximum bitrate of the requested video
$min\_rate \in \mathbb{R}_{\geq 0}$	minimum bitrate of the requested video
$flowinuse_l \in \mathbb{R}_{\geq 0}$	currently active flows at link $l \in \mathscr{L}$
$C_l \in \mathbb{R}_{\geq 0}$	Capacity of link $l \in \mathscr{L}$

Table 3.7 Perfect joint optimisation Input Parameters

Name	Description
$x_l^s \in \mathbb{R}_{\geq 0}$	video traffic coming from the node $s \in \mathscr{N}$ that flows over link $l \in \mathscr{L}$
$x_s \in \mathbb{R}_{\geq 0}$	video traffic coming from the node $s \in \mathscr{N}$
$q_l \in \mathbb{R}_{\geq 0}$	congestion metric for link $l \in \mathscr{L}$
$f_l \in \mathbb{R}_{\geq 0}$	traffic flow intensity at link $l \in \mathscr{L}$
$p_s \in \{0;1\}$	binary variable indicating whether node $s \in \mathscr{N}$ is selected

Table 3.8 Perfect joint optimisation Decision Variables

the server selection and routing problems are studied in the domain where we determine the greediness of our formulation does not significantly impact the results.

Next section presents an evaluation of the models discussed above. This evaluation aims to provide answers to the following two questions:

- Can we expect better multipath performance for our metrics of interest when CP has complete visibility over the ISP network information?
- How far the performance of *non-cooperative no-visibility* and *Full-visibility* models is from the theoretical optimal boundary, which is established by the Pareto curve of the perfect joint optimal solutions (Eq. 3.4)?

3.4 Performance of today's video delivery system against theoretical boundary

Figures that are presented in this section are of two kinds: Pareto curves, considering time averages, and time series, allowing to analyze the underlying phenomena of our model when interacting with the request arrivals through our discrete event simulator. Several metrics are considered for each.

We will show three types of time average (Pareto) figures, all of them being plotted against the average total network congestion. The X-axis will contain the following metrics:

- **Achieved served rate**: instantaneous obtained rate averaged over all active requests, and averaged over time. These figures will be presented with confidence intervals, which are acquired after running the same experiment five times over different random number generator init number.

- **Maxfrac**: average fraction of requests obtaining the maximum rate (taking into account available access link bandwidth that can be lower than the maximum video bitrate).

- **Relfrac**: average fraction of requests obtaining a rate higher than without multipath subtracted with the fraction of requests obtaining a rate lower than without multipath.

We run five repetitions of each experiment with different random generator seed, which impacts the generation of the input workload trace. All depicted points at time average figures for ISP – CP objectives tradeoff are the average values of all five results, with 0.95 confidence intervals in both dimensions. The theoretical boundary is, however, obtained for only one sample of those repetitions for the sake of clarity.

Mentioned metrics will be presented for the theoretical performance boundary (also referred to as Full-collaboration), so will be the following relation strategies between ISP and CP: non-collaborative no-visibility, with and without multipath (CP has no visibility over exact path bandwidths) and non-collaborative Full-visibility (CP has complete and perfect visibility over path bandwidths). The Full-collaboration front is constructed from different values of weight γ; these values are noted besides the resulting curve points.

In the time series, we will show the following five metrics at each figure (except for Fig. 3.7). Every time series figure will be plotted from a single sample of multiple experiment runs.

- **Average achieved request bitrate**: in our system, every request either gets assigned a feasible bitrate, or is rejected and is assigned zero bitrate. At each request arrival, the average rate obtained by all currently active requests is logged, then a running average of them within a moving window of 30 requests is plotted in the top figure of the time series.

- **Number of active requests**: running average of amount of currently active requests in the system. In the second row of time series figures.

- **Running average of the share of clients served by the outside multipath server**. In the third row of time series figures.

- **Total congestion**: running average of value of the sum of d_l, for all $l \in \mathscr{L}$, for both ISPs. In the fourth row of the time series figures.

- **Requests rejects**: running average of rejects (i.e., if CP deems the request's video minimal bitrate cannot be satisfied).[4] In the fifth row of the time series figures.

Environment settings are the same as in Section 3.2, namely: partially-meshed tree network topology, 1 Gbps link capacity, 20 Mbps access link capacity for every request, one available video class with bitrates of lowest and highest quality representations being 5 and 50 Mbps correspondingly, 30 Minutes re-routing period, client arrival intensity is $\lambda = 0.2$ (we do not consider low load arrival profile in this section).

3.4.1 Impact of choosing a greedy formulation

Fig. 3.7 illustrates the limitation of the simplified formulation by comparing the performance of Full-collaboration (green lines) with non-collaborative no-visibility no-MP case (blues lines). This figure shows the following metrics, row by row:

- Running average of achieved video rate;
- Running average of normalised number of currently served requests;
- Instantaneous share of clients served by the outside multipath server;
- Running average of total congestion.

The first and last rows of the figure show that Full-collaboration does not achieve a higher obtained rate on the long run, but yields more congestion. This is due to the interplay between the blindness of the chosen formulation of Full-collaboration with the traffic level to handle. In this case, the requests durations are 4, 15 and 60 minutes. In the shown period, the number of active requests does not reach steady state, the current congestion information Full-collaboration relies on is therefore inaccurate for the future, and therefore takes a wrong decision, that is decision which is not going to yield higher rate but costs higher congestion.

We acknowledge this is the limitation of our Full-collaboration modelling, and as a result, investigate its predictive power regarding the collaboration gains in its validity domain only. Let us mention that a study of the interest of collaboration depending on the traffic intensity would be possible with a more complex Full-collaboration problem considering re-allocation

[4] In this analysis, by reject we only mean a definitive reject from the CP, if it estimates that no source can support a minimum required video bitrate for the request. In case if CP, for example, wrongly (by having inaccurate bandwidth estimations) decides that some source can serve a request, whereas ISP (with his complete view of network state) happens not to be able to support the minimal bitrate of requested video, such a request will be considered accepted in our experimental analysis

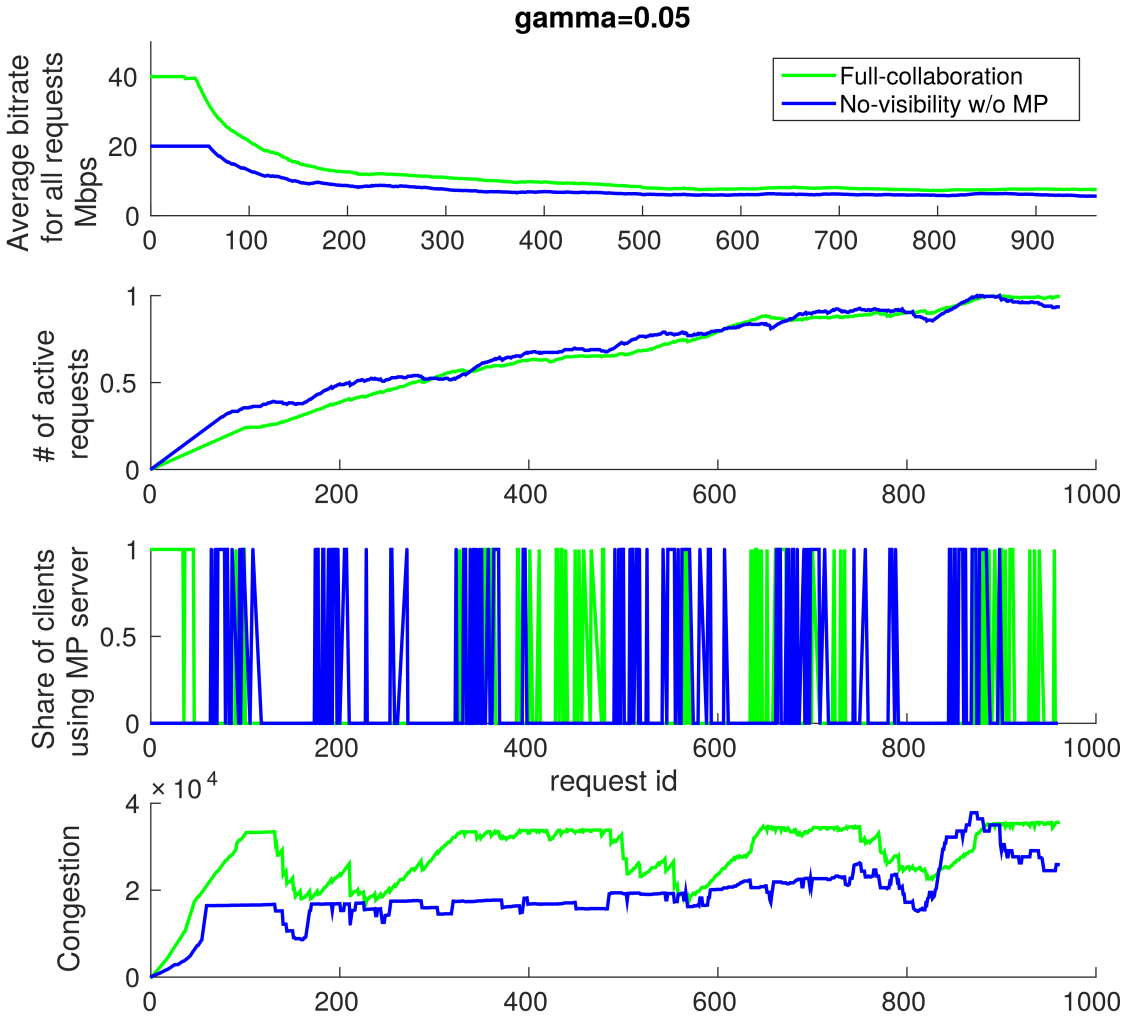


Fig. 3.7 Time series of studied system metrics for requests of 4, 15 and 60 minutes minutes long.

of several requests at the same time to avoid the shortcomings of greedy assignment in high traffic intensity scenario.

We, therefore, verify next that the validity domain corresponds to reduced traffic intensity where the decision of Full-collaboration consistently yields better results than non-collaborative no-visibility. Such settings are obtained for shorter requests, more precisely, of 2 minutes long – as illustrated in Fig. 3.8.

3.4.2 Observations on instantaneous behaviour of considered relation strategies

Fig. 3.8 shows that Full-collaboration is able to sustain similar or higher rate than with non-collaborative no-visibility, without increasing the total congestion. The congestion graph shown in the fourth row illustrates a drawback of the non-collaborative no-visibility case: the CP bases its server assignment decision only on the data it can collect. We assume it collects passive measurements by running an EWMA of the rate obtained on each source-destination pair and serving as $bwest_{sd}$. However, in order to react to traffic intensity decrease and allow to select again higher-hierarchy servers, the estimates need to be re-initialized periodically (here we set the initialization value of $bwest_{sd}$ to the max request rate 50 Mbps and the re-initialization period set to 200s). This phenomenon is seen in the third row of Fig. 3.8. Consequently, the congestion periodically increases with non-collaborative no-visibility, as the re-initialization does not necessarily correspond to the correct available bandwidth.

Perfect estimation is represented by the Full-visibility case, where the CP knows the exact traffic going through each link at each point in time. In this case, the obtained rate is as good as in the previous case, but the congestion is higher, as not considered by the CP but for its a posteriori impact on the available bandwidth. In particular, we see from Fig. 3.8 that Full-collaboration (for $\gamma = 0.07$ here) is able to obtain a similar rate at a cost of half the average congestion entailed by non-collaborative no-visibility case or third that entailed by Full-visibility.

The fifth row of Fig. 3.8 shows that while Full-collaboration yields no rejected requests, the non-collaborative no-visibility sometimes have rejects but Full-visibility, by greedily filling up the pipes, have a consistent fraction of rejected requests. The third row of Fig. 3.8 represents the fraction of requests assigned to the top server (that allowing MP). We observe that, while Full-collaboration maintains this fraction between 0 and 0.5, non-collaborative no-visibility spikes to 1 at the bwest re-initialization times, and Full-visibility maintains this fraction close to one and constantly higher that 0.7. These levels explain the levels of congestion observed in the fourth row.

Let us comment on the behaviour of non-collaborative no-visibility. First, we observe that the re-routing period does not impact the results. This is due to the first routing where a traffic matrix with a flow from each server to each destination exists yields, on this specific partially-meshed tree topology, to exploit all the paths. Second, the re-initialization period for $bwest_{sd}$ and the choice of the EWMA parameter are important. The EWMA parameter is and set to 0.9 (0 means $bwest_{sd}$ remains at the initialization value and does not get updated, while 1 means $bwest_{sd}$ is set to the last sample value). The re-initialization is set to 200s.

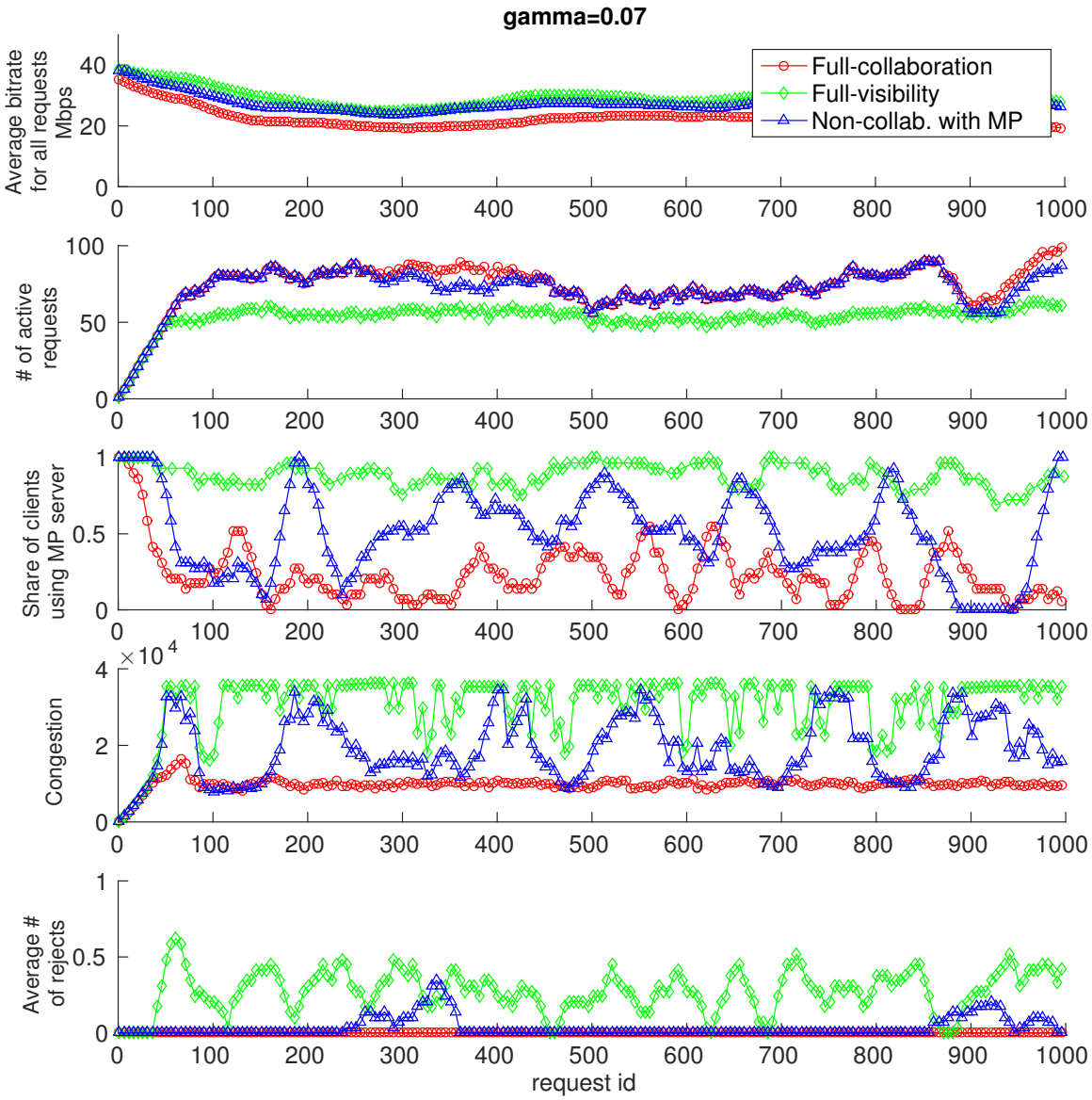


Fig. 3.8 Time series of studied system metrics for requests of 2 minutes long. All CP – ISP relation strategies are considered with multipath enabled

However, these parameters are difficult to fix or even learn in real systems, and this is where collaboration can be helpful for the CP.

3.4.3 Performance of Full-collaboration Pareto front as compared to other relation strategies

Figure 3.9 represents the served rate-congestion tradeoff, where, for the reasons discussed at the beginning of the section, the optimal is considered to be Full-collaboration, hence referred

to as the Pareto curve. As expected in this representation, we find again that Full-collaboration allows to save a significant fraction of network congestion without sacrificing served rate. For example, point $\gamma = 0.04$ shows that the same average rate (about 26 Mbps) can be obtained with as much as 50% less congestion than Full-visibility with MP. For the same point $\gamma = 0.04$ allows to achieve the same average served rate than with non-collaborative no-visibility with MP, with about 25% less congestion.

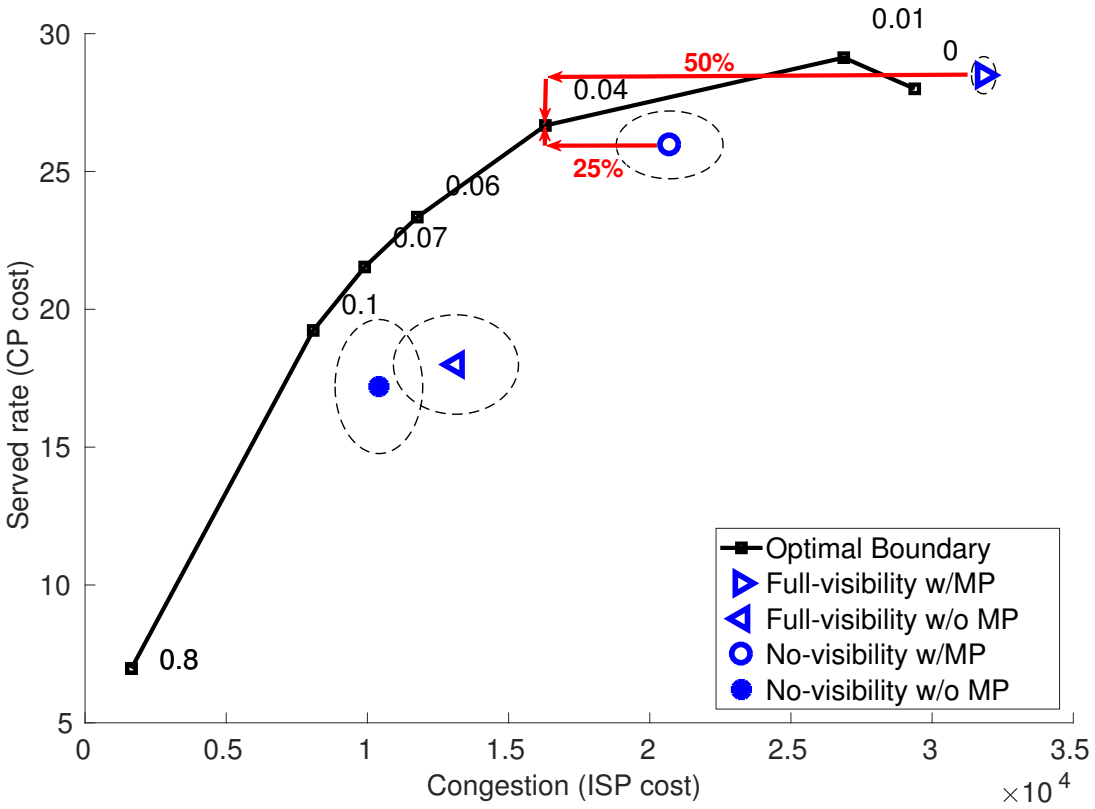


Fig. 3.9 Performance of different strategies as compared to themselves and the theoretical boundary (Full-collaboration) at base settings, 2 minutes long requests. Confidence intervals in both dimensions are shown as ellipses around respective points

If the Full-visibility model here does not exhibit a clear advantage for the CP for any of the considered metrics (except a marginal advantage over non-collaborative no-visibility for the served rate), [45] has shown the interest of a better visibility into the ISP state when more refined model of request prediction are considered.

Finally, two other metrics are represented in Fig. 3.10 and 3.11. Comparing the non-collaborative no-visibility w/MP with Full-collaboration with $\gamma = 0.06$, Fig. 3.10 shows that over 50% congestion can be saved still ensuring that about 35% of the requests will benefit

from the maximum rate of 40Mbps. As well, Fig. 3.11 shows that a net rate of about 30% of requests benefit from MP, achieving a higher rate then with low-level caches only.

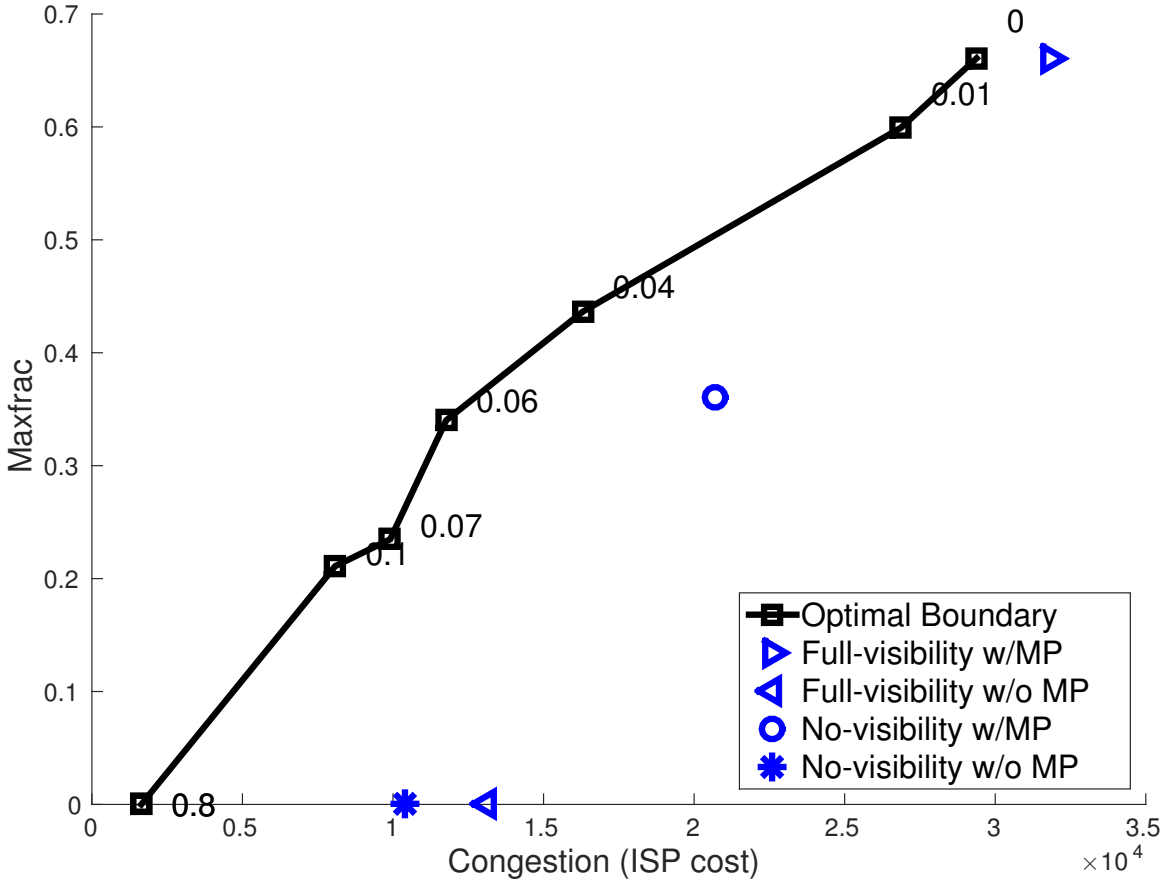


Fig. 3.10 Average fraction of requests obtaining maximum video rate, for different strategies (taking into account available access link bandwidth), 2 minutes long requests

3.5 Conclusion

Motivated by the foreseen increase in video traffic and more bandwidth-demanding immersive media applications, allowing a user to take advantage of several interfaces at the same time appears as a viable and cheap solution. In this chapter, we have assessed the potential benefit of such a solution for both main actors of a video delivery system – Internet Service Provider and Content Provider. Most notably, we have demonstrated that under low-load profile (i.e., when our network is well dimensioned to accept all incoming video requests), multipath increases average served bitrate of videos by up to 50% while accepting slightly lower amount of video requests. At the same time, this does not come for free: in our

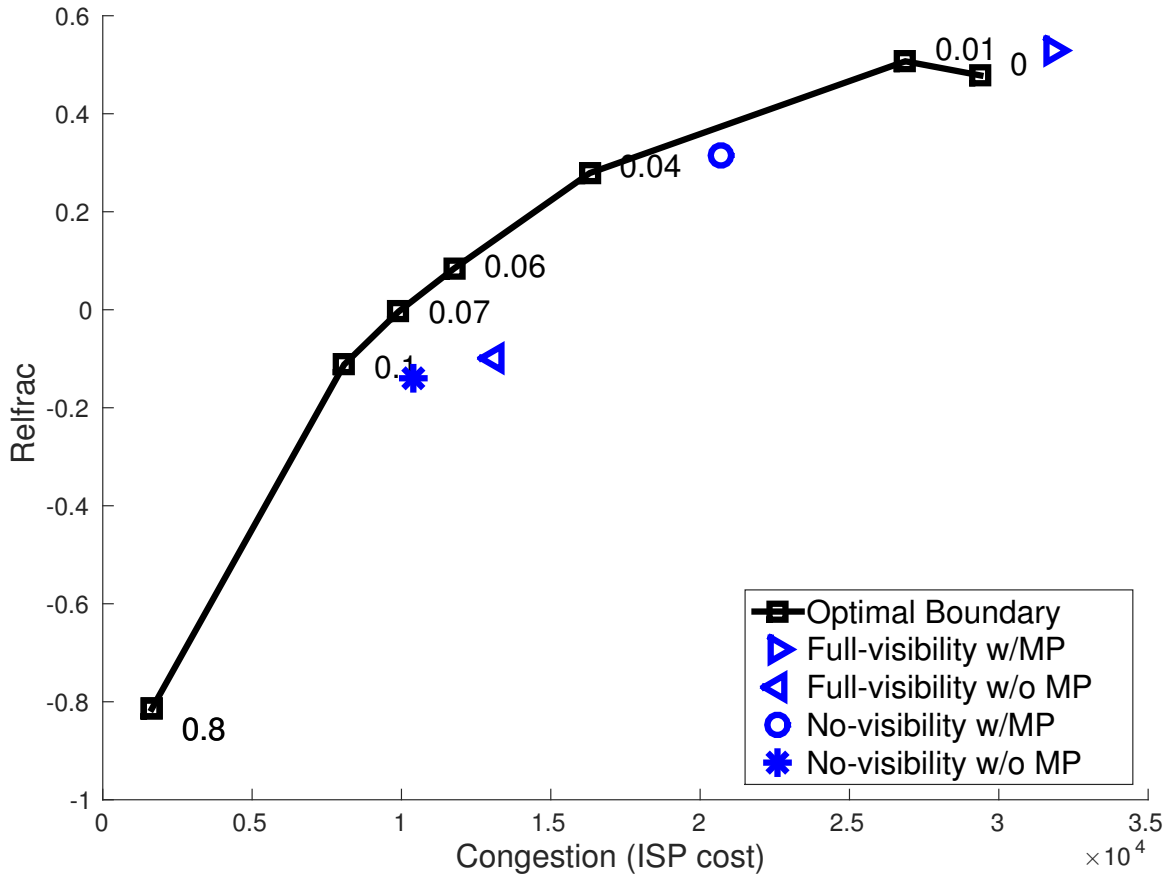


Fig. 3.11 Average fraction of requests obtaining video rate higher than without multipath minus fraction of requests obtaining a rate lower than without multipath, for different strategies, 2 minute long requests

partially-connected meshed tree topology, multipath flows have to pass through upper part of the backhaul network, which would have been avoided if in-network caches were used to source the video. As a result, using multipath puts a considerable strain onto the upper part of topology, which is a rather negative point: this part of a tree-type topology provides connectivity of the ISP to the outside Internet, so excess congestion at those links would inevitably mean QoS/QoE degradation for external services.

