Managed Video Services over Software Defined Networks

by

K. Tolga Bağcı

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Electrical and Electronics Engineering

July 9, 2018

Managed Video Services over Software Defined Networks

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

K. Tolga Bağcı

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Dr. A. Murat Tekalp (Advisor)

Assoc. Prof. Dr. Öznur Özkasap

Assoc. Prof. Dr. Sinem Çöleri Ergen

Assoc. Prof. Dr. Berk Canberk

Asst. Prof. Dr. Ali C. Beğen

Canım oğlum ve sevgili eşime...

Acknowledgements

I would like to express my sincere gratitude and profound respect to my advisor Prof. Murat Tekalp for his strong guidance, inspiration, everlasting patience and valuable support. I am also grateful to members of my thesis committee Assoc. Prof. Öznur Özkasap, Assoc. Prof. Sinem Ergen, Assoc. Prof. Berk Canberk and Asst. Prof. Ali Beğen for their valuable comments and serving on my defense committee.

First and foremost, I thank to my beloved wife Şebnem. There would be none of this without her endless support and patience. Next, I thank to the most beautiful thing in my life, my little boy Toprak, who gave me happiness even in the most difficult times. I am grateful to my sister Tuğba and brother in law Tamer for their significant support and motivation. I thank to my mother Gülseren and my father Hüseyin for their love, support and vision. Next, I would like to express my gratitudes to Yalçın, Seyhan and Tuğtekin for all the fun that we had. I also thank to MVGL friends and Özhakiki Canlar.

I also acknowledge the support of TUBITAK (The Scientific and Technological Research Council of Turkey) Grant #113E254 and #115E299. I am also thankful to TUBITAK BIDEB who provided me scholarship during my PhD study.

Abstract

The best effort nature of the current Internet leads to inefficiencies for all parties involved in a video service; namely, network service providers (NSP), over-the-top (OTT) video service providers (VSP), and users (clients). From the perspective of the NSP, increasing volume of video services does not contribute to their revenues since NSPs commonly charge a flat monthly subscription fee for the best-effort Internet access. From the perspective of OTT VSP, unlike IPTV services that are offered over a managed private IP network, they cannot promise specific quality of experience (QoE) to their users as they rely on the best-effort service offered by some NSP. From the perspective of end-users, they do not have a choice to receive video services with some level of QoE for an extra-cost. In this thesis, recognizing these shortcomings, we investigate the vision of open multi-service inter-networking, including different service-levels and service-level awareness of users, and propose managed video service architectures using software defined networking (SDN) that is a programmable networking paradigm and a key technology for 5G networks.

First, we propose centralized and distributed architectures for collaboration between NSP, VSP and users to provide NSP-managed or VSP-managed streaming services over SDN with quality-of-service (QoS) reserved network slices. We show that QoS reservation alone is not sufficient to overcome QoE fluctuations per user and unfairness between heterogeneous video clients, and clients also need to employ TCP receive-window adaptation knowing their fair-share bitrate through collaborative streaming service models that are NSP-managed (centralized) and VSP-managed (centralized or distributed).

Second, we propose a video service architecture and a novel resource al-

location optimization framework to enable NSPs to offer value-added video services (VAVS) over single operator SDN including multiple service-levels and associated business models. To this effect, we introduce a new batch-optimization framework, where resource (path, bitrate and admission control) allocations for a small group of flows (consisting of new service requests and some existing ones) are performed simultaneously. In order to compute dynamic resource allocations online, we propose a heuristic group-constrained-shortest-path procedure that aims for a fair allocation of resources among a group of requests with the same service level, while maximizing the total NSP revenue.

Third, we propose an SDN-enabled distributed open exchange framework for dynamic optimization of end-to-end (e2e) QoS paths over multi-operator networks, which enables individual NSPs to offer inter-operator services with e2e QoS guarantees while allowing each NSP to manage their own network resources. The SDN controllers of all NSPs communicate with each other to advertise a set of QoS-enabled paths across their own network with ask prices, and each NSP dynamically selects the best e2e price-performance path for its customers by bidding on a subset of these advertised paths, and form e2e paths by concatenating them.

Özetçe

Mevcut İnternet'in en iyi çabalı hizmeti doğası, bir video hizmetinde yer alan tüm taraflar, yani ağ hizmeti sağlayıcıları (NSP), üst düzey (OTT) video servis sağlayıcıları (VSP) ve kullanıcılar (istemciler), için verimsizliklere yol açmaktadır. NSP tarafından bakılırsa, NSP'ler genellikle en iyi çabalı İnternet hizmeti için aylık bir abonelik ücreti aldığından, video hizmetlerinin artan hacmi gelirlerine katkıda bulunmaz. OTT VSP'nin perspektifinden bakılırsa, özel bir IP ağı üzerinden yönetilen IPTV hizmetlerinden farklı olarak, kullanıcılarına belirli bir deneyim kalitesi (QoE) vaat edemezler çünkü NSP'ler tarafından sunulan en iyi çabalı İnternet üzerinden servis vermektedirler. Son kullanıcılar tarafından bakılır ise, ekstra bir ücret karşılığında bir miktar QoE ile video hizmetleri alma seçeneği mevcut sistemlerde mümkün değildir. Bu tezde, bu eksikliklerin farkında olarak, farklı hizmet düzeyleri ve hizmet seviyesinde kullanıcı farkındalığı da dahil olmak üzere, açık çoklu-hizmet arası ağ oluşturma vizyonunu uygulayarak, 5G ağları için anahtar bir teknoloji olan yazılım tanımlı ağlar (SDN) üzerinde yönetilebilen video hizmeti mimarileri önerilmektedir.

Öncelikle, NSP, VSP ve kullanıcılar arasında, hizmet kalitesi (QoS) ayrılmış ağ dilimleriyle SDN üzerinden NSP tarafından yönetilen veya VSP tarafından yönetilen akış hizmetleri sağlamak için merkezi ve dağıtılmış mimariler öneriyoruz. QoS rezervasyonunun, kullanıcı başına QoE dalgalanmalarını ve heterojen video istemcileri arasındaki adaletsizliği gidermek için tek başına yeterli olmadığını ve son kullanıcıların adil paylaşım bit hızlarını bilerek TCP gelen veri penceresi uyarlamalarını kullanmaları gerektiğini gösteriyoruz. Bunun için NSP tarafından yönetilen (merkezi) ve VSP tarafından yönetilen (merkezi veya dağıtılmış) hizmet modellerini öneriyoruz.

İkinci olarak, NSP'lerin birden fazla hizmet seviyesi ve ilgili iş modelleri de dahil olmak üzere tek operatörlü SDN üzerinde katma değerli video hizmetleri (VAVS) sunabilmelerini sağlamak için bir video hizmeti mimarisi ve yeni bir kaynak ayırma optimizasyonu çerçevesi öneriyoruz. Bu amaçla, küçük bir grup akış için (yeni servis talepleri ve bazı mevcut olanlardan oluşan) kaynakların (yol, bit hızı ve kabul kontrolü) tahsislerinin aynı anda gerçekleştirildiği yeni bir parti optimizasyon çerçevesini sunuyoruz. Dinamik kaynak tahsislerini çevrimiçi olarak hesaplamak için, toplam NSP gelirini en üst düzeye çıkarırken, aynı hizmet düzeyine sahip bir grup talepler arasında kaynakların adil bir şekilde tahsis edilmesini amaçlayan, sezgisel grup kısıtlamalı en kısa yol prosedürünü öneriyoruz.

Üçüncü olarak, her bir NSPnin kendi ağ kaynaklarını yönetirken bireysel olarak e2e QoS garantileriyle birlikte operatörler arası hizmetler sunmasını sağlayan, çoklu operatör ağları üzerinden uçtan uca (e2e) QoS yollarının dinamik optimizasyonu için SDN üzerinde dağıtımlı bir açık değişim çerçevesi öneriyoruz. Önerilen sistemde, tüm NSP'lerin SDN denetleyicileri, birbirleriyle iletişim kurarak, kendi ağları üzerinden bir dizi QoS yolunun satış fiyatlarının bildirilmesi için iletişim kurarlar. Her bir NSP, duyurusu yapılan QoS yollarının bir alt kümesine teklif vererek müşterileri için en iyi e2e fiyat performans yolunu dinamik olarak belirleyerek, NSP ağları üzerindeki bu yollar birleştirilir ve uçtan uca yollar oluşturulur.

Contents

Chapter 1

Introduction

1.1 Motivation

The video streaming traffic over the Internet protocol (IP) is expected to reach 80 percent of all IP traffic by 2019 increasing from 67 percent in 2014 [1]. Video streaming services over the Internet can be delivered in two ways, namely IPTV and over-the-top (OTT). IPTV services are offered over managed private IP networks that is owned or leased by the local network service provider (NSP) hosting the video content as well. Since the network resources are managed and IP packets never travel over the open Internet in IPTV, quality of service (QoS) enabled differentiated video services can be offered to the IPTV clients based on their subscription plans. On the other hand, OTT enables delivery of video services offered by the third party video service providers (VSP) which are either subscription-based, e.g., Netflix, Hulu, Amazon Instant Video, or free video services e.g., Youtube, Vimeo, over the open unmanaged Internet. However, the best effort nature of the current Internet leads to inefficiencies for all parties (NSP, VSP and end-users) involved in a video service.

Today, the main revenue stream of the NSPs are the clients paying flat subscription fees in exchange for the best-effort Internet access. Although video streaming traffic over the IP is increasing, it is not contributing to the revenue of the NSPs since network usage charges are not directly related to the amount of network resources consumed. Accordingly, the OTT video

service providers are becoming overwhelming loads on the telecom operators as they naturally do not invest on the network and eat up the resources. Moreover, OTT VSPs are not able to offer differentiated services to its users and cannot promise specific quality-of-experience (QoE) as they rely on the best-effort service offered by some NSP. Similarly, whether the clients consume heavy data or not, they charged with a fixed fee where light users subsidize heavy users when the total revenue from the clients in return for the consumed resources are considered [2].

In the future Internet, all three parties that are involved in the delivery of the OTT video services can benefit from new business models enabled by the next generation network management technologies. Firstly, NSPs can support differentiated services for OTT video streaming traffic so that the premium subscribers of OTT VSPs receive video service with high QoE. For example, high throughput service to ultra-HD video-on-demand (VoD) and low delay service to real-time communication (RTC) provide sufficient level of QoE to OTT premium subscribers. Secondly, clients can be charged according to the on-demand service-level they choose. Thirdly, NSPs will be able to monetize the OTT video traffic and increase the revenues. On the other hand, differentiated traffic volume should be limited to a certain level providing fairness to the best-effort Internet services under Net-neutrality principles.

Over the years, there have been many traffic engineering (TE) proposals to improve network performance and provide QoS over an operator network, including integrated services (IntServ), differentiated services (DiffServ), multi-label packet switching (MPLS) [3], application layer traffic optimization (ALTO) [4], and path computation element (PCE) [5]. However, these services are not widely deployed, except for closed networks, due to scalability concerns. Recently, software-defined networking (SDN), a programmable networking paradigm and a key technology for 5G networks, has emerged as a promising alternative to realize service-aware routing, flow-level quality assurance and efficient dynamic resource management. SDN is composed of two planes: a data plane comprised of switches, which perform data forwarding, and a control plane connecting all switches to a controller,

which determines flow tables and sends them to the switches. Hence, packet forwarding and route determination tasks are decoupled. The controller is a software component that is the brain of the network. It has global visibility of the network topology and resources within a domain, meaning, it can collect near real-time data from switches about the state of the switch and network. Accordingly, it can modify flow tables to optimize network utilization and avoid congestion. The controller communicates with the switches by means of a southbound interface such as OpenFlow [6] to specify or modify flow tables, and sometimes to alter the packet header fields. Besides network programmability, SDN enables an abstraction of the physical network, such that one physical network can support a variety of services over customized network slices. Recently, several SDN-enabled video services with QoS over an operator network have been proposed [7], [8], [9], [10], [11].

In order to offer (i) OTT VSPs the same benefits as IPTV by enabling them to offer their users standard and premium content, e.g., HD and UHD with some QoE, (ii) users the choice of service-level that they are willing to pay for, and (iii) NSPs additional revenues from these value added services, efficient management of future open multi-service networks has crucial importance. This thesis investigates architectures for managed and value-added video services over SDN.

1.2 Contributions

The main contributions of the thesis are as follows:

- Assuming that specific resources to individual and/or groups of flows are reserved by one of SDN-QoS mechanisms, we address how can we i) minimize video-quality fluctuations per-flow, ii) maximize video-quality fairness among heterogeneous flows, and iii) maximize total goodput over the reserved resources. The three way interaction between resource reservation at network level and TCP congestion control at transport level and dynamic adaptive streaming over HTTP (DASH) quality level adaptation at the application level has not been studied before. It turns

out that resource reservation alone is not sufficient to optimize QoE variability, fairness, and resource utilization in DASH video because TCP flows compete for more bandwidth due to nature of TCP congestion control. We show the need to control TCP receive-window size *rwnd* when the TCP rate reaches the rate reserved for that flow in order to properly manage QoE variability and fairness. We then propose collaborative streaming architectures, namely the NSP-managed (centralized) and the VSP-managed (centralized and distributed), where NSP, VSP, and users interact. The proposed architectures are mainly motivated by (i) QoS implementations require involvement of the NSP, (ii) clients need to know their reserved bitrate in order to implement TCP rate control by *rwnd* management. List of our papers resulting from the research on collaborative video architectures over future networks are [7] and [8].

- Focusing on NSP-managed architectures, we propose a service architecture and an implementation to realize value-added video services over an open multi-service network, where users can select the service-level that they desire to pay for. In order to compute network resources by NSP, we introduce a batch-optimization framework for dynamic resource allocation for a group of flows with multiple service-levels to maximize revenues for the NSP. Moreover, a divide-and-conquer procedure to compute the optimal solution to the multi-service traffic engineering optimization problem approximately. In order to compute resource allocation online, a fast heuristic group constrained-shortest-path (GCSP) procedure that performs close to the approximately optimal method. We demonstrate that the processing the service requests in batches significantly improves revenue, fairness, and utilization compared to sequential processing of incoming service requests when network is in congested mode of operation. List of our papers resulting from the research on value-added video services over SDN are [10] and [12].

- It is likely that in a streaming video service the VSP and the client or in

a video-conference service two or more peers reside in different NSP networks. This necessitates a cooperation between multiple NSPs to offer a video service with end-to-end (e2e) QoS. We propose a distributed framework to enable dynamic e2e inter-NSP path optimization and real-time e2e service-aware flow management in multi-operator SDN. The proposed system (i) is based on cooperation between SDN controllers of different NSPs as opposed to being a centralized exchange that relies on a third party physical infrastructure, (ii) is based on the open market cooperation model with dynamic inter-domain path pricing as opposed to static pre-negotiated pricing and potentially leads to lower prices due to competition among multiple QoS path providers, and (iii) supports inter-operator services with multiple e2e service levels for different value-added services on the same physical network. List of our papers resulting from the research on video services over multi-operator SDN are [13], [14] and [15].

1.3 Organization

The rest of the thesis is organized as follows:

- Chapter 2 presents NSP-managed video streaming service model with centralized collaboration between the NSP, VSP, and users. We then present VSP-managed service model with centralized and distributed architectures. It is also shown that QoS reservation alone is not sufficient to overcome QoE fluctuations per client and unfairness between heterogeneous video clients, and clients also need to employ TCP receive-window adaptation knowing their reserved bandwidth. We published the research presented in Chapter 2 in paper [7].

- Chapter 3 adopts NSP-managed architecture and presents a novel framework for resource allocation optimization of a group of service requests enabling NSPs to offer value-added video services with multiple service levels and maximize revenues. We also present a method to compute optimal solution approximately and a fast heuristic method to compute

resources online. We published the research presented in Chapter 3 in paper [10].

- Chapter 4 extends to study in Chapter 3 and presents dynamic e2e path optimization framework for multi-operator networks which is based on cooperation between SDN controllers of different NSPs and enables open market model with dynamic inter-domain path pricing. We submitted the research presented in Chapter 4 in paper [15].

- Chapter 5 presents the concluding remarks.

Chapter 2

Managed Video Service Architectures over Future Networks

2.1 Introduction

We are witnessing increasing consumption of Hyper-Text Transfer Protocol (HTTP) adaptive video streaming over the Transmission Control Protocol (TCP) on the open (unmanaged) Internet. However, the best-effort nature of the Internet results in a varying degree of unpredictable end-to-end (e2e) quality of service (QoS), which is characterized by unpredictable variations in TCP throughput and packet loss rates. This in turn causes unsatisfactory quality of experience (QoE) for end-users (clients), because of video stalls (due to client play-out buffer drainage) and frequent video quality variations (due to TCP throughput fluctuations) [16]. We foresee that, in addition to classical best-effort over-the-top (OTT) video services, managed video services over reserved network slices [17] will also be offered over future networks. This chapter discusses new models and collaborative streaming architectures for such services.

In HTTP adaptive streaming (HAS), the video is encoded in fixed-duration segments that are made available at the video server at multiple bitrates

(quality levels), and clients request video segments at a bitrate which best fits their play-out-buffer status and an estimate of available network throughput. Segment durations vary between 1-15 secs, but are fixed and synchronized for multiple representations of the same video at different bitrates. The MPEG dynamic adaptive streaming over HTTP (DASH) standard specifies the format of video content and associated manifest file so that different HAS players can play the same content [18].

Since DASH uses TCP transport, there are two separate flow/rate control mechanisms applied at two different levels: (i) TCP congestion/flow control at the transport level, and (ii) DASH video rate adaptation by the client at the application level. These two rate control mechanisms are unaware of each other, and they poorly interact especially in the presence of network congestion [16]. As a result, DASH flows competing for a specific bottleneck link throughput (i) experience unstable bitrates per flow (leading to QoE variability), (ii) result in unfair distribution of link capacity among competing heterogeneous flows (leading to QoE unfairness), (iii) cause underutilization of the bottleneck link capacity. An evaluation of three TCP-based rate-control schemes for adaptive streaming using scalable H.264/SVC video coding in terms of the quality of delivered video has been conducted in [19]. The DASH video bitrate adaptation at the application layer can only alleviate the undesired effects of TCP goodput fluctuations by matching video bitrate to available TCP goodput. While this is effective to avoid video-stall events in most cases, it comes at the cost of video quality fluctuations that are known to be disturbing to the end-users. One way to overcome quality fluctuations is to stabilize the TCP goodput to solve the problem at the transport level. Furthermore, it is not possible to provide rate-fairness among heterogeneous DASH users in a low capacity network slice without regulating their TCP goodput, since the standard TCP rate control tends to equalize the rate of competing TCP flows regardless of possibly differing rate requirements of these flows. Note that application layer DASH rate adaptation cannot solve this problem.

QoE refers to user satisfaction of a streaming session, which depends on many factors such as the quality of video compression, transmission losses,

quality of display device, and other factors like user engagement in content. On the other hand, video quality, which depends on encoding bitrate as well as content complexity, is often measured in terms of peak-signal-to noise ratio (PSNR) or the structural similarity metric (SSIM) [20]. There are many measures of QoE, both subjective and objective, in the literature [21–24]. In terms of objective measures, it is generally agreed that QoE for a streaming session depends on both the mean and variance of video quality, as well as frequency and duration of stalls, assuming all other factors are fixed. Analysis of all factors affecting QoE in DASH video can be found in [25]. Metrics for measuring QoE are discussed in [23]. QoE evaluation of different bitrate adaptation logics are presented in [24, 26].

The concept of QoE fairness considers distribution of available bitrate among heterogeneous video clients in a fair manner such that clients with different video resolutions experience similar QoE. That is, instead of sharing the available bitrate equally between multiple DASH clients, users receiving video at HD resolution are allocated a higher share of the bitrate than users receiving video at SD or lower resolutions. We can classify related works for achieving QoE-fairness between competing TCP flows as i) NSP-enforced solutions, ii) VSP-assisted solutions [27], and iii) features built-into the client player [28, 29]. NSP-enforced approaches include i) traffic shaping at the network edges by ACK-pacing or active window management (AWM), and ii) network-wide solutions involving SDN and QoS provisioning. Since the TCP congestion window cannot be larger than TCP receiver window size *rwnd*, the network operator can perform traffic shaping by controlling the downstream flow rate (send rate) for each flow by either manipulating *rwnd* field in ACK headers at an appropriate router or buffering ACK messages at routers along congested paths to delay them, and hence, slow down senders in a fair manner. The load-adaptive ACK pacer [30] adapts to aggregate traffic fluctuations at a router, rather than per flow state as in [27], and triggers ACK pacing when queue fullness exceeds a threshold. It has been demonstrated that AWM, activated based on queue fullness in a router, achieves QoE fairness among multiple competing DASH streams in a DVB-T broadcast network [31]. The AWM method has also been implemented at the

access router of the VSP, where computation of *rwnd* relies only on local data available within the VSP router [27]. Traffic shaping at home gateways has also been proposed to provide QoE-fairness between multiple competing heterogeneous DASH flows in home networks [32] [33].

Future networks deploying SDN will make e2e QoS provisioning more practical and feasible since the entire topology of the network is visible to an SDN controller [34], which makes localization and solution of the congestion and QoE-fairness problems more accurate. Approaches to provision e2e QoS over SDN has been surveyed in [35]. Methods to provide QoS to DASH flows include (i) metering (policing) flows at select switches, (ii) use ACK pacing techniques (traffic shaping) for flow-based rate control [27], and (iii) allocating and managing special queues for specific flows [36].

Related works that employ SDN to offer QoS to DASH flows in order to achieve network-wide QoE fairness include [37] [38] [39] [40] [41] [42] [43]. These methods typically involve a central optimization problem to allocate resources to different DASH flows. Georgopoulos et al. [37] propose an OpenFlow-assisted QoE fairness framework (QFF) that monitors the status of all DASH applications in a network and dynamically allocates fair network resources to each client. Nam et al. [38] propose an SDN application to monitor network and dynamically change routing paths using multi-protocol label switching (MPLS) traffic engineering (TE) to provide better QoE performance. Liotou et al. [44] introduce a QoE-Service module using TE over a SDN to enhance the performance of OTT applications. Ramakrishnan et al. [39] propose an SDN-based VQOA module that implements multi-client bandwidth allocation optimization for video rate selections among competing clients in order to increase overall QoE performance of the system. Petrangeli et al. [40] propose QoE-driven rate adaptation heuristic for fair adaptive video streaming. Kleinrouweler et al. [41] develop an SDN architecture for a DASH Service Manager, that signals desired bitrates to DASH clients to adapt and perform TE to overcome the negative effect of TCP. Mu et al. [42] introduce a user-level fairness model, UFair, and its hierarchical variant, UFairHA, which orchestrate DASH media streams using SDN, and incorporate three fairness metrics (video quality, switching impact, and cost

efficiency) to achieve user-level fairness. Bentaleb et al. [43] manage and allocate network resources dynamically for each client based on its expected QoE. However, none of these works provide an analysis of the interaction between QoS resource reservation, TCP congestion control, and the DASH video rate adaptation mechanism.

Assuming that we reserve specific resources to individual and/or groups of flows by one of SDN-QoS mechanisms, this chapter proposes NSP-managed and VSP-managed collaborative streaming architectures to address: how can we i) minimize video-quality fluctuations per-flow, ii) maximize video-quality fairness among heterogeneous flows, and iii) maximize total goodput over the reserved resources. The three way interaction between resource reservation at network level and TCP congestion control at transport level and DASH quality level adaptation at the application level has not been studied before. It turns out that resource reservation alone is not sufficient to optimize QoE variability, fairness, and resource utilization in DASH video because TCP flows compete for more bandwidth due to nature of TCP congestion control. We show the need to control TCP receive-window size when the TCP rate reaches the rate reserved for that flow in order to properly manage QoE variability and fairness in Section 2.2. We then propose collaborative streaming architectures to inform clients about their fair-share bitrates: the NSP-managed architecture is described in Section 2.3 and the VSP-managed centralized and distributed architectures are described in Section 2.4. The main novelties presented in this chapter are incorporation of client-side TCP receive window management in central collaboration architectures and introducing a novel VSP-managed distributed collaboration architecture. The proposed collaborative streaming approach is motivated by (i) QoS implementations require involvement of the NSP, (ii) clients need to know their fair-share bitrate in order to implement TCP rate control by *rwnd* management, and (iii) pure network-based solutions have difficulty in identifying video flows and the resources they need when HTTPS is used. Experimental procedures and results are described in Section 2.5 and conclusions are presented in Section 2.6.

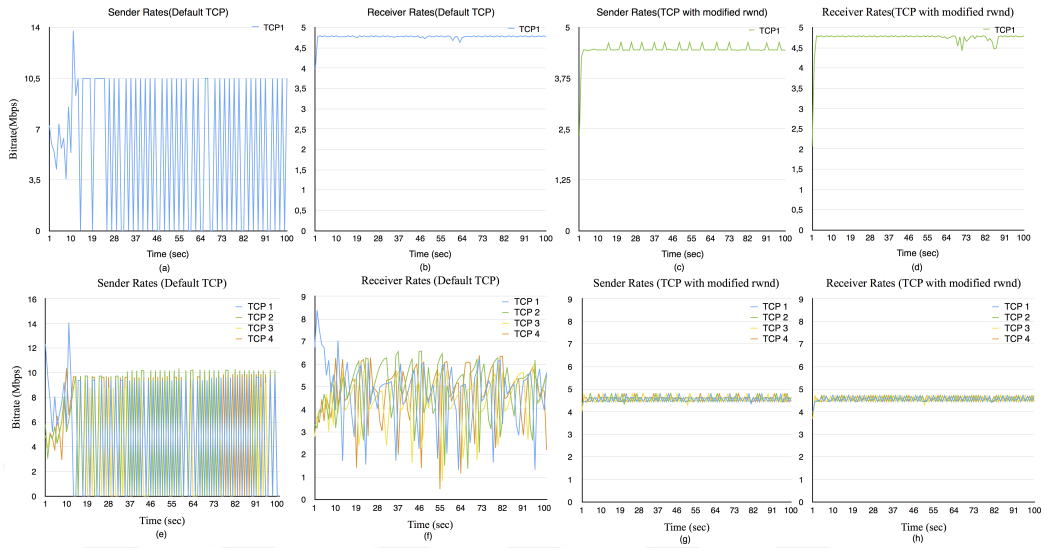

Figure 2.1: Instantaneous TCP rates for default TCP vs. TCP with rwnd control: a) sender and b) receiver rates for a single standard TCP flow in a reserved 5 Mbps network slice, c) sender and d) receiver rates for a single TCP flow with rate control in a reserved 5 Mbps slice, e) sender and f) receiver rates for four standard TCP clients in a reserved 20 Mbps slice, g) sender and h) receiver rates for four TCP clients with rate control in a reserved 20 Mbps slice.

2.2 Resource Reservation and TCP Performance

This section first introduces two managed video service models over future networks and relevant network slicing options in Section 2.2.1 and then discuss the relationship between resource reservation and TCP performance in Section 2.2.2.

2.2.1 Managed Video Services over Dynamic Network Slices

Two potential models for managed video services over future networks are: i) NSP-managed video services, where users (clients) can purchase value-added connectivity services from their NSP to consume video with some service-level agreement, and ii) VSP-managed video services that can be purchased

from a VSP, where VSPs provide such services over resource-reserved network slices that are provisioned by one or more NSP.

Ideal collaboration among a group of DASH clients requires that these services be delivered over network slices with reserved throughput, since we want to make sure that DASH flows not only don't compete with each other, but they are isolated and do not compete with other flows as well. In both service models, reserved network slices are provisioned by the NSP according to limited client information, i.e., source and destination IP addresses and video resolution per client, that the VSP shares with the NSP. In the NSP-managed services, the fair-share bitrate for each client and the optimal routing are determined centrally by the NSP itself and clients are assigned to network slices according to their video resolution/bitrate as explained in Section 2.3. In the VSP-managed service model, the NSP first performs best-effort routing for current VSP clients and then on each link clients of the same VSP are assigned to a slice whose capacity is determined by the NSP according to resolution/bitrate of clients on that link. In this case, the VSP manages the fair-share bitrate for each client as detailed in Section 2.4. We note that network slices are dynamic as clients can enter/exit randomly and network congestion conditions vary in time.

2.2.2 Interaction Between Resource Reservation and TCP

In scheduling multiple DASH streams over a resource-reserved slice, with no client-side TCP receive-window (rwnd) management from the application, there is three-way interaction between network resource reservation, TCP rate control and application level DASH video rate adaptation algorithm, where standard methods for application level DASH rate adaptation are based on the TCP goodput estimated by the client. The main goal of this section is to show that we can stabilize TCP goodput per client by enabling client-side rwnd control, provided that the slice capacity does not change in time and there are no entry/exit of clients. Under these conditions, the three-way interaction can essentially be considered as a two-way interaction since

the TCP goodput would not vary. However, when the NSP cannot guarantee a fixed rate to a customer/client or a fixed slice capacity to a VSP (soft QoS) and/or there are entry/exit of clients, TCP goodput per client can still change in time and we again have three-way interaction but in a much more controlled manner compared to basic competitive streaming, since with the knowledge of their fair-share bitrate, clients can now set their DASH video rate adaptation and TCP receive-window size in a consistent manner.

We now analyze the interaction between resource reservation at the network level and TCP congestion control at the transport level. Suppose we reserve a 20 Mbps slice in a network and route 4 homogeneous DASH streams over this reserved capacity where the sender/receiver rates are shown in Figure 2.1(e-f). Then, the fair share for each DASH flow would be 5 Mbps. It turns out that network resource reservation alone is not sufficient to achieve smooth throughput around 5 Mbps, hence, smooth QoE, for these DASH streams because TCP is unaware of the reserved bitrate, and the flows continue to compete for more bandwidth due to nature of TCP congestion control mechanism. Although the average bitrate (goodput) for each flow comes near their fair share value over the long run, significant fluctuations of the instantaneous bitrate for each flow results in underutilization of the reserved slice capacity and also variations in the application level DASH video quality levels. Hence, we pose the following question: Assuming we reserve a specific end-to-end (e2e) throughput for a set of DASH streams by one of well-known QoS mechanisms, how can we i) minimize instantaneous throughput fluctuations, ii) maximize video-quality fairness among flows, and iii) maximize utilization of the allocated average bitrate.

The answer is that we need to apply rate control for each TCP flow when their rates reach the fair-share rate reserved for that flow in order to properly manage QoE variability and network utilization. We demonstrate that this can be achieved by client-side receive window management for each TCP flow. The send and receive rates (averaged over 1 sec.) of four TCP flows over a shared slice configured at 20 Mbps are shown in Figure 2.1(e-h) with and without setting optimal receive window sizes. It can be seen that setting *rwnd* to an optimal value, given the fair-share bitrate, at receiving

clients decreases TCP throughput variation significantly and achieves nearly a flat rate for each client near their fair-share value. Moreover, comparing the send rates shows that TCP rate control also helps avoiding bursts at the ingress queues of switches. This observation applies even to the case of dynamic per-flow queue management with one flow per queue (slice). We see from Figure 2.1(a)-(d) that even in the case of a single flow per slice, TCP rate control is needed for a smooth send rate, which helps reducing the flow processing at the sender side switch.

In the following, we propose two different video service models, an NSP-managed service architecture in Section 2.3 and VSP-managed centralized and distributed collaborative streaming architectures in Section 2.4, in order to provide clients their fair-share bitrate values to guide them set both the application-level DASH rate adaptation and TCP receive-window (rwnd) values from the application level.

2.3 NSP-Managed Video Services

In NSP-managed video services, the NSP performs per-client path/bitrate computation and resource provisioning based on client information provided by the VSP. The architecture of the proposed system is presented in Section 2.3.1. QoS provisioning (path/rate calculation) for each client and queue management by the NSP are described in Section 2.3.2 and TCP receive-window size and DASH rate adaptation by the clients are described in Section 2.3.3.

2.3.1 System Architecture

In the proposed system, the NSP deploys three units, Traffic Engineering Manager (TEM), VSP Interface (VSP-I), and Value-Added Services Manager (VAS-M) connected to its SDN controller. The complete system architecture with enumerated information flow is depicted in Figure 2.2.

The information flow for NSP-managed video services consist of the following steps: 1) The client requests value-added services from its NSP through

the VAS-M. 2) The client requests a video from the VSP. 3) VSP passes source and destination IP and video resolution information to the NSP. 4) NSP-TEM computes path/routing and bitrate for client. 5) NSP takes necessary queue management actions to reserve an appropriate bitrate resources for client. 6) NSP passes reserved bitrate information to client. 7) Client sets the TCP receive-window size and DASH video adaptation rate based on this information.