Such a multipath extension is, however, not straightforward, owing to the reliance of the current content delivery architecture on in-network caches. If the latter initially preclude the use of MP, we show the interest for the CDN and client to make an informed decision on which server to select. Depending on the network state, the ISP may hence have an interest in impacting the server selection decision. These ideas, most probably, require a certain degree of collaboration between the actors.

Curious to discover whether such a collaboration can make sense from a standpoint of both actors' performance (in terms of their objectives), we have formulated a perfect theoretical boundary for a full collaboration between ISP and CP. Numerical evaluation of this boundary demonstrated a significant room for improvement of non-collaborative MP (about 33%) and no-MP (over 40%) strategies, in terms of ISP congestion and for a workload with several minute long requests durations. Such a performance boundary being theoretical, we are now interested in designing a realistic collaborative strategy. The design of such cooperation mechanism is the objective of the next chapter.

Chapter 4

Collaboration schemes between CDN and ISP for better quality-congestion tradeoffs

Previous chapter has demonstrated the effect of enabling multipath data transport in a video delivery system that also makes use of caches. We have demonstrated that its benefit in general can be limited by the lack of interoperation between the Content Provider and Internet Service Providers. Such a conclusion is, however, claimed basing on the performance of a perfect fully-collaborative optimisation problem, which essentially wipes the frontier between both actors and refers to them as a single entity.

Such an idealised case is certainly possible in theory, especially considering the recent trend of the large national ISPs to also enter the market of content production and provision ([87]). Nevertheless, many CPs today are acting independently from ISPs and at most have their own CDNs to improve accessibility of their services, so the problem of collaboration calls for a scheme more feasible, than presented in previous chapter, yet as efficient. Any kind of technological cooperation in the Internet is hard to achieve, so, generally the less cooperation is entailed, the more realistic it is to implement. On the other hand, demonstrated benefits and recent reports on technological enablers for the cooperation in video delivery (e.g., [60]) could give both actors a solid incentive to work in that direction. This chapter is, therefore, challenging the task of designing such a mutually-interesting collaboration.

In this endeavour we source our inspiration from the work of Jiang et al. in [63] that has been referenced in the previous chapter. In the same way as they do, we apply Nash bargaining in the design of the collaboration in order to ensure optimality and fairness of a resultant strategy. The contributions of this chapter are the following:

- Acknowledging the intrinsic difficulty in establishing collaboration between different business-entities in the Internet, we design two collaboration schemes that involve a different amount of interaction and test them against the perfect collaboration;
- We discuss the real-world implementability of those collaboration schemes.

Game theory (and Nash bargaining in particular) has been quite a popular approach in studying cooperation in the Internet, especially for large-scale systems with competing entities. DiPalantino and Johari [39] has presented a work very close to [63]: they model an interaction of a traffic engineering problem, representing an ISP, and a user problem, that selects a server for its content. They also conclude that there exists a mutually-beneficial optimum for both parties' performance. A part of this work is devoted to multi-ISP generalisation of their interaction; they consider the ISPs to be in charge of their intra-domain routing, while users would implicitly decide the intra-domain traffic allocation. Over the course of their analysis, authors conclude that even though an equilibrium exists for such a system, in practice it might not hold due to selfishness of ISPs and their subsequent price inflations. In the previous chapter we have dealt with two ISPs that implement multipath connectivity; in these chapter, however, we will consider both ISP network being governed by the same entity for the sake of simplicity in formulations. We estimate, however, that our assumption of ISPs willing to accept as much traffic as economically possible should not also entail simplifications in competition behaviour.

Lee et al. [86] use game-theoretic approach to tackle the interplay between multiple ISPs of different tiers, with a focus, among other, on revenue maximisation and resource distribution. As a contribution, they provide a distributed algorithm for resource allocations with quick convergence and a model to estimate revenues. Shrimali et al. [145] offer a work on designing an inter-domain traffic engineering protocol for a set of ISPs that share some flows between them. They apply Nash bargaining so as to maximise the utility of all their ISPs while maintaining fairness. Findings of both cited works in this paragraph demonstrate benefits from collaboration in their respective assumptions and objectives.

Quite naturally, game theory and Nash bargaining can be used to model economic interaction of Internet actors. Kibilda et al. [73] demonstrate a case of virtual network providers that seek to either build their own infrastructure or paying mobile network operators to host their services. Authors develop a bargaining problem that incorporates, from one side, revenue interests of the physical carrier, and from other side – financial limitations of the virtual carrier on renting the infrastructure, given the anticipated volume of infrastructure resources. An interesting finding is that in case if virtual carrier's service areas have a high

degree of spatial clustering, building his own infrastructure becomes more attractive thus harming the ability of the physical carrier to set higher prices for his hosting services.

Relatively few works shed light upon the economics of QoE, which can be regarded as a form of regulation of client's demand (which could be useful for cooperation in the Internet). Though cooperation between service and content providers is the central matter of this chapter, we do not specifically assess how to find an economically viable tradeoff between both actors; the following works could, however, examplify methods of achieving this.

Reichl et al. [131] propose fixed-point modelling of the willingness of a video consumer to pay for different QoE levels in a video use-case. In their environment, client demand for video services drives the QoE price changes, which, in turn, regulates that demand. They prove that a convergence towards a fixed point in such a model exists, and empirically validate their theoretical work. Work [100] of Mäki et al. goes further in their analysis of the willingness-to-pay: they, notably, investigate how is relation of the former to QoE affected by tariff changes, as well as by cultural differences.

Relation of QoE and service pricing can also be studied by coupling it through user churn. Ahmad et al. [5] present a model of collaboration between over-the-top (OTT) content providers and ISPs that is focused on maximising their revenues that naturally incorporates users' QoE and their churn due to dissatisfaction. Their results demonstrate that in their settings with a service with two quality levels, lack of collaboration between OTT CP and ISP leads to insufficient QoE provision for premium users that results in total revenue drop, while their form of collaboration alleviates this issue. Basing on this, they conclude that collaboration is useful for revenue maximisation in QoE-based service delivery. Though their collaboration is centralised and they do not proceed with its decoupling, they argue that collaboration between different entities is difficult to achieve in reality due to the lack of standard cooperation interfaces between them.

Work of Heegaard et al. [50] demonstrates how increased amount of information sharing can benefit revenue generation of multiple operators. They develop a game-theoretical base for revenue performance of actors by defining the Service Level Agreements (SLAs) for QoS guarantees and penalties for their nonfulfillment. With several levels designating the amount of information shared between the parties, they show that there exists an optimal level of sharing that minimises under- or overestimation of SLA non-fulfillment risks, thus improving operators' operational efficiency and, hence, increasing revenues.

The chapter is organised in the following way. Section 4.1 presents the design for ISP-CP collaboration. Section 4.2 shows the experimental settings and demonstrates experimental

results for the proposed collaboration schemes. Section 4.3 discusses those obtained results to evaluate the benefit of the collaborative schemes over current common practice.

4.1 Design of collaboration schemes between ISP and CP

We intend to design ISP-CP collaboration models to achieve two goals: (i) active leverage over congestion at the ISP, and (ii) better CP estimates of the ISP state. Instrumenting the ISP with (i) means that ISP would be able to express his interests in the process of video delivery. Obtaining (ii) for the CP would mean estimating faster and more accurately the traffic situation at the ISP, which will allow CP to take informed decisions as in compliance with the ISP network state and its interests.

These two features constitute the collaboration schemes we are going to design in this section.

Ahead of presenting our design, it is important to note that in this chapter we consider the client to be connected to two separate networks which are, however, managed by a single ISP entity - as opposed to the previous chapter, where two separate networks were also managed by different entities. We consider that ISPs are not willing to offload offered traffic to another ISP unless they are severely congested, so we do not expect our decision to affect the accuracy of competition modelling (as compared to a proper two-ISP case). As we will see later, collaboration requires information exchange between the parties, so such an assumption greatly decreases the bulkiness of our formulations while keeping our considerations intact and valid for either case.

4.1.1 Giving the ISP power to refuse CP's subflow establishment choices

Generally, increase in the traffic volume for a service means more clients for it, so it should normally be regarded as a positive thing. Consequently, the job of an ISP is to fulfil the demand for a service as it brings benefit for both ISP and its services (a CP in our case). As we have seen in the previous chapter, enabling multipath incurs extra load at the ISP networks regardless of the settings in our study. As it happens, not always such an increase in traffic demand brings clearly better service performance (one can refer to the high load case at Fig. 3.3 in terms of video bitrate). Besides, even when the service actually achieves a clear improvement in its performance by increasing network demand, ISPs are providing their resources to a multitude of services; it is, therefore, the first and foremost interest of the ISP to keeping their network free enough in order to guarantee that every services is satisfied.

In other words, for our hypothetical environment, ISPs are not likely to be interested in unmoderated increase in CP's video traffic demand that is incurred by use of multipath.

These considerations lie in the foundation for the first collaboration scheme, which is supposed to be "light" on the amount of exchanged information. The idea behind it is to grant the ISP a leverage in moderating the CP's video traffic bitrate (by cutting MPTCP subflows). Upon receiving a video request that is destined to the CP, ISP performs a server selection for it considering the exact traffic information in its own network. In case it evaluates that the gain in video request bitrate when using multipath for this request will not be significant enough with respect to the potential increase in congestion level, it will prefer the low-level caches to be selected by the CP for request servicing. At the same time, this estimations will not be shared with the CP, so should it select a multipath server (in contrary to ISP's preference), one of its multipath connections will be "cut" [112] by the ISP - which will impact the request's served bitrate. Due to the CP maintaining EWMAs of connection bandwidths until clients' premises, such MP "cuts" will be implicitly reported back to the CP, and it will consider them as a lack of bandwidth at the network. Hence, next time CP will be more moderate in its request management.

It can be noted that such a traffic load management in the ISP can also modulate the actual flow bitrate of requests, and not only the server selection. This study is, nonetheless, limited to the leverage over the server selection only as to assess the interaction between multipath and caching.

The benefit of this collaboration scheme is two-fold: it attempts to introduce some sort of fairness in operation between ISP and CP (as before, in problem formulations, we considered the ISP to have no word in CP's decision making), meanwhile, it does not require any explicit information exchange between both. We discuss the caveats of real-life implementation of such a scheme in the end of the section. In this study, we refer to this proposed collaboration scheme as "Collaboration-light".

In this collaboration, CP is solving a non-collaborative no-visibility problem, formulated in Equation 3.1.

Since ISP has to perform a server selection with extended network visibility, we can formulate its optimisation problem in the same way as the perfect collaboration (Equation 3.4), as they optimise for essentially the same task. The formulation is expressed in Equation 4.1, while input parameters and variables to this model are explained in Tables 4.1 – 4.2. Naming conventions are those of Eq. 3.4.

Collaboration-light for ISP: (4.1a)

$$\max\Big(\sum_{d\in\mathscr{D}} x_d - \gamma\cdot\sum_{l\in\mathscr{L}} q_l\Big) \tag{4.1b}$$

s.t. (4.1c)

$$x_{d_1} + x_{d_2} \geq min\_rate \qquad \mathscr{D} = d_1, d_2 \tag{4.1d}$$

$$x_{d_1} + x_{d_2} \leq max\_rate \qquad \mathscr{D} = d_1, d_2 \tag{4.1e}$$

$$x_s \geq min\_rate \cdot p_s \qquad \forall s \in \mathscr{N} \setminus \mathscr{D} \tag{4.1f}$$

$$x_s \leq max\_rate \cdot p_s \qquad \forall s \in \mathscr{N} \setminus \mathscr{D} \tag{4.1g}$$

$$\sum_{s\in\mathscr{N}} p_s = 1 \tag{4.1h}$$

$$p_s = 0 \qquad \forall s \in \mathscr{N} \setminus \mathscr{I} \tag{4.1i}$$

$$f_l \leq C_l, \qquad \forall l \in \mathscr{L} \tag{4.1j}$$

$$f_l - \sum_{s\in\mathscr{N}} x_l^s = flowinuse_l, \qquad \forall l \in \mathscr{L} \tag{4.1k}$$

$$f_l \leq q_l, \qquad \forall l \in \mathscr{L} \tag{4.1l}$$

$$3\cdot f_l - q_l \leq 2/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.1m}$$

$$10\cdot f_l - q_l \leq 16/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.1n}$$

$$70\cdot f_l - q_l \leq 178/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.1o}$$

$$500\cdot f_l - q_l \leq 1468/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.1p}$$

$$5000\cdot f_l - q_l \leq 16318/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.1q}$$

$$\sum_{l\in In(i)} x_l^s - \sum_{l\in Out(i)} x_l^s = \begin{cases} +x_i & \text{if } i \in \mathscr{D} \\ -x_i & \text{otherwise} \end{cases}, \qquad \forall s, i \in \mathscr{N} \tag{4.1r}$$

Name	Description
$max\_rate \in \mathbb{R}_{\geq 0}$	maximum bitrate of the requested video
$min\_rate \in \mathbb{R}_{\geq 0}$	minimum bitrate of the requested video
$flowinuse_l \in \mathbb{R}_{\geq 0}$	currently active flows at link $l \in \mathscr{L}$
$C_l \in \mathbb{R}_{\geq 0}$	Capacity of link $l \in \mathscr{L}$

Table 4.1 Collaboration-light Input Parameters

The difference between Full-collaboration and the presented Collaboration-light would be in how to handle this optimisation problem. In case of perfect collaboration, the result of

Name	Description
$x_l^s \in \mathbb{R}_{\geq 0}$	video traffic coming from the node $s \in \mathscr{N}$ that flows over link $l \in \mathscr{L}$
$x_s \in \mathbb{R}_{\geq 0}$	video traffic coming from the node $s \in \mathscr{N}$
$q_l \in \mathbb{R}_{\geq 0}$	congestion metric for link $l \in \mathscr{L}$
$f_l \in \mathbb{R}_{\geq 0}$	traffic flow intensity at link $l \in \mathscr{L}$
$p_s \in \{0;1\}$	binary variable indicating whether node $s \in \mathscr{N}$ is selected

Table 4.2 Collaboration-light Decision Variables

optimisation was applied to both CP and ISP as it is supposed that they act together; for the discussed "light" collaboration scheme, only the ISP uses the outcome of the optimisation, while the CP does its own server selection. Taking this difference into account, Algorithm 2 summarises the processing for **every** incoming request within Collaboration-light strategy (on top of the General ISP and CP actions, as outlined in Algorithm 1).

Algorithm 2 Processing an incoming request in Collaboration-light

Input: incoming request with max_rate and min_rate; utilisation of links C_l at ISP; CP's bandwidth estimations bw_est_{sd}

(I) ISP actions:

(i) Receives the request to be transported to CP and determines its max_rate, min_rate, and destinations d_1 and d_2

(ii) Computes 4.1 and obtains p_s^{ISP} for all sources s

(II) CP actions:

(i) Receives the request, sets its max_rate, min_rate, and determines its destinations d_1 and d_2

(ii) Computes 3.1 and obtains p_s^{CP} for all its sources s

(iii) Starts streaming video content to the client from the source which $p_s^{CP} = 1$

(III) Further ISP actions:

(i) Receives traffic for the request and determines its source $s \in \mathscr{N}$ and its bitrate x_s

if s is the outside MP server **and** $p_s^{ISP} \cdot x_s == 0$ for newly determined s **then**

(a) Cut the second MP subflow

else

(b) Transport the content traffic in its entirety

end if

4.1.2 Giving the CP visibility over ISP decisions: an active collaboration scheme

While the Collaboration-light has a benefit of not requiring any information share, it does not in fact establish a meaningful collaboration between ISP and CP. The perfect collaboration presented before did promise improvements for their metrics of interest, but this was achieved due to both parties collaborating to the most possible extent. The ultimate goal of this chapter is to attain the performance of the perfect collaboration, and this will definitely require a certain extent of information sharing; we aim at making the amount of information shared as small as possible.

The idea behind such a collaboration is the following. We consider the perfect collaboration scheme as a base, but artificially let the ISP and CP make their own decisions – however, the variables they are deciding on belong to the same decision space. Let us denote the variable representing a video request flow as x_{sd} (note that it can have one or two subflows, depending on the type of the source node). The considered problem will then look the following way:

$$\max \quad (\sum_{s,d \in \mathscr{V}} x_{sd} - \gamma \cdot \sum_{l \in \mathscr{L}} q_l) + \mu \cdot \sum_{s,d \in \mathscr{V}} x_{sd} \tag{4.2a}$$

s.t. (4.2b)

server unicity constraint: $\sum p_s = 1$; (4.2c)

client video bitrate is within $[min\_rate, max\_rate]$; (4.2d)

client video bitrate is less or equal to bandwidth estimation; (4.2e)

flows do not surpass any of the link capacities; (4.2f)

congestion linearisation; (4.2g)

flow conservation. (4.2h)

The resultant objective function is again a weighted sum of ISP and CP objectives (like in the original perfect collaboration formulation), thus again producing an infinite set of Pareto-optimal solutions. This time we are not, however, interested in exploring the Pareto curve of those solutions (which is computationally expensive); instead, we aim at finding a weight between the objectives that would provide both optimality of the problem *and* its fairness for both actors.

Fairness in ISP – CP collaboration

In order to achieve this, we can address Nash bargaining [63, 114, 21], which guarantees that a product of objective improvements of both (in our case) actors is optimal and fair. The objective improvement here means that Nash bargaining does not just consider the isolated performance of the actors, but rather the gains obtained from collaboration by each actor. This initial performance can be defined as, for instance, the one which they find themselves without any collaboration; alternatively, it could be set according to political reasoning of the parties and solidified within a Service Level Agreement. Equation 4.3 can therefore illustrate the objective function of a Nash bargaining for our task:

$$\max \quad ((\sum_{s,d} x_{sd} - \gamma \cdot \sum_{l} q_l - ISP_0) \cdot (\sum_{s,d} x_{sd} - CP_0)) \tag{4.3a}$$

s.t. (the same as in Eq. 4.2) (4.3b)

Where ISP_0 and CP_0 are the initial performance of, respectively, ISP and CP when no collaboration is applied.

One could note that this transformation looses convexity due to the presence of the product between two objectives. In order to alleviate this issue, we can pass to an equivalent problem of sum of the logarithms of the objectives, which brings the convexity back without affecting the decision space. This transformation is presented at Formulation 4.4.

$$\max \quad (log\,(\sum_{s,d} x_{sd} - \gamma \cdot \sum_{l} q_l - ISP_0) + log\,(\sum_{s,d} x_{sd} - CP_0)) \tag{4.4a}$$

s.t. (the same as in 4.2) (4.4b)

Design of a distributed optimisation problem

So far, formulations for a more extended collaboration were considering CP and ISP working over the same decision space, thus implementing, in general, a more complicated full-collaboration problem. On the contrary, one of the crucial assumptions of a collaboration strategy is to consider parties being independent. To achieve this, we can make both our actors have their own decision realms and variables; this can be easily done having formulated the Equation 4.4 as the coupling between both actors is represented by a simple sum of two objectives.

Now each party has to perform server selection, and they do it independently over their own realms. Ideally, we would want both actors to take the same server selection decisions so that the problem would be equivalent to the full collaboration model. To achieve this, we can use an auxiliary variable [121] p_s from previous problem formulations: though its main use-case was to ensure the server unicity, the variable itself is indicating the actual server selected. In the (simplified) decoupled collaborative formulation in Eq. 4.5 we assume that each party uses its own indicator variable (p_s^{CP} for CP or p_s^{ISP} for ISP; note also that their indices denote the same physical nodes); Equation 4.5d can then be put in place to guarantee that the server selections of both ISP and CP are coherent.

$$\max \quad (log\,(\sum_{s,d} x_{sd} - \gamma \cdot \sum_{l} q_l - ISP_0) + log\,(\sum_{s,d} x_{sd} - CP_0)) \tag{4.5a}$$

s.t. (the same as in 4.2) (4.5b)

$$x_{sd}^{ISP} = x_{sd}^{CP}, \qquad \forall s,d \tag{4.5c}$$

$$p_s^{CP} = p_s^{ISP} \tag{4.5d}$$

Though decoupled at the objective function, the formulation above becomes coupled through a consistency constraint. It is hence still centralised, i.e., the optimisation problem has to be solved at a central entity as constraint 4.5d is defined using variables of both actors. Ultimately, our goal is to depart from a centralised optimisation problem – ISP and CP are, after all, physically separate entities with potentially non-overlapping interests.

To continue decoupling the above problem into two separate ones, we must relax these two auxiliary constraints, which are complicating our problem. Luckily, we can rely on dual decomposition in this question. In addition, we know that, in our formulation, video bitrate x_{sd} depends on the server selection p_s, so we can make a simplification and remove the constraint on video bitrate – we now assume that it will be respected as long as constraint on server selection is respected too.

Now, once we have formulated the joint collaboration model, we can continue with its decomposition. As seen clearly in Formulation 4.4, both components of the objective function are optimising over a different set of variables. The problem could be decoupled right away, but constraint 4.5d is complicating the process. It has to be, therefore, relaxed into the objective function, and dual decomposition can help us with this. Let us define a Lagrangian of 4.4 as

$$\begin{aligned} &L(q_l, x_{sd}, p_s^{ISP}, p_s^{CP}, \eta_s) = \\ &(log\ (\textstyle\sum_{s,d} x_{sd} - \gamma \cdot \sum_l q_l - ISP_0) + log\ (\textstyle\sum_{s,d} x_{sd} - CP_0)) - \textstyle\sum_s \eta_s \cdot (p_s^{CP} - p_s^{ISP}) \end{aligned} \tag{4.6}$$

Lagrangian multiplier $\eta_s, \forall s \in \mathscr{N}$ is introduced to reflect the price of not satisfying the consistency constraint, which is now relaxed. Note that, as Eq. 4.5d is an equality constraint, η_s can be positive as well as negative, trying to make the CP match the tradeoff decided by the ISP (for example, if the CP has chosen server 1 (top server) while ISP would have chosen lower-level caches, then η_1 gets positive, and the CP is enticed not to choose server 1 at the next request).

The dual objective for this decomposition will be:

$$\eta_s = argmin \quad L(q_l, x_{sd}, p_s^{ISP}, p_s^{CP}, \eta_s) \tag{4.7a}$$

The problem 4.4 can now be decomposed into two separate primal problems, one for each of the parties, and a dual problem coordinating their decisions. Considering notation conventions for ISP problem are those of Eq. 3.4, while CP formulation used conventions from Eq. 3.1, the CP problem will be:

Distributed bargaining collaboration for CP: (4.8a)

$$\max \quad (log\ (\sum_{s,d} x_{sd} - CP_0) - \sum_s \eta_s \cdot (p_s^{CP})) \tag{4.8b}$$

s.t. (4.8c)

$$x_{sd} \leq bw\_est_{sd}, \qquad \forall s \in \mathscr{S}, d \in \mathscr{D} \tag{4.8d}$$

$$\sum_d x_{sd} \geq min\_rate \cdot p_s \qquad \forall s \in \mathscr{S} \tag{4.8e}$$

$$\sum_d x_{sd} \leq max\_rate \cdot p_s \qquad \forall s \in \mathscr{S} \tag{4.8f}$$

$$\sum_s p_s^{CP} = 1 \tag{4.8g}$$

And the decoupled problem for ISP is:

Distributed bargaining collaboration for ISP: (4.9a)

$$\max \quad (log\,(\sum_{d} x_d - \gamma \cdot \sum_{l} q_l - ISP_0) + \sum_{s} \eta_s \cdot (p_s^{ISP})) \tag{4.9b}$$

s.t. (4.9c)

$$x_{d_1} + x_{d_2} \geq min\_rate \qquad \mathscr{D} = d_1, d_2 \tag{4.9d}$$

$$x_{d_1} + x_{d_2} \leq max\_rate \qquad \mathscr{D} = d_1, d_2 \tag{4.9e}$$

$$x_s \geq min\_rate \cdot p_s^{ISP} \qquad \forall s \in \mathscr{N} \setminus \mathscr{D} \tag{4.9f}$$

$$x_s \leq max\_rate \cdot p_s^{ISP} \qquad \forall s \in \mathscr{N} \setminus \mathscr{D} \tag{4.9g}$$

$$\sum_{s \in \mathscr{N}} p_s^{ISP} = 1 \tag{4.9h}$$

$$p_s^{ISP} = 0 \qquad \forall s \in \mathscr{N} \setminus \mathscr{I} \tag{4.9i}$$

$$f_l \leq C_l, \qquad \forall l \in \mathscr{L} \tag{4.9j}$$

$$f_l - \sum_{s \in \mathscr{N}} x_l^s = flowinuse_l, \qquad \forall l \in \mathscr{L} \tag{4.9k}$$

$$f_l \leq q_l, \qquad \forall l \in \mathscr{L} \tag{4.9l}$$

$$3 \cdot f_l - q_l \leq 2/3 \cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.9m}$$

$$10 \cdot f_l - q_l \leq 16/3 \cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.9n}$$

$$70 \cdot f_l - q_l \leq 178/3 \cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.9o}$$

$$500 \cdot f_l - q_l \leq 1468/3 \cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.9p}$$

$$5000 \cdot f_l - q_l \leq 16318/3 \cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.9q}$$

$$\sum_{l \in In(i)} x_l^s - \sum_{l \in Out(i)} x_l^s = \begin{cases} +x_i & \text{if } i \in \mathscr{D} \\ -x_i & \text{otherwise} \end{cases}, \qquad \forall s, i \in \mathscr{N} \tag{4.9r}$$

Meanwhile, multiplier η_s has to be updated by the ISP in a dual problem for each request k coming by using iterative gradient descent and shared with the CP. Equation 4.10 illustrates this update; $\beta > 0$ in its formulation is a sufficiently small step size [121].

$$\eta_s(k+1) = \eta_s(k) + \beta(p_s^{CP} - p_s^{ISP}) \tag{4.10a}$$

Bringing linearity into distributed collaboration problem

One could have noted that primal distributed problems Eq. 4.9 and Eq.4.8 became non-linear (due to presence of the logarithmic function) once we attempted to bring convexity into bargaining-based collaboration formulation. Solving such optimisations would be easier if they were linear; on the other hand, it would also be good to keep the idea of bargaining. We, therefore, decided to not only simplify the bargaining formulation, but also linearise contending metrics: network congestion for ISP and video bitrate for CP. In this way, CP's and ISP's own objectives would remain the same, but the collaboration itself could be expressed as a weighted sum of those objectives, like in full-collaboration:

$$\max \quad (\sum_{s,d} x_{sd} - \gamma \cdot \sum_{l} q_l) \tag{4.11a}$$

s.t. (4.11b)

server unicity constraint: $\sum p_s = 1$; (4.11c)

client video bitrate is within $[min\_rate, max\_rate]$; (4.11d)

client video bitrate is less or equal to bandwidth estimation; (4.11e)

flows do not surpass any of the link capacities; (4.11f)

congestion linearisation; (4.11g)

flow conservation. (4.11h)

ISP and CP are coupled in this formulation through the objective function. We can relax this coupling, and to keep decisions of both parties intact with each other, auxiliary variables for server selection and resulting video bitrates are put in place:

$$\max \quad ((\sum_{i,d} x_{sd} - \gamma \cdot \sum_{l} q_l)) \tag{4.12a}$$

s.t. (the same as in 4.11), and (4.12b)

$$x_{sd}^{ISP} = x_{sd}^{CP}, \qquad \forall s, d \tag{4.12c}$$

$$p_s^{ISP} = p_s^{CP}, \qquad \forall s \tag{4.12d}$$

Relaxation of the auxiliary constraints is done in exactly the same way as for the bargaining-enabled formulation, with the Lagrangian multipliers η_s. Note that we remove the first constraint (with x_{sd}) just like in the mentioned formulation. The resulting problems

for CP and ISP is expressed next; please note that notation conventions for ISP problem are those of Eq. 3.4, while CP formulation used conventions from Eq. 3.1.