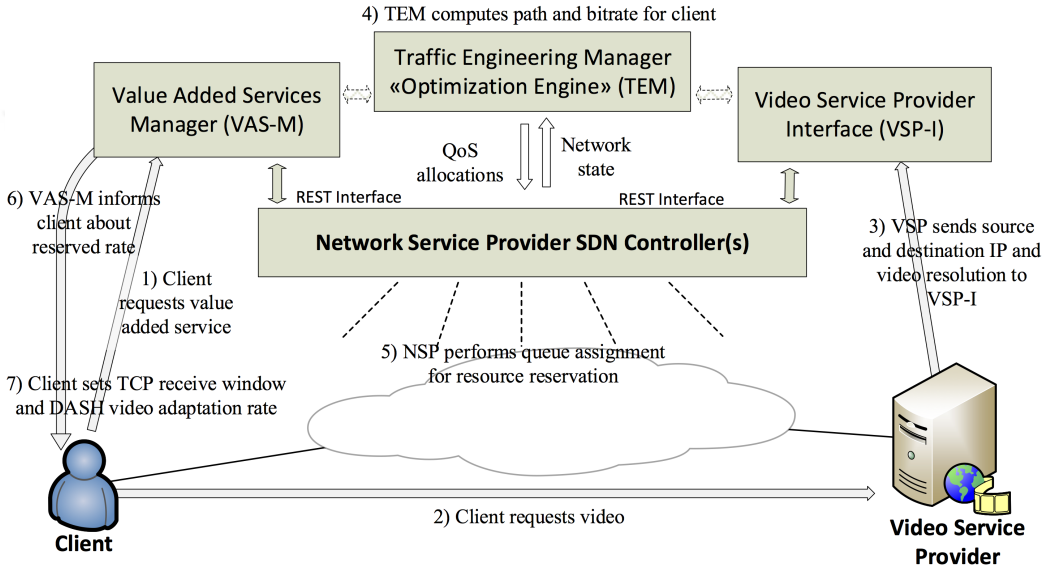

Figure 2.2: System architecture for NSP-managed video services.

We note that the TEM periodically (re-)computes resource allocations and routing decisions based on the up-to-date network state and client/video parameters. Thus, whenever a new DASH client requests a video stream or finishes streaming or network congestion occurs, this collaborative process provides the best instantaneous rate decisions to the clients.

2.3.2 Path / Rate Calculation and Queue Management by NSP

We assume that the source (video server) and destination (client) are in the same SDN network. Resources to be allocated to each client by the NSP-TEM are an e2e path p_i (a set of links from the source to the destination) and

a bitrate b_i, which are computed by the constrained shortest path algorithm that is shown in Algorithm 1. The inputs to Algorithm 1 are a set (L) of links with available capacities and a request $r_i = (s, d, b_i^{req}, \delta)$, where s, d denote source and destination IP addresses, b_i^{req} is the desired bitrate, and δ is the bandwidth search precision for client i. The algorithm first filters the link set L by eliminating those links whose capacity are less than the requested video bitrate b_i^{req} to find a link set L^* of links with sufficient capacity to serve client i (Line 5). Then, using the filtered link set L^*, a directed graph G is formed and the shortest path p_i^* over G is calculated (Lines 6 and 7). If there exists a path p_i^* with the bitrate b_i^{alloc}, then client i is routed over p_i^* (Line 8-10). If there exists no path with the bitrate b_i^{alloc}, then b_i^{alloc} is decremented by δ Mbps (Line 12) and the procedure is repeated until a path p_i^* is found with the maximum bitrate possible.

Algorithm 1: NSP - Path and Rate Calculation Algorithm

```
1  Input: L, r_i, δ
2  Output: p_i, b_i
3  b_i^alloc = b_i^req;
4  do
5      Create L* the set of links with sufficient capacity;
6      Create graph G using links in L*;
7      Find the shortest path p_i* in G;
8      if p_i* exists then
9          p_i = p_i*;
10         b_i = b_i^alloc;
11     else
12         b_i^alloc = max(0, b_i^alloc − δ);
13 while p_i* does not exist;
```

The NSP employs queue management methods for QoS (throughput) reservation at the network level. Configuring multiple queues at switch ports for packet egress provides per-flow rate-limiting ability and enables weighted fairness among flows. Having received per-flow routes and optimal downstream bandwidths to be allocated for each DASH client from the TEM, the SDN controller configures QoS queues at the ports of network edge

switches. Queue management is handled by configuration protocols such as OF-Config [45] or OVSDB [46] based on controller/switch compliancy since both parties need to support the same switch configuration protocol. Once queues are implemented, by using OpenFlow queuing action, traffic flows can be routed over particular queues satisfying the related bitrate requirements. Queues can be configured in two different ways: (i) dynamic configuration of a new queue per flow [36], or (ii) static configuration of a fixed number of queues with some maximum bitrates [12].

In the dynamic configuration, queues are configured on-demand per-flow. For each incoming video service request, a new queue at a specific rate is created and removed once the flow expires. Deployment of soft virtual switches, such as Open vSwitch (OVS) [47], running over high performance computers with sufficient resources at the network edges makes realization of on-demand per-flow queue management feasible. In this approach, since each flow is assigned to a flow-specific queue, traffic burst in one flow does not affect the QoS of other flows as long as the sum of capacities of all queues over a port stay under the link capacity. However, TCP rate control by clients helps reducing flow processing requirements at the ingress switches as demonstrated in Figure 2.1(a)-(d).

In the static configuration, a fixed number of shared queues with certain operating rates are configured once, and incoming video service requests are dynamically assigned to these queues. The number and rates of queues to be configured by the NSP are determined by taking the adaptation levels of DASH contents into consideration since the NSP and VSP collaborate. The idea is to configure specific queues for subsets of these adaptation levels. Let B be the set of average bitrates for different adaptation levels, and B_i is the average bitrate for the i^{th} adaptation level, where i is between 1 and N_{AL}, the number of adaptation levels. We partition B into non-overlapping N_Q subsets. Let B_k^n represent adaptation bitrate k in the n^{th} subset. We configure N_Q queues, such that the rate for n^{th} queue is set to $max(B_k^n) \cdot U_{max}$, where U_{max} is the number of DASH clients that can be served at the highest quality level by n^{th} queue. It is common for VSPs to encode DASH content at different bitrates ranging from 0.3 Mbps up to 20 Mbps at 10 adaptation

levels covering multiple video resolutions, e.g., SD, HD, Full HD, and 4K. For example, these adaptation levels can be partitioned into four non-overlapping subsets and four different queues can be configured where each queue is used for flows of a particular resolution consisting of multiple adaptation levels. We note that assigning multiple flows having similar bitrate requirements to a shared queue enables basic QoE fairness among DASH clients with standard TCP streams. The necessity of TCP rate control to achieve smooth QoE per flow and video-quality fairness has been demonstrated in Figure 2.1(e-h).

2.3.3 TCP Receive-Window / DASH-Rate Adaptation at Clients

It is important to note that resource reservation by queue management alone is not sufficient to control per-flow QoE variability. This is because the standard TCP congestion control is not aware of the reserved network resources and continues to increase its rate until experiencing losses, which results in large fluctuations in the per-flow instantaneous throughput. The throughput fluctuations result in video-quality variations as a consequence of application layer adaptation mechanism at the clients. This is especially noticeable when the number of flows assigned to a particular queue is close to U_{max}, i.e., queue utilization approaches its maximum level.

The video-quality variability can be avoided and quality-fairness can be achieved by applying rate control to each TCP flow by *rwnd* management at clients when the TCP rate reaches the rate reserved for that flow. Once clients are informed by VAS-M about their reserved bandwidths based on the current network utilization, they can dynamically manage their TCP window sizes. Each client periodically estimates the round-trip-time (RTT) to VSP content server (VSP-CS) so that *rwnd* is set to the optimal value $bandwidth \cdot RTT$ and advertised in TCP ACKs.

2.4 VSP-Managed Video Services

In VSP-managed video services, the NSP does not perform per-client resource reservation as was the case in Section 2.3, instead it collaborates with the VSP and provisions a reserved slice to groups of VSP clients over each link. The proposed architecture for VSP-managed video services is introduced in Section 2.4.1. Computation of fair-share bitrates for each heterogeneous DASH client over a slice reserved for the VSP clients on each link is either performed by the VSP centrally, which is described in Section 2.4.3 or by means of a distributed inter-client collaboration scheme, which is presented in Section 2.4.4.

2.4.1 System Architecture

In the proposed system architecture, the VSP supports a Management Server (VSP-M) to manage the collaborative streaming service, and the NSP supports a Traffic Engineering Manager (TEM) for resource provisioning and a VSP Interface (VSP-I) to communicate with the VSP. The proposed system architecture together with the flow of information is depicted in Figure 2.3.

The information flow for VSP-managed video services consist of the following steps: 1) The client requests video service from VSP through the VSP management server (VSP-M). 2) VSP passes source and destination IP to the NSP. 3) NSP-TEM computes shortest-path route for client. 4) NSP passes to the VSP an updated list of all VSP clients on each slice/link. Note that our use of the term slice refers to a particular reserved capacity for the VSP over a single link and not to an end-to-end path. 5) VSP re-computes the bitrate it requests from the NSP for its slice over each link (knowing the bitrate requirements of all of its clients), 6) NSP updates reserved bitrate for the slice allocated to the VSP over each link and informs the VSP. The capacity of reserved slices are determined by the NSP according to how many other non-VSP users are on each link. It is important that the VSP employs a reserved slice in order to isolate VSP clients performing TCP receive window management from other non-VSP users. 7) VSP manages these slices (see Algorithm 2 and its description in Section 2.4.2), where in centralized

collaboration, the VSP computes the fair-share bitrate for each client over each reserved slice and informs clients about their fair-share bitrate (see Section 2.4.3); or alternatively, in the distributed collaboration architecture, the VSP forms groups of clients and lets them collaborate among each other to determine their fair-share bitrates (see Section 2.4.4), 8) Clients set the TCP receive-window size and DASH video adaptation rate based on this information.

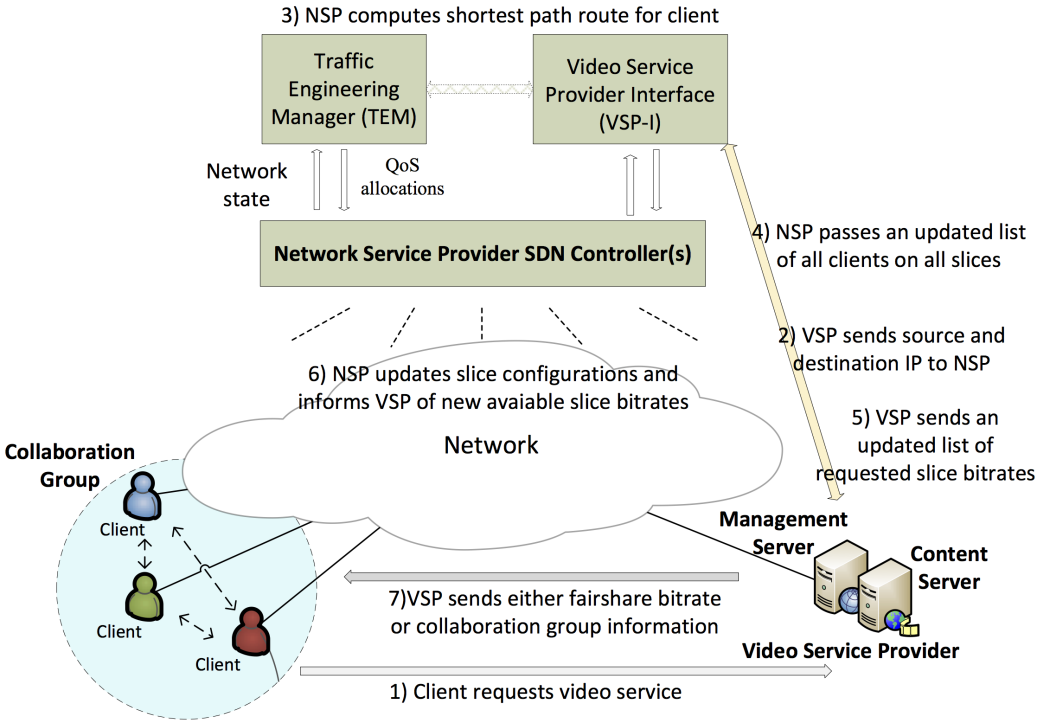

Figure 2.3: System architecture for VSP-managed video services.

2.4.2 Slice Management by VSP

An end-to-end path from a video server to a client consists of multiple links/slices and the VSP need to manage its clients over each slice. In step 7 above, the VSP manages the fair-share bitrates of clients over the set of slices that the NSP allocates to the VSP (step 6 above). The procedure for VSP slice management is described by means of Algorithm 2, which is run by the VSP-M. The input to Algorithm 2 is a list of slices $L = \{l_1, \ldots, l_N\}$

Algorithm 2: VSP - Slice Management Algorithm

Input: L
Output (Centralized Mode): br_k
Output (Distributed Mode): D_j, C_j
$qV = 0.96$;
$r_i = (\frac{qV - c_i}{a_i})^{\frac{1}{b_i}}, i = 1, \ldots, M$;
do
  foreach *slice* l_j *in* L **do**
    Count N_i^j clients on l_j with class $i = 1, \ldots, M$;
    Compute congestion ratio $CR_j = \frac{\sum_{i=1}^{M} r_i \cdot N_i^j}{C_j}$;
  Sort links in L based on CR;
  Create $D = \bigcup_{i=1}^{N} D_i$;
  Initialize br_k for each client with id k in D;
  if *Centralized Mode* **then**
    foreach *slice* l_j *in* L **do**
      $B_j \leftarrow$ Call Alg. 3 with inputs C_j and N_i^j;
      foreach *bitrate* br_k^j *in* B_j **do**
        if $br_k^j < br_k$ **then**
          $br_k = br_k^j$;
          Notify client k about br_k;
  if *Distributed Mode* **then**
    Initialize an empty set D^*;
    foreach *slice* l_j *in* L **do**
      Remove client ids in D^* from D_j;
      Adjust C_j considering the removed clients;
      Notify clients in D_j about $k's$ in D_j and C_j;
      Add client ids in D_j to D^*;
  Sleep for Δt seconds;
  Update link capacities in L;
while *true*;

at each link i, such that $l_i = (C_i, D_i)$, where C_i is the reserved capacity for slice i and D_i represents the set of VSP clients on slice i. We denote the list of all current VSP clients by $D = \bigcup_{i=1}^{N} D_i$. Algorithm 2 involves the following steps:

2.4.2.1 Determining Fair-share Bitrates for Different Video Resolutions

This step explains the theoretical foundation behind the steps 4 and 5 of Algorithm 2. In order to achieve QoE fairness among heterogeneous DASH clients, the fair-share bitrate for each DASH client should depend on their video resolution such that all clients receive the same or similar video quality regardless of the resolution of the video they receive.

The empirical relationship between video quality, measured by Structural Similarity Index (SSIM), and bitrate for $M = 3$ different video resolutions is depicted in Figure 2.4 [37] [42]. These curves can be compactly represented by three "video quality coefficients" by fitting a function of the form $qV = a \cdot r^b + c$ [37] [42], where the coefficients a, b, and c can be computed by least squares curve fitting. The resulting coefficients for $M = 3$ different video resolutions are provided in Table 2.1. Then, given a particular quality value qV^*, the fair-share bitrates r_i for resolution type $i = 1, \ldots, M$ can be calculated from

$$r_i = \left(\frac{qV^* - c_i}{a_i}\right)^{\frac{1}{b_i}}, i = 1, \ldots, M \tag{2.1}$$

where a_i, b_i, and c_i denote coefficients for resolution type i shown in Table 2.1.

Table 2.1: Coefficients of the video quality function [37].