Linearised distributed collaboration for ISP: (4.13a)

$$\max \ \ ((\sum_{d\in\mathscr{D}} x_d - \gamma\cdot\sum_{l} q_l) + \sum_{s} \eta_s \cdot p_s^{ISP}) \tag{4.13b}$$

s.t. (4.13c)

$$x_{d_1} + x_{d_2} \geq min\_rate \qquad \mathscr{D} = d_1, d_2 \tag{4.13d}$$

$$x_{d_1} + x_{d_2} \leq max\_rate \qquad \mathscr{D} = d_1, d_2 \tag{4.13e}$$

$$x_s \geq min\_rate \cdot p_s^{ISP} \qquad \forall s \in \mathscr{N} \setminus \mathscr{D} \tag{4.13f}$$

$$x_s \leq max\_rate \cdot p_s^{ISP} \qquad \forall s \in \mathscr{N} \setminus \mathscr{D} \tag{4.13g}$$

$$\sum_{s\in\mathscr{N}} p_s = 1 \tag{4.13h}$$

$$p_s = 0 \qquad \forall s \in \mathscr{N} \setminus \mathscr{I} \tag{4.13i}$$

$$f_l \leq C_l, \qquad \forall l \in \mathscr{L} \tag{4.13j}$$

$$f_l - \sum_{s\in\mathscr{N}} x_l^s = flowinuse_l, \qquad \forall l \in \mathscr{L} \tag{4.13k}$$

$$f_l \leq q_l, \qquad \forall l \in \mathscr{L} \tag{4.13l}$$

$$3\cdot f_l - q_l \leq 2/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.13m}$$

$$10\cdot f_l - q_l \leq 16/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.13n}$$

$$70\cdot f_l - q_l \leq 178/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.13o}$$

$$500\cdot f_l - q_l \leq 1468/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.13p}$$

$$5000\cdot f_l - q_l \leq 16318/3\cdot C_l, \qquad \forall l \in \mathscr{L} \tag{4.13q}$$

$$\sum_{l\in In(i)} x_l^s - \sum_{l\in Out(i)} x_l^s = \begin{cases} +x_i & \text{if } i \in \mathscr{D} \\ -x_i & \text{otherwise} \end{cases}, \qquad \forall s, i \in \mathscr{N} \tag{4.13r}$$

In Eq. 4.13, constraints are presented as following.

- Eqs. 4.13d – 4.13e ensure that a single video flow never goes beyond the limits of mimimum- and maximum-quality representation bitrate for the given video request.
- Eqs. 4.13f – 4.13g guarantee that for all s which p_s is zero, x_s will also be zero.
- Eqs. 4.13h guarantees that only one server among S will be selected.

- Eqs. 4.13i declares that for all network nodes outside of $\mathscr{I}$, p_s never goes beyond zero, since they are not designated to serve video content.
- Eqs. 4.13j ensures no flow f_l surpasses the capacity C_l of the link it goes through.
- Eqs. 4.13k defines flows f_l through the sum of portions of all flows between every pair of nodes that go over link l.
- Eqs. 4.13l – 4.13q present the linearisation of our congestion function [44, 63].
- Eqs. 4.13r is a flow conservation constraint.

Linearised distributed collaboration for CP: (4.14a)

$$\max \ (\sum_{s,d} x_{sd} - \sum_{s} \eta_s \cdot p_s^{CP}) \tag{4.14b}$$

s.t. (4.14c)

$$x_{sd} \leq bw\_est_{sd}, \qquad \forall s \in \mathscr{S}, d \in \mathscr{D} \tag{4.14d}$$

$$\sum_{d} x_{sd} \geq min\_rate \cdot p_s \qquad \forall s \in \mathscr{S} \tag{4.14e}$$

$$\sum_{d} x_{sd} \leq max\_rate \cdot p_s \qquad \forall s \in \mathscr{S} \tag{4.14f}$$

$$\sum_{s} p_s^{CP} = 1 \tag{4.14g}$$

In Eq. 4.14, constraints are presented as following.

- Eq. 4.14d guarantees that no bitrate x_{sd} surpasses the bw_est_{sd} as the result of selection process.
- Eq. 4.14e and Eq. 3.1e guarantee that bitrates x_{st} between any server s and client t (which is the sum of x_{sd}) respect the bitrates of minimum- and maximum-quality representations of a requested video, while also ensuring that $x_{st} = 0$ for all s, p_s of which are zeros.
- Eq. 4.14g guarantee that only one server can be selected.

Multipliers η_s being updated in the same way as before (Equation 4.10):

$$\eta_s(k+1) = \eta_s(k) + \beta(p_s^{CP} - p_s^{ISP}) \tag{4.15a}$$

Formulation that is expressed in Equations 4.13 – 4.15 is now a linear and distributed collaboration problem, which is used in our evaluation. This collaboration strategy is referred to as *Collaboration-extended*.

Each of those distributed models is solved by its corresponding entity independently, even though their decisions are coupled. To cope with this coupling, explicit signaling is required. Note, that this collaboration model is not real-time, i.e., CP can only know the ISP decision (by η_s) only after its decision. The next incoming request, therefore, will be processed with the newly-obtained information taken into account.

That being said, the presented Collaboration-extended strategy can be summarised in Algorithm 3.

Algorithm 3 Processing an incoming request in Collaboration-extended

Input: incoming request with *max_rate* and *min_rate*; utilisation of links C_l at ISP; CP's bandwidth estimations bw_est_{sd}; prices η_s for all sources obtained after serving previous request; weight γ defining the importance of congestion in ISP's part of collaboration

(I) ISP actions:

(i) Receives the request to be transported to CP and determines its *max_rate*, *min_rate*, and destinations d_1 and d_2

(ii) Computes 4.13 and obtains p_s^{ISP} for all sources i

(II) CP actions:

(i) Receives the request, sets its *max_rate*, *min_rate*, and determines its destinations d_1 and d_2

(ii) Computes 4.14 and obtains p_s^{CP} for all its sources s

(iii) Starts streaming video content to the client from the source which $p_s^{CP} = 1$

(III) Further ISP actions:

(i) Receives traffic for the request and determines its source $s \in \mathcal{N}$ and its bitrate x_d

if s is the outside MP server **and** $p_s^{ISP} \cdot x_d == 0$ for newly determined s **then**

(a) Cut the second MP subflow

else

(b) Transport the content traffic in its entirety

end if

(ii) Update prices η_s for all sources $s \in \mathcal{N}$, as outlined in (A)

(iii) Send prices η_s for all sources $s \in \mathcal{N}$ to CP

(A) Update of prices η_s by the ISP:

for all sources $s \in \mathcal{N}$ **do**

$\eta_s \leftarrow \eta_s + \beta(p_s^{CP} - p_s^{ISP})$

end for

Finally, it is important to mention that in our evaluation we make the choice not to have the ISP include his pricing constraint η_s (equivalently, the relaxation term in its objective), as we choose to identify how close the scheme is able to bring the CP to the ISP-decided

rate-congestion tradeoff. This is also motivated by the fact that, unlike in [63], the ISP objective considers two terms: the clients' QoE (served rate) traded with the congestion with the γ parameter (which we fix to a certain value). The objective for the ISP becomes then the following (Eq. 4.16):

Linearised distributed collaboration for ISP: (4.16a)

$$\max \quad (\sum_{d} x_d - \gamma \cdot \sum_{l} q_l) \tag{4.16b}$$

s.t. (the same as in 4.13) (4.16c)

4.1.3 Discussion on the real-world implementation of collaboration

Both of proposed collaboration models rely on several parameters that have to be learned or shared between the parties, such as target bitrates of requested videos, or bandwidth estimations. This subsection discusses how can such parameters be acquired in a real-world implementation.

Link bandwidths for ISP

One of the input parameters for the ISP is available link bandwidths in its topology (or, more accurately, link utilisation figures). In our study we assume that all links that constitute this topology are point-to-point with a fixed capacity, so obtaining their utilisation is relatively easy. On the other hand, clients could be connected to the ISP using wireless technologies, the channel's capacity of which until the ISP access node is required as a part of topology. While we do not focus on an exact access technology, it is important to discuss how can such a metric be obtained.

An obvious solution would be to run bandwidth probe flows; while being invasive to network performance, such a technique can be used for any technology and give relatively accurate estimations quickly. It could be interesting, nevertheless, to be able to collect such measurements using the instruments that are provided with access technologies.

For mobile wireless connectivity such as LTE, this task is complicated by the changing nature wireless medium quality. Estimating real-time channel characteristics (like Bit-Error Rate and Signal-to-Noise Ratio) is an important routine for the system's functioning [105, 72]; knowing other characteristics of the wireless setup in question, an instantaneous eNB-to-client bitrate can be calculated [16]. Another option in this case would be to monitor the

bitrate of client's flows or resort to general statistics for the given base station equipment and environment [119].

Local wireless connectivity, such as IEEE 802.11, is not usually exposed to the ISP. On the other hand, most advanced revision of this technology (802.11ad) allow for capacity of up to few Gbps, which greatly surpasses the fastest home fibre-optic connections (e.g., 1 Gbps offered by Google Fiber), so estimating home wireless capacity can be unnecessary.

Wired access technologies are usually are either point-to-point (FTTx, ADSL), or shared with high degree of provisioning (local Ethernet connectivity), therefore their physical capacity is usually fully disposable to the client - even though their actual access bandwidth to the Internet may be restricted by the ISP according to the its tariff policy.

Path bandwidths for the CP

CP makes his own server selections to maximise the resulting client's QoE. In our simplified formulation, we consider QoE to be constituted by video bitrate only, so CP has to rely on bandwidth measurements. Since we consider the ISP topology to be hidden from the CP, it can only deal with bandwidths of the paths from its caches/servers until the clients or their premises, and therefore several considerations has to be taken into account.

First, it is hardly possible to keep track of available bandwidth until every client, past and future. Instead one could imagine aggregating these this statistics by, for example, geographical allocation (ideally – grouping by base stations in mobile networks or aggregate routers in wired ones), because clients residing in the same physical area would probably have close bandwidth measurements until a distant node (like an outside CDN server). This is the approach we implement in our numerical evaluation: CP, when estimating client-server bandwidths, in fact groups all video clients by their respective access nodes and maintains those estimation for each pair of every access-node and cache/server. In reality, this could potentially be done basing on IP address subnets, though today it requires a certain support from the ISP as the era of strict geographical assignment of IP address is long gone. Nevertheless, such a problem of estimating the ISP states has been investigated for critical operations by CDNs or client-CDN intermediaries such as Conviva (like their Precision Video technology, also in [45]),

Speaking of the bandwidth estimation themselves, a good solution would be to exploit ongoing video transmissions. Together with client aggregation one could build up statistics for the clients in the same area and to use the resulting estimate to make decisions for future clients connecting from the same area. Another, easier way to estimate client's bandwidths would be to run probe flow before the session start, but that would inevitably increase video startup times.

Minimum and maximum video bitrates for the ISP

In order to make server selection, ISP has to know the target bitrates of requested videos. While input parameters for ISP models mention both maximum and minimum bitrates, it is certainly more important to know the former as it tells whether the request can be supported by the network.

With the proliferation of end-to-end encryption in the Internet, there is no practical possibility to peek into the client's traffic as to identify its requests. This leaves us with two possibilities: either CP could share this information per-request (for example, using MPEG-SAND [158]), or ISP could infer them since using publicly available estimations (like [3]). The former method seems like an inexpensive and efficient solution: with increasing estimated minimum bitrate by a certain small "safety" margin, ISP could make sure that most of the videos would comply with his estimation. Since what we are interested in is the server selection, and not the resulting estimated request bitrate, that overestimation margin should not impact negatively the accuracy of ISP in this regard.

There exist works that attempt to tackle the problem of HTTPS encryption with regards to video traffic management [28, 88], which gives a promise that already in near future it would be possible to infer video characteristics without breaking encryption or invading privacy.

Exchanged weights

Parameters η_s are unique to our collaboration scheme. In our collaboration, CP has to know exact ISP decisions (and not just the fact of a error), so these parameters have to be shared explicitly. This could be achieved by the means of MPEG-SAND, or a proprietary information exchange framework.

4.2 Experimenting with ISP-CP collaboration

In order to evaluate the discussed-above models, they have been implemented in Matlab using CPLEX as linear optimisation solver. For the performance to be comparable to the evaluation of the previous chapter, we have applied the same network topology (Fig. 4.1) and settings, namely:

- Clients connected simultaneously two two ISPs that have partially-connected tree topology; each of the ISPs has a single-link connectivity to the outside multipath-capable server. Entire topology is constructed of 1 Gbps links.

- CP's interest in improving the QoE of its users, under which we understand the achieved video bitrate. ISP, in its turn, is interested in reducing congestion in its network; the congestion is quantified as proposed in [44], such that the its value would increase abruptly when link utilisation would approach its capacity.
- Each ISP has in-network caches high in the topology (resembling Packet Gateway (P-GW) placement), that replicate the outside server's content catalogue.
- As to reduce the amount of variability, we set fixed the request bitrate to 50 Mbps and access capacity to 20 Mbps.
- After identifying the weakness of greedy approach in collaboration schemes (see Sec. 3.4), we set the arrival rate of requests to $\lambda = 0.2$ and maximum request duration to 2 minutes.

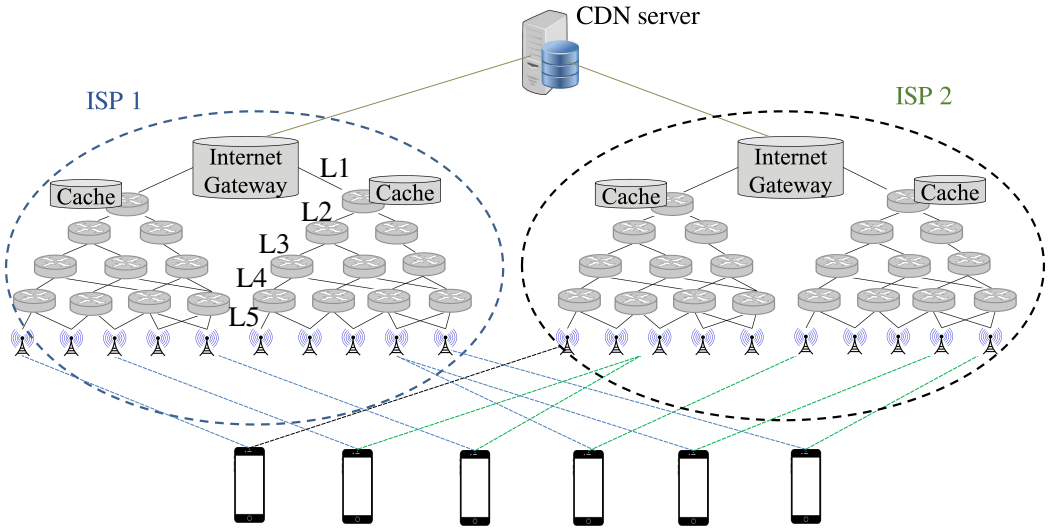


Fig. 4.1 Clients connected to two access networks, with in-network caches and the external CDN server. "L1", "L2" and so on indicate the ISP topology level.

To illustrate our evaluation, we will plot two kinds of figures: time series and time average. We will show three types of time average figures, all of them being plotted against the average total network congestion. The second axis will contain the following metrics:

- **Achieved** served rate: instantaneous obtained rate averaged over all active requests, and averaged over time. These figures will be presented with confidence intervals of 0.95, which are acquired after running the same experiment five times over different random

number generator init number. Random generator initialisation here the generation of input workload trace.

- **Maxfrac**: average fraction of requests obtaining the maximum rate (taking into account available access link bandwidth that can be lower than the maximum video bitrate). They will not have confidence intervals shown.

- **Relfrac**: average fraction of requests obtaining a rate higher than with without multipath subtracted with the fraction of requests obtaining a rate lower than without multipat. They will not have confidence intervals shown.

Mentioned metrics will be presented against the theoretical performance boundary (also referred to as Full-collaboration in Chapter 3), as well the following relation schemes between ISP and CP: non-collaborative no-visibility, with and without multipath (CP has no visibility over exact path bandwidths), non-collaborative Full-visibility (CP has complete and perfect visibility over path bandwidths), Collaboration-light with multipath, and Collaboration-extended with multipath.

In the time series, every figure is plotted from a single sample of multiple experiment runs. The following metrics will be shown:

- **Average achieved request bitrate**: in our system, every request either gets assigned a feasible bitrate, or is rejected and is assigned zero bitrate. At each request arrival, the average rate obtained by all currently active requests is logged, then a running average of them within a moving window of 30 requests is plotted in the top figure of the time series.

- **Number of active requests**: running average of amount of currently active requests in the system. In the second row of the first set of time series.

- **Running average of the share of clients served by the outside multipath server**. In the second fifth of the second set of time series

- **Total congestion**: running average of value of the sum of d_l, for all $l \in \mathscr{L}$, for both ISPs. In the fourth row of the time series figures.

- **Requests rejects**: running average of rejects (i.e., if CP deems the request's video minimal bitrate cannot be satisfied). In the second fifth of the second set of time series.

We first present the basic scenario, with settings as above, and then modify some of them as to observe the sensitivity of the devised collaborative schemes against modified setting.

For the sake of figure clarity, time series plots only contain curves for the following strategies: non-collaborative no-visibility with MP, non-collaborative Full-visibility (formulated at Eq. 3.3), Full-collaboration (Eq. 3.4), and Collaboration-extended (Eq. 4.16 and Eq. 4.14). Results for collaboration-light are reported only in the time average curves. All figures below are obtained when γ of the fullcollab problem considered by the ISP is set to 0.07 (being the middle ground in terms of ISP ocngestion).

4.2.1 Performance of collaboration schemes at base settings

Fig. 4.2 shows that the video rate obtained with Collaboration-extended lies quite close to that planned on by the ISP considering Full-collaboration, as are the rates Full-visibility and non-collaborative no-visibility with MP. However, the level of congestion obtained by Collaboration-extended is much lower and closer to Full-collaboration than non-collaborative no-visibility with MP and non-collaborative full-visibility. The level of accepted requests is also similar, with no rejects in case of Collaboration-extended model. The third row of Fig. 4.2 shows that, contrary to non-collaborative no-visibility with MP, the fraction of requests assigned to server 1 with Collaboration-extended matches much more that of Full-collaboration, thus better utilising the in-network caches. Let us here recall that the server assignment is always made by the CDN. The ISP preference, output of the Full-collaboration problem, only impacts the consistency costs η_s, for all $s \in \mathscr{S}$, as well as the CP's bandwidth estimation as one of the subflow the CDN server expected to be useful to serve the request finally got a zero-rate.

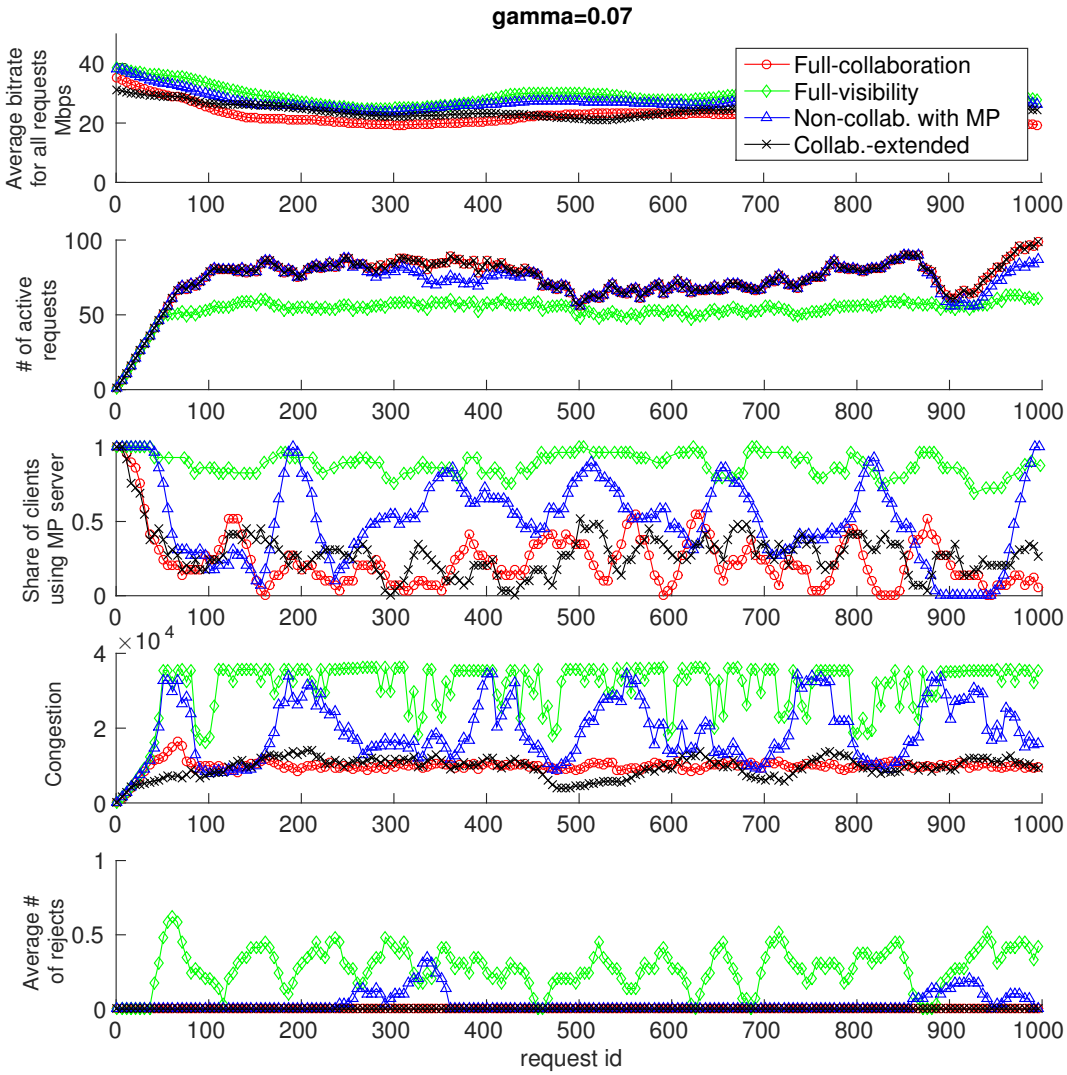


Fig. 4.2 Base experiment with requests of 2 minutes long: Evolution of system's metrics over time.

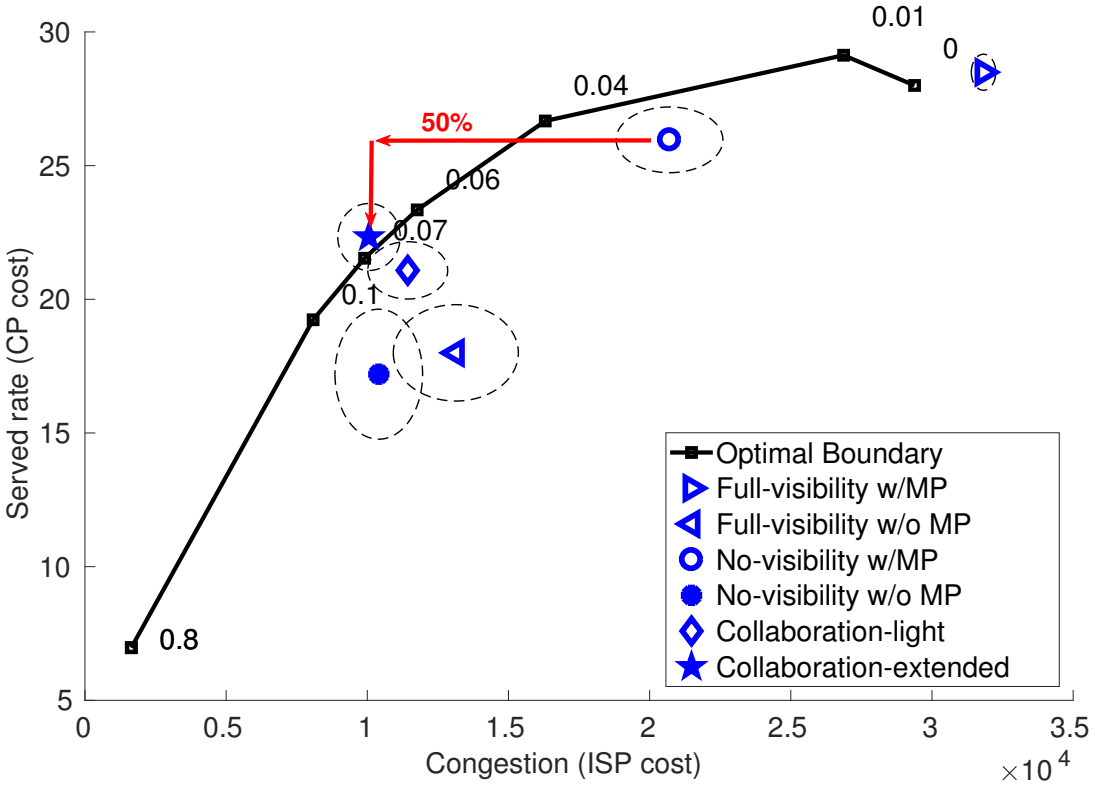


Fig. 4.3 Performance of different strategies as compared to themselves and the theoretical boundary (full-collaboration) at base settings, 2 minutes long requests. Confidence intervals in both dimensions are shown as ellipses around respective points.

Fig. 4.3 shows that Collaboration-light allows to bring the cooperative performance closer to the Pareto front, as well as to the tradeoff desired by the ISP. Moreover, Collaboration-extended is even more efficient in doing so as it allows to reduce the congestion level by a significant amount (illustrated to 50% in this case), as compared to non-collaborative no-visibility with MP, while maintaining a rather elevated served rate, as compared to Collaboration-light.

In addition, one could have noted that we used a linear QoE mapping function – i.e., in our considerations, the QoE is exactly equal to the obtained video bitrate. In reality, visual quality has a logarithmic relation to the video bitrate [130, 133]. If we would apply it in our modelling, we could perhaps expect the gain of collaboration in terms congestion to be even more pronounced, at even less loss of video visual quality (due to it having a logarithmic mapping).

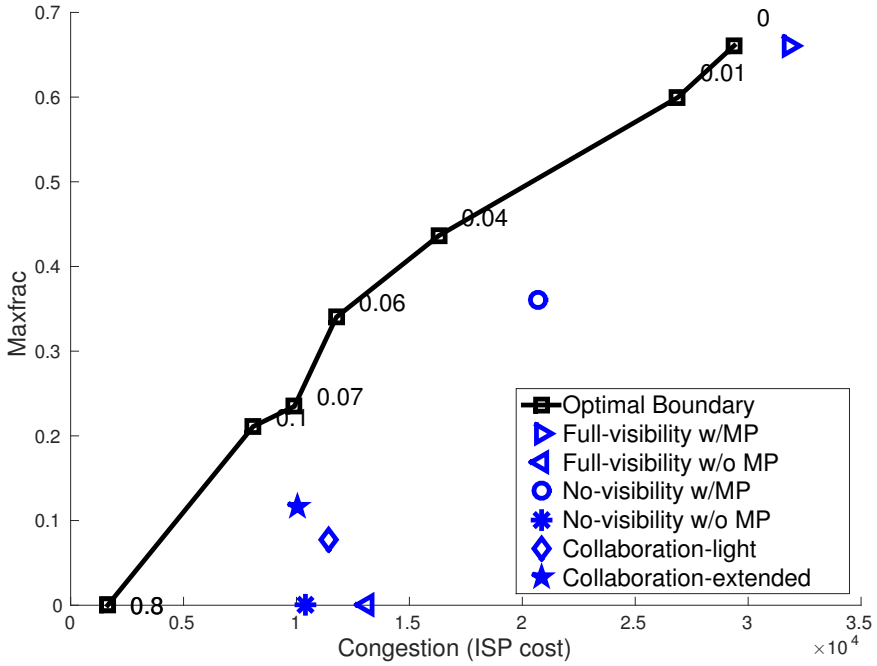


Fig. 4.4 Average fraction of requests obtaining maximum video rate, for different strategies at base settings (taking into account available access link bandwidth), 2 minutes long requests

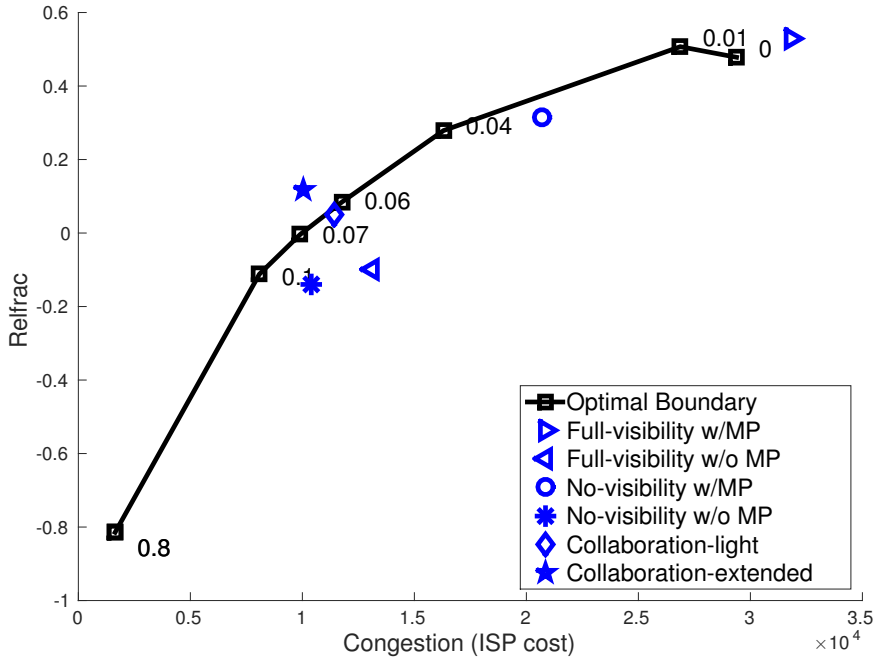


Fig. 4.5 Average fraction of requests obtaining video rate higher than without multipath minus fraction of requests obtaining a rate lower than without multipath, for different strategies at base settings, 2 minute long requests

When looking at other metrics than average served rate, Fig. 4.5 shows that Collaboration-extended can also get close to the desired tradeoff in terms of net fraction of requests obtaining more rate than without MP. However, Fig. 4.4 shows that Collaboration-extended (as well as Collaboration-light) does not reach the expected level of fraction of requests obtaining the max rate and corresponding to the desired tradeoff. This can be explained by the fact that Collaboration-extended strives to make the CDN follow the decisions of the ISP in average over time, so average congestion can hence be met. However, matching the number of requests obtaining the max rate would mean fitting the same server selection stochastic process over time, which Collaboration-extended does not.

4.2.2 Sensitivity analysis

Initial experimental results have demonstrated the performance of collaboration models for settings with a rather limited degree of variability. This subsection will gradually demonstrate how do the devised collaboration schemes perform with more variable, realistic workloads and settings.

Variability in video bitrates

First we start with unlocking variability at the video bitrates. Previously, we considered that Content Provider offers only one video with a highest-quality representation bitrate of 50 Mbps; now we extend amount of videos to three. These three videos can represent different video classes in reality, which are characterised by a different amount of action – and thus by a different bitrate of the highest-quality representation. In order to be comparable to the previous results, the average bitrate has to be conserved – so Figures 4.6 – 4.7 demonstrate the performance of different (no-)collaboration schemes with a set of videos, highest-quality representation of which are encoded with bitrates of 20 Mbps, 50 Mbps and 80 Mbps (at a normal distribution between those values). The bitrate of the lowest-quality representation remains 10% of the highest-quality bitrate.

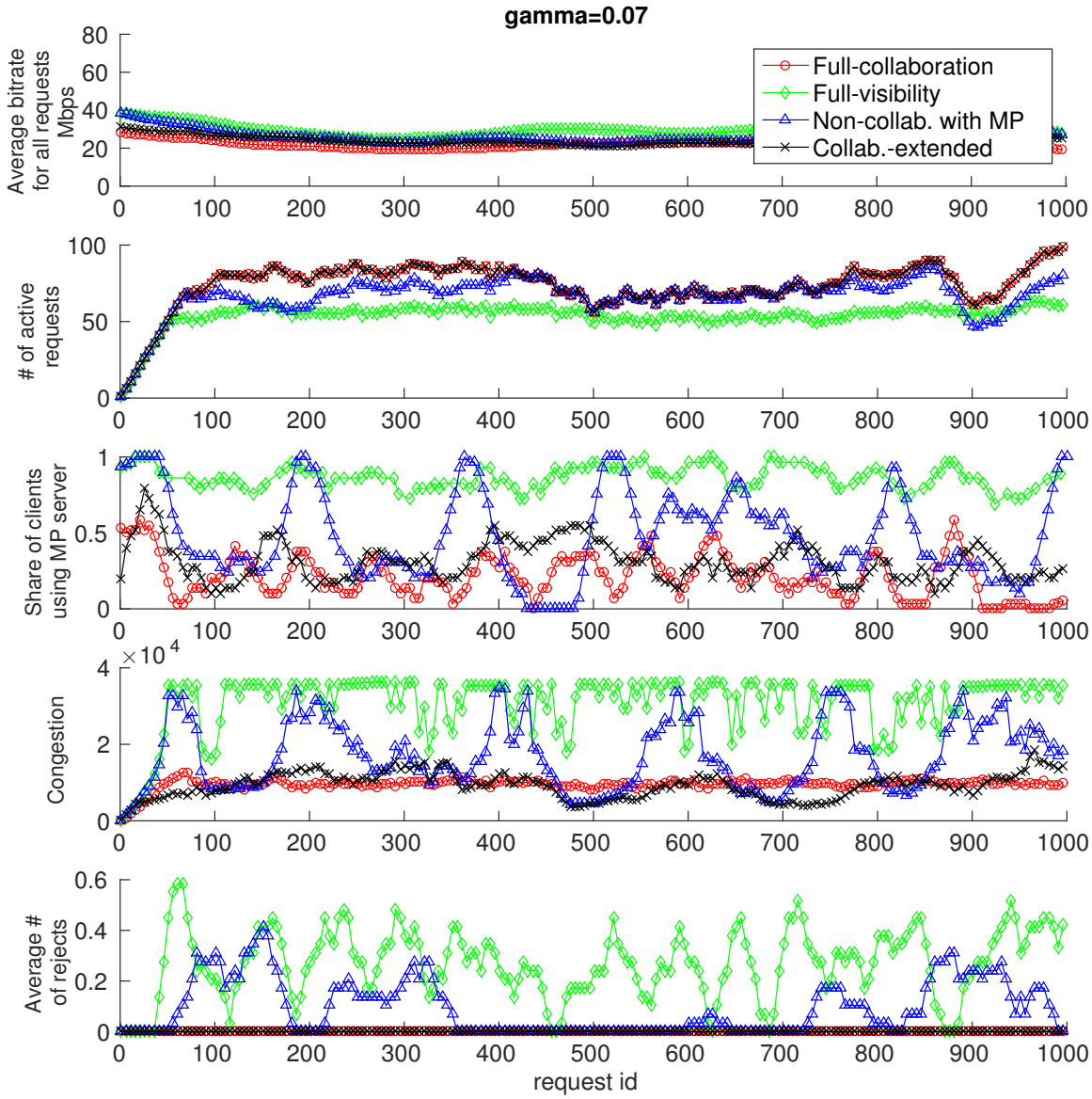


Fig. 4.6 Case of varying maximum video bitrate: Evolution of system's metrics over time.

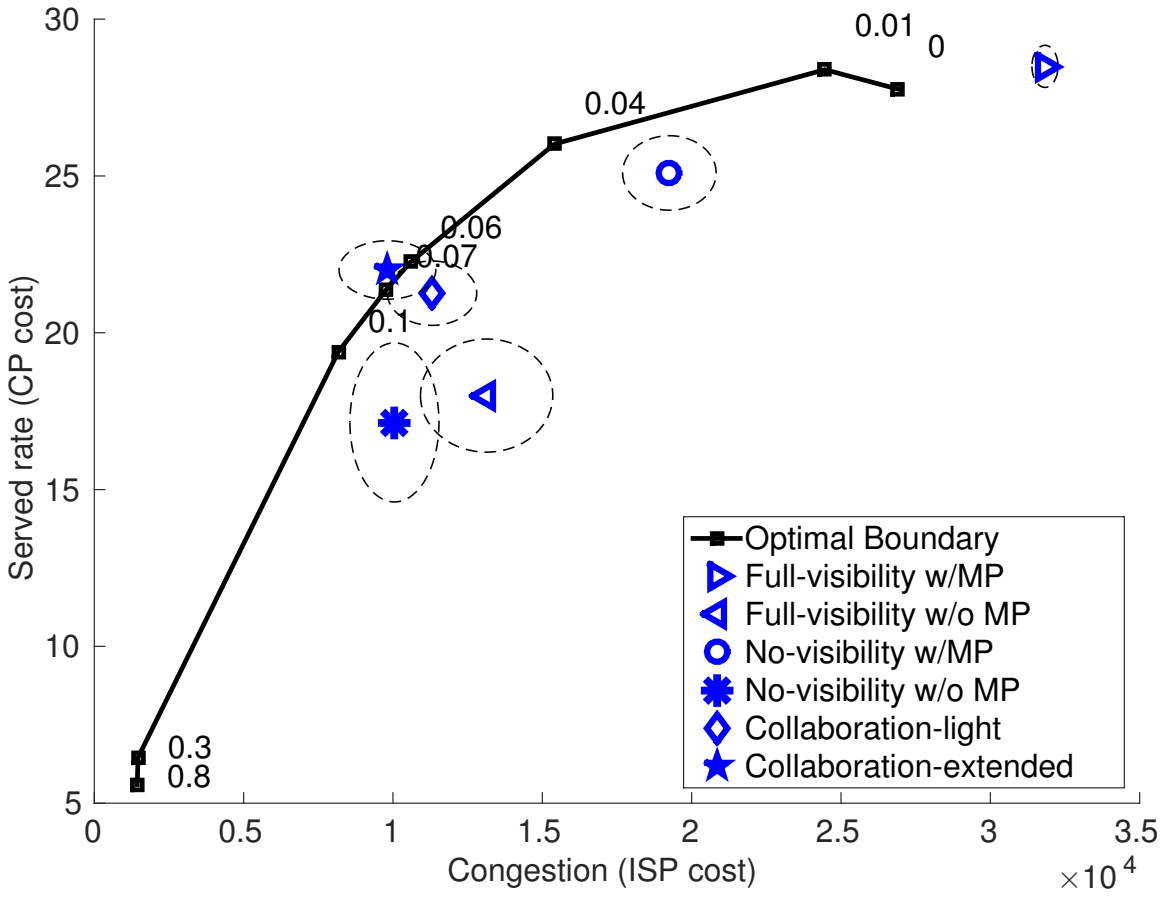


Fig. 4.7 Performance of different strategies as compared to themselves and the theoretical boundary (full-collaboration) with varying maximum video bitrate, 2 minutes long requests.

Figures suggest that this degree of variability in video bitrates produces results very similar to those of the base experiment for our collaboration schemes, except that at such settings the no-MP no-visibility case can approach the desired tradeoff - though sacrificing the request reject rates (as shown at row 5 of Fig.4.6).

Variability in access link bandwidth

Next we assess the isolated impact of variable access link bandwidth. As in the previous clause, we maintain the same average value of the varying parameter, so in this case the access link bandwidth will have normally distributed values of 5, 10, 20, 30, 35 Mbps while the maximum video quality remains fixed at 50 Mbps. Figures 4.8 – 4.9 demonstrate the performance of different (no-)collaboration strategies.

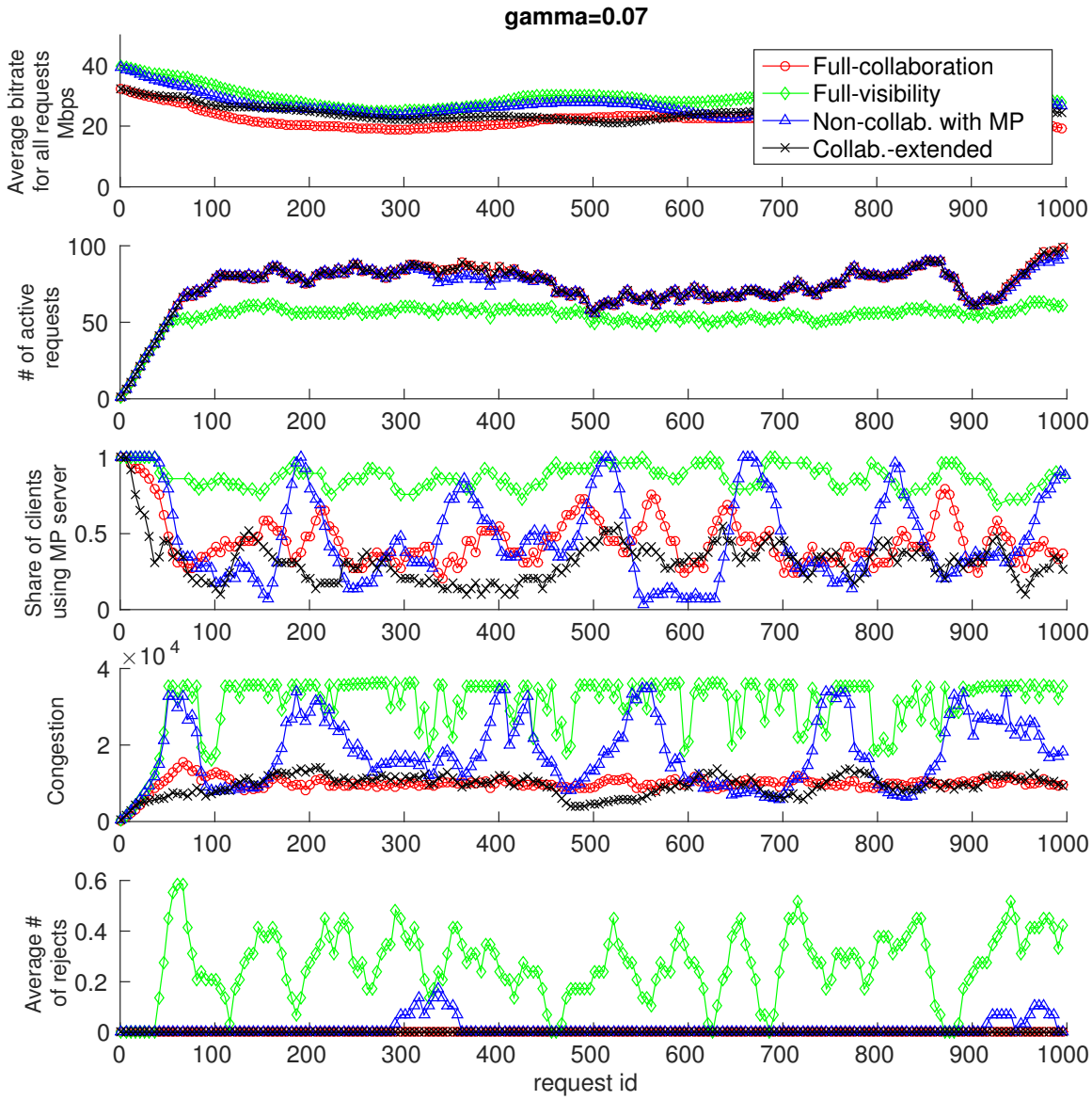


Fig. 4.8 Case of varying access connectivity capacity: Evolution of system's metrics over time.

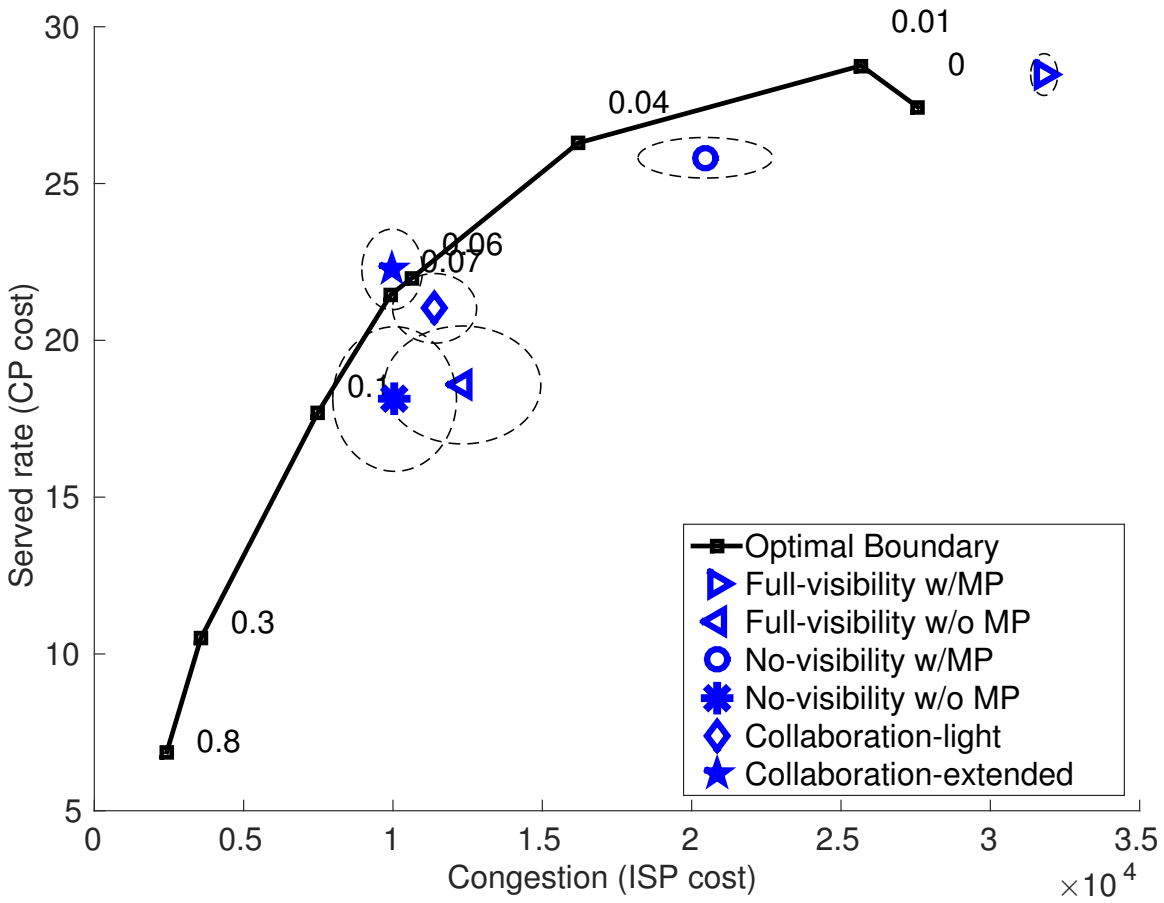


Fig. 4.9 Performance of different strategies as compared to themselves and the theoretical boundary (full-collaboration) with varying access connectivity capacity, 2 minutes long requests.

As opposed to the previous clause, variation in immediate capacity of the access connectivity brings some change in performance of collaboration strategies. Moreover, it's interesting to note that, judging from confidence intervals, variability of the results does not change as compared to base scenario, even though intuitively a case with fixed bandwidth and video qualities would have been expected to be rather stable.

Apart from that, we can notice on Fig. 4.9 that variable access capacity manages to bring the no-MP no-visibility strategy quite close to the target tradeoff with no increase in request rejects, though both proposed collaborative strategies remain even closer and demonstrating less discrepancy in try-to-try results. Since Fig 4.8 only shows performance of MP-enabled strategies, the results shown are qualitatively equivalent to those of previously-discussed settings.

Performance with all above settings being variable

Now let us see how the devised collaboration schemes compare to non-collaborative strategies in more a more realistic scenario – with both maximum video bitrate and immediate access capacity being variable. Figures 4.10 – 4.11.

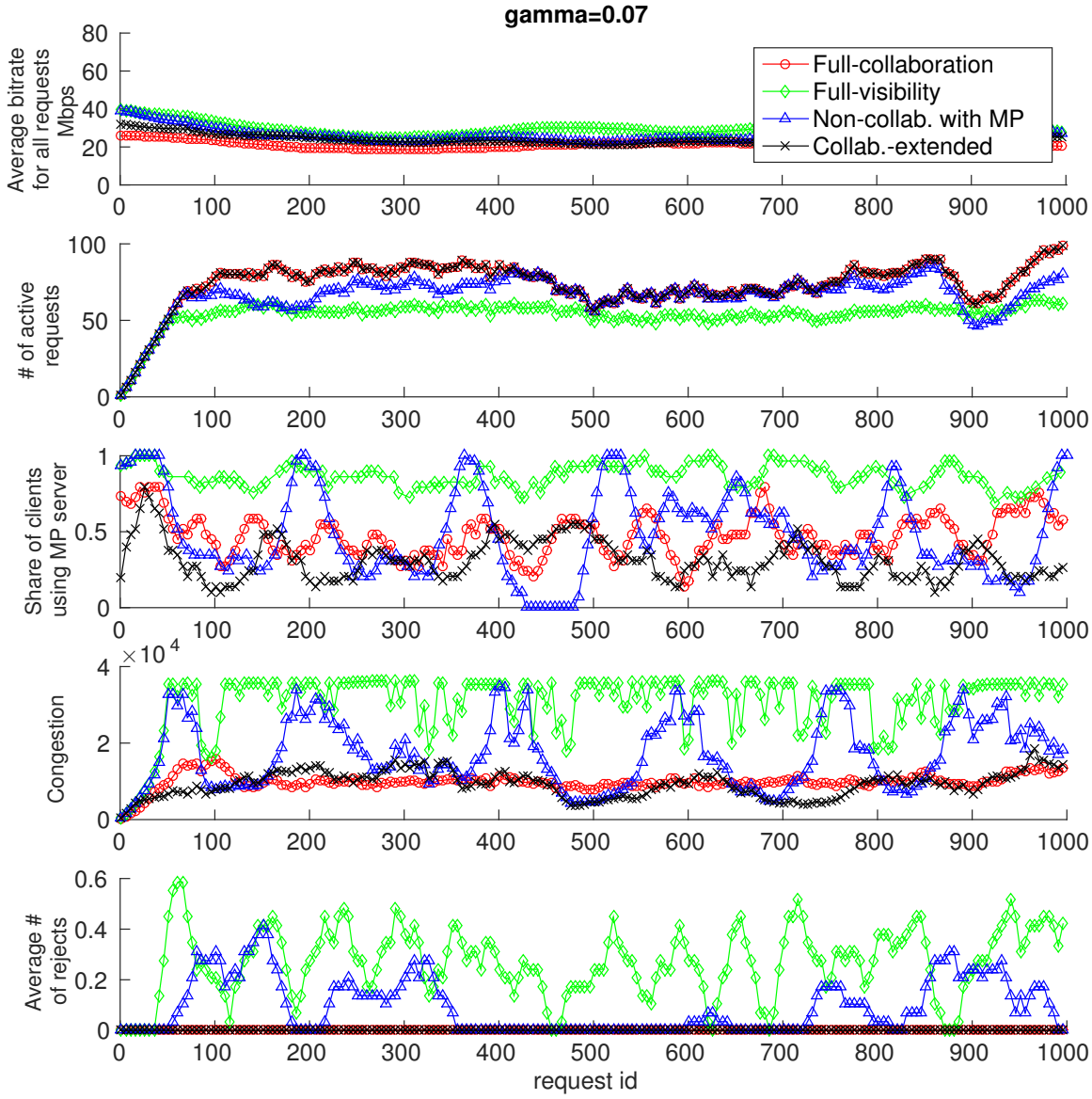


Fig. 4.10 Case of varying maximum video bitrate *and* access capacity: Evolution of system's metrics over time.