	Power Series Model			Goodness of Fit	
Device Type	a	b	c	Adjusted R^2	RMSE
1080p	-3.035	-0.5061	1.022	0.9959	0.006011
720p	-4.85	-0.647	1.011	0.9983	0.002923
360p	-17.53	-1.048	0.9912	0.9982	0.002097

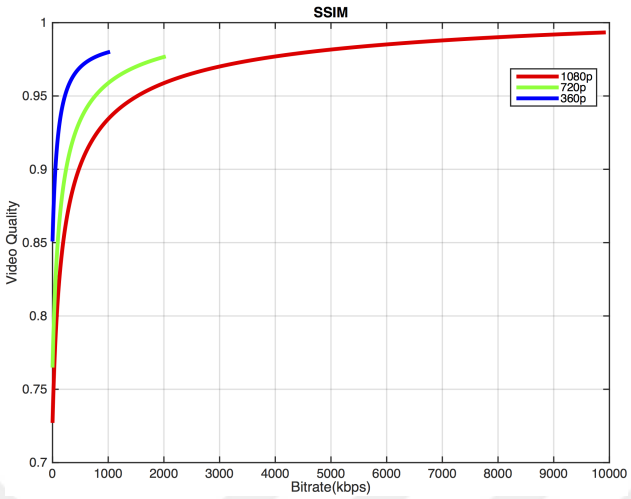

Figure 2.4: Mapping between video quality and bit-rate according to three types of display resolution [37]

2.4.2.2 Sort the Slice List According to Congestion Ratio

Steps 6-10 of Algorithm 2 computes the congestion ratio for each slice. The congestion ratio CR_j for a slice j is defined as the total requested traffic, given by the sum of fair-share bitrates of clients over slice j, divided by the capacity C_j of slice j. Hence, it is given by $CR_j = \frac{\sum_{i=1}^{M} r_i \cdot N_i^j}{C_j}$, where the fair-share bitrates r_i are computed from Eqn. 2.1 by setting qV to a reasonable value. VSP performs slice management starting from the highest congested slice since it is more likely to create bottleneck along the path. Hence, the VSP sorts slices based on CR in decreasing order (Line 10).

2.4.2.3 Computation of Fair-Share Bitrates

In the centralized architecture, VSP computes the fair-share bitrate of each client after traversing over all slices in the sorted link set L starting from the highest congested link. Using the current capacity C_j of slice j and the number N_i^j of clients with resolution type i over slice j as inputs to Algorithm 3 (described in Section 2.4.3), VSP computes the fair-share bitrates B_j of all clients on slice j (Line 15). After processing each slice j, the algorithm checks if the fair-share bitrate br_k^j of client k is less than its present value br_k. Since the e2e bitrate of each client is determined by the smallest rate over all slices,

Algorithm 3: VSP - Centralized Fair Share Algorithm

Input: C, $N_i, i = 1, \ldots, M$
Output: $r_i, i = 1, \ldots, M$
$\epsilon = 100$;
$\Delta qV^* = 0.01$;
$qV^* = 0.8$;
$total = 0$;
while $|C - total| > \epsilon$ **do**
    $r_i = (\frac{qV^* - c_i}{a_i})^{\frac{1}{b_i}}$, $i = 1, \ldots, M$;
    $total = \sum_{i=1}^{M} N_i \cdot r_i$;
    if $total < C$ **then**
        $qV^* = qV^* + \Delta qV^*$;
    else
        $\Delta qV^* = \frac{\Delta qV^*}{2}$;
        $qV^* = qV^* - \Delta qV^*$;

the VSP notifies client k about its new fair-share bitrate if br_k^j is smaller than the present value. (Line 16-19). We note that our experiments indicate that the smallest bitrate is always the one computed over the highest congested slice.

In the distributed architecture, VSP forms groups of clients after going over all slices in the sorted set L. Starting with the highest congested slice, VSP forms a new group using the client set D_j of slice j. It excludes clients in the list D^* that have previously been assigned to a group from the client list D_j and adjusts the slice capacities C_j accordingly for all j. Once a group is formed, VSP notifies all clients in the group D_j about the list of peers indexed by k and reserved group capacity C_j of slice j (Line 21-26) so that clients perform distributed collaboration to compute their fair-share bitrates as described in Section 2.4.4.

Since the network state and/or the number of DASH clients vary over time, the per-client fair-share bitrate calculation in the centralized architecture and grouping in the distributed architecture must be dynamic. Hence, the procedure needs to be periodically repeated at every Δt seconds, i.e., the VSP periodically triggers slice management using the updated link informa-

tions and clients (Line 27-29). The while loop runs as long as there are VSP customers using the managed video service. However, the outputs are made available periodically every Δt seconds.

2.4.3 Centralized Collaboration

In the centralized collaboration architecture, the fair-share bitrates for each client is computed by the VSP Management Server as described in Algorithm 3. The inputs to the algorithm are the number of clients N_i for each video resolution type $i = 1, \ldots, M$ and the slice capacity C. The algorithm aims to find the maximum video quality qV^* such that the sum of the fair-share bitrates $r_i, i = 1, \ldots, M$ corresponding to this qV^* for all M clients over the slice is equal to the capacity C. This is achieved by means of an iterative search procedure in Lines 7-14 of Algorithm 3. We set the parameters convergence margin ϵ and search step size ΔqV^* in Lines 3-4. The variables total bitrate *total* and video quality qV^* are initialized in Lines 5-6. The initial qV^*, that is, the initial common video quality value for all clients is set to a low value so that the resulting initial total bitrate does not exceed the slice capacity. The bitrates r_i for different video resolutions corresponding to a given qV^* at current iteration are obtained by Eqn. 2.1, where a_i, b_i, and c_i for video resolution type i are provided in Table 2.1 (Line 8). The total bitrate *total* for all clients given the current qV^* value is computed in Line 9. If the total bitrate does not exceed slice capacity C, we increment the value of qV^* by the stepsize ΔqV^*. Otherwise, stepsize ΔqV^* is halved, and qV^* is decremented by the new ΔqV^*. Iterations continue until the total bitrate converges to within ϵ neighborhood of C. Once the fair-share bitrates r_i are computed, the VSP informs clients about their fair-share bitrates, so that each client updates its TCP receive-window size *rwnd* and DASH video adaptation rate as described in Section 2.3.3.

2.4.4 Distributed Client-Collaboration

In the proposed distributed collaboration scheme, once clients receive grouping information (IP addresses and resolution types/coefficients of peering

clients) and reserved capacity for their group from the VSP, they collaborate with each other to determine their own fair-shares of the capacity reserved for the group rather than competing with each other.

Implementation of inter-client collaboration requires modification of the DASH client as depicted in Figure 2.5. The collaboration functionality is provided by (i) a new messenger module, which manages communication of a DASH client with other members of the collaboration group and the VSP-M for receiving/sending messages, and (ii) buffer state aware fair-share computation (Algorithm 4) module.

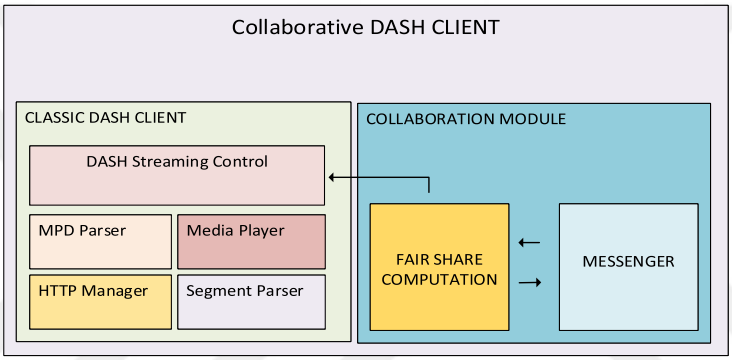

Figure 2.5: A proprietary DASH client with inter-client collaboration function.

Given that each DASH client has access to the video quality coefficients of all group members, the reserved capacity C for the group, and most recent buffer durations d_{ij} for client j with video resolution i, each client can compute its own bitrate using Algorithm 4 that is similar to Algorithm 3 but additionally uses playback buffer information of clients in the collaboration group. The idea is to scale the bitrates to be allocated for each client according to the respective playback buffer conditions. DASH clients with buffer status lower than the average buffer fullness $\overline{d}_i$ of VSP-clients with a particular resolution type i on a given group are more likely to experience a stall event. Therefore, DASH clients with buffer status higher than this average level should be allocated a proportionally smaller bitrate than their fair-share bitrate in order to allow a higher share of the slice capacity to clients with buffer status lower than the average. To this effect, we compute the average

Algorithm 4: Clients - Distributed Fair Share Algorithm

Input: C, d_{ij}
Output: r_i, α_{ij}
$\epsilon = 100$;
$\Delta qV^* = 0.01$;
$qV^* = 0.8$;
$total = 0$;
$\overline{d}_i = \frac{\sum_{j=1}^{N_i} d_{ij}}{N_i}, i = 1, \ldots, M$;
while $|C - total| > \epsilon$ **do**
    $r_i = (\frac{qV^* - c_i}{a_i})^{\frac{1}{b_i}}, i = 1, \ldots, M$;
    $\alpha_{ij} = 2 - \frac{d_{ij}}{\overline{d}_i}, i = 1, \ldots, M$ and $j = 1, \ldots, N_i$;
    $W_i = \sum_{j=1}^{N_i} \alpha_{ij}, i = 1, \ldots, M$;
    $total = \sum_{i=1}^{M} W_i \cdot r_i$;
    if $total < C$ **then**
        $qV^* = qV^* + \Delta qV^*$;
    else
        $\Delta qV^* = \frac{\Delta qV^*}{2}$;
        $qV^* = qV^* - \Delta qV^*$;

buffer fullness measure $\overline{d}_i = \frac{\sum_{j=1}^{N_i} d_{ij}}{N_i}$ (Line 7), where N_i denotes the number of clients with resolution type i in the group and d_{ij} is the normalized buffer duration such that $0 \leq d_{ij} \leq 1$ for all ij. Deviation factor $\frac{d_{ij}}{\overline{d}_i}$ represents a measure of relative buffer state of client j among the client with resolution type i, hence, the buffer scale factor α_{ij} is computed as $2 - \frac{d_{ij}}{\overline{d}_i}$ (Line 10). In this way, total weight W_i for resolution type i is computed and buffer status aware total bitrate equation is formed (Line 10-11). Similar to Algorithm 3, Algorithm 4 iteratively searches for the highest possible fair-share bitrates r_i. As a result, client j with resolution type i computes its buffer-aware fair-share bitrate as $r_i \cdot \alpha_{ij}$ and updates its TCP receive-window size and DASH video adaptation rate accordingly.

Accurate collaboration among DASH clients requires periodic communication, which is possible by soft synchronization. For synchronization, each client receives the current time-stamp from the VSP Management Server only once when they first connect to the server, and use it (after adjusting by the estimated RTT) as their reference wall-clock. Thus, the communication overhead for collaboration synchronization is negligible. Periodicity of collaboration is essential since the state of client buffers vary with time, and DASH clients should inform other group members about their buffer status to adjust their fair-share according to the new conditions. For effective collaboration, clients need recent playback buffer information of other clients in the group. To this end, clients periodically send unicast messages to notify each other about their present playback buffer status so that they can collaboratively set their TCP receive window sizes and decide next segments adaptation level based on group QoE fairness. For a group of N collaborating clients, there exist $N \cdot (N-1)$ messages at each collaboration cycle. Since each message carries only a single number (representing playback buffer duration) of 4 bytes and the collaboration period is typically 5-10 seconds, the communication overhead due to inter-client information exchange is also very small.

2.5 Evaluation

2.5.1 Experimental Setup

We evaluate the performance of NSP-managed and VSP-managed (both centralized and distributed collaboration) video services over an SDN, whose data plane consists of a partial mesh network generated by Mininet [48] using OVS switches [47], which are connected to a Floodlight [49] controller. Network topology contains 11 switches, where each switch has at least one client as shown in Figure 2.6. We run our experiments on network slices with reserved capacity; hence, the only source of cross-traffic is other collaborating DASH clients using the same network slice. In our implementation, TEM and VSP-I units of the NSP expose resource allocation/routing and NSP-VSP collaboration functionalities, respectively, as web services employing JSON based APIs.

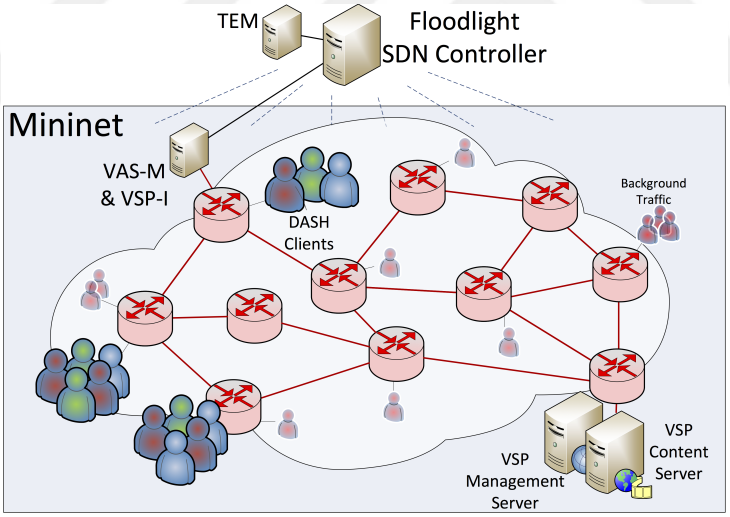

Figure 2.6: Exemplary network used for evaluations.

The VSP is represented by one of the hosts that runs a management server (VSP-M) and a content server using HTTP 1.1 with persistent connections. Content server stores the media presentation description (MPD) files and encoded segments. The test video is *Tears of Steel* with duration 12 minutes and variable bitrate (VBR) encoding. The available resolutions, adaptation

Table 2.2: Resolutions and encoding parameters for test video content.

Content	Resolution	Bitrates (Mbps)	segment (sec)
Tears of Steel	1080p	1.5, 2.4, 3, 4, 6, 10	2, 4, 10
	720p	0.50, 0.80, 1.5, 2.4	
	360p	0.50, 0.80	

level average bitrates and encoding parameters are shown in Table 2.2.

Our DASH client uses Java sockets for communication with the VSP-M and the group members (in distributed collaboration), and also to set calculated receive window sizes. Clients estimate RTT as the time difference between the instant when a segment request is sent to the content server and the instant when the first byte of the response is received. Since we evaluate the performances of the competitive and the proposed collaborative approaches, and not the effect of the application layer adaptation heuristic, our client implementation uses a conventional application layer rate adaptation method which is based on the instantaneous ready-to-play buffer duration. Hence, the same application layer adaptation algorithm is triggered when comparing different approaches.

2.5.2 Results

We perform four different sets of experiments. The first set of results compare the performance of the proposed collaborative architectures (NSP-managed, VSP-managed centralized, VSP-managed distributed) with those of SDN-based QoE-aware DASH streaming (available in the literature) and basic competitive DASH streaming over a fully utilized network slice. The second set of results show the performance of both centralized and distributed VSP-managed collaborative streaming when clients enter and exit at random times. The third set of results show the robustness of VSP-managed distributed collaboration in a pessimistic scenario with a large number of clients in a group and limited available reserved slice capacity. Finally, the fourth set of results study the effects of DASH segment size and network slice capacity on the performance of competitive vs. collaborative streaming.

In particular, each experiment is repeated 10 times and we report the

mean and variance, in the case of video quality qV, of the results of these runs, as well as mean number of stalls (MNS) and the mean duration of stalls (MDS) in seconds with different number of video resolutions and clients, varying DASH segment durations, and varying reserved link capacity. We also define a new measure, called low-quality penalty (LQP), which quantifies segments that are received with bitrate c_i which are below the fair share bandwidth b_i for client i. The LQP score of client i for a content with T segments can be calculated as $\frac{\sum_t max(b_i - {c_i}^t, 0)}{T}$, where ${c_i}^t$ denotes the average bitrate of the segment received by client i at time t. The metric MQP, which we report, is the mean of LQPs over all clients that are served.

In the first group of results, we compare the performances of i) Competitive: Basic DASH streaming (with no resource reservation and no TCP receive-window (rwnd) control), ii) SDN-DASH: DASH streaming with SDN-based centralized fair-share bitrate calculation per-client (but no client-side rwnd control), where clients only use this bitrate in their application layer DASH rate adaptation heuristic ([37]), iii) NSP-managed: DASH streaming with fair-share bitrate calculation per-client, where clients use this bitrate for client-side rwnd control and application layer DASH rate adaptation, iv) VSP-managed (Centralized): DASH streaming with VSP-based centralized fair-share bitrate calculation for a group of DASH clients over a slice, and v) VSP-managed (Distributed): A group of clients over a reserved slice communicate with each other to compute their own fair-share bitrates. We first study the case of homogeneous clients, where all clients receive 1080p content, and then the case of heterogeneous clients, where clients receive 1080p, 720p, or 360p video.

In the homogeneous clients scenario, we have four 1080p DASH clients over a 20 Mbps slice so that the fair share bitrate for each client is 5 Mbps. Each DASH client streams *Tears Of Steel* with 4 sec. segments. In the cases of competitive DASH and SDN-DASH streaming, application-level DASH adaptation bitrates for all clients fluctuate frequently between 2.4 Mbps (lowest) and 10 Mbps (highest) adaptation levels. In contrast, in all types of collaborative architectures, we observe that bitrate adaptation for all DASH clients only vary between 4 and 6 Mbps, where the fair-share bitrate for each cli-

Table 2.3: Comparison for four homogeneous 1080p clients.