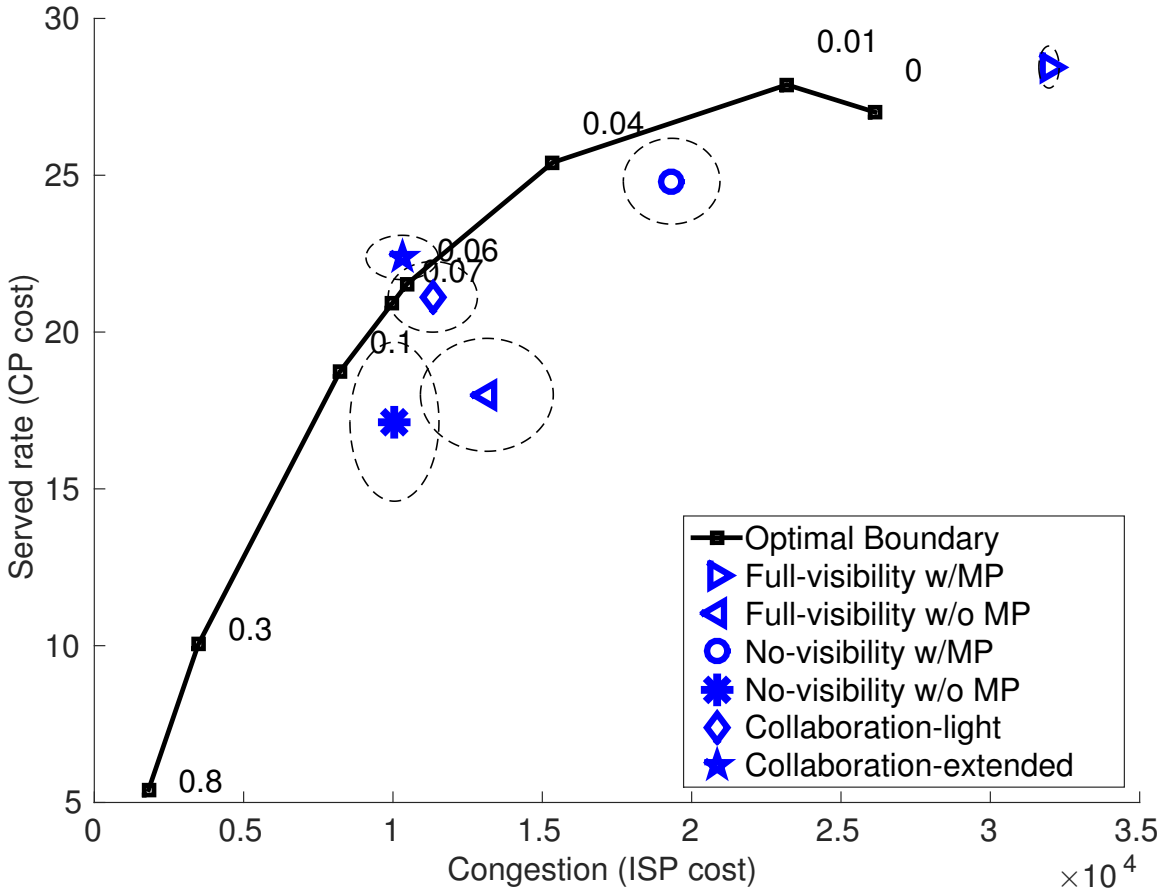


Fig. 4.11 Performance of different strategies as compared to themselves and the theoretical boundary (full-collaboration) with varying maximum video bitrate *and* access connectivity capacity, 2 minutes long requests.

Obtained results can be characterised in a way as a junction between performance character of two previous settings: as in variable maximum video bitrate case, the no-MP no-visibility scheme approaches the target tradeoff, but it does come at the expense of increased request reject rate (as opposed to the variable access capacity case). Anyhow, the proposed collaborative strategies manage to perform well in any of those settings: keeping rather close achieved video bitrate as the no-visibility with MP case, they outperform the former in network congestion and manage to do it with zero request rejects.

4.2.3 Reducing CP's bandwidth estimation re-initialisation timeout

When explaining the modeling of CP's behaviour in Chapter 3, we have mentioned that it has to maintain bandwidth estimations until clients' premises, and it uses EWMA to manage them. We argued that it is useful to frequently re-initialise the meterings as to avoid it

becoming stale. It is interesting, however, to see how do collaboration models respond to increasing this timeout. In previous results, the timeout value has been set to 200 seconds, and Figures 4.12 – 4.13 demonstrate the system's performance with timeout equal to 1000 seconds, for base settings.

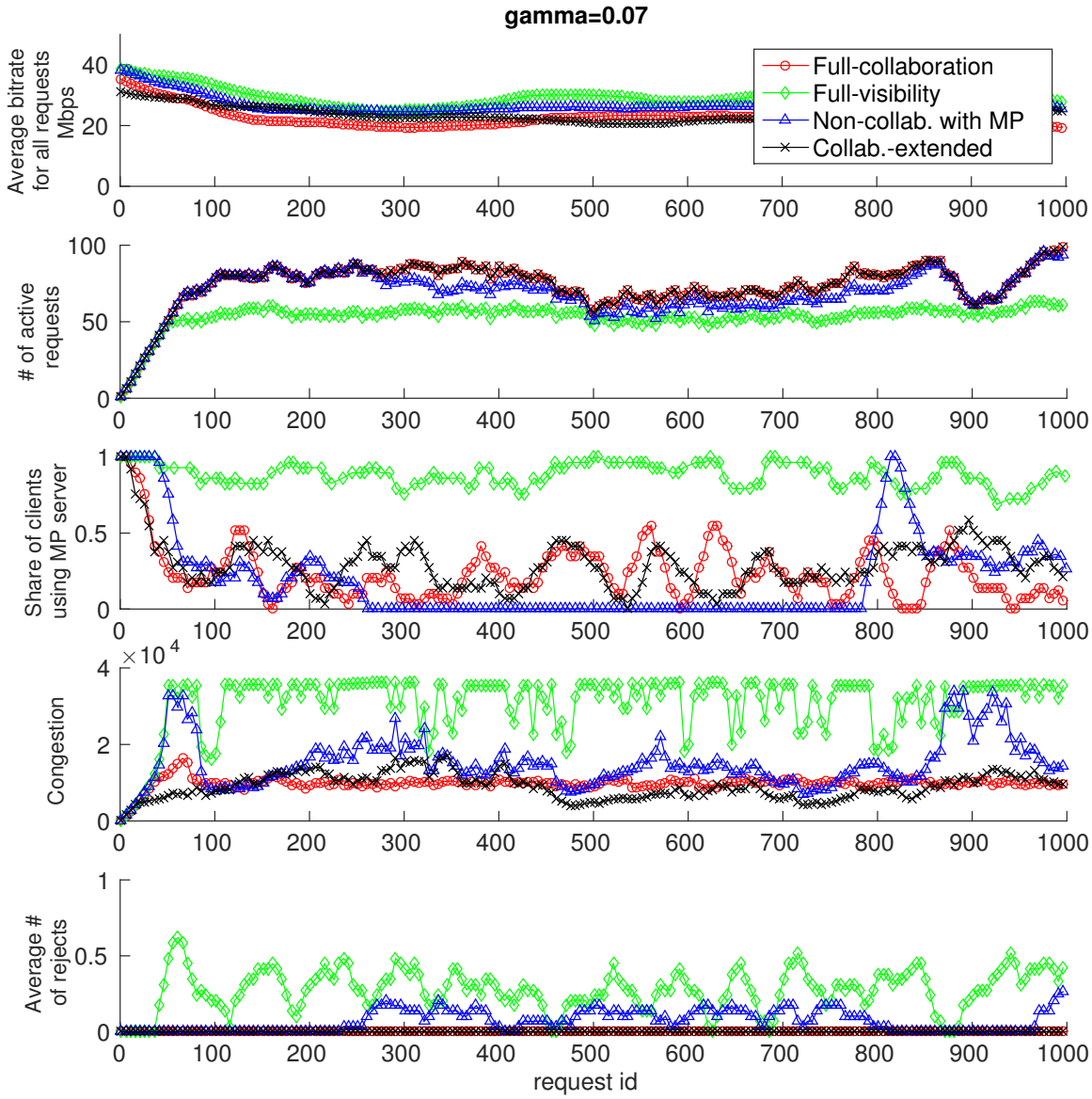


Fig. 4.12 Case of increased bandwidth estimation timeout at CP: Evolution of system's metrics over time, at base settings.

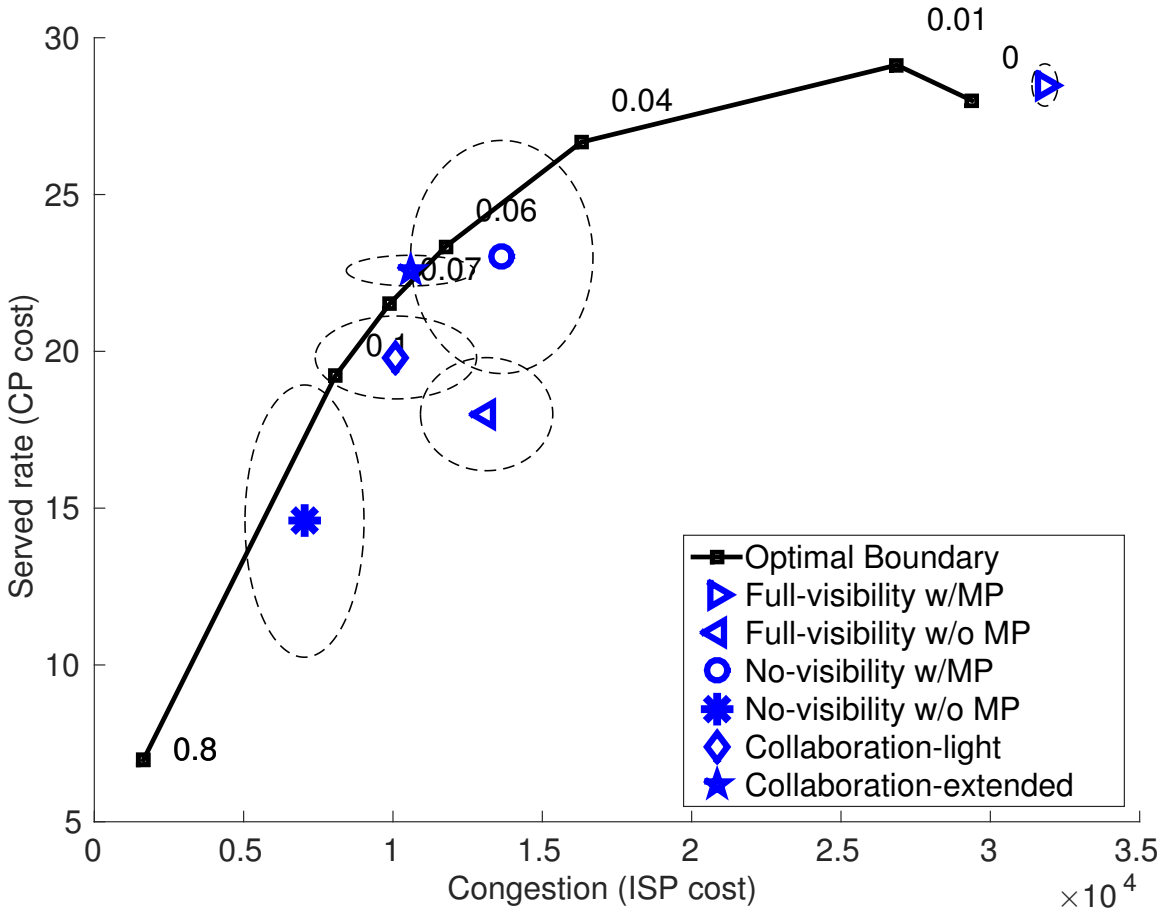


Fig. 4.13 Performance of different strategies as compared to themselves and the theoretical boundary (full-collaboration) at base settings but with CP's bandwidth estimation re-initialisation happening every 1000 seconds, 2 minutes long requests.

Frequent re-initialisations happened to reduce the performance of multipath no-visibility case in terms of actors' objectives, by so being somewhat favourable for the collaboration models. With a timeout of 1000 seconds situation changes, and we see on Fig. 4.13 how no-visibility case approaches both Collaboration-extended and the theoretical boundary of full collaboration, though with quite a significant variability. At the same time, the time series plots (Fig. 4.12, row 5) show that this improvement does not come for free for the multipath no-visibility scheme: the reject rate is also increased. This can be explained by the fact that infrequent bandwidth measurement re-initialisation, as expected, affects the accuracy of CP's bandwidth estimations. Row 3 of Fig. 4.12 shows that the estimation for the multipath server (which is referred to as "level 1") becomes too low to be selected by CP, so its utilisation goes down to zero until the values is re-initialised again, while in reality it is free enough to accept requests. Rows 3 and 5 also demonstrate themselves to be connected,

suggesting that it is not possible to accept all requests without using all available caches and servers – at current settings.

4.3 Discussion

Performance of both devised collaboration schemes is promising in several regards. First of all, both of the proposed schemes demonstrate a clear improvement in terms of actors' objectives (ISP congestion and average achieved video bitrate) in comparison to no-collaboration setups. Moreover, performance of collaboration schemes in fact achieves the theoretical boundary that has been established in previous chapter for performance our system. Having a random component in server selection and workload generation, variability in results between different sample experiments of collaboration schemes consistently manages to be smaller than those of no-collaboration strategies, giving an advantage of a more stable solution in terms of performance.

There is another important metric for the CP, that has not however been a part of its decision strategy: request acceptance rate. Even though the network we run our experiments on is well-dimensioned to accept all incoming requests in our workload, non-collaborative approach does not manage to always do so. This is explained by the inherent weakness of bandwidth estimation method at the CP side: in order to know how much available bandwidth there is between a given cache/server and client premises, CP has to assign a video flow which will serve as a probe for it. As discussed before, once the estimation becomes too low (due to a certain critical number of client flows being assigned to it), the path stops being selected and hence probed by client flows. This dictates the need to re-initialise the estimations as to stimulate exploration of available paths. A timely exploration of a path that was estimated to be overutilised (but has been freed since last probing) will lead to better request acceptance rate, but will also inevitably compromise objectives of the CP and ISP as the re-initialised values will often be incorrect. On the other hand, as we have seen at Figure 4.12, increasing the timeout leads to severe underutilisation of certain servers and comes at the expense of decreased acceptance rate. We can imagine that increasing the timeout even more will finally cause all possible path estimations to become stale so any request acceptance would become impossible, even though the network itself would be able to do so. We can conclude, therefore, that a good value of the re-initialisation timeout depends on several factors such as incoming workload, network set-up and the priorities of the CP in terms of its objectives. It becomes not easy to select a good value for it, yet to maintain it with changing workload as to keep the system efficient.

Our collaboration scheme alleviates this issue. Once its bandwidth estimations become stale or get re-initialised, CP is making an ill-informed decision which is much more likely not to coincide with the one of ISP. In such a case, price η_s received from ISP will modulate CP's future decision by helping him to quickly "catch up" with estimating ISP state. Because of this, as we can see on the time series plots, both collaborative strategies manage to perform equally well regardless of the selected re-initialisation timeout, deeming unnecessary the need to tune this value. Collaboration also lifts the need to balance between congestion performance and request acceptance, which is apparent in a non-collaborative setup: regardless of timeout value, both collaborative strategies manage to accept all requests at the examined workload, while keeping actors' objectives well-satisfied (i.e., close to be optimal and fair).

In addition to the mentioned advantages, Collaboration-extended almost follows the server/cache selection pattern of the target Full-collaboration: it keeps upper server being selected on average about 40% of the time, which can be seen at time series figures in previous section. This is good as selecting caches means less total network congestion at ISP and potentially less latency between client and the source of its video (which is not, however, a subject of study in this chapter). Most importantly, this result does not come at a compromise for the request acceptance (as compared to the non-collaborative approach) while being close to the target theoretical boundary for the achieved video bitrate and network congestion. This becomes possible due to both actors being in sync with their decisions, which is a feature of the extended collaborative scheme.

In terms of the metric called Maxfrac, performance of our collaboration schemes does not manage to achieve the theoretical boundary. Maxfrac is the ratio of requests obtaining the maximum achievable video bitrate (this accounting for the access link capacity, if smaller); Fig. 4.4 shows that at the same level of congestion both of devised collaboration strategies have about twice as less such requests compared to the Full-collaboration boundary. This is, in fact, good news: judging from the average bitrate numbers for collaborations and the boundary, we can conclude that such a difference in Maxfrac comes from the Full-collaboration having complete visibility over the network state, so it can easily assign first coming requests to the best paths (which will let them achieve the maximum available video bitrate). The network capacity is, however, finite, so assigning the first requests maximum video bitrate leads to later requests only allowing themselves much lower bitrates. The average value remains the same, but unfairness between different request sessions becomes rather significant – which is, arguably, a negative point.

It is interesting to note that the Collaboration-light already achieves a significant improvement over non-collaborative setup, and empowering our video delivery system with actual collaboration (in our interpretation, Collaboration-extended) gives somewhat marginal

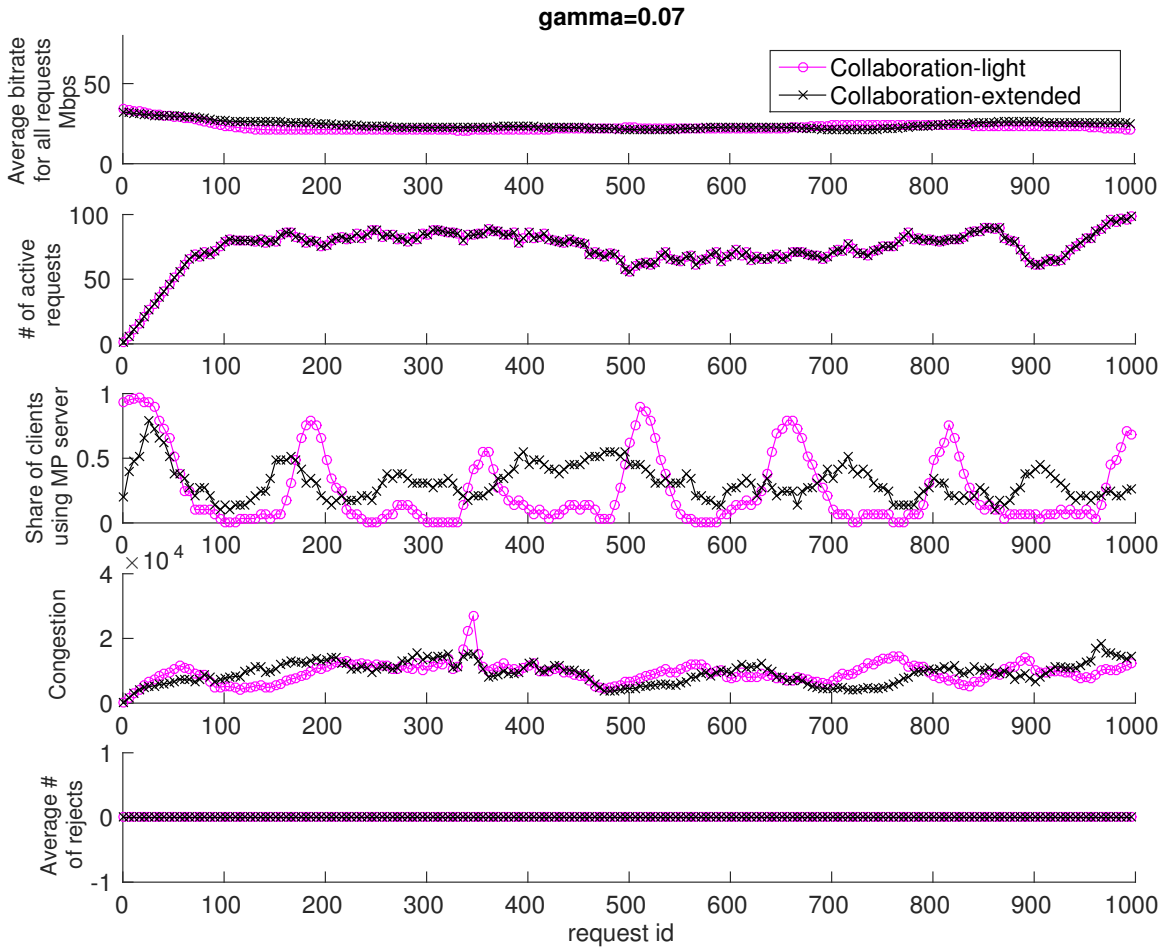


Fig. 4.14 Comparison of Collaboration-light and Collaboraton-extended in case of variable maximum video quality *and* access capacity: Evolution of system's metrics over time.

improvement in actors' objectives while being able to accept all incoming requests. Figure 4.14 compares both of our collaboration schemes against each other in a time series when both maximum video bitrate and access link capacity is variable. As one can see, both curves almost follow each other for every metric, with a slightly higher congestion at times and a different character of server utilisation for the light collaborative scheme. Since in Collaboration-light the CP does not exchange decision information with the ISP, its decisions are still sensitive to the bandwidth estimation re-initialisation (as can be noticed with the seasonality in the upper server selection at row 3 of Fig. 4.14), but request rejection (row 5) remains zero while network congestion and video bitrate approach the theoretical boundary. It appears that the ability of the ISP to downgrade CP's multipath flows to singlepath (when he deems them too expensive in terms of its congestion) becomes the ultimate tool for

improving general performance of a video delivery system (considering our settings and assumptions).

While the Collaboration-extended brings even more improvement and, importantly, stability across different experiment runs, it requires both actors to implement an interaction as to exchange the decision information. Regardless of whether it is done using MPEG-SAND or a proprietary solution, such an agreement can be difficult to manage both technologically and politically. Yet, we make the demonstration in this work that for realistic access topologies and video request patterns, collaboration yields substantial gains in congestion (40-50%) at sustained video rate.

Chapter 5

Conclusions and future work

5.1 Conclusions

In the shed of a dramatic increase in video demands in the Internet that is already putting Internet Service Provider infrastructures to an exercise, research community is seeking solutions to decrease the strain of video traffic onto today's networks. This thesis is an attempt to contribute towards this goal. To achieve this, we focused on several novel techniques and concepts: Quality of Experience (QoE), video caching, HTTP Adaptive Streaming (HAS), and multipath (MP) transport with an expectation of them proving to be useful in our endeavour.

In **Chapter 2**, we started with exploring the problem of caching in HAS. Upon experimenting with Rate- and Buffer-based, and Fast MPC quality adaptation algorithms in a presence of a proxy cache, we have confirmed the reported issue of bitrate oscillations. Applying Mixed-Integer Linear Programming, we have designed a theoretical optimal boundary for a HAS system performance in a presence of a cache, and have set the QoE performance of mentioned quality adaptation algorithms against this boundary. With an objective of finding whether cache-awareness can help alleviate bitrate oscillations, we have focused on studying it with respect to Fast MPC adaptation algorithm that was among the most versatile and QoE-oriented in that period. Our numerical evaluations have demonstrated that cache-awareness for Fast MPC successfully decreases quality instability and stall duration, even in harsh and realistic network conditions (when the link bandwidth is not known and varies). We have also discussed the possibilities of achieving cache-awareness for a quality adaptation algorithm using technologies being today at our disposal.

One of these opportunities for cache-awareness happens to be a cooperation between entities in a video delivery system. We turned our research into this direction so as to study its perspectives. At the same time, we have noticed that multipath transport has a potential to

improve QoE for modern video applications such as ultra-high resolution video. Having these two techniques in mind, we have continued our research – now considering large-scale video delivery systems. First, in **Chapter 3**, we have applied Mixed-Integer Linear Programming in the task of modelling the behaviour of Internet Service Provider (ISP) and Content Provider (CP), together with their respective objectives: reducing network congestion (by optimising routing) and increasing video bitrate (by selecting an optimal cache/server). In our system, ISPs rely on in-network caching so as to reduce the backhaul traffic. On the other hand, CP is always interested in providing as high QoE as possible to their clients, which is why it is interested in selecting multipath servers providing a useful bandwidth aggregation property. We discussed the potential contradiction between these two new operational points, and by analysing the results of a numerical evaluation of their performance, we have demonstrated that this contradiction indeed takes place: using multipath does improve clients' achieved video rate, but also causes a significant increase in ISP congestion.

Continuing in line with our initial focus on collaboration, we studied whether it can bring improvement in ISP and/or CP performance. For that, we have designed two theoretical ISP – CP collaboration cases: a *Full-visibility*, where CP is artificially made to have access to the ISP network state, and *Full-collaboration*, which is a theoretical optimal boundary for cooperative performance of ISP and CP. By comparing the performance of new operational points and Full-visibility case against the boundary, we have demonstrated that collaboration promises quite a significant improvement in both ISP and CP objectives. Moreover, we have noticed that, in our settings, perfect visibility over ISP network does bring benefits for the CP that is exercising it, but on the expense of excessively harming the ISP, thus affecting fairness between two parties.

After identifying the potential in collaboration between ISP and CP, in **Chapter 4** we proceeded to designing realistic collaboration strategies, aiming to approach the theoretical performance boundary. We proposed two strategies: *Collaboration-light* and *Collaboration-extended*. In the first strategy, we consider that ISP can now impact the decision of CP server selection by "cutting" MP subflows in case it estimates that the associated increase in its congestion does not justify using MP by the CP. Considering that Collaboration-light is not an explicit collaboration by its design, we proceed with developing an algorithm that would feature joint ISP and CP decision making. Inspiring from Nash bargaining and applying Lagrangian Decomposition, we designed Collaboration-extended – a distributed collaborative algorithm with limited amount of information exchanged between parties. It aims at providing optimal joint actors' performance *and* fairness between their decisions. After developing a linear, easily-solvable version of Collaboration-extended, we have compared both our strategies against basic operational points in a non-collaborative scenario, and against the

optimal boundary. Our results have shown that Collaboration-extended potently approaches the optimal front in a fair trade-off region in a variety of realistic settings, by so fulfilling its purpose. Another interesting finding is that Collaboration-light, being simpler in its design (due to no need of information exchange between parties[1]), manages to approach the extended version in its performance. These findings enabled us to conclude on the considerable benefits of building a collaboration between ISP and CP in a multipath-enabled video delivery system. In order to assist the reader in understanding the opportunities of implementing collaboration in real networking systems, we provide an overview of up-to-date techniques that can be used for the discussed purpose.

Finally, in **Appendix A** we briefly present our efforts in obtaining an accessible HAS- and multipath-enabled network simulator that could be used in related studies so as to fill the gap between numerical evaluation approach and full-fledged real-world infrastructure studies. We explain the difficulties in achieving this with commonly-used NS-3 network simulator and its existing HAS and MPTCP models. In spite of difficulties, we have successfully obtained a distribution of this simulator that includes functional community models of MPTCP and MPEG-DASH, with our contribution being the modifications to these models so as to enable them to work together in a single distribution. We released this distribution for the community in a hope that it facilitates the research in video delivery.

Overall, this thesis presents a justification of the interest of establishing collaboration in a video delivery system. We show that it gives a potential for better coordination between different actors of the system, which inevitably results in better QoE, fairness, or both.

5.2 Future work

This thesis has demonstrated the potential of collaboration in improving video QoE and decreasing infrastructure costs, but did not discuss the financial side of this problem. In Chapters 3 and 4, ISP and CP find themselves in a slight competition, for which we have estimated a tradeoff. Ultimately, it would be interesting to connect actors' performance to financial metrics (such as infrastructure costs, or increase in revenues from improving video quality) so as to underpin and define the performance tradeoff.

In this thesis, we have used application-level metrics (such as video bitrate) so as to express the QoE. Such an assessment does not incorporate users' subjective experience, not only simplifying the analysis, but also slightly offsetting the relevance of the conclusions with respect to the subjective user experience (that is different for every person). Development of a subjective QoE model for this thesis' use-cases and linking it together with the mentioned-

[1] save for ISP needing to know maximum and minimum video bitrate

above financial considerations could result in a valuable contribution on QoE-based design of a collaborative video delivery system.