Method	Mean QV	Var QV	MNS	MDS	MQP
Competitive	0,9752	0,00006	1	3,5	1,13
SDN-DASH	0,9765	0,00006	1	1,2	0,94
NSP-Managed	0,9776	0,00004	0	0	0,85
VSP-Managed (Centralized)	0,9776	0,00004	0	0	0,85
VSP-Managed (Distributed)	0,9777	0,00003	0	0	0,83

ent is 5 Mbps. We also observe from Table 2.3 that both NSP-managed and VSP-managed (centralized and distributed) collaborative frameworks provide better results (for all measures) than competitive and SDN-DASH streaming. In particular, the MDS and MQP values for competitive DASH clients are significantly higher than those of both types of collaboration. We also note that the VSP-managed distributed collaborative framework achieves the highest mean video quality (and smallest variance) and does not experience any stalls since all clients in a group are aware of buffer status of each other.

In the heterogeneous clients scenario, we present results for three clients with video resolutions 1080p, 720p and 360p, respectively, streaming the video *Tears Of Steel* with 4 sec. segments over a reserved network slice with capacity 7 Mbps. In VSP-managed centralized collaborative streaming, the fair share bitrates for these clients can be computed as 4.02 Mbps for 1080p, 2.09 Mbps for 720p and 0.86 Mbps for 360p clients, which correspond to the fair video quality value $qV = 0.9765$ (see Algorithm 3). Clearly, how close we can achieve this qV value depends on how close the determined bitrate values are to the DASH adaptation bitrate sets of a particular content (available at the content server), since each DASH client starts streaming with the nearest adaptation level to its fair-share bitrate. Table 2.4 shows that the ensemble mean of quality values of different resolution video clients over 10 runs are

closer to the computed qV value in collaborative streaming (compared to competitive streaming), which is an indicator that we can achieve fairness among heterogeneous DASH clients. In the case of 360p client, we see that the competitive streaming is as good as collaborative streaming since the 360p client gets an unfair share of the bitrate at the expense of 720p and 1080p clients due to the nature of TCP congestion control. A single realization of quality qV vs. time for three different clients in the cases of competitive, VSP-centralized, and VSP-distributed are shown in Figures 2.7a, 2.7b and 2.7c, respectively. In general, the variance of quality is smaller in collaborative streaming compared to that of the competitive streaming with the exception of 360p resolution (discussed above). We also observe that DASH clients with 720p and 1080p resolutions in the competitive systems experience more stall events. All collaborative systems show comparable results for DASH clients with 360p and 720p resolutions. Our experiments with other video indicate that these observations are independent of the content and bitrates.

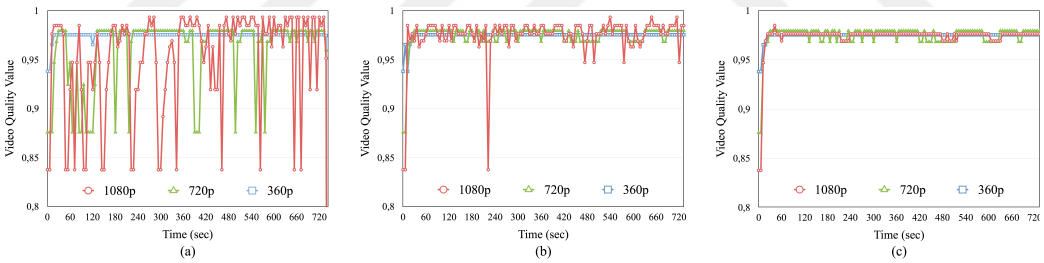

Figure 2.7: Single realizations of competitive vs. collaborative streaming results for three heterogeneous clients using Tears of Steel: a) competitive, b) VSP-managed centralized collaboration, c) VSP-managed distributed collaboration.

In the second set of results, we compare the performance of competitive vs. VSP-managed collaborative DASH clients in a dynamic scenario where clients enter/exit at random times. In this scenario, the reserved slice capacity for DASH flows is 10 Mbps, and we initially have 3 heterogeneous DASH clients that stream *Tears Of Steel* with 4 sec segments. In the centralized collaboration, the fair-share bitrates (without considering client buffer state) are computed by the VSP as 5.5 Mbps for 1080p, 2.9 Mbps for 720p and

Table 2.4: Comparison for three heterogeneous clients using 'Tears of Steel' with 4 sec. segments.

Method	Resolution	Mean QV	Var QV	MNS	MDS	MQP
Competitive	1080p	0,9503	0,0005	8	3,5	1,69
	720p	0,9642	0,0003	4	2,3	0,53
	360p	0,9746	0,00001	0	0	0,42
SDN-DASH	1080p	0,9637	0,0001	5	3	0,7
	720p	0,9668	0,0002	4	1,7	0,48
	360p	0,9746	0,00001	0	0	0,42
NSP-Managed	1080p	0,9720	0,00055	1,3	1,5	0,3
	720p	0,9738	0,00020	1,2	1,2	0,26
	360p	0,9746	0,00002	0	0	0,42
VSP-Managed (Centralized)	1080p	0,9703	0,00002	1	1,6	0,55
	720p	0,9721	0,00006	0	0	0,35
	360p	0,9746	0,00001	0	0	0,42
VSP-Managed (Distributed)	1080p	0,9715	0,00002	1	1,5	0,34
	720p	0,9735	0,00005	0	0	0,28
	360p	0,9746	0,00001	0	0	0,42

1.5 Mbps for 360p to achieve a common quality $qV = 0.9832$. In the both centralized and distributed collaboration, when a new client enters or exits the new fair share bitrates are recomputed centrally or by group collaboration, respectively. After 100 sec, a new DASH client starts streaming the same content at 1080p resolution with 10 sec segments which decreases qV by 0.009 for all clients over the same slice. After 200 sec, another client starts streaming 360p resolution, which causes a further decrease in qV by 0.002, and the fair-share bitrates become 3.4 Mbps for 1080p, 1.8 Mbps for 720p, and 0.7 Mbps for 360p. One of the 1080p clients quits streaming at time 300 sec., which results in updating fair-share bitrates to 4.9 Mbps for 1080p, 2.6 Mbps for 720p, and 1.2 Mbps for 360p with a common $qV = 0.981$. We see from Figure 2.8(a) that competitive DASH clients experience qv fluctuations throughout a session since they are not aware of the dynamically entering/exiting DASH clients, while Figure 2.8(b-c) show that both central and distributed collaboration results in more stable qv values in time as they keep up with computed qV values throughout a session. In addition, we see from Table 2.5 that competitive 1080p clients suffer considerably from stall events. Furthermore, competitive 1080p clients experience undesired high video quality fluctuations, while collaborative clients provide much better stall results and smaller qv variance for 1080p clients.

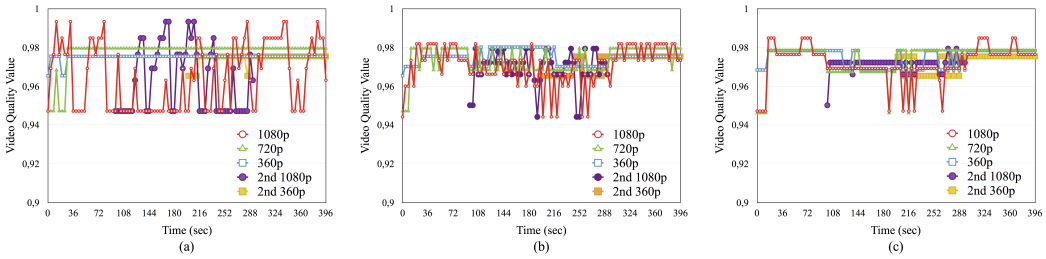

Figure 2.8: Single realizations of competitive vs. collaborative streaming results when clients enter/exit randomly: a) competitive, b) VSP-managed central collaboration, c) VSP-managed distributed collaboration.

In the third set of results, we consider "a pessimistic scenario" for VSP-managed distributed streaming with 12 heterogeneous clients, where 6 clients are 1080p, 4 clients are 720p, and two clients are 360p, in a group that share

Table 2.5: Comparison when clients enter/exit randomly.

Method	Resolution	Mean QV	Var QV	MNS	MDS
Competitive	1080p	0,9651	0,0002	28	111,6
	720p	0,9778	0,00005	4	12,2
	360p	0,9752	0,00002	0	0
VSP-Managed (Centralized)	1080p	0,9709	0,00009	6	11,6
	720p	0,9733	0,0001	3	4,2
	360p	0,9749	0,00001	0	0
VSP-Managed (Distributed)	1080p	0,9719	0,00008	5	8,5
	720p	0,9742	0,00006	2	2,8
	360p	0,9750	0,00002	0	0

a low capacity network slice, where the slice capacity is set equal to the sum of the lowest available adaptation rates required to serve 12 users, i.e., the slice capacity is set equal to 21.5 Mbps. All clients stream *Tears Of Steel* with 10 second segments. Graph of a single realization of video quality vs. time for competitive vs. distributed-collaborative streaming clients is shown in Figure 2.9. We see even in this resource limited scenario that we achieve fairness, that is, there does not appear any critical difference between the mean quality values of different resolution DASH clients. Furthermore, in the competitive framework shown in Figure 2.9 (a), the variance of video quality of each DASH client is very large compared to the case of collaborative clients, and the mean video quality values given in Table 2.7 for different resolution flows are unacceptably low for 1080p DASH clients. In general, collaborative streaming achieves outstanding results compared to competitive streaming.

Table 2.6: Analysis of the effect of DASH segment duration.

Method	MNS		MDS		MQP	
	Segment Duration		Segment Duration		Segment Duration	
	2 sec.	10 sec.	2 sec.	10 sec.	2 sec.	10 sec.
Competitive	28.2	4	18	3	1.55	0.96
VSP-Managed (Centralized)	4.5	0	0.8	0.	0.83	0.84

In collaborative streaming, we observe adverse effect of limited slice re-

Table 2.7: Comparison for 12 heterogeneous clients sharing a 21.5 Mbps network slice.

Resolution	competitive					VSP-managed distributed collaboration				
	mean qv	var qv	MNS	MDS	MQP	mean qv	var qv	MNS	MDS	MQP
1080p	0.9255	0.0028	5.7	3.6	0.65	0.9530	0.0004	0.45	1.2	0.42
720p	0.943	0.0014	3.4	2.24	0,19	0.9467	0.0002	0	0	0.12
360p	0.972	0.0001	1.9	1.54	0.01	0.9411	0.00009	0	0	0.09

sources in the case of 360p clients, where the fair-share bitrate is computed as 0.45 Mbps while the lowest available video adaptation level bitrate is 0.5 Mbps. As a result, it can be seen from Figure 2.9(b) that 360p clients experience lower overall mean qV values compared to 720p and 1080p clients. Competitive streaming achieves better quality values for 360p clients at the expense of 720p and 1080p clients as expected because of almost equal sharing of the slice capacity among heterogeneous clients.

In the fourth set of experiments, we consider two limiting cases: i) what happens as we increase the capacity of the reserved slice and ii) what happens when the DASH segment duration increases in competitive vs. collaborative streaming. To answer the first question: Inspection of Table II shows that we need at least 13.2 Mbps slice capacity to stream video to a 1080p, a 720p and 360p client at the highest quality level. In order to analyze the effect of increasing slice capacity on the comparison of competitive vs. collaborative heterogeneous streaming, we gradually increase the slice capacity from 5.5 Mbps up to 14.5 Mbps while keeping the number of the DASH clients fixed at three. Figure 2.10 shows achievable qv values for different resolution competitive (dotted lines) and collaborative (solid lines) clients. We see that in competitive streaming 360p client operates above its fair-share bitrate until the slice capacity reaches 6.5 Mbps and 720p client operates above its fair-share until the slice capacity reaches 11.5 Mbps at the expense of the 1080p client. We also observe in collaborative streaming that qv values for all three clients remain close until the slice capacity reaches 6.5 Mbps, showing that we can achieve fairness among clients when we have a limited capacity slice. We next observe that when we have sufficient slice capacity (exceeding 6.5 Mbps) to serve 360p and 720p clients at their highest quality levels (2.4 Mbps and

0.8 Mbps, respectively), then 1080p client starts receiving video at gradually higher bitrates. Finally, as the slice capacity increases such that we have unconstrained capacity, the quality levels achieved by competitive streaming converges to those of collaborative streaming. To answer the second question: we observe from Table 2.6 that as the segment duration increases from 2 sec. to 10 sec. there is considerable improvement in mean number and duration of stalls in competitive streaming. This is due to longer chunk durations allow more smoothing (averaging of available goodput during the chunk duration), while smaller chunk durations generate more traffic overhead due to increased number of chunk requests and leads to under-utilization of link capacity due to effect of RTT. However, collaborative streaming still yields significantly better results than those of competitive streaming in both cases.

Overall, we see that the proposed collaborative architectures provide significant improvements in the mean and variance of video quality values as well as the number and duration of stall events compared to competing heterogeneous DASH clients. Although the fair quality value, hence fair-share bitrates for each client, is directly related to the reserved slice capacity and the number of heterogeneous DASH clients in a reserved slice, we can see from the results that it is also important that the encoding rates of the available adaptation levels match the fair-share bitrates for heterogeneous clients in order to achieve the maximum possible fair quality value for all clients. Furthermore, the adjustment of the TCP receive window size is a key factor to prevent rate instability per client and maximize utilization of reserved resources. Finally, comparing the cases of 2 sec. and 10 sec. segments, we observe that QoE variability increases when shorter duration segments are employed in the competitive (default TCP) clients.

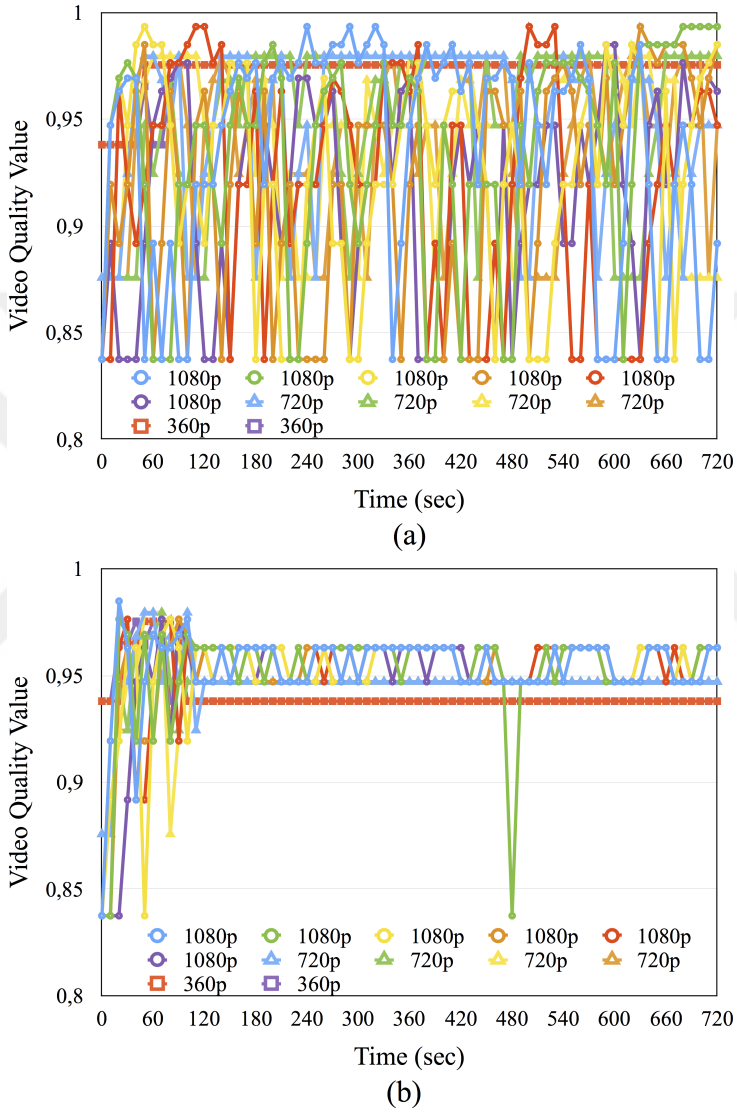

Figure 2.9: Single realizations of competitive vs. collaborative streaming results for 12 heterogeneous clients: a) competitive, b) VSP-managed distributed collaboration.