In order to focus on the intrinsic effects of collaboration in a video-delivery system, we have agreed to simplify our modelling with respect to exact TCP and HAS processes. Now having achieved our goal in modelling, we can proceed with lifting these assumptions in order to have a more realistic assessment of ISP – CP collaboration. The NS-3 distribution we have provided as a part of our work could be of great help in this task.

We have discussed that information exchange (which is required for effective collaboration in our use-cases) can be achieved with MPEG-SAND, that is defining standardised interfaces exactly for these purposes. This could be applied to designing bandwidth predictors for HAS quality adaptation. When using such an interface in its full potential (e.g., to report about other ongoing transmissions in the same area), we could, perhaps, envision a very simple and light quality adaptation that would rely on the predictor for its performance.

Appendix A

An ns-3 distribution supporting MPTCP and MPEG-DASH obtained by merging community models

As volumes of online video streaming continue to grow, researchers seek ways to make it more efficient. Apart from developing better client- or server-side solutions, the research on the video delivery over the network remains a difficult-to-approach domain owing to the need to run large-scale experiments. Such experiments should involve modern video delivery technologies like HTTP Adaptive Streaming (HAS) [148, 33] and Content Delivery Networks (CDN), as well as complex network topologies. For these reasons, most of the research in this domain remains done by teams in collaboration with large CDNs and content providers (e.g., Akamai and Netflix).

Despite that ns-3 is an alternative to having a real deployment, existing ns-3 models are quite diverse and have a rather loose compatibility between each other. It results that researchers often have to spend efforts to make several ns-3 modules gleaned from the Internet work together.

MPEG-DASH and MPTCP are two technologies that one would naturally study together. However, even though ns-3 community provides MPEG-DASH and MPTCP implementations, it happens that they are not directly usable due to incompatibilities. In this work, we present an ns-3 distribution that packages the AMuSt Framework DASH implementation [81] and the ns-3 MPTCP implementation from University of Sussex [71] such that they can be used together to nourish the flourishing research on Internet video streaming.

In the following, we present the efforts that were needed to package MPTCP and MPEG-DASH models together into ns-3. As we show further, the seemingly simple task of merging community models turned out to be tricky as they were based on different ns-3 releases. In

order to make the two models work together, we first had to overcome the incompatibilities between different ns-3 versions, and then to extend the MPTCP implementation of ns-3 to support functionalities needed by MPEG-DASH module.

This work is made as part of a work on simulating a large-scale multipath-enabled video delivery system which is now under development. The source code of the resulting distribution is available on GitHub at:
https://github.com/vitaliipoliakov/ns3-dash-mptcp

A.1 Selecting models for Multipath and MPEG-DASH

Multipath transport protocol today is represented by Multipath TCP (MPTCP). Several implementations of it exist in ns-3 [36]. Naturally, one would want to use the newest and most advanced implementation by Coudron and Secci [36]; however, in spite of offering benefits like full compliance with MPTCP specifications and compatibility with the ns-3 TCP socket API, we have found the current release of this implementation to be working unreliably in our scenario (excessive DUPACK's and inadequate subflow management when used with data transmissions bigger than 1 MB). Given our time frame, we moved to an older implementation developed by the University of Sussex [71]. This older implementation, though being developed for ns-3.19, has proven to work reliably with big amounts of data to transmit, and also responding reasonably well to path asymmetry (both bandwidth and delay) and it turned out to be faster for us to adapt it to our needs than modifying the implementation by Coudron and Secci.

A plethora of MPEG-DASH models have been implemented by the ns-3 community; in our work we use the implementation of Kreuzberger et al. [81], the AMuST-DASH framework, because it is directly built on top of a largely adopted Linux DASH library developed by Bitmovin [1]. This ns-3 module has been developed for ns-3.24.

A.2 Putting MPTCP and MPEG-DASH together

In the following we describe the main manipulations that we did to make our two models work together.

[1] https://github.com/bitmovin/libdash

A.2.1 Modifications for MPTCP

The selected implementation of MPTCP seems to be focused on making the connection initiator to be the transmitting entity, therefore some of the features of the socket do not work when the other party has to transmit data as well. In addition to that, the implementation does not directly allow one to send real data from applications. Here we discuss modifications which mitigate those limitations; unless otherwise noted, they only concern the file `src/internet/model/mp-tcp-socket-base.cc` (together with its header file) of the original distribution.

Connection receiver's limitations

Distinction between the connection initiator and connection receiver is represented by socket's methods which can only be called by the particular endpoint according to the logical flow of MPTCP. The methods specific to the connection receiver are underdeveloped, so the following modifications are required to allow it to transmit data.

The `ProcessSynRcvd` method sets up the MPTCP socket upon receiving an incoming session and respond back; however, the implementation seems to never change the default value (0 Bytes) of congestion window of the socket. We initialize the congestion window to a value of one (1) MSS as it is done for the connection initiator methods; note that it is, however, not an optimal value [34].

The `SendPendingData` checks whether the currently selected MPTCP subflow has already been established; otherwise, the next subflow is selected. However, in the latter case the connection receiver's socket fails to change the subflow and hence ceases to transmit. We have not been able to identify the cause of this, though calling `getSubflowToUse` method inside of the (mentioned above) conditional clause solves the issue without any side effects.

Transmitting real data

The `Add` and `Retrieve` methods implemented for the socket's TX and RX buffers (file `mp-tcp-typedefs.cc`) do not allow handling actual user data. In fact, they accept/return merely the amount of bytes as an argument/output correspondingly. As MPEG-DASH is based on packet contents, we have therefore added the `AddRealData` and `RetrieveRealData` methods, which effectively handle real data in the MPTCP socket.

Work in progress

The RTO functionality of the socket implementation does not seem to be finished on the connection receiver side: the corresponding method (`ReTxTimeout`) can only be executed

when called by the initiator. Removing this limitation, nevertheless, allows the socket to run without visible issues, though it has to be tested more.

Upon receiving the unordered data the socket proceeds with storing it in the right order. One of the conditional clauses checks if the Data Sequence Number (DSN) of the received mapping is smaller than the last stored DSN. In case it is, it checks the same for corresponding Subflow Sequence Numbers (SSN) and exits the program if the last is false. To the best of our knowledge, in MPTCP the latter condition cannot be false if the former condition is true; nevertheless, we have observed this happen for unknown reasons. Unable to solve it, we have removed the last condition and have not observed any issues with data ordering since then.

A.2.2 Modifications for AMuSt-DASH

The selected implementation of MPTCP does not replace the stock TCP socket, but instead comes in complement to it. Therefore, all applications written willing to use MPTCP must be updated to use this specific multipath socket – including the AMuSt-DASH model. As mentioned before, the MPTCP we use makes it fairly easy to convert applications for multipath: though the API itself is rather different, the specifics of MPTCP protocol – like subflow establishment – are done entirely by the socket itself without any need to program them in the application. As a result, the application will follow exactly the same steps for establishing/receiving an MPTCP connection than for TCP but with minor differences in methods called, as explained below.

First, each method of the application handling a socket has to cast it to an MPTCP socket using a `DynamicCast` directive.

Second, the selected implementation of MPTCP does not have a `Send(Ptr<Packet>)` method. Instead, one needs to fill the TX buffer with data using `FillBuffer`, and then call the `SendBufferedData` method to initiate data transmission.

A.3 Compiling the merged distribution

The two models mentioned above have been developed for different ns-3 versions which are not fully compatible with each other. This leads to compilation errors once the files are merged together; here we discuss them and explain fixes.

Since the implementation of MPTCP is rather complicated and spans across multiple files, we have decided to use its distribution as a substrate and merge the AMuSt-DASH (which is an application) into it. This requires copying the AMuSt-DASH-specific files (i.e., models, headers, and helpers) from `applications` module into the distribution supporting MPTCP,

without forgetting to update `src/applications/wscript` and `src/wscript` scripts to correctly build the modified modules.

The header file `src/internet/model/tcp-socket.h` has been changed between ns-3.19 and ns-3.24 as the TCP states declaration has been rethought; the version included in ns-3.19 (MPTCP) declares the states outside of the `TcpSocket` class, while the one in ns-3.24 (DASH) has them inside. We have adopted the newer version of the file, which hence required updating seldom references to the TCP states in the TCP/MPTCP socket of the substrate distribution such that they are accessed from the `TcpSocket` namespace.

Next, AMuSt-DASH requires custom string handling functions: `string_ends_width()`, `zlib_compress_string()`, and `zlib_decompress_string()`. These functions are implemented in file *src/core/model/string.cc* of the original AMuSt-DASH distribution and can safely be added in the module. It has to be noted, however, that two last functions depend on ZLib library [2], so its support has to be added in the `src/core/wscript` (exactly as it is done in the DASH distribution).

A.4 Conclusion

In spite of the several unclear points (that we have mentioned above), and the fact that the resulting distribution is based on ns-3.19, our tests until this moment suggest that the distribution is rather stable and usable for performing experiments involving MPEG-DASH and MPTCP.

Ultimately, we would like to create a stand-alone module holding the mentioned implementations of MPEG-DASH and MPTCP, that could be used with an ns-3 distribution of choice. Unfortunately, the implementation of MPTCP requires modifications to essential files such as `tcp-l4-protocol.cc`, which might differ between releases of ns-3. Therefore, creating a proper module of this kind will most likely require having MPTCP supported by the official releases of the simulator.

[2] https://zlib.net/

Appendix B

Traductions Françaises

B.1 Introduction

Depuis que les réseaux informatiques ont quitté les bien contrôlés laboratoires de la recherche universitaire, la bataille entre les fournisseurs de services Internet (FAI) et les demandes croissantes des utilisateurs a été féroce. Depuis le début, la réponse à la croissance de la demande a été de sur-approvisionner les réseaux des FAI. Cette course a été l'un des principaux moteurs des progrès des technologies de transport de données pour les réseaux locaux, métropolitains et étendus. Cependant, deux facteurs principaux rendent une telle approche insuffisante pour répondre aux attentes des utilisateurs déjà aujourd'hui.

Premièrement, la prolifération de l'accès Internet à large bande dans les années 2000 et l'émergence de plates-formes de vidéo sur Internet (notamment Youtube et les fournisseurs de vidéo à la demande en ligne) ont multiplié les demandes de vidéos. En tant qu'application consommant beaucoup de bande passante et associée à des volumes de demande sans précédent, Cisco prévoit que le trafic vidéo représentera 82% du trafic Internet d'ici 2021, contre 60% déjà observé en 2016 [35]. Ce n'est pas une limite: une très récente prolifération de périphériques permettant de visualiser et de créer du contenu vidéo Ultra Haute Définition (4096 lignes ou plus) avec des vitesses de tramage élevées (plus de 60 images par seconde) et les applications de réalité virtuelle suggèrent que les volumes de trafic vidéo seront loin d'être aujourd'hui. Le même rapport de Cisco [35] prédit par exemple une multiplication par 20 du trafic VR (réalité virtuelle) et AR (réalité augmentée) d'ici 2021.

Deuxièmement, le nombre d'utilisateurs d'Internet ne semble pas encore avoir atteint sa limite. Aujourd'hui, environ la moitié seulement de la population de la Terre a accès à Internet (4,2 milliards de personnes, selon Internet World Stats [1]), vivant pour la plupart dans

[1] https://www.internetworldstats.com/stats.htm

des pays développés ou telles en développement actif. La connectivité Internet devenant une commodité de nos jours, on peut peut-être s'attendre à ce que le reste de la population humaine devienne connecté dans les années à venir (mais peut-être pas les plus proches). En outre, le rapport [20] suggère que le nombre d'appareils compatibles Internet par habitant augmentera jusqu'à 3,5 d'ici 2021, conséquence du développement du secteur de l'Internet des objets.

Ces deux facteurs montrent à quel point l'avenir de la demande Internet peut devenir insupportable, tant pour les FAI que pour les avancées technologiques dans le secteur des télécommunications. Dans cette perspective, une partie de la recherche actuelle en matière d'interconnexion de réseaux tente de relever les défis à venir en améliorant l'efficacité de l'utilisation des infrastructures, plutôt qu'en augmentant ses capacités, en particulier dans le domaine de la vidéo sur Internet.

L'une des techniques de base permettant de réduire le volume de trafic vidéo circulant sur les réseaux est la mise en cache dans le réseau[97, 163, 120]. Etant principalement développée comme une solution pour améliorer la latence d'accès du client aux services Web, l'idée de stocker du contenu à proximité de l'utilisateur s'est avérée avoir un potentiel dans le domaine de la diffusion vidéo. Très souvent, les topologies de réseau sont conçues avec une plus grande capacité proche de l'utilisateur [25, 37]. Placer en cache dans la partie du réseau bien approvisionnée par l'utilisateur et bien fournie peut aider le FAI à éviter plusieurs fois le trafic vidéo (et autre) sur l'ensemble de son réseau de backhaul – mais plutôt à le transporter une seule fois puis diffusez à partir du cache pour les autres utilisateurs intéressés. La recherche dans la domaine du caching a récemment été renforcée avec l'introduction de Mobile Edge Computing (MEC [17, 56]) et de Edge Caching [94]. Dans le contexte de la diffusion vidéo, ces idées insistent sur la possibilité d'utiliser le bord du réseau (principalement sans fil) pour mettre en cache du contenu vidéo et effectuer un transcodage afin de réduire le volume de trafic vidéo dans le backhaul et au-dessus.

Jusqu'à récemment, il n'existait aucune norme spécifique pour la transmission vidéo sur Internet et ses fonctionnalités, en particulier pour l'adaptation de la qualité et la possibilité d'être mis en cache. Le protocole en temps réel (RTP [140]) et sa famille ont été conçus pour faciliter la diffusion en continu de contenu multimédia sur des réseaux Best-Effort. Cela inclut la livraison en Multicast, la surveillance des flux, l'identification de la payload et le Sequence Timestamping. Cependant, étant un protocole OSI, RTP n'a pas abordé le format de stockage du contenu ni son adaptation en termes de qualité. HTTP pourrait également être utilisé pour transporter des supports, tout en le considérant comme un contenu Web normal. Les headers *Range-request* peuvent être utilisés pour exercer un contrôle limité sur la lecture vidéo. Non seulement cela compliquait la tâche de la mise en cache (avec des

objets trop grands ou trop uniques, comme les bytes range-request de requête pouvant être différentes pour chaque utilisateur), mais cela rendait également l'adaptation de la qualité plutôt complexe. L'émergence du concept de streaming adaptatif HTTP [122, das] a permis de progresser considérablement vers la résolution de ces deux tâches. Les fichiers vidéo étant transmis par HTTP et stockés dans des segments prédéfinis (également appelés *chunks*) qui sont identiques pour tout les utilisateurs au sein du service, il est finalement devenu possible de mettre en cache facilement (même de manière transparente) du contenu vidéo et de adapter leur qualité sans calculs compliqués des range-requests. Une partie indissociable de HAS est son algorithme d'adaptation de la qualité; Ayant débuté avec de simples estimations du débit descendante (algorithme à base de débit, Rate-Based algorithm), les recherches dans ce sens ont fait beaucoup de chemin pour même exploiter le machine learning pour la sélection de qualité [102].

Avec le développement du transport de données par trajets multiples (*multipath*, tels que MPTCP [15]) et le concept émergent de Multi-RAT [27], une nouvelle perspective sur la diffusion de contenu lourd aux clients s'est ouverte. À l'origine, le transport multipath visait principalement la résilience [90, 137]; cependant, notre objectif est d'utiliser son potentiel d'agrégation de la bande passante afin d'améliorer la connectivité des clients. Aujourd'hui, malgré les progrès récents des technologies de la radio mobile, la bande passante moyenne de la liaison descendante LTE est inférieure à 15 Mbps dans de nombreux pays développés (par exemple, 13,4 Mbps en France et 12,3 Mbps aux Etats-Unis [fra]), ce qui ne permet pas la distribution de futures applications vidéo comme décrit ci-dessus [3]. Même l'avènement de la 5G laisse entrevoir des améliorations de la bande passante d'accès radio pour les utilisateurs, bien que l'observation des chiffres LTE juste au-dessus permette de se demander si la promesse d'un débit de liaison descendante de 1 Gbps serait valable dans un environnement réel avec des appareils grand public classiques. Étant donné que la consommation de vidéo sur mobile devient de plus en plus répandue, le développement d'une solution permettant de réduire la connectivité mobile lente (en moyenne) devient une effort importante. MPTCP donne la possibilité d'utiliser plusieurs connectivités sans fil (par exemple, Wi-Fi et LTE) simultanément afin d'obtenir une bande passante de connectivité globale plus élevée qui peut aider à servir des applications vidéo exigeantes.

Différement des avancées technologiques susmentionnées dans la diffusion vidéo sur Internet, le concept de qualité d'expérience aide les chercheurs à mesurer et à quantifier les améliorations apportées au domaine de la diffusion vidéo (entre autres). Contrairement au concept de qualité de service (QoS), une latence faible ou une bande passante élevée risquent de ne pas toujours offrir une bonne expérience utilisateur [75]. Une évaluation subjective, telle que le Mean Opinion Score (MOS [129, 154]), ou des métriques vidéo au niveau de

l'application, telles que la durée des interruptions ou le Peak Signal-To-Noise Ratio (PSNR) des images vidéo, peuvent également être utilisées afin d'évaluer l'expérience utilisateur exacte. La focalisation directe sur ces dernières fournit donc une bonne base pour concevoir et tester de nouvelles techniques de diffusion vidéo.

Cette thèse présente un travail sur l'intersection de ces technologies (caching, HAS, transport par Multipath), dans le but d'améliorer la qualité de l'expérience dans les applications vidéo dans la perspective des défis futurs mentionnés. Nous considérons que le caching est le moyen important pour réduire le volume de trafic vidéo dans les réseaux de FAI, tandis que HAS est la clé pour fournir aux utilisateurs de vidéo autant de QoE que possible compte tenu de l'évolution du niveau de QoS dans les réseaux au mieux. Le transport par Multipath, à son tour, devrait permettre la consommation de types de vidéo à très haut débit. Leur utilisation ensemble pour atteindre notre objectif s'est toutefois révélée difficile. Les problèmes et les contributions de cette thèse sont résumés comme suit:

- Il a été signalé que l'algorithme d'adaptation de la qualité HAS Rate-Based peut être trompé par la présence d'en cache transparent dans le réseau; cet effet a été nommé *bitrate oscillations* [84]. Cela est dû à l'incapacité d'un tel algorithme de prédire la provenance d'un segment vidéo entrant (de cache proche ou de serveur distant) et, par conséquent, de son débit de téléchargement. Nous menons une étude de ce phénomène dans le but d'identifier l'ampleur de son effet sur la QoE vidéo. Pour ce faire, nous posons les questions suivantes. Si nous imaginions un algorithme d'adaptation de qualité parfaitement compatible avec le cache et connaissant toujours le débit de téléchargement des segments futurs quelle que soit la qualité, quel en serait le résultat en termes de métriques QoE dans le cas de la présence en cache, et à quelle distance limite parfaite les algorithmes couramment implémentés effectueraient-ils? Étant donné que le nouvel algorithme d'adaptation de la qualité FastMPC rapide a été développé en tant qu'algorithme basé sur l'optimisation, quel serait son rendement en présence de cache? Pour répondre aux questions posées, cette thèse propose les contributions suivantes:

  1. En appliquant le Mixed-Integer Linear Programming (MILP), nous définissons une limite de performance pour un algorithme d'adaptation de la qualité en présence de cache;
  2. Nous testons les algorithmes d'adaptation Rate-Based et Buffer-Based par rapport à cette limite dans un banc d'essai expérimental réaliste (basé sur un lecteur multimédia VLC modifié), en surveillant d'importants paramètres de QoE (qualité vidéo, durée des interruptions et changements de qualité) et en observant une

poténtiel pour l'amélioration de leurs performances – par rapport à la limite théorique;

3. Nous testons un nouvel algorithme d'adaptation FastMPC et constatons que celui-ci, bien que très proche de la limite, offre des performances sous-optimales à certains clients. Nous examinons ensuite si la connaissance de l'état due le cache (c'est-à-dire connaissance de la provenance des segments futurs, mais pas leur débit exact) peut contribuer à atténuer cet inconvénient – et en concluons qu'elle peut effectivement réduire le nombre de changements de qualité pour ces clients et améliorer de manière générale performances des autres en termes de changements de qualité et de durée des interruptions;
4. Nous discutons de la possibilité d'implémentation de la reconnaissance du cache.

- Nous voyons un fort potentiel d'utilisation de la connectivité multiple sur le périphérique du client en termes d'agrégation de bande passante (pouvant conduire à une QoE de service plus élevée). D'autre part, les fournisseurs du contenu (*Content Providers*, CP) et leurs réseaux de distribution de contenu (*Content Delivery Networks*, CDN) déploient souvent leurs propres caches chez les FAI en plus des serveurs déployés dans leurs propres locaux. Étant donné que les caches des FAI ne sont pas conçus pour être accessibles de l'extérieur de leurs réseaux, nous pouvons soit servir les clients à partir de caches internes au réseau en utilisant le transport à un seul chemin (pression de trafic moindre pour le FAI mais peut-être moins la QoE des clients), soit renoncer aux avantages de la mise en cache afin d'utiliser le serveur extérieur en utilisant le transport Multipath (contrainte de trafic FAI plus élevée mais possiblement plus de QoE d'utilisateur). Le fait d'avoir deux parties, à savoir le FAI et le CP, avec des tels intérêts potentiellement contradictoire, peut rendre le fonctionnement d'un système de diffusion vidéo sous-optimal dans un scénario à Multipath. Néanmoins, le transport Multipath apporte de nouveaux régimes opérationnels à un système de diffusion vidéo en introduisant une telle contradiction. Quels avantages ces nouveaux régimes opérationnels apportent-ils aux différents intérêts du fournisseur de contenu (qualité client supérieure) et du FAI (nécessité de servir des services nouveaux et plus exigeants, qualité pour tous les services, congestion réduite)? Ou, en d'autres termes, quelle stratégie convient le mieux à ces deux acteurs et pour quels cas d'utilisation: servir à partir du haut du réseau pour potentiellement atteindre une bande passante supérieure, ou plus près du bord du client pour réduire la congestion du réseau? Ensuite, dans le cas où CP et FAI collaboraient afin d'obtenir une performance mutuellement bénéfique d'un système de transmission vidéo en termes d'objectifs, en quoi cette

performance serait-elle différente du scénario actuel (où fournisseurs FAI et CP indépendants ne collaborent pas?) Pour étudier ces nouvelles options, cette thèse fournit les contributions suivantes:

1. À l'aide de MILP, nous définissons une limite théorique pour une performance parfaite commune FAI – CP en termes des leurs objectifs (congestion du réseau et débit vidéo) dans un système avec Multipath;
2. De la même manière, nous modélisons le comportement de deux acteurs d'un système de diffusion vidéo: un FAI et un CP. Nous effectuons une évaluation numérique des nouveaux régimes opérationnels, apportés par le Multipath, par rapport aux objectifs des acteurs. Nous constatons que, selon la limite parfaite, les performances des deux acteurs peuvent être considérablement améliorées si nous nous écartons des stratégies non collaboratives, en réduisant jusqu'à 50% la congestion du réseau dans le même niveau de qualité vidéo dans nos paramètres de simulation;
3. Nous concevons deux schémas de collaboration qui impliquent une interaction différente et les testons par rapport à la limite théorique parfaite. Notre évaluation prouve que ces stratégies, malgré leur légèreté et leur répartition (grâce à Lagrangian Relaxation), apportent une amélioration substantielle de la performance des FAI et CP par rapport à la stratégie non collaborative, ainsi qu'à l'équité entre leurs objectifs;
4. Nous discutons de l'applicabilité réelle de ces schémas de collaboration dans le monde réel.

- La recherche sur les systèmes de transmission vidéo impliquant une HAS et de Multipath nécessite souvent de se concentrer sur les grands systèmes. Les grands groupes de recherche ayant des relations dans le secteur des fournisseurs de contenu peuvent utiliser de vraies infrastructures pour mener leurs recherches. D'autres chercheurs se retrouvent avec des simulations ou des évaluations numériques pour ce type d'études. Cependant, il n'existe aucun simulateur de réseau open-source qui prendrait en charge le HAS et le transport Multipath. Il existe plusieurs modèles de communautés isolées pour MPEG-DASH et MPTCP (représentant les deux technologies) pour le simulateur de réseau NS-3 [51], mais nous avons trouvé très difficile de les intégrer ensemble afin d'obtenir une distribution avec les deux fonctions intégré. Un simulateur de réseau doté de telles fonctionnalités aiderait la communauté à poursuivre ses recherches de qualité sur la diffusion vidéo à grande échelle, au-delà des évaluations numériques. Dans cette

thèse, nous présenterons brièvement nos efforts pour produire une distribution NS-3 incluant MPEG-DASH et MPTCP.

Le manuscrit est organisé en quatre chapitres, mis à part cette introduction, et une annexe.

Un bref aperçu de l'évolution de la diffusion vidéo sur Internet et de l'état actuel des connaissances, qui fait l'objet du chapitre 1 juste après cette introduction.

Adaptation de la qualité HAS et caching: au chapitre 2, nous confirmons que, dans certaines conditions, l'adaptation de la qualité basée sur le débit (Rate-Based) de HAS peut être trompée par la présence du cache, ce qui entraîne des oscillations du débit. Nous formulons une limite optimale pour l'exécution d'un algorithme hypothétique d'adaptation parfaitement compatible avec le cache pour notre scénario. Nous testons ensuite les algorithmes Rate-Based et Buffer-Based par rapport à ce problème et comparons leurs performances QoE à la limite. Après cela, nous présentons un algorithme basé sur l'optimisation récemment proposé appelé Fast MPC et procédons à son évaluation par rapport à la limite optimale. Nous constatons que Fast MPC souffre toujours de la présence du cache dans nos paramètres réalistes, nous discutons d'une extension compatible avec le cache. Nous démontrons que la reconnaissance du cache peut considérablement réduire le nombre de changements de qualité et la durée de interruptions des clients parmi les premiers à demander une vidéo populaire au réseau extérieur. Nous expliquons comment il est possible d'obtenir la reconnaissance du cache pour un algorithme d'adaptation de la qualité et concluons que l'infrastructure de caching peut tirer avantage d'une gestion par le fournisseur du contenu (CP) ou d'une signalisation explicite ou implicite avec le CP ou ses utilisateurs.

Interaction entre Multipath et Caching: après la conclusion précédente, au chapitre 3, nous prenons un peu de recul et considérons un système de diffusion vidéo au sens large. Nous considérons un système activé par Multipath où les FAI et CDN peuvent potentiellement avoir des préférences différentes sur les trajets multiples. Nous essayons d'évaluer l'avantage (ou son absence) de l'utilisation du transport de données par Multipath pour un tel système. Nous commençons par modéliser les problèmes d'optimisation des deux acteurs, FAI et CDN, puis en concevant un banc d'essai de simulation utilisant une topologie d'opérateur mobile. Ensuite, afin de déterminer les avantages potentiels de la collaboration entre FAI et CP, nous définissons une limite théorique pour leur performance commune afin de déterminer s'il existe une marge d'amélioration – ce qui se trouve être assez important (en termes d'objectifs des acteurs).

Stratégies de collaboration: au chapitre précédent, nous avons démontré l'intérêt de la collaboration entre les fournisseurs de services Internet et CDN pour mieux tirer parti du caching et du Multipath. Au chapitre 4, nous développons l'idée de collaboration en introduisant deux modèles distribués conjoints avec différents volumes d'échange d'informations.

Ensuite, nous procédons à une évaluation numérique des stratégies élaborées. Enfin, nous discutons de la possibilité de mise en œuvre de nos deux stratégies de collaboration en tenant compte des contraintes politiques et technologiques réelles.

Efforts in NS-3: dans l'Annexe A, nous présentons brièvement les motivations et passons aux détails de la distribution résultante et de ses limites.

B.2 Résumé du Chapitre 2

Les bitrate oscillations [85] apparaissent lorsque la capacité de connectivité entre le client et le cache est supérieure à celle de la connectivité jusqu'au serveur vidéo. Parce que le client ne connaît pas la provenance du prochain chunk vidéo demandé, il ne peut pas prédire correctement son débit de téléchargement dans de telles conditions. Cela conduit à des sur- et sous-estimations de la qualité vidéo à sélectionner. Le chapitre commence par développer une limite théorique de performance d'un algorithme de sélection de qualité HAS. Inspirés de Yin et al. [174], nous définissons cette limite comme suit:

$$\max \sum_{r=1}^{R} \Big(\sum_{k=1}^{K} Q_{r,k} - \lambda \sum_{k=1}^{K-1} \Delta Q_{r,k+1} - \mu \sum_{k=1}^{K} dBp_{r,k} - \mu^{s} T^{s} \Big) \tag{B.1}$$

L'équation ci-dessus maximise la somme pondérée de différentes mesures de QoE pour tous les chunks vidéo $k \in K$ sur plusieurs sessions vidéo consécutives $r \in R$. Les paramètres de qualité d'expérience pris en compte sont la qualité vidéo, son instabilité, la durée des interruptions et le délai de démarrage. Leurs poids définissent leur importance dans la **fonction résultante**.

Nous testons ensuite la gravité de cet effet sur les algorithmes actuels basés sur le taux (RBA) et sur le buffer (en particulier la famille BBA [59]). Nous réalisons cette expérience dans un environnement virtualisé contrôlé et utilisons un logiciel VLC instrumenté en tant que logiciel de lecture vidéo. Les Illustrations 2.2 – 2.4 illustrent les possibilités d'amélioration en termes de métriques de qualité d'expérience pour RBA et BBA en présence d'un cache.

Nous poursuivons notre étude en évaluant l'algorithme FastMPC [174], basé sur la résolution d'un problème d'optimisation optimisant la QoE pour une courte fenêtre de séquences de séquences vidéo. Nous effectuons nos évaluations dans un environnement de simulation Python. Les figures 2.9 – 2.11 montrent comment FastMPC souffre également des oscillations du débit, principalement pour les premiers clients.

Nous définissons ensuite un prédicteur de bande passante parfaitement compatible avec le cache pour FastMPC, qui connaît toujours la provenance de tous les fragments vidéo (mais pas leur bande passante exacte, étant différente de la limite théorique). L'illustration 2.12

montre l'amélioration des performances du FastMPC sensible au cache, suivie d'une analyse de sensibilité.

Nous concluons que la connaissance du cache pour FastMPC peut considérablement réduire l'instabilité de la qualité pour plusieurs clients qui regardent la même vidéo, tout en augmentant considérablement l'occupation de buffer de lecture (ce qui permet de faire face aux changements de bande passante dus à la nature best-effort du réseau). Nous expliquons également qu'il est difficile d'obtenir une reconnaissance du cache sans une collaboration entre l'infrastructure de mise en cache et les clients.

B.3 Résumé du Chapitre 3

Afin d'étudier à grande échelle un système de diffusion vidéo par multipath, nous appliquons MILP pour modéliser le comportement de ses principaux acteurs: FAI et CP. Nous considérons le réseau comme décrit à l'Illustration B.1, qui contient un serveur multipath externe, des caches dans les locaux des FAI et des clients connectés aux deux FAI en même temps.

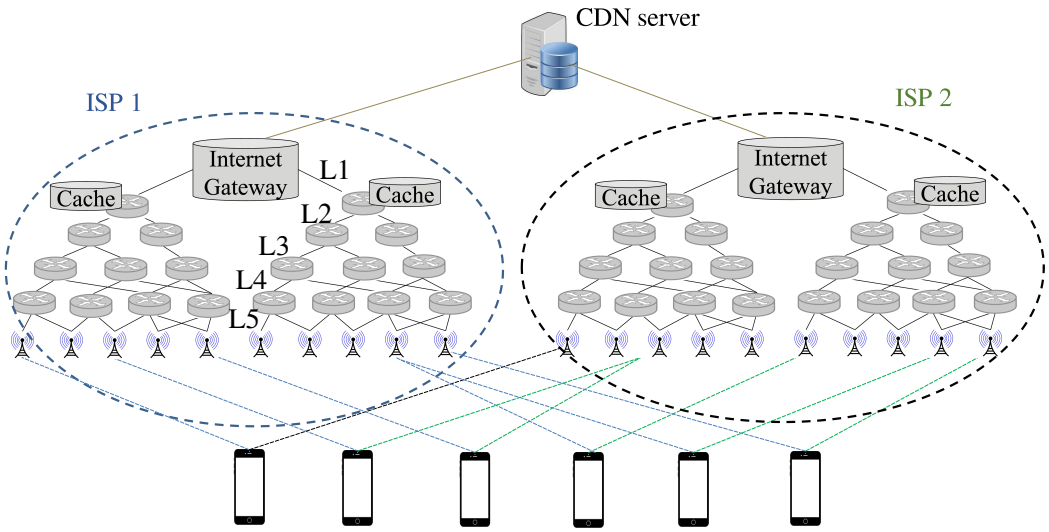


Fig. B.1 Clients connected to two access networks, with in-network caches and the external CDN server. "L1", "L2" and so on indicate the ISP topology level.

Dans notre scénario, les FAI effectuent une optimisation du routage périodique afin de réduire la congestion de leur réseau [63], modélisée comme dans l'équation 3.2. Le CP est chargé de la sélection du serveur ou cache afin d'optimiser la qualité vidéo obtenue par le client, conformément à l'équation de formulation 3.1. Ensemble, ils forment un système de diffusion vidéo dont le fonctionnement peut être exprimé sous la forme d'algorithme 1.

En utilisant notre évaluation numérique testée, nous testons si le multitrajet peut apporter des améliorations de performances en termes d'économie de congestion et / ou de qualité vidéo des clients atteinte. Dans nos configurations réalistes, nous concluons que lorsque le réseau est bien dimensionné pour la charge vidéo entrante, Multipath contribue à améliorer l'objectif du CP (qualité vidéo), mais engendre également un congestion important des réseaux des FAI (aggravant ainsi l'objectif de ce dernier) (Figures 3.3 et 3.4).

Nous estimons que l'incohérence des décisions des deux parties conduit à une telle inéquité lors de l'utilisation de Multipath. Afin de vérifier cette hypothèse, nous modélisons deux autres modes de fonctionnement du système: Full-visibility (lorsque le CP connaît l'état du réseau du FAI, Equation 3.3) et Full-collaboration, où la sélection du serveur est effectuée avec prise en compte des effets de la décision sur la congestion du FAI (équation 3.4). La collaboration complète est modélisée comme une limite optimale pour la performance conjointe FAI – CDN, en définissant ainsi une **courbe de Pareto**, utilisée pour comparer les autres approches de leur performance. L'Illustration 3.9 présente cette comparaison; on peut en conclure qu'une collaboration parfaite peut apporter des améliorations significatives en termes d'objectifs des deux acteurs, par rapport aux approches non collaboratives. Conscient de ce résultat, le chapitre suivant présente les travaux sur la conception d'algorithmes réalistes permettant d'implémenter la collaboration entre les FAI et le CP.

B.4 Résumé du Chapitre 4

Nous avons l'intention de concevoir des modèles de collaboration entre FAI et CP pour atteindre deux objectifs: (i) effet de levier actif sur la congestion au niveau des FAI et (ii) de meilleures estimations de l'état du FAI par le CP. Instrumenter le FAI avec (i) signifie que ce dernier serait capable d'exprimer ses intérêts dans le processus de diffusion vidéo. Obtenir (ii) pour le CP signifierait une estimation plus rapide et plus précise de la situation du trafic au niveau du FAI, ce qui permettra au CP de prendre des décisions en connaissance de cause conformément à l'état du réseau du FAI et à ses intérêts.

Nous proposons deux modèles de collaboration: Collaboration-light et Collaboration-extended. Le premier ne met en œuvre que le premier objectif du passage précédent – sur la congestion des FAI. Dans le chapitre précédent, nous avons remarqué que le Multipath introduit une équité dans le système dans nos paramètres, à la fois bénéfique pour le CP et préjudiciable pour le FAI. Par conséquent, nous donnons au FAI la possibilité de couper les sous-flux Multipath si l'estimation que l'avantage de Multipath pour la CP est trop onéreuse pour FAI. Nous formulons ce comportement dans l'équation 4.1 et décrivons son fonctionnement dans l'algorithme 2.

Ensuite, nous procédons à la mise en œuvre du deuxième objectif. Inspirant de [63] et de la Nash bargaining [114], nous développons un modèle de décision conjointe distribué qui doit échanger un nombre limité d'informations entre les parties afin d'informer le CP sur l'état du réseau de FAI sans divulguer sa topologie ni les occupations exactes des ses canals de communication. Ce modèle est formulé dans les équations 4.8, 4.9 et 4.10. Sachant qu'une telle formulation est compliquée à résoudre du fait de sa non-linéarité, nous la simplifions à cet égard. Les formulations résultantes, présentées aux équations 4.13, 4.14, et 4.15 implémentent les simplifications, et l'algorithme 3 présente le Collaboration-extended pour une collaboration conjointe FAI-CP.

Nous procédons ensuite à une évaluation numérique des modèles conçus et à une analyse de sensibilité. La figure B.2 montre comment nos stratégies se comparent aux points opérationnels de base et à la limite optimale. Ces résultats démontrent que notre solution est capable d'atteindre la limite optimale en termes de performance des acteurs, en leur apportant une équité sans compromettre leurs performances.

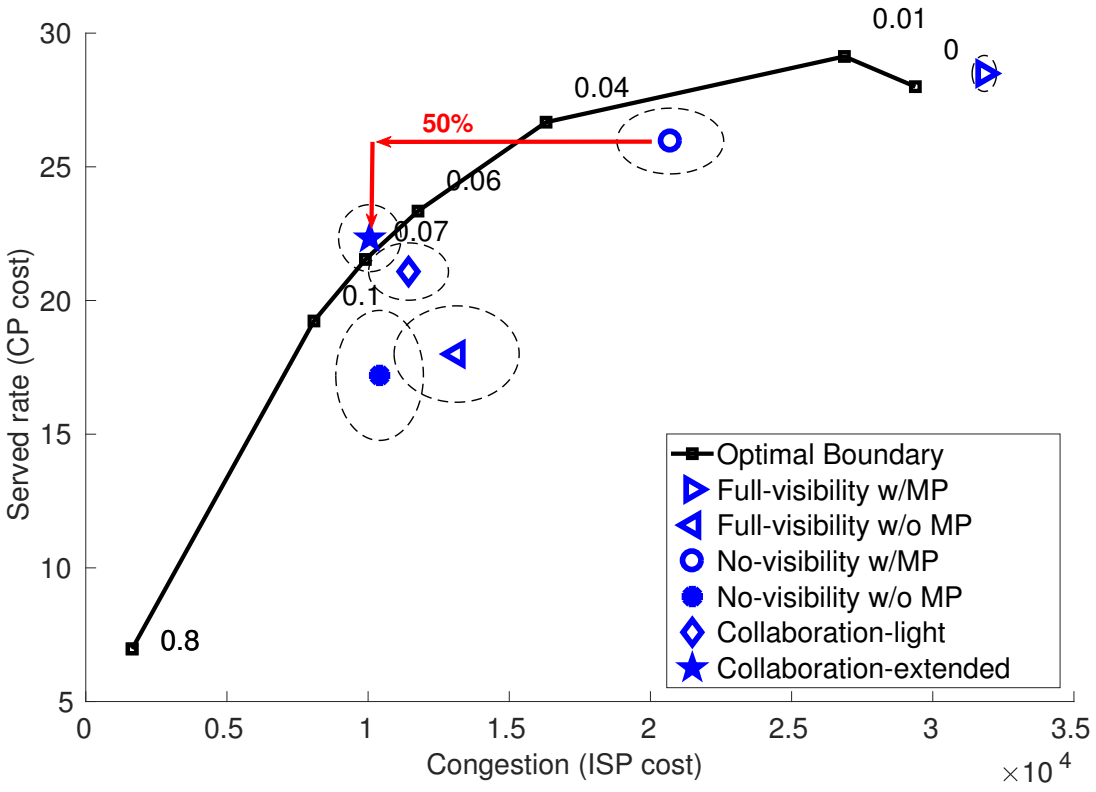


Fig. B.2 Performance of different strategies as compared to themselves and the theoretical boundary (full-collaboration) at base settings, 2 minutes long requests. Confidence intervals in both dimensions are shown as ellipses around respective points.

Enfin, nous discutons de l'applicabilité dans le monde réel d'une telle collaboration faisant appel à des techniques et approches modernes.

B.5 Conclusion

Face à l'augmentation spectaculaire des demandes vidéo sur Internet, qui mettent déjà à rude épreuve les infrastructures des fournisseurs de services Internet, la communauté des chercheurs est à la recherche de solutions pour réduire l'impact du trafic vidéo sur les réseaux actuels. Cette thèse est une tentative de contribuer à cet objectif. Pour ce faire, nous nous sommes concentrés sur plusieurs techniques et concepts novateurs: qualité d'expérience (QoE), caching, streaming adaptatif HTTP (HAS) et transport par trajets multiples (Multipath), en espérant qu'ils s'avéreraient utiles pour notre effort.

Dans **Chapitre 2**, nous avons commencé par explorer le problème de la caching dans HAS. Après avoir expérimenté des algorithmes d'adaptation de la qualité basés sur le débit et sur le buffer, ainsi que le FastMPC, en présence d'un cache proxy, nous avons confirmé le problème signalé des bitrate oscillations. En appliquant la MILP, nous avons conçu une limite optimale théorique pour les performances d'un système HAS en présence d'en cache et avons comparé les performances QOE des algorithmes d'adaptation de la qualité mentionnés. Dans le but de déterminer si la connaissance du cache peut aider à réduire les oscillations du débit, nous nous sommes concentrés sur son utilisation pour l'algorithme d'adaptation FastMPC, l'un des plus polyvalents et orienté sur la QoE de cette période. Nos évaluations numériques ont démontré que la connaissance du cache pour Fast MPC réduit efficacement l'instabilité de la qualité et la durée des interruptions, même dans des conditions de réseau difficiles et réalistes (lorsque la bande passante de la liaison n'est pas connue et varie). Nous avons également discuté des possibilités d'atteindre la connaissance du cache pour un algorithme d'adaptation de la qualité utilisant les technologies dont nous disposons aujourd'hui.

Une de ces possibilités de reconnaissance de l'état du cache se trouve être une coopération entre entités dans un système de diffusion vidéo. Nous avons orienté nos recherches dans cette direction afin d'en étudier les perspectives. Dans le même temps, nous avons constaté que le transport Multipath pouvait potentiellement améliorer la qualité d'expérience des applications vidéo modernes telles que la réalité virtuelle. Gardant ces deux techniques à l'esprit, nous avons poursuivi nos recherches sur les systèmes de diffusion vidéo à grande échelle. Tout d'abord, dans **Chapitre 3**, nous avons appliqué la MILP à la tâche de modélisation du comportement du FAI et du fournisseur de contenu (Content Provider, CP), ainsi que leurs objectifs respectifs: réduire la congestion du réseau (par optimisation du routage) et augmentation du débit vidéo (en sélectionnant en cache/serveur optimal). Dans notre système,

les FAI s'appuient sur la caching sur le réseau pour réduire le trafic de backhaul. D'autre part, le CP est toujours intéressé à fournir à ses clients la QoE la plus élevée possible. C'est pourquoi il est intéressé par la sélection de serveurs Multipath externes offrant une propriété utile d'agrégation de bande passante. Nous avons discuté de la contradiction potentielle entre ces deux nouveaux régimes opérationnels et, en analysant les résultats d'une évaluation numérique de leurs performances, nous avons montré que cette contradiction existait bel et bien: l'utilisation de la Multipath améliorait le débit vidéo atteint des clients, mais provoquait également une augmentation importante de la congestion des FAI.

Dans la continuité de notre objectif initial de collaboration, nous avons étudié si cela pouvait améliorer les performances des FAI et/ou du CP. Pour cela, nous avons conçu deux cas de collaboration FAI – CP théoriques: un *Full-visibility*, où CP est artificiellement créé pour avoir accès à l'état du réseau du FAI, et *Full-collaboration*, qui est une limite optimale théorique pour la performance coopérative des FAI et CP. En comparant les performances des nouveaux régimes opérationnels et des cas de Full-visibility par rapport à la limite, nous avons démontré que la collaboration promettait une amélioration significative des objectifs des FAI et du programme de partenaires. De plus, nous avons constaté que, dans nos contextes, le Full-visibility sur le réseau des FAI présente des avantages pour le CP qui l'exerce, mais aux dépens d'un préjudice excessif pour le fournisseur de services, affectant ainsi l'équité entre les deux parties.

Après avoir identifié le potentiel de collaboration entre FAI et CP, dans **Chapitre 4**, nous avons procédé à la conception de stratégies de collaboration réalistes, visant à approcher la limite de performance théorique. Nous avons proposé deux stratégies: *Collaboration-light* et *Collaboration-extended*. Dans la première stratégie, nous considérons que le FAI peut désormais influer sur la décision de sélection du serveur du CP en «coupant» les sous-flux Multipath du CP au cas où il estime que l'augmentation correspondante de sa congestion ne justifie pas leur l'utilisation par le CP. Considérant que Collaboration-light n'est pas une collaboration explicite par sa conception, nous développons un algorithme qui prendrait en compte la prise de décision conjointe entre FAI et CP. Inspirant des Nash bargaining et de l'application de Lagrangian decomposition, nous avons conçu Collaboration-extended, un algorithme collaboratif distribué avec une quantité limitée d'informations échangées entre les parties. Son objectif est de fournir des améliorations en performances QoE *et* en équitabilité pour les acteurs, conjoints entre leurs décisions. Après avoir développé une version linéaire et facile à résoudre de Collaboration-extended, nous avons comparé nos stratégies aux régimes opérationnels de base dans un scénario non collaboratif et à la limite optimale. Nos résultats ont montré que Collaboration-extended est capable d'approcher la limite optimal dans une région de compromis équitables, dans divers contextes réalistes, en remplissant ainsi son

objectif. Une autre conclusion intéressante est que Collaboration-light, étant plus simple dans sa conception (ne nécessitant aucun échange d'informations entre les parties [2]), parvient à approcher la version étendue dans ses performances. Ces résultats nous ont permis de conclure aux avantages considérables de la mise en place d'une collaboration entre FAI et CP dans un système de transmission vidéo par trajets multiples. Afin d'aider le lecteur à comprendre les possibilités de mise en œuvre de la collaboration dans des systèmes de réseau réels, nous fournissons un aperçu des techniques actuelles pouvant être utilisées dans le but discuté.

Enfin, dans **Annexe A**, nous décrivons brièvement nos efforts pour obtenir un simulateur de réseau accessible utilisant la HAS et le Multipath qui pourrait être utilisé dans des études connexes afin de combler le fossé entre l'approche de l'évaluation numérique et les études d'infrastructure dans le monde réel, à part entière. Nous expliquons les difficultés rencontrées pour y parvenir avec le simulateur de réseau NS-3 couramment utilisé et ses modèles HAS et MPTCP existants. Malgré les difficultés, nous avons réussi à obtenir une distribution de ce simulateur comprenant des modèles de communauté fonctionnelle de MPTCP et MPEG-DASH, notre contribution étant les modifications apportées à ces modèles afin de leur permettre de travailler ensemble dans une distribution unique. Nous avons publié cette distribution pour la communauté dans l'espoir que cela facilitera la recherche en diffusion vidéo.

Dans l'ensemble, cette thèse présente une justification de l'intérêt d'établir une collaboration dans un système de diffusion vidéo. Nous montrons que cela offre un potentiel pour une meilleure coordination entre les différents acteurs du système, ce qui entraîne inévitablement une meilleure QoE, une meilleure équité, ou les deux.

B.6 Travail futur

Cette thèse a démontré le potentiel de la collaboration pour améliorer la QoE vidéo et réduire les coûts d'infrastructure, mais n'a pas abordé l'aspect financier de ce problème. Aux chapitres 3 et 4, FAI et CP se retrouvent dans une légère concurrence, pour laquelle nous avons estimé un compromis. En fin de compte, il serait intéressant de relier la performance des acteurs à des métriques financières (telles que les coûts d'infrastructure ou l'augmentation des revenus grâce à l'amélioration de la qualité vidéo) afin de renforcer et de définir le compromis de performance.

Dans cette thèse, nous avons utilisé des métriques au niveau de l'application (telles que le débit vidéo) afin d'exprimer la QoE. Une telle évaluation ne tient pas compte de l'expérience

[2] sauf pour les FAI ayant besoin de connaître le débit vidéo maximal et minimal

subjective des utilisateurs, non seulement en simplifiant l'analyse, mais en compensant légèrement la pertinence des conclusions par rapport à l'expérience subjective de l'utilisateur (différente pour chaque personne). Le développement d'un modèle QoE subjectif pour les cas d'utilisation de cette thèse et sa liaison aux considérations financières susmentionnées pourraient apporter une contribution précieuse à la conception d'un système de diffusion vidéo collaborative basé sur la QoE.

Afin de nous concentrer sur les effets intrinsèques de la collaboration dans un système de diffusion vidéo, nous avons convenu de simplifier notre modélisation en ce qui concerne les processus TCP et HAS exacts. Maintenant que notre objectif de modélisation a été atteint, nous pouvons **lever** ces hypothèses afin d'obtenir une évaluation plus réaliste de la collaboration FAI – CP. La distribution NS-3 que nous avons fournie dans le cadre de notre travail pourrait être d'une grande aide dans cette tâche.

Nous avons discuté de l'échange d'informations (qui est nécessaire pour une collaboration efficace dans nos cas d'utilisation) peut être réalisé avec MPEG-SAND, c'est-à-dire la définition d'interfaces normalisées exactement à ces fins. Cela pourrait être appliqué à la conception de prédicteurs de bande passante pour l'adaptation de la qualité HAS. Lorsqu'on utilise une telle interface dans toute son potentiel (par exemple, pour signaler d'autres transmissions en cours dans le même domaine), on pourrait peut-être envisager une adaptation de qualité très simple et légère qui reposerait sur le prédicteur pour ses performances.

List of publications

Poliakov, V., Sassatelli, L., & Saucez, D. (2016, April). Impact of caching on http adaptive streaming decisions: towards an optimal. *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*.

Poliakov, V., Sassatelli, L., & Saucez, D. (2016, December). Case for caching and Model Predictive Control quality decision algorithm for HTTP Adaptive Streaming: is cache-awareness actually needed? In *Globecom Workshops (GC Wkshps),* 2016 IEEE.

Poliakov, V., Sassatelli, L., & Saucez, D. (2018, May). Adaptive Video Streaming, Multipath and Caching: Can Less Be More? In *2018 IEEE International Conference on Communications (ICC)*. 2018 IEEE.

Poliakov, V., Saucez, D., & Sassatelli, L. (2018, June). An ns-3 distribution supporting MPTCP and MPEG-DASH obtained by merging community models. In *WNS3 2018-Workshop on ns-3*.

References

[fra] Global state of mobile networks (08/2016). www.opensignal.com/reports/2016/08/global-state-of-the-mobile-network. Accessed: 10/2017.

[das] Iso 23009–1: 2014: Information technology-dynamic adaptive streaming over http (dash).

[3] (2017). Youtube recommended upload encoding settings.

[4] Adhikari, V. K., Guo, Y., Hao, F., Hilt, V., and Zhang, Z.-L. (2012). A tale of three cdns: An active measurement study of hulu and its cdns. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 7–12. IEEE.

[5] Ahmad, A., Floris, A., and Atzori, L. (2016). Qoe-centric service delivery: A collaborative approach among otts and isps. *Computer Networks*, 110:168–179.

[6] Ali, W., Shamsuddin, S. M., and Ismail, A. S. (2011). A survey of web caching and prefetching. *Int. J. Advance. Soft Comput. Appl*, 3(1):18–44.

[7] Almeroth, K. C. and Ammar, M. H. (1996). The use of multicast delivery to provide a scalable and interactive video-on-demand service. *IEEE Journal on Selected Areas in Communications*, 14(6):1110–1122.

[8] Apostolopoulos, J. G. and Trott, M. D. (2004). Path diversity for enhanced media streaming. *IEEE Communications Magazine*, 42(8):80–87.

[9] Aroussi, S. and Mellouk, A. (2014). Survey on machine learning-based qoe-qos correlation models. In *Computing, Management and Telecommunications (ComManTel), 2014 International Conference on*, pages 200–204. IEEE.

[10] Arzani, B., Gurney, A., Cheng, S., Guerin, R., and Loo, B. T. (2014). Impact of path characteristics and scheduling policies on mptcp performance. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pages 743–748. IEEE.

[11] Balachandran, A., Sekar, V., Akella, A., and Seshan, S. (2013a). Analyzing the potential benefits of cdn augmentation strategies for internet video workloads. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 43–56. ACM.

[12] Balachandran, A., Sekar, V., Akella, A., and Seshan, S. (2013b). Analyzing the potential benefits of cdn augmentation strategies for internet video workloads. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 43–56. ACM.

[13] Bang, Y., Rhee, J.-K. K., Park, K., Lim, K., Nam, G., Shinn, J. D., Lee, J., Jo, S., Koo, J.-R., Sung, J., et al. (2016). Cdn interconnection service trial: implementation and analysis. *IEEE Communications Magazine*, 54(6):94–100.

[14] Baraković, S. and Skorin-Kapov, L. (2013). Survey and challenges of qoe management issues in wireless networks. *Journal of Computer Networks and Communications*, 2013.

[15] Barré, S., Paasch, C., and Bonaventure, O. (2011). Multipath tcp: from theory to practice. *NETWORKING 2011*, pages 444–457.

[16] Basukala, R., Mohd Ramli, H., and Sandrasegaran, K. (2009). Performance of well known packet scheduling algorithms in the downlink 3gpp lte system. In *Malaysia International Conference on Communications*. IEEE.

[17] Beck, M. T., Werner, M., Feld, S., and Schimper, S. (2014). Mobile edge computing: A taxonomy. In *Proc. of the Sixth International Conference on Advances in Future Internet*, pages 48–55. Citeseer.

[18] Begluk, T., Husić, J. B., and Baraković, S. (2018). Machine learning-based qoe prediction for video streaming over lte network. In *INFOTEH-JAHORINA (INFOTEH), 2018 17th International Symposium*, pages 1–5. IEEE.

[19] Bentaleb, A., Begen, A. C., and Zimmermann, R. (2016). Sdndash: Improving qoe of http adaptive streaming using software defined networking. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1296–1305. ACM.