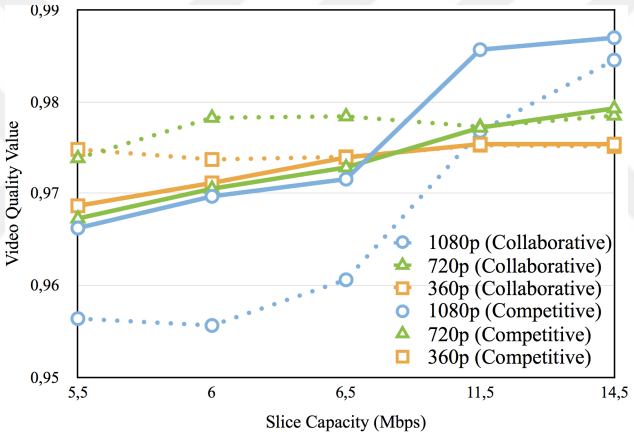

Figure 2.10: Convergence of video quality in competitive vs. collaborative streaming as the slice capacity increases.

2.6 Conclusions

Future networks deploying SDN will make e2e QoS provisioning more practical and feasible. Hence, it will be possible to offer NSP-managed DASH services to individual clients or VSP-managed DASH services to a group of clients over network slices with reserved capacity. We observe that reserving network resources alone is not sufficient to i) minimize video-quality variability per client, ii) maximize video-quality fairness among heterogeneous clients, and iii) maximize total goodput over the reserved slice capacity, because of the nature of the TCP congestion control mechanisms. Hence, we propose and implement SDN-assisted centralized and distributed architectures for collaboration between NSP, VSP, and DASH clients to estimate a fair-share rate for each client, so that clients can set their TCP receive-window size and DASH video adaptation rate consistently in order to avoid unnecessary packet losses and TCP goodput fluctuations. Although various architectures for centralized collaboration between the NSP, VSP and clients have been proposed before, our proposal to include client-side TCP receive-window size control to stabilize TCP goodput at the transport level and perform application level DASH rate adaptation that is consistent with TCP goodput is novel. Moreover, the concept of inter-client (distributed) collaboration is a novel approach to achieve QoE fairness between heterogeneous DASH clients. Furthermore, the proposed dynamic grouping of clients provides robustness of collaboration against changing number of clients sharing the same link as well as changing network conditions. Experiments show that the proposed managed DASH service architectures provide less frequent video quality variations per client in smaller amounts and almost no freezes compared to competitive DASH streaming contributing towards a higher QoE.

Chapter 3

Value-Added Video Services over Single Operator SDN

3.1 Introduction

The best-effort nature of the current Internet leads to inefficiencies for all parties involved in a video service; namely, network service providers (NSP), over-the-top (OTT) video service providers (VSP), and users (clients). From the perspective of the NSP, increasing volume of video services does not contribute to their revenues since NSPs commonly charge a flat monthly subscription fee for the best-effort Internet access. From the perspective of OTT VSP, unlike IPTV services that are offered over a managed private IP network, they cannot promise specific quality of experience (QoE) to their users as they rely on the best-effort service offered by some NSP. From the perspective of end-users, they do not have a choice to receive video services with some level of QoE for an extra cost. Recognizing these shortcomings, the vision of open multi-service internetworking, including different service-levels, service-level awareness of users, and associated business models, was recently discussed in a whitepaper [2]. A more detailed discussion of related work on value-added services can be found in Section 3.2.3.

Software defined networking (SDN) is a programmable networking paradigm and a key technology for 5G networks. We propose to build the proposed

value-added video services framework over SDN since it provides a practical framework for dynamic QoS provisioning since the controller can have an overall view of all network resources, including switch/queue states and traffic load. A more detailed analysis of some of the recent methods is presented in Section 3.2.2.

In the current Internet, the state-of-the-art in resource allocation and path computation is to process each service request sequentially on first-come first serve basis. While this may be an acceptable approach with a single-level of service (best-effort) for all users, it is not the optimal approach in future multiple service-level networks considering fairness among users and the total revenue for the NSP, especially when the network is in congestion mode. We propose an alternative batch optimization of a small group of service requests simultaneously, which helps accommodating service requests that otherwise may be denied, and also increase revenues for the NSPs. Resource allocation for a hard QoS request is considered together with admissions control, while soft QoS requests are allocated to the extent possible. We elaborate more on this in Section 3.3.4 and Section 3.4.

This chapter proposes (i) a service architecture and an implementation to realize value-added video services over an open multi-service network, where users can select the service-level that they desire to pay for, and (ii) a dynamic batch-optimization framework in order to provide the users with the service level they ordered while utilizing the network resources most efficiently and providing the NSP the highest possible revenue. The proposed architecture and methodology applies to both streaming video services as well as real-time communications services. The main novelties presented by this chapter are:

- a batch-optimization framework for dynamic resource allocation for a group of flows with multiple service-levels to maximize revenues for the NSP,
- a divide-and-conquer procedure to compute the optimal solution to the multi-service traffic engineering optimization problem approximately,
- a fast heuristic group constrained-shortest-path (GCSP) procedure that performs close to the above approximately optimal method,

- demonstrating that batch optimization significantly improves revenue, fairness, and utilization compared to sequential processing in congested mode of operation,
- an architecture for users to interact with the NSP to choose the service level they wish to pay for, and
- a method to implement multiple service levels at SDN switches.

Efficient management of future open multi-service networks will offer (i) OTT VSPs the same benefits as IPTV by enabling them to offer their users standard and premium content, e.g., HD and UHD with some QoE, (ii) users the choice of service-level that they are willing to pay for, and (iii) NSPs additional revenues from these value-added services.

The rest of this chapter is organized as follows: Related works are reviewed in Section 3.2. The architectural elements of new value-added video services over SDN are presented in Section 3.3. Section 3.4 introduces the proposed dynamic batch-optimization of resource allocation for a group of flows. Setup for experimental evaluation and results are shown in Section 3.5. Conclusions are presented in Section 3.6.

3.2 Related Works

This section discusses recent works that are related to our proposed solution. Section 3.2.1 overviews QoS and resource allocation methods over the Internet, Section 3.2.2 presents recent approaches to QoS over SDN, and Section 3.2.3 reviews the concept of value-added services by network operators.

3.2.1 QoS and Resource Allocation

The need for differentiated quality connectivity has been recognized and researched for many years. Quality of service (QoS) refers to the ability to manage throughput, delay, jitter, and packet loss of certain flows by providing them the network resources they need. Classical methods to provide

QoS include integrated services (Intserv) that offers hard QoS but suffers from scalability issues, and differentiated services (Diffserv) that only offers soft QoS by marking the type of service (ToS) in the IP header. The use of the ToS byte together with multi-protocol label switching (MPLS) [3], also known as MPLS Diffserv, is common in private networks. The IETF has published standards to help QoS provisioning using these methods, including Application Layer Traffic Optimization (ALTO) [4], which makes network data available to applications, and Path Computation Element to faciliate computation of routes as a network service [5]. However, none of these have been designed for dynamic resource reconfiguration.

Recent research on resource allocation and QoS routing can be classified as sequential (first-come first-served) and group-based optimization methods. Among the sequential (one client at a time) processing algorithms, Kuipers et al. [50] present an overview of early constraint-based path selection algorithms for QoS routing, Xue et al. [51] discuss various approximation algorithms for multi-constrained path (MCP) problems and propose an improved approximation scheme, and Chen et al. [52] presents two approximation algorithms for the constrained shortest-path (CSP) problem. Group-based methods optimize resource allocation of a group of clients simultaneously. Zhang et al. [53] propose a multi-service class based admission control and routing scheme to maximize revenue by guaranteeing the connection blocking probabilities of higher priority service or handoff connections in IEEE 802.16-based wireless networks. Zhu et al. [54] proposes to jointly optimize path provisioning to multiple heterogeneous clients. Wen et al. [55] surveys existing research on cloud mobile media and formulates a group-based resource allocation scheme for the cost of ownership of cloud resources for VSP operators. Adaptive video coding [56] and aggregation of bandwidth [57] were also considered for QoS in video services.

3.2.2 QoS over SDN

A survey of QoS approaches in OpenFlow-enabled SDN environments can be found in [35]. Strategies for QoS provisioning over SDN include queue management over the shortest-path, flow re-routing when capacity along the

shortest-path is not sufficient, and a combination of queue management and flow re-routing. Flow-based QoS provisioning process consists of two main steps: i) end-to-end (e2e) path computation and resource allocation to QoS flows at the traffic engineering manager (TEM) of the NSP, which passes the results to the SDN controller, and ii) QoS implementation at SDN switches along the selected path.

A survey of the state-of-the-art traffic engineering methods for SDN can be found in [58]. Egilmez et al. [59] present a framework for sequential optimization of forwarding decisions to enable dynamic QoS over OpenFlow networks and discuss application of this framework to QoS-enabled streaming of scalable encoded videos with two QoS levels. Huang et al. [60] propose joint optimization of rules allocation and traffic engineering in order to use ternary content-addressable memory (TCAM) resources at the switches more efficiently. Egilmez and Tekalp [61] present centralized and distributed architectures and an optimization framework for e2e QoS in multi-operator SDN. SDNHAS is a system for group optimization of resource allocation for HTTP adaptive streaming [62]. Yang et al. [63] model the joint admission and routing problem as a Markov decision process to maximize the revenue.

Assuming path computation and resource allocation per flow has been determined by the NSP-TEM, the controller pushes this information to switches in the form of flow table updates. It is possible to implement QoS at the switches by various mechanisms, including (i) metering (policing) flows at select switches, using functions supported by OpenFlow, (ii) traffic shaping for flow-based rate control [31], (iii) allocating and managing special queues for specific flows. It has been generally concluded that traffic shaping is a better traffic management alternative than policing [64]. However, none of these works explicitly account for real-time dynamic resource allocation for value-added services as proposed in Section III.

3.2.3 Value-Added Services

Although video traffic over networks is increasing significantly over the years, this is not contributing to the revenues of the NSPs since they commonly col-

lect flat monthly fees from users for the best-effort Internet access, which is not directly related to the amount of network resources the users consume. Accordingly, OTT VSPs are becoming overwhelming loads on the NSPs, NSPs do not have an incentive to implement QoS solutions, and light users subsidize heavy users when the total revenue for the consumed network resources are considered.

Recognizing these facts, the concept of open multi-service internetworking that combines the benefits of the basic Internet access service of today and the innovation potentials of future value-added connectivity services was proposed [2]. The main pillars of open multi-service internetworking are:

- Implementation of different service levels (traffic modes) to meet the specific needs of different applications: For example, Basic Quality (BQ, aka. best-effort), which is the current Internet service, Improved Quality (IQ) for premium video streaming services with soft service-level (QoS) guarantees, and Assured Quality (AQ) for premium real-time communications services with hard QoS guarantees.

- Service-level awareness: The users should know the expected performance of the network at a particular location and time for different service levels. Service-level indicators should be available in real-time, to properly inform users about what service-level choices are available to them at what price to better serve their needs.

- Value-added connectivity services (VACS): These are managed quality connectivity offerings from any end-point to any other end-point of the network. Future open VACS can be facilitated by deployment of new AQ traffic exchanges and IQ traffic exchanges in addition to the current best-effort traffic exchanges.

- Business models and charging principles: The charges for a VACS should be proportional with the resources consumed by the service, across the chain of service elements. In general, charging may be coordinated by the policy control and enforcement modules of the edge NSPs that serve as the end-points in a specific VACS.

With the adoption of these pillars in future networks, it will become essential for NSPs to optimize dynamic resource allocation for VACS to increase its revenues driven by new service pricing models, which is the subject of this chapter.

Note that recent changes in the Net Neutrality regulation in the USA and regulation passed by the European Parliament in November 2015 aims to allow NSPs to offer such value-added services other than basic Internet access that are optimized for specific content. On the other hand, basic Internet services and fairness principles must be protected by imposing algorithmic constraints to enforce fair resource allocation.

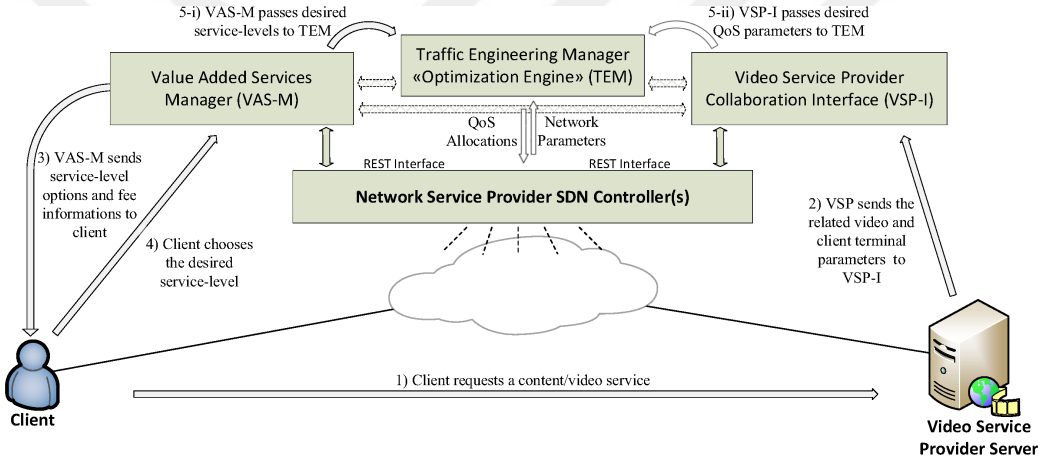

Figure 3.1: Information flow between the user, VSP and NSP for requesting a video service with a user-specified service-level.

3.3 Value-Added Video Services over SDN

Advances in networking technologies make it possible for the NSPs to offer, in collaboration with VSPs, new premium value-added streaming video services, including video-on-demand or live streaming services, driven by user QoE preferences and new pricing models. This section discusses the elements of an NSP-managed premium video streaming service. The service architecture is summarized in Section 3.3.1. Available service levels and an example

scenario for applicable charges are discussed in Section 3.3.2. An example scenario for the interaction between user, NSP and VSP for service request, including determination of a nominal requested bitrate per user, is described in Section 3.3.3. The main contribution of this chapter is dynamic batch-optimization of path and actual bitrate (per client) for a group of clients simultaneously in order to maximize total NSP revenue, which is introduced in Section 3.3.4. Finally, implementation of resource management at the network level is described in Section 3.3.5.

3.3.1 Service Architecture

The main building blocks of the proposed NSP-managed video service architecture are three servers that are connected to the SDN controller at the NSP through its north-bound interface as depicted in Fig. 3.1. They are: i) Video Service Provider Interface (VSP-I) that enables the NSP to communicate with the VSP management server to receive specific user service parameters, ii) Value-Added Services Manager (VAS-M) that allows interaction between the NSP and users so that the NSP provides users available service-level options and pricing information and collects their service-level selection, and iii) Traffic Engineering Manager (TEM) unit that performs dynamic resource allocation and path optimization given the present state of the network, user-selected service-level provided by VAS-M, and user video parameters provided by VSP-I. Hence, a value-added video service is characterized by a user-selected service-level and a VSP-determined requested user bitrate.

3.3.2 Service Levels and Charges

The service levels specify whether the user wishes to pay for hard or soft QoS according to the requirements of its application and his/her budget or wants simply the best-effort service at no extra cost. We define two premium service-levels, namely Assured-Quality (AQ) and Improved-Quality (IQ) service, in addition to the standard best-effort service. The AQ service offers hard QoS with a guaranteed set of QoS parameters subject to admission

control. It is mainly intended for mission critical applications such as tele-surgery or tele-presence at a price. The admission control is administered as an outcome of the resource allocation optimization that is described in Section 3.4. On the other hand, the IQ service offers soft QoS. In IQ services, the NSP-TEM aims to allocate the nominal network resources requested by the VSP to the maximum extent possible; however, no guarantees are provided. The pricing of premium service levels depends on the business model of the NSP. A possible pricing model would be to charge users in proportion to the network resources they use, but not exceeding the amount agreed by them at the time of service initiation. For example, an NSP may apply per-megabyte charges for IQ users, and charge AQ users twice the amount of an IQ service for the same megabyte usage.