[20] Bicheno, S. (2017). Can broadband keep up with the multi-screen family?

[21] Binmore, K., Rubinstein, A., and Wolinsky, A. (1986). The nash bargaining solution in economic modelling. *The RAND Journal of Economics*, pages 176–188.

[22] 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.

[23] Bouten, N., Latré, 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.

[24] Brooks, P. and Hestnes, B. (2010). User measures of quality of experience: why being objective and quantitative is important. *IEEE network*, 24(2).

[25] Carofiglio, G., Gallo, M., Muscariello, L., and Perino, D. (2015). Scalable mobile backhauling via information-centric networking. In *IEEE Int. Workshop on LAN and MAN*.

[26] Casas, P., D'Alconzo, A., Wamser, F., Seufert, M., Gardlo, B., Schwind, A., Tran-Gia, P., and Schatz, R. (2017). Predicting qoe in cellular networks using machine learning and in-smartphone measurements. In *Quality of Multimedia Experience (QoMEX), 2017 Ninth International Conference on*, pages 1–6. IEEE.

[27] Chandrashekar, S., Maeder, A., Sartori, C., Höhne, T., Vejlgaard, B., and Chandramouli, D. (2016). 5g multi-rat multi-connectivity architecture. In *Communications Workshops (ICC), 2016 IEEE International Conference on*, pages 180–186. IEEE.

[28] Chen, J., Ammar, M., Fayed, M., and Fonseca, R. (2016). Client-driven network-level qoe fairness for encrypted'dash-s'. In *Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*, pages 55–60. ACM.

[29] Chen, M. and Zakhor, A. (2004). Rate control for streaming video over wireless. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1181–1190. IEEE.

[30] Chen, S., Shen, B., Yan, Y., Basu, S., and Zhang, X. (2004). Srb: Shared running buffers in proxy to exploit memory locality of multiple streaming media sessions. In *null*, pages 787–794. IEEE.

[31] Chen, Y.-C., Lim, Y.-s., Gibbens, R. J., Nahum, E. M., Khalili, R., and Towsley, D. (2013). A measurement-based study of multipath tcp performance over wireless networks. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 455–468. ACM.

[32] Cho, K., Lee, M., Park, K., Kwon, T. T., Choi, Y., and Pack, S. (2012). Wave: Popularity-based and collaborative in-network caching for content-oriented networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 316–321. IEEE.

[33] Christopher, M. (2015). Mpeg-dash vs. apple hls vs. microsoft smooth streaming vs. adobe hds.

[34] Chu, J., Cheng, Y., Dukkipati, N., and Mathis, M. (2013). Increasing tcp's initial window.

[35] Cisco (2017). Vni global IP traffic forecast, 2016 - 2021.

[36] Coudron, M. and Secci, S. (2017). An implementation of multipath tcp in ns3. *Computer Networks*, 116:1–11.

[37] Croy, P. (2011). Lte backhaul requirements: A reality check. *Aviat Networks*.

[38] De Moor, K., Ketyko, I., Joseph, W., Deryckere, T., De Marez, L., Martens, L., and Verleye, G. (2010). Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting. *Mobile Networks and Applications*, 15(3):378–391.

[39] DiPalantino, D. and Johari, R. (2009). Traffic engineering vs. content distribution: A game theoretic perspective. In *INFOCOM 2009, IEEE*, pages 540–548. IEEE.

[40] Dobrian, F., Sekar, V., Awan, A., Stoica, I., Joseph, D., Ganjam, A., Zhan, J., and Zhang, H. (2011). Understanding the impact of video quality on user engagement. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 362–373. ACM.

[41] Ernst, J. B., Kremer, S. C., and Rodrigues, J. J. (2014). A survey of qos/qoe mechanisms in heterogeneous wireless networks. *Physical Communication*, 13:61–72.

[42] 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 *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 419–426. IEEE.

[43] Ford, A., Raiciu, C., Handley, M., and Bonaventure, O. (2013). TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824.

[44] Fortz, B. and Thorup, M. (2000). Internet traffic engineering by optimizing ospf weights. In *INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*, volume 2, pages 519–528. IEEE.

[45] Ganjam, A., Siddiqui, F., Zhan, J., Liu, X., Stoica, I., Jiang, J., Sekar, V., and Zhang, H. (2015). C3: Internet-scale control plane for video quality optimization. In *NSDI*, volume 15, pages 131–144.

[46] Ghadiyaram, D., Bovik, A. C., Yeganeh, H., Kordasiewicz, R., and Gallant, M. (2014). Study of the effects of stalling events on the quality of experience of mobile streaming videos. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 989–993. IEEE.

[47] Grimes, T. and Mai, K. (2006). Digital rights management. US Patent 7,036,011.

[48] Guo, Y., Suh, K., Kurose, J., and Towsley, D. (2003). P2cast: peer-to-peer patching scheme for vod service. In *Proceedings of the 12th international conference on World Wide Web*, pages 301–309. ACM.

[49] Han, H., Shakkottai, S., Hollot, C. V., Srikant, R., and Towsley, D. (2006). Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet. *IEEE/ACM Transactions on networking*, 14(6):1260–1271.

[50] Heegaard, P. E., Biczók, G., and Toka, L. (2016). Sharing is power: Incentives for information exchange in multi-operator service delivery. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–7. IEEE.

[51] Henderson, T. R., Lacage, M., Riley, G. F., Dowell, C., and Kopena, J. (2008). Network simulations with the ns-3 simulator. *SIGCOMM demonstration*, 14(14):527.

[52] 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 *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*, pages 1–6. IEEE.

[53] Hoßfeld, T., Hock, D., Tran-Gia, P., Tutschku, K., and Fiedler, M. (2008). Testing the iqx hypothesis for exponential interdependency between qos and qoe of voice codecs ilbc and g. 711. In *Proceedings of the 18th ITC Specialist Seminar on Quality of Experience*, pages 105–114.

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

[55] Hossfelt, T., Skorin-Kapov, L., Heegaard, P. E., Varela, M., and Chen, K.-T. (2016). On additive and multiplicative qos-qoe models for multiple qos parameters. In *Proceedings of the 5th ISCA/DEGA Workshop on Perceptual Quality of Systems PQS 2016*. ISCA.

[56] Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., and Young, V. (2015). Mobile edge computing—a key technology towards 5g. *ETSI white paper*, 11(11):1–16.

[57] Hua, K. A., Cai, Y., and Sheu, S. (1998). Patching: A multicast technique for true video-on-demand services. In *Proceedings of the sixth ACM international conference on Multimedia*, pages 191–200. ACM.

[58] Hua, K. A. and Sheu, S. (1997). Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems. In *ACM SIGCOMM Computer Communication Review*, volume 27, pages 89–100. ACM.

[59] Huang, T.-Y., Johari, R., McKeown, N., Trunnell, M., and Watson, M. (2015). A buffer-based approach to rate adaptation: Evidence from a large video streaming service. *ACM SIGCOMM Comp. Comm. Review*, 44(4):187–198.

[60] Jiang, J., Liu, X., Sekar, V., Stoica, I., and Zhang, H. (2014a). Eona: Experience-oriented network architecture. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, page 11. ACM.

[61] Jiang, J., Sekar, V., Milner, H., Shepherd, D., Stoica, I., and Zhang, H. (2016). Cfa: A practical prediction system for video qoe optimization. In *NSDI*, pages 137–150.

[62] Jiang, J., Sekar, V., and Zhang, H. (2014b). Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. *IEEE/ACM Transactions on Networking (TON)*, 22(1):326–340.

[63] Jiang, W., Zhang-Shen, R., Rexford, J., and Chiang, M. (2009). Cooperative content distribution and traffic engineering in an isp network. In *ACM SIGMETRICS Performance Evaluation Review*, volume 37, pages 239–250. ACM.

[64] Jin, S. and Bestavros, A. (2002). Cache-and-relay streaming media delivery for asynchronous clients. Technical report, Boston University Computer Science Department.

[65] Johnson, D. L., Belding, E. M., and Van Stam, G. (2012). Network traffic locality in a rural african village. In *Proceedings of the fifth international conference on information and communication technologies and development*, pages 268–277. ACM.

[66] Juhn, L.-S. and Tseng, L.-M. (1997). Harmonic broadcasting for video-on-demand service. *IEEE transactions on broadcasting*, 43(3):268–271.

[67] Juluri, P. and Medhi, D. (2015). Cache'n dash: Efficient caching for dash. In *Proceedings of the 2015 ACM SIGCOMM*, pages 599–600. ACM.

[68] Kalva, H., Adzic, V., and Furht, B. (2012). Comparing mpeg avc and svc for adaptive http streaming. In *Consumer Electronics (ICCE), 2012 IEEE International Conference on*, pages 158–159. IEEE.

[69] Kameda, T. and Sun, Y. (2003). Survey on vod broadcasting schemes.

[70] Khan, S., Peng, Y., Steinbach, E., Sgroi, M., and Kellerer, W. (2006). Application-driven cross-layer optimization for video streaming over wireless networks. *IEEE communications Magazine*, 44(1):122–130.

[71] Kheirkhah, M., Wakeman, I., and Parisis, G. (2015). Multipath-tcp in ns-3. *arXiv preprint arXiv:1510.07721*.

[72] Khlifi, A. and Bouallegue, R. (2011). Performance analysis of ls and lmmse channel estimation techniques for lte downlink systems. *arXiv preprint arXiv:1111.1666*.

[73] Kibilda, J., Malandrino, F., and DaSilva, L. A. (2016). Incentives for infrastructure deployment by over-the-top service providers in a mobile network: A cooperative game theory model. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE.

[74] Kilkki, K. (2008). Quality of experience in communications ecosystem. *J. UCS*, 14(5):615–624.

[75] Kim, H. J. and Choi, S. G. (2010). A study on a qos/qoe correlation model for qoe evaluation on iptv service. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 2, pages 1377–1382. IEEE.

[76] Kim, J., Chen, Y.-C., Khalili, R., Towsley, D., and Feldmann, A. (2014). Multi-source multipath http (mhttp): A proposal. *ACM SIGMETRICS Performance Evaluation Review*, 42(1):583–584.

[77] Kim, T. and Ammar, M. H. (2005). Optimal quality adaptation for scalable encoded video. *IEEE Journal on Selected Areas in Communications*, 23(2):344–356.

[78] Knoche, H., De Meer, H. G., and Kirsh, D. (1999). Utility curves: Mean opinion scores considered biased. In *Quality of Service, 1999. IWQoS'99. 1999 Seventh International Workshop on*, pages 12–14. IEEE.

[79] Korhonen, J., Burini, N., You, J., and Nadernejad, E. (2012). How to evaluate objective video quality metrics reliably. In *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*, pages 57–62. IEEE.

[80] Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.

[81] Kreuzberger, C., Posch, D., and Hellwagner, H. (2016). Amust framework - adaptive multimedia streaming simulation framework for ns-3 and ndnsim.

[82] Kreuzberger, C., Rainer, B., and Hellwagner, H. (2015). Modelling the impact of caching and popularity on concurrent adaptive multimedia streams in information-centric networks. In *IEEE ICMEW, June 2015*, pages 1–6. IEEE.

[83] Krishnan, S. S. and Sitaraman, R. K. (2013). Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. *IEEE/ACM Transactions on Networking (TON)*, 21(6):2001–2014.

[84] Lee, D. H., Dovrolis, C., and Begen, A. C. (2014a). Caching in http adaptive streaming: Friend or foe? In *ACM NOSSDAV*, page 31.

[85] Lee, D. H., Dovrolis, C., and Begen, A. C. (2014b). Caching in http adaptive streaming: Friend or foe? In *ACM NOSSDAV*, page 31.

[86] Lee, S. C., Jiang, J. W., Chiu, D.-M. C., and Lui, J. C. (2008). Interaction of isps: Distributed resource allocation and revenue maximization. *IEEE Transactions on Parallel and Distributed Systems*, 19(2):204–218.

[87] Lenzner, R. (2017). AT&T Proposed Time Warner Takeover Illustrates Intense Hunger For Content.

[88] Li, F., Chung, J. W., and Claypool, M. (2018a). Silhouette: Identifying youtube video flows from encrypted traffic. In *Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 19–24. ACM.

[89] Li, J., Aurelius, A., Du, M., Wang, H., Arvidsson, A., and Kihl, M. (2013). Youtube traffic content analysis in the perspective of clip category and duration. In *IEEE NoF*.

[90] Li, L., Xu, K., Li, T., Zheng, K., Peng, C., Wang, D., Wang, X., Shen, M., and Mijumbi, R. (2018b). A measurement study on multi-path tcp with multiple cellular carriers on high speed rails. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 161–175. ACM.

[91] 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.

[92] Liotou, E., Samdanis, K., Pateromichelakis, E., Passas, N., and Merakos, L. (2018). Qoe-sdn app: A rate-guided qoe-aware sdn-app for http adaptive video streaming. *IEEE Journal on Selected Areas in Communications*.

[93] Little, T. D. and Venkatesh, D. (1994). Prospects for interactive video-on-demand. *IEEE multimedia*, (3):14–24.

[94] Liu, D., Chen, B., Yang, C., and Molisch, A. F. (2016). Caching at the wireless edge: design aspects, challenges, and future directions. *IEEE Communications Magazine*, 54(9):22–28.

[95] Liu, J., Xu, J., et al. (2004). Proxy caching for media streaming over the internet. *IEEE Communications magazine*, 42(8):88–94.

[96] 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.

[97] Luotonen, A. and Altis, K. (1994). World-wide web proxies. *Computer Networks and ISDN systems*, 27(2):147–154.

[98] Maggi, L., Gkatzikis, L., Paschos, G., and Leguay, J. (2015). Adapting caching to audience retention rate: Which video chunk to store? *arXiv preprint arXiv:1512.03274*.

[99] Magharei, N. and Rejaie, R. (2009). Prime: Peer-to-peer receiver-driven mesh-based streaming. *IEEE/ACM Transactions on Networking (TON)*, 17(4):1052–1065.

[100] Mäki, T., Zwickl, P., and Varela, M. (2016). Network quality differentiation: regional effects, market entrance, and empirical testability. In *IFIP Networking Conference (IFIP Networking) and Workshops, 2016*, pages 476–484. IEEE.

[101] Mandryk, R. L., Inkpen, K. M., and Calvert, T. W. (2006). Using psychophysiological techniques to measure user experience with entertainment technologies. *Behaviour & information technology*, 25(2):141–158.

[102] Mao, H., Netravali, R., and Alizadeh, M. (2017). Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 197–210. ACM.

[103] McKeown, N. (2009). Software-defined networking. *INFOCOM keynote talk*, 17(2):30–32.

[104] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.

[105] Mehmood, A. and Cheema, W.-A. (2009). Channel estimation for lte downlink.

[106] Mehr, S. K., Juluri, P., Maddumala, M., and Medhi, D. (2018). An adaptation aware hybrid client-cache approach for video delivery with dynamic adaptive streaming over http. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5. IEEE.

[107] Mirza, M., Sommers, J., Barford, P., and Zhu, X. (2007). A machine learning approach to tcp throughput prediction. In *ACM SIGMETRICS Performance Evaluation Review*, volume 35, pages 97–108. ACM.

[108] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.

[109] Mok, R. K., Chan, E. W., and Chang, R. K. (2011). Measuring the quality of experience of http video streaming. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 485–492. IEEE.

[110] Mok, R. K., Luo, X., Chan, E. W., and Chang, R. K. (2012). Qdash: a qoe-aware dash system. In *Proceedings of the 3rd Multimedia Systems Conference*, pages 11–22. ACM.

[111] Moller, S., Engelbrecht, K.-P., Kuhnel, C., Wechsung, I., and Weiss, B. (2009). A taxonomy of quality of service and quality of experience of multimodal human-machine interaction. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pages 7–12. IEEE.

[112] Nam, H., Calin, D., and Schulzrinne, H. (2016). Towards dynamic mptcp path control using sdn. In *NetSoft Conference and Workshops (NetSoft), 2016 IEEE*, pages 286–294. IEEE.

[113] Nam, H., Kim, K.-H., Kim, J. Y., and Schulzrinne, H. (2014). Towards qoe-aware video streaming using sdn. In *Global Communications Conference (GLOBECOM), 2014 IEEE*, pages 1317–1322. IEEE.

[114] Nash Jr, J. F. (1950). The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162.

[115] Nikravesh, A., Guo, Y., Qian, F., Mao, Z. M., and Sen, S. (2016). An in-depth understanding of multipath tcp on mobile devices: Measurement and system design. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 189–201. ACM.

[116] Niven-Jenkins, B., Le Faucheur, F., and Bitar, N. (2012). Content distribution network interconnection (cdni) problem statement. Technical report.

[117] Nygren, E., Sitaraman, R. K., and Sun, J. (2010). The akamai network: a platform for high-performance internet applications. *ACM SIGOPS Operating Systems Review*, 44(3):2–19.

[118] Orsolic, I., Pevec, D., Suznjevic, M., and Skorin-Kapov, L. (2017). A machine learning approach to classifying youtube qoe based on encrypted network traffic. *Multimedia tools and applications*, 76(21):22267–22301.

[119] Østerbø, O. (2011). Scheduling and capacity estimation in lte. In *Proceedings of the 23rd International Teletraffic Congress*, pages 63–70. International Teletraffic Congress.

[120] Pallis, G. and Vakali, A. (2006). Insight and perspectives for content delivery networks. *Communications of the ACM*, 49(1):101–106.

[121] Palomar, D. P. and Chiang, M. (2006). A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(8):1439–1451.

[122] Pantos, R. and May, W. (2017). Http live streaming. Technical report.

[123] Pathak, A., Zhang, M., Hu, Y. C., Mahajan, R., and Maltz, D. (2011). Latency inflation with mpls-based traffic engineering. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 463–472. ACM.

[124] Pathan, M. and Buyya, R. (2008). A taxonomy of cdns. In *Content delivery networks*, pages 33–77. Springer.

[125] Peirens, B., Detal, G., Barre, S., and Bonaventure, O. (2016). Link bonding with transparent multipath tcp. RFC Draft.

[126] Perkis, A., Munkeby, S., and Hillestad, O. I. (2006). A model for measuring quality of experience. In *Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic*, pages 198–201. IEEE.

[127] Poon, W.-F. and Lo, K.-T. (1999). Design of multicast delivery for providing vcr functionality in interactive video-on-demand systems. *IEEE Transactions on Broadcasting*, 45(1):141–148.

[128] Qadir, Q. M., Kist, A. A., and Zhang, Z. (2015). Mechanisms for qoe optimisation of video traffic: A review paper. *Australasian Journal of Information, Communication Technology and Applications*, 1(1):1–18.

[129] Rec, I. (2008). P. 10: Vocabulary for performance and quality of service, amendment 2: New definitions for inclusion in recommendation itu-t p. 10/g. 100. *Int. Telecomm. Union, Geneva*.

[130] Reichl, P., Egger, S., Schatz, R., and D'Alconzo, A. (2010). The logarithmic nature of qoe and the role of the weber-fechner law in qoe assessment. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE.

[131] Reichl, P., Maillé, P., Zwickl, P., and Sackl, A. (2013a). A fixed-point model for qoe-based charging. In *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*, pages 33–38. ACM.

[132] Reichl, P., Tuffin, B., and Schatz, R. (2013b). Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience. *Telecommunication Systems*, 52(2):587–600.

[133] Reichl, P., Tuffin, B., and Schatz, R. (2013c). Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience. *Telecommunication Systems*, 52(2):587–600.

[134] Rejaie, R., Handley, M., and Estrin, D. (2000). Layered quality adaptation for internet video streaming. *IEEE Journal on Selected Areas in Communications*, 18(12):2530–2543.

[135] Riiser, H., Vigmostad, P., Griwodz, C., and Halvorsen, P. (2013). Commute path bandwidth traces from 3g networks: analysis and applications. In *Proceedings of the 4th ACM Multimedia Systems Conference*, pages 114–118. ACM.

[136] Samain, J., Carofiglio, G., Tortelli, M., and Rossi, D. (2018). A simple yet effective network-assisted signal for enhanced dash quality of experience. In *Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 55–60. ACM.

[137] Sato, M., Nakajima, S., and Suzuki, K. (2012). Ethernet link aggregation. US Patent 8,274,980.

[138] Schatz, R., Hoßfeld, T., Janowski, L., and Egger, S. (2013). From packets to people: quality of experience as a new measurement challenge. In *Data traffic monitoring and analysis*, pages 219–263. Springer.

[139] Schlinker, B., Kim, H., Cui, T., Katz-Bassett, E., Madhyastha, H. V., Cunha, I., Quinn, J., Hasan, S., Lapukhov, P., and Zeng, H. (2017). Engineering egress with edge fabric. In *Proceedings of the ACM SIGCOMM 2017 Conference (SIGCOMM'17). ACM, New York, NY, USA*.

[140] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. (2003). Rtp: A transport protocol for real-time applications. Technical report.

[141] 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.

[142] Sen, S., Rexford, J., and Towsley, D. (1999). Proxy prefix caching for multimedia streams. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1310–1319. IEEE.

[143] Seo, S. (2015). Kt's giga lte. *Presentation at IETF*, 93.

[144] Shan, Y. (2005). Cross-layer techniques for adaptive video streaming over wireless networks. *EURASIP Journal on Advances in Signal Processing*, 2005(2):871928.

[145] Shrimali, G., Akella, A., and Mutapcic, A. (2010). Cooperative interdomain traffic engineering using nash bargaining and decomposition. *IEEE/ACM Transactions on Networking*, 18(2):341–352.

[146] Siekkinen, M., Hoque, M. A., and Nurminen, J. K. (2016). Using viewing statistics to control energy and traffic overhead in mobile video streaming. *IEEE/ACM Transactions on Networking*, 24(3):1489–1503.

[147] Skorin-Kapov, L., Varela, M., Hoßfeld, T., and Chen, K.-T. (2018). A survey of emerging concepts and challenges for qoe management of multimedia services. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(2s):29.

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

[149] Song, J.-W., Park, K.-S., and Yang, S.-B. (2006). An effective cooperative cache replacement policy for mobile p2p environments. In *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, volume 2, pages 24–30. IEEE.

[150] Sousa, I., Queluz, M. P., and Rodrigues, A. (2017). A survey on qoe-oriented wireless resources scheduling. *arXiv preprint arXiv:1705.07839*.

[151] Spagna, S., Liebsch, M., Baldessari, R., Niccolini, S., Schmid, S., Garroppo, R., Ozawa, K., and Awano, J. (2013). Design principles of an operator-owned highly distributed content delivery network. *IEEE Communications Magazine*, 51(4):132–140.

[152] Spiteri, K., Urgaonkar, R., and Sitaraman, R. K. (2016). Bola: Near-optimal bitrate adaptation for online videos. In *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*, pages 1–9. IEEE.

[153] Stampleman, J. B. (1992). *Scalable video compression*. PhD thesis, Massachusetts Institute of Technology.

[154] Streijl, R. C., Winkler, S., and Hands, D. S. (2016). Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227.

[155] Subramanya, S. and Yi, B. K. (2006). Digital rights management. *IEEE Potentials*, 25(2):31–34.

[156] Sutton, R. S., Barto, A. G., Bach, F., et al. (1998). *Reinforcement learning: An introduction*. MIT press.

[157] Takahashi, A., Hands, D., and Barriac, V. (2008). Standardization activities in the itu for a qoe assessment of iptv. *IEEE Communications Magazine*, 46(2).

[158] Thomas, E., van Deventer, M., Stockhammer, T., Begen, A. C., and Famaey, J. (2015). Enhancing mpeg dash performance via server and network assistance.

[159] Torres, R., Finamore, A., Kim, J. R., Mellia, M., Munafo, M. M., and Rao, S. (2011). Dissecting video server selection strategies in the youtube cdn. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 248–257. IEEE.

[160] Viswanathan, S. and Imielinski, T. (1996). Metropolitan area video-on-demand service using pyramid broadcasting. *Multimedia systems*, 4(4):197–208.

[161] Vlavianos, A., Iliofotou, M., and Faloutsos, M. (2006). Bitos: Enhancing bittorrent for supporting streaming applications. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–6. IEEE.

[162] Volk, M., Sterle, J., Sedlar, U., and Kos, A. (2010). An approach to modeling and control of qoe in next generation networks [next generation telco it architectures]. *IEEE Communications Magazine*, 48(8).

[163] Wang, J. (1999). A survey of web caching schemes for the internet. *ACM SIGCOMM Computer Communication Review*, 29(5):36–46.

[164] Wang, W.-H., Palaniswami, M., and Low, S. H. (2003). Optimal flow control and routing in multi-path networks. *Performance Evaluation*, 52(2):119–132.

[165] Wang, Y., van der Schaar, M., Chang, S.-F., and Loui, A. C. (2005). Classification-based multidimensional adaptation prediction for scalable video coding using subjective quality evaluation. *IEEE transactions on circuits and systems for video technology*, 15(10):1270–1279.

[166] Wang, Z., Huang, J., and Rose, S. (2017). Evolution and challenges of dns-based cdns. *Digital Communications and Networks*.

[167] Wierzbicki, A., Leibowitz, N., Ripeanu, M., and Wozniak, R. (2004). Cache replacement policies revisited: The case of p2p traffic. In *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on*, pages 182–189. IEEE.

[168] Wilson, G. M. and Sasse, M. A. (2000). Do users always know what's good for them? utilising physiological responses to assess media quality. In *People and computers XIV—Usability or else!*, pages 327–339. Springer.

[169] Winkler, S. and Mohandas, P. (2008). The evolution of video quality measurement: from psnr to hybrid metrics. *IEEE Transactions on Broadcasting*, 54(3):660–668.

[170] Wohlfart, F., Chatzis, N., Dabanoglu, C., Carle, G., and Willinger, W. (2018). Leveraging interconnections for performance: the serving infrastructure of a large cdn. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 206–220. ACM.

[171] Wu, K.-L., Yu, P. S., and Wolf, J. L. (2001). Segment-based proxy caching of multimedia streams. In *Proceedings of the 10th international conference on World Wide Web*, pages 36–44. ACM.

[172] 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.

[173] Yin, H., Liu, X., Zhan, T., Sekar, V., Qiu, F., Lin, C., Zhang, H., and Li, B. (2009). 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. ACM.

[174] Yin, X., Jindal, A., Sekar, V., and Sinopoli, B. (2015). A control-theoretic approach for dynamic adaptive video streaming over http. In *Proceedings of the 2015 ACM SIGCOMM*, pages 325–338. ACM.

[175] Yunfeng, H. (2015). Cache-friendly rate adaptation for dynamic adaptive streaming over http (dash).

[176] Zhang, G., Liu, W., Hei, X., and Cheng, W. (2015). Unreeling xunlei kankan: Understanding hybrid cdn-p2p video-on-demand streaming. *IEEE Transactions on Multimedia*, 17(2):229–242.

[177] Zhu, J., He, J., Zhou, H., and Zhao, B. (2013). Epcache: In-network video caching for lte core networks. In *Wireless Communications & Signal Processing (WCSP), 2013 International Conference on*, pages 1–6. IEEE.

[178] Zinner, T., Jarschel, M., Blenk, A., Wamser, F., and Kellerer, W. (2014). Dynamic application-aware resource management using software-defined networking: Implementation prospects and challenges. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–6. IEEE.