3.3.3 User-VSP-NSP Interaction

The information flow between the user, VSP, and various modules of the NSP for service set-up and service-level selection is illustrated in Fig. 3.1. For a premium streaming service, service set-up consists of the following steps: i) User sends a message to the VSP management server to request a particular content, ii) The VSP sends the service parameters, including the nominal video bitrate, as well as client information such as IP address, to the VSP-I of the NSP. Note that no information about the content itself is shared due to confidentiality between the VSP and user, iii) VSP-I passes the user IP address to VAS-M, so that the NSP can contact the user to offer premium service-level options and related pricing information, iv) the user selects the desired service-level according to his/her QoE expectations and the pricing information, v) VAS-M transfers the desired service-level information to the TEM, while the VSP-I passes the video service parameters to the TEM.

The SDN controller passes network state information to the TEM periodically, and triggers optimization engine to perform per-flow dynamic e2e QoS (re-)allocation and path (re-)computation as described in Section 3.4. The computed resource allocation and path decisions are passed back to the SDN controller, which then updates the flowtables at the switches and dynamically reconfigures queues at the switches as described in Section 3.3.5.

In the example interaction scenario above, the user only selects the desired service level and the VSP passes the nominal video bitrate information to the NSP as a default value. For more advanced user interaction, these VSP-selected nominal bitrates can be presented for user approval along with other available bitrate options by the NSP in the service-level selection screen. Users will then have a choice of receiving service at higher or lower bitrates along with pricing information for these alternative bitrates.

The streaming VSP determines the nominal bitrate for each service request according to the display resolution (e.g., 360p, 720p, 1080p) of the user and the genre type (e.g., cartoon, movie, sports) of the requested video. It is well-known that there is a nonlinear relationship between the bitrate and QoE experienced by the user depending on its display resolution often characterized by utility curves [62, 65]. For example, the QoE experienced by a user receiving cartoon content at HD (1080p) resolution at 5 Mbps may be equivalent to QoE experienced by another user receiving the same content over a mobile phone at 360p resolution at less than 1 Mbps. Furthermore, a user receiving a sports video at HD resolution may need more than 8 Mbps to experience the same QoE with another user receiving cartoon content at the same resolution at 5 Mbps. Therefore, the VSP determines the nominal bitrate for each service request according to resolution and genre type of the service request in order to deliver fair QoE to all users. While managed streaming video services are associated with specific bandwidth requirements, managed RTC video services also have low delay requirements.

3.3.4 Dynamic Batch-Optimization of Resource Allocation

Suppose that the network operates at some load level, which represents all AQ, IQ and best-effort services that are already allocated. We assume new service requests arrive and ongoing services expire according to Poisson distributions with different time-varying parameters for the different service levels [66, 67]. If the rates of the incoming service requests are equal to those of that expire, then the network is in steady-state, which is the normal mode

of operation. On the other hand, if the rates of incoming service requests are higher than those that expire, then the network will enter congestion mode.

In this work, we propose to accumulate incoming service requests within independent small time intervals of duration Δt, and perform batch path-computation optimization for these requests simultaneously, which includes sequential processing as a special case with batch size equal to one and Δt goes to zero. For each interval, there are three possibilities: i) There are enough resources to satisfy the new service requests along respective shortest paths (normal-mode): Path computation for each new request can be serviced by the Dijkstra shortest-path (SP) algorithm. ii) There aren't enough resources along respective shortest paths for all requests, but there are enough resources elsewhere in the network (near congestion-mode): An optimization framework is required to find the best alternate route (constrained shortest path) that satisfies the service level and resource requirements for new service requests. iii) There aren't enough resources anywhere in the network to accommodate the new requests (congestion-mode): An optimization framework is required to relocate some IQ service requests to alternate paths and/or reduce their service rates to make room available for all or some of new service requests to the extent possible.

We can now address the question of "Why do we propose a dynamic batch-optimization framework to decide for resource allocation to a group of incoming service requests?" With the introduction of multiple service levels and associated pricing policy, processing a small number of requests simultaneously every Δt seconds helps accommodating service requests that otherwise would be denied in the congestion mode, and proves to be a profitable option for NSPs as an alternative to sequential processing. For example, when the network is in congestion mode, optimization of a group of requests simultaneously may result in accommodating AQ service requests that arrive later than IQ service requests at the expense of providing lesser than the full requested capacity to IQ services. If service requests were processed sequentially, such an AQ service would have been denied because an IQ service that arrived prior to the AQ request would have used the available capacity. Hence, the goal of processing service requests in small groups is to satisfy all

service requests within a small interval Δt to the maximum-possible extent in such a way to maximize the revenue of the NSP. Furthermore, group processing of service requests may be essential in disaster/emergency scenarios in order to accommodate critical service requests.

3.3.5 Resource Reservation at the Network-Level

This section discusses how the resources allocated to individual service requests (as the output of the optimization framework proposed in Section 3.4) are implemented or executed at the network level.

We employ queue management methods for resource reservation at the network level. Configuring multiple queues at switch ports for packet egress provides per-flow rate-limiting ability and enables weighted fairness among flows. Having received per-flow routes and downstream bitrates to be allocated for each premium user from the TEM, the SDN controller of the NSP configures QoS queues at the ports of network edge switches. Queue management is handled by configuration protocols such as OF-Config [45] or OVSDB [46] based on controller/switch compliancy since both parties need to support the same switch configuration protocol. Once queues are implemented, by using OpenFlow queuing action, special traffic flows can be routed over particular queues satisfying the related bandwidth requirements. Queues can be configured in two different ways: (i) static configuration of a fixed number of queues with some maximum bitrates, or (ii) dynamic configuration of queues on-demand and allocating one or more flows in each queue.

In our implementation, we adopt dynamic queue configuration, where queues are configured at variable rates per group-of-flows or per-flow. Each video service request is assigned to a queue at a specific rate (optimally computed by TEM). Deployment of soft virtual switches, such as Open vSwitch (OVS) [47], running over high performance computers with sufficient resources at the network edges allows efficient realization of dynamic queue management. Since each flow is assigned to a flow-specific queue, traffic bursts in one queue do not affect the QoS of other flows as long as the sum

of capacities of all queues over a port stay under the link capacity.

3.4 Dynamic Batch-Optimization of Resource Allocation with Multiple Service-Levels

This section first presents the formulation of the dynamic TE optimization problem with multiple service-levels. We then discuss an approximate solution strategy and a fast heuristic method to solve the problem in real-time. The mathematical notation used in Sections 3.4.1, 3.4.2 and 3.4.3 are summarized in the top, middle and bottom parts of Table 3.1, respectively.

3.4.1 Problem Formulation: The Objective

Since the definition of the objective depends on the service-level parameters, we start by introducing service-level constraints. Next, we define the objective, followed by path constraints and capacity constraints.

Service-Level Constraints: Let IQ and AQ denote sets of users requesting IQ and AQ service levels, respectively. Each user $i \in U$, where U represents the set of premium users, is either a member of IQ or AQ, i.e., U is the union of disjoint sets IQ and AQ. The best-effort users are not included in the formulation of the optimization problem, since their flows are always routed through the shortest path without any QoS specification. Eqn. 3.1 and 3.2 express the IQ service-level constraints for user i, which state that TEM allocates as much bandwidth as possible up to a desired bandwidth with no guarantees, given by

$$b_i \leq bd_i, \qquad \forall i \in IQ \tag{3.1}$$

$$b_i \geq 0, \qquad \forall i \in IQ \tag{3.2}$$

where the variable b_i denotes e2e bitrate allocated to user i and bd_i is a constant representing the requested/nominal bitrate for user i. In the AQ service-level, we define a binary admission control variable, aq_i, since AQ

service level has strict e2e resource reservation requirements and it may not always be possible to satisfy them for all users in AQ. Then, the AQ service-level constraints, Eqns. 3.3 and 3.4, can be defined as

$$b_i \leq aq_i \cdot bd_i^{max}, \quad \forall i \in AQ \tag{3.3}$$

$$b_i \geq aq_i \cdot bd_i^{min}, \quad \forall i \in AQ \tag{3.4}$$

where bd_i^{min} and bd_i^{max} denote the range for the desired/nominal bandwidth for client i. Note that the binary variable, aq_i, in Eqns. 3.3 and 3.4 make AQ service-level request all-or-nothing unlike the IQ service-level. That is, user i either gets the service with at least the minimum desired bandwidth guaranteed, e.g., $aq_i = 1$ and $bd_i^{min} \leq b_i \leq bd_i^{max}$, or its service request is denied, e.g., $aq_i = 0$ and $b_i = 0$.

Objective: The objective of the optimization problem is to maximize the total revenue, TR, of the NSP from the premium service users. We define two parameters p^{IQ} and p_i^{AQ} for premium service fees, where p^{IQ} is the cost per megabit for IQ service users, and p_i^{AQ} is the cost for i^{th} AQ service user depending on the bitrate requested. The revenue from an IQ service user i is proportional to amount of bandwidth allocated to that user and can be written as $b_i \cdot p^{IQ}$. Note that the total amount that user i in IQ agrees to pay is bounded by $bd_i \cdot p^{IQ}$ as TEM can allocate at most bd_i bandwidth due to constraint in Eqn. 3.2. On the other hand, the revenue from an AQ service user i is $aq_i \cdot p_i^{AQ}$, that is, the NSP earns from AQ customers only when the desired QoS is guaranteed for user i, i.e., $aq_i = 1$. Therefore, the total revenue TR of the NSP from premium services is given by

$$TR = \sum_{i \in IQ} b_i p^{IQ} + \sum_{i \in AQ} aq_i p_i^{AQ} \tag{3.5}$$

Path Constraints: Let $G(S, L)$ be a directed weighted multi-graph representing a network topology with full-duplex links, where S is the set of nodes (switches) and L is the set of directed edges (links). Let $l_{ij}^{s_a s_b}$ be a binary variable representing the assignment decision of user i's flow to the j^{th} link outbound from switch s_a and inbound to switch s_b where link id j is between 1 and $N_{s_a s_b}$ (number of direct links between switches s_a and s_b).

The idea is to find a sequence of l_i's generating a path from the source to the destination, which satisfies certain QoS for each user i. For each flow i, Eqn. 3.6 states that there is always an exit from the source switch s^{src_i} and Eqn. 3.7 states that there is always an entry to the destination switch s^{dst_i} as follows:

$$\sum_{s \in S} \sum_{j=1}^{N_{s^{src_i}s}} l_{ij}^{s^{src_i}s} = 1, \quad \forall i \in U \tag{3.6}$$

$$\sum_{s \in S} \sum_{j=1}^{N_{ss^{dst_i}}} l_{ij}^{ss^{dst_i}} = 1, \quad \forall i \in U \tag{3.7}$$

Similarly, Eqn. 3.8 and Eqn. 3.9 state that entry to the source switch s^{src_i} and exit from the destination switch s^{dst_i} are not allowed, respectively, as follows:

$$\sum_{s \in S} \sum_{j=1}^{N_{ss^{src_i}}} l_{ij}^{ss^{src_i}} = 0, \quad \forall i \in U \tag{3.8}$$

$$\sum_{s \in S} \sum_{j=1}^{N_{s^{dst_i}s}} l_{ij}^{s^{dst_i}s} = 0, \quad \forall i \in U \tag{3.9}$$

In order to complete a path from the source s^{src_i} to the destination s^{dst_i} for flow i, whenever a flow enters into any intermediate switch $s^* \in S/s^{src_i}s^{dst_i}$, there must be an exit from that switch. Therefore, flow assignment to intermediate switches for an e2e path is defined by the constraint:

$$\sum_{s \in S} \sum_{j=1}^{N_{s^*s}} l_{ij}^{s^*s} - \sum_{s \in S} \sum_{j=1}^{N_{ss^*}} l_{ij}^{ss^*} = 0, \quad \forall i \in U, \forall s^* \in S/s^{src_i}s^{dst_i} \tag{3.10}$$

In order to avoid loops, we need to constrain the number of visits to a particular switch. Any switch s' can be entered or exited at most once. These constraints are specified as follows:

$$\sum_{s \in S} \sum_{j=1}^{N_{ss'}} l_{ij}^{ss'} \leq 1, \quad \forall i \in U, \forall s' \in S \tag{3.11}$$

$$\sum_{s \in S} \sum_{j=1}^{N_{s's}} l_{ij}^{s's} \leq 1, \quad \forall i \in U, \forall s' \in S \tag{3.12}$$

Furthermore, a flow cannot go from one switch to another and come back to the same switch for any switch pair s_a and s_b.

$$\sum_{j=1}^{N_{s_a s_b}} l_{ij}^{s_a s_b} + \sum_{j=1}^{N_{s_b s_a}} l_{ij}^{s_b s_a} \leq 1, \quad \forall i \in U, \forall s_a, s_b \in S \tag{3.13}$$

$$\sum_{s_a \in S'} \sum_{s_b \in S'} \sum_{j=1}^{N_{s_a s_b}} l_{ij}^{s_a s_b} \leq \left|S'\right| - 1, \quad \forall i \in U, S' \subsetneq S \tag{3.14}$$

Note that Eqns. 3.11, 3.12 and 3.13 avoid loops along the e2e path, while Eqn.3.14 eliminates free loops that are not connected to the desired e2e path.

Capacity Constraints: Total allocated bitrate for all flows on a link exiting from switch s_1 and entering to switch s_2 must be less than the capacity of that link. We first define the capacity constraint as

$$\sum_{i \in U} l_{ij}^{s_1 s_2} b_i \leq C_j^{s_1 s_2}, \quad \forall s_1, s_2 \in S, \forall j \in [1, N_{s_1 s_2}] \tag{3.15}$$

where $l_{ij}^{s_1 s_2}$ is 1 or 0 indicating the decision of putting flow i on the link or not. Then, $l_{ij}^{s_1 s_2} b_i$ is the bitrate allocated for user i on the link j outbound from s_1 and inbound to s_2.

Observe that the formulation becomes a mixed integer non-linear programming (MINLP) problem due to the capacity constraint in Eqn.3.15. The formulation can be transformed into mixed-integer linear programming (MIP), which speeds up the solution considerably, by introducing slack variables $a_{ij}^{s_1 s_2}$, where $a_{ij}^{s_1 s_2} = l_{ij}^{s_1 s_2} b_i$. Then, the capacity constraint in Eqn.3.15 can be re-expressed as a number of linear constraints as follows:

$$\sum_{i \in U} a_{ij}^{s_1 s_2} \leq C_j^{s_1 s_2}, \quad \forall s_1, s_2 \in S, \forall j \in [1, N_{s_1 s_2}] \tag{3.16}$$

$$a_{ij}^{s_1 s_2} \geq 0, \quad \forall i \in U, \forall s_1, s_2 \in S, \forall j \in [1, N_{s_1 s_2}] \tag{3.17}$$

$$a_{ij}^{s_1 s_2} \leq b_i, \quad \forall i \in U, \forall s_1, s_2 \in S, \forall j \in [1, N_{s_1 s_2}] \tag{3.18}$$

$$a_{ij}^{s_1 s_2} \leq l_{ij}^{s_1 s_2} bd_i, \quad \forall i \in U, \forall s_1, s_2 \in S, \forall j \in [1, N_{s_1 s_2}] \tag{3.19}$$

$$a_{ij}^{s_1 s_2} \geq b_i - bd_i(1 - l_{ij}^{s_1 s_2}), \quad \forall i \in IQ, \forall s_1, s_2 \in S, \forall j \in [1, N_{s_1 s_2}] \tag{3.20}$$

$$a_{ij}^{s_1 s_2} \geq b_i - bd_i^{max}(1 - l_{ij}^{s_1 s_2}), \quad \forall i \in AQ, s_1, s_2 \in S, j \in [1, N_{s_1 s_2}] \tag{3.21}$$

To summarize, we formulate a MIP problem with the objective of maximizing the NSP revenue TR in Eqn. (3.5) under the service-level constraints Eqns. (3.1-3.4), path constraints Eqns. (3.6-3.13), and linearized capacity constraints Eqns. (3.16-3.21). Eventually, for each service request i, the solution for the optimization problem provides i) a sequence of links $l_{ij}^{s_a s_b}$ forming an e2e path from the source to destination for each flow i, (ii) a bitrate b_i over this path, and iii) in the case AQ service, a decision, aq_i, indicating whether the requested service can be provided or not (admission control).

The solution of the problem when optimizing a group of service requests simultaneously over the whole network is NP-complete. There are two parameters that affect the run-time: (i) the size of the network, and (ii) number of service requests. Dependence of the run-time on these parameters over a partially-connected network with random mesh topology (averaged over 10 independent runs) and 70 Mbps bi-directional links is illustrated in Fig. 3.2, where the average switch connectivity degree is 4 and service-level choices and corresponding traffic classes are generated randomly. It can be seen that the run-time for processing more than 10 service requests simultaneously over large networks can be long. Hence, in Section 3.4.2 we resort to an approximate method for solving the optimization problem over large networks in a reasonable amount of time. Then, in Section 3.4.3 we propose a fast heuristic method that runs in near real-time and performs very close to the approximate optimal solution.

3.4.2 Approximate Group Optimization (AGOP) Method

In this section, we present a method to compute the optimal solution approximately in a reasonable amount of time, which we intend to use as a benchmark for the fast heuristic solution that we propose in Section 3.4.3. The approximation is based on the divide-and-conquer strategy, where we partition the network topology into smaller sub-networks. The problem with large number of service requests is handled by processing groups of service requests in small time intervals Δt.

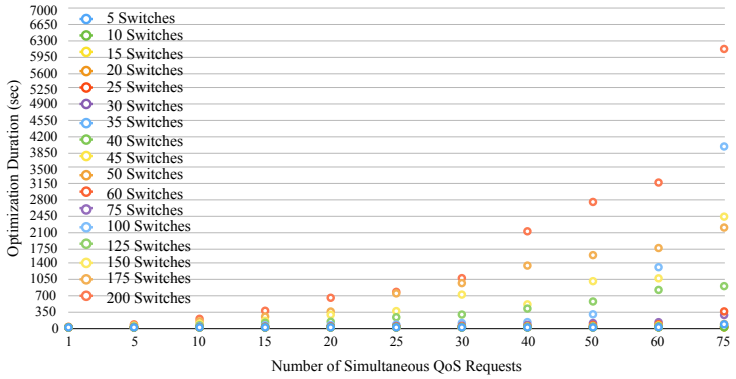

Figure 3.2: Optimization run-time for different topology sizes and service request numbers.

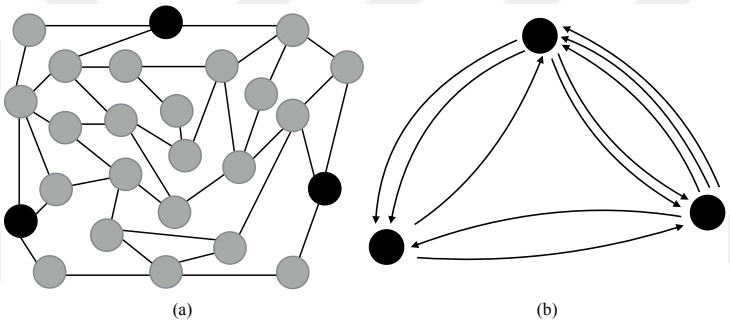

Figure 3.3: Domain topology aggregation: (a) Actual domain topology with directed links, (b) Aggregated representation with multiple directed virtual links among border gateway switches (denoted by black nodes).

An operator's network topology may be partitioned into a certain number of smaller domains by considering geographical locations as in the transit-stub network model [68]. We assume that the segmentation of the network graph into local domains is already known. In the proposed approach, optimization of resources for each domain is performed independently in parallel. The actual topology of each domain is represented by an aggregated graph with only the border gateways (switches) and multiple directed virtual links connecting these border switches, where each virtual link represents a different physical route between the border gateways as depicted in Fig. 3.3. The aggregation of each domain is performed using the Algorithm 5.

Let $G(S, L)$ be the directed weighted graph representation of the actual

topology of a particular domain where S and L represent sets of switches and links, respectively. Let S^* be the subset of S including only the border switches and BP be the set consisting of all pairs of border switches in S^*. Algorithm 5 is iterative and runs as long as there exists unassigned resources along a directed path p_n connecting border pair $bp_n \in BP$ (Line 3). For each border pair bp_n (Line 4), we find the current shortest path p_n^* for the current border switch pair bp_n (Line 5). If no path is found, we remove bp_n from BP (Line 14). If p_n^* exists, we compute the end-to-end available bandwidth b_n, which is the minimum available bandwidth of all links l along p_n^* (Line 7) and create a directed virtual link with bandwidth b_n connecting the switches bp_n and add it to bp_n's virtual bandwidth set in B_V (Line 8). Then, b_n value is deducted from the available bandwidths b_l of links l along the path (Line 10) and remove any link l if it has no more available bandwidth (Line 12). Once all aggregated domain representations are computed, we combine them using inter-domain links across border switches of different domains.

Algorithm 5: Domain Aggregation Algorithm

```
1  Input: G and BP
2  Output: B_V
3  while any p_n in G exists for any bp_n ∈ BP do
4      foreach border pair bp_n ∈ BP do
5          Find shortest path p_n^* in G;
6          if p_n^* exists then
7              Compute the b_n of p_n^*;
8              Add b_n to bp_n's virtual bandwidth set in B_V;
9              foreach link l along the p_n^* do
10                 b_l = b_l − b_n;
11                 if b_l = 0 then
12                     Remove l from G;
13         else
14             Remove bp_n from BP;
```

The proposed solution strategy for optimization over the partitioned network is as follows: i) Assign source and destination switches for each service

request to a border switch which has the highest capacity path between itself and the actual source or destination switch. ii) Having assigned source and destination switches to border switches for every service request, an optimization is first performed using the formulation in Section 3.4.1 over the whole network consisting of multiple domains with aggregated representations. As a result of this step, entry and exit border switches for each service flow at each domain along the respective e2e inter-domain paths are determined. iii) Then, because there exists single links from one switch to another in the actual domain topologies, $N_{s_1 s_2}$ is set as 1 in the formulation for all pair of switches (s_1, s_2); and each domain performs optimization over its actual topology simultaneously, given the source and destination switches for each flow passing through. iv) Once path and bandwidth allocations for each flow in each domain are calculated, the final physical e2e routes are obtained by concatenation of the individual paths provided by respective domains and e2e bandwidth for each service flow is determined as the minimum of bandwidths computed by domains along the route of that flow.

Another factor affecting the run-time of optimization is the number of service requests being optimized simultaneously. The full optimization framework considers the route and bandwidth allocations for new service requests together with those of all flows that already exist in the network and try to satisfy requirements of all flows from all service-levels. In order to reduce the number of service requests to be processed, we can keep the routes and allocations of existing AQ flows fixed, as they have strict requirements, and solve the optimization problem for all new service requests together with only a small group of existing IQ service flows.

In order to analyze the run-time and performance gap with the true optimal solution, we perform experiments over a network consisting of 25 switches and 10 simultaneous service requests. When the network resources are sufficient to fulfill all desired bandwidth requests, AGOP achieves 100% performance compared to true optimal solution and its runtime is 1 sec while the true optimal solution runs in 4 sec. On the other hand, if the network is in congested mode and all service requests cannot be fulfilled, total revenue obtained by AGOP is on the average 85% of the true optimal solution. How-

ever, its run-time is 2-5 sec. while the run-time for the true optimal solution is about 3 mins.

3.4.3 Group Constrained-Shortest-Path (GCSP) Method

The run-time of the approximate solution discussed in the previous section ranges from several seconds up to a minute for different set-ups. In this section, we propose a fast heuristic solution, which typically takes less than 100 msec. to optimize a group of service requests. The shortest-path (SP) algorithm for path computation is well-known and commonly used in the current Internet. We propose an extension of the SP algorithm called the group constrained-shortest-path (GCSP) algorithm. A special case of the GCSP algorithm, for group size equal to one, is called the constrained shortest path (CSP) algorithm.

The GCSP method performs group based processing of service requests as presented in Algorithm 6. A group R of service requests is formed by accumulating new service requests R^{in} within a time duration Δt and optionally including some of the existing IQ services, R^{ex} (Line 7). The inputs to the algorithm are R^{in}, R^{ex}, the set of switches S, and the set of links L, which represents all links in the network along with their available bandwidths. If some of the existing IQ services are being re-processed, e.g., R^{ex} is non-empty, the bandwidth for particular links in L corresponding to the links along the respective e2e paths for services in R^{ex} are incremented by the amounts that were allocated to those services (Line 3-6), i.e., the bandwidth for services in R^{ex} is reset for the purposes of optimization.

Once the final group R is formed, requests are sorted first according to the service-levels and then the desired bandwidths within the each service-level (Line 8). In order to accommodate AQ service requests to the maximum possible extent and maximize total revenue, AQ requests are processed prior to IQ requests. In addition, service requests requiring smaller bandwidth are also listed prior to those with higher bandwidth requirements. The idea is to increase the number of AQ requests admitted as well as increase the utility

Algorithm 6: Group Constrained Shortest Path Algorithm

Input: R^{in}, R^{ex}, S, L
Output: P, B
foreach *service request* $r_x \in R^{ex}$ **do**
    foreach *link* l *along* p_x **do**
        $b_l = b_l + b_x^{alloc}$;
        Update l in L;
$R = R^{in} \cup R^{ex}$;
Sort service requests in R;
foreach *service request* $r_i \in R$ **do**
    $b_i^{alloc} = b_i^{req}$;
    if *requested service-level is IQ* **then**
        Generate R^*;
        $b_{R^*}^{tot} = \sum_{k \in R^*} b_k^{req}$;
        $c^* = \sum_{l \in L^*} b_l$;
        $w^* = min(1, c^*/b_{R^*}^{tot})$;
        $b_i^{alloc} = b_i^{alloc} \cdot w^*$;
    do
        Create graph G using links in L and switches S;
        foreach *link* l *in* G **do**
            if $b_l < b_i^{alloc}$ **then**
                Remove l from G;
        Find shortest path p_i^* in G;
        if p_i^* *exists* **then**
            Add p_i^* to P for r_i;
            Add b_i^{alloc} to B for r_i;
            foreach *link* l *along the* p_i^* *in* L **do**
                $b_l = b_l - b_i^{alloc}$;
        else
            if *requested service-level is AQ* **then**
                $b_i^{alloc} = 0$;
                Add b_i^{alloc} to B for r_i;
                break;
            if *requested service-level is IQ* **then**
                $b_i^{alloc} = max(0, b_i^{alloc} - \delta)$;
    while p_i^* *does not exist*;

of IQ requests. For each of the ordered service requests r_i in R, TEM aims to reserve a bandwidth b_i^{alloc} which can be at most the requested bandwidth b_i^{req} (Line 10). In order to increase the fairness among the IQ requests in the group being processed, TEM scales the bandwidths to be allocated with a certain scaling parameter w^* for particular IQ requests (Line 11-16) so that processing an IQ request prior to others does not adversely affect the utility of the next IQ requests to be processed. To calculate the scaling factor w^*, TEM first forms a sub-group R^* including current request being processed along with some other particular IQ requests for which the data traffic originates from the same switch (Line 12). The total amount of the desired bandwidths $b_{R^*}^{tot}$ for these requests in R^* is calculated (Line 13) along with the remaining capacity c^* of the outbound links from the associated source switch (Line 14). Then, w^* is calculated as the ratio of c^* and $b_{R^*}^{tot}$. If there is no congestion, w^* is set to 1 for these requests in R^* keeping the initial requested bandwidth values (Line 15-16). Note that due to ordering in R, bandwidth scaling only takes effect after all AQ requests are processed.

Once bandwidth to be allocated b_i^{alloc} is initialized for the current request being processed, TEM first creates a graph G using the switches S and the links in L, then removes certain links with less remaining bandwidth than b_i^{alloc} (Line 18-21). Using the current graph G, shortest path p_i^* is computed, and b_i^{alloc} and p_i^* are stored for the current request r_i in the output maps for allocated bandwidths and calculated e2e paths maps, B and P, respectively (Line 22-25). Then, capacities of the links in L along the path p_i^* are updated by extracting the b_i^{alloc} from the currently available bandwidths (Line 26-27). Since particular links were removed from G (Line 21), it is possible that path p_i^* does not exist. In such a case, if the requested service-level of r_i is AQ, then TEM is not able to provide the requested bandwidth and no resource allocation is performed denying the request (Line 29-32). On the other hand, if the service-level is IQ, TEM tries to allocate a lower bandwidth so that it updates b_i^{alloc} where δ is the bandwidth decrease step size (e.g., 0.5 Mbps) (Line 34). Having an updated bandwidth value to be allocated for r_i and links with recent capacities in L, a new graph G is created and the same procedure is repeated until a path p_i^* is found (Line 35).

Considering the computational complexity, for each request in R, the worst-case complexity of forming R^* is $O(R)$ (Line 11-16). The complexity of creating graph G and finding a single shortest path for the request is $O(|S+L|+|L|log|S|)$ with binary heap implementation (Line 18-22). In the worst-case, for a request with desired bandwidth b^{req}, a path is found in b^{req}/δ iterations (Line 17-35). Therefore, the worst-case complexity of finding paths and bandwidths to be allocated for all requests is $O(R^2+Rb^{req}/\delta+|S+L|+|L|log|S|)$. We also note that although the algorithm can be improved with binary search, we observe that resource allocations in our problem are generally done in small number of iterations (Line 17-35).

We note that the CSP algorithm is a special case of the GCSP when R^{in} contains a single request and R^{ex} is empty, where incoming service requests are processed individually as soon as a request arrives, i.e., Δt approaches zero.

Table 3.1: Mathematical Notations

Symbol	Definition
AQ	the set of assured quality service users
IQ	the set of improved quality service users
U	the set of all premium service users
bd_i	the requested/nominal bitrate for user i
b_i	the e2e bandwidth allocated to user i
aq_i	the AQ service level admission control variable
p^{IQ}	the per megabit cost for IQ service level
p_i^{AQ}	the bitrate specific cost for i^{th} AQ service user
TR	the total revenue from users in AQ and IQ
G	the directed weighted multi-graph topology
S, s_a	the set of switches, a switch element
$s^{src_i}, s^{dst_i}, s^*$	the source, destination and any intermediate switch along the path of flow i
L	the set of directed links
$l_{ij}^{s_a s_b}$	the link assignment decision for user i's flow
$N_{s_a s_b}$	the number of directed links between s_a and s_b
$C_j^{s_1 s_2}$	the capacity for directed link j between s_1 and s_2

BP	the set of border switch pairs
bp_n	a border switch pair
p_n	a directed path connecting switches in pair bp_n
p_n^*	the shortest path between switches in pair bp_n
b_n	a virtual link bandwidth connecting switches bp_n
B_V	the set of virtual bandwidths between switches bp_n

R^{in}, R^{ex}	the sets if incoming and existing requests
R, r_i	the union of R^{in} and R^{ex}, a request element
R^*	a sub-group of R for which the data traffic originates from the same switch
$b_{R^*}^{tot}$	the total amount of desired bandwidths in R^*
w^*	the bandwidth scaling parameter
P	the set of computed paths for requests
p_i^*	the computed path for request i
B	the set of computed bandwidths for requests
b_i^{alloc}	the allocated bandwidth for request i

3.5 Evaluation

We evaluate the performance of the proposed GCSP method by comparing it with those of shortest path (SP), CSP [54] and AGOP. The AGOP method is used as a benchmark to compare the total NSP revenue obtained by using the GCSP since AGOP is designed to maximize the NSP revenue.

3.5.1 Experimental Setup

The experiments are performed over a network where we adopt the transit-stub (domain) topology model generated in the same way as in the GT-ITM [69]. We simulated a network with 210 switches based on the GT-ITM model, where the average number of connections for each switch is approximately 3, creating many possible paths from source to destination nodes. We configure the link speeds as 100 Mbps and create 70% initial traffic load. A smaller partially connected mesh topology with 20 OVS switches is also realized by using Mininet [70] to validate our service architecture in an emulation environment.

We use Floodlight SDN controller [49] which we have extended with i) a resource monitoring module that periodically calculates the traffic level and delay in each link, and ii) a queue management module that enables dynamic queue creation using OVSDB similar to [71].

We implemented the NSP units TEM, VAS-M and VSP-I as web services running over independent servers. They employ custom JSON-based APIs enabling communication between each other and data plane entities, e.g., clients and VSP management servers. TEM exposes different web services to the controller for each of the proposed methods. In the optimization engine implementation, TEM uses GAMS Java API [72] so that for each resource allocation cycle it i) dynamically creates the optimization task using the model described in Sec. 3.4.1 along with the parameters for the network state and service requests, ii) runs the process using the CPLEX solver, and iii) collects the outputs from the GAMS database. For all methods, TEM informs the SDN controller with the resource allocation decisions and VAS-M about the service admission decisions. We note that TEM unit runs over a

high-performance server with 8 cores at 2.4 GHz and 64 GB of RAM.

The nominal video bitrates used in our experiments are 15-18 Mbps for UHD, 3-7 Mbps for HD, and 1-3 Mbps for SD resolutions, respectively, as suggested by YouTube.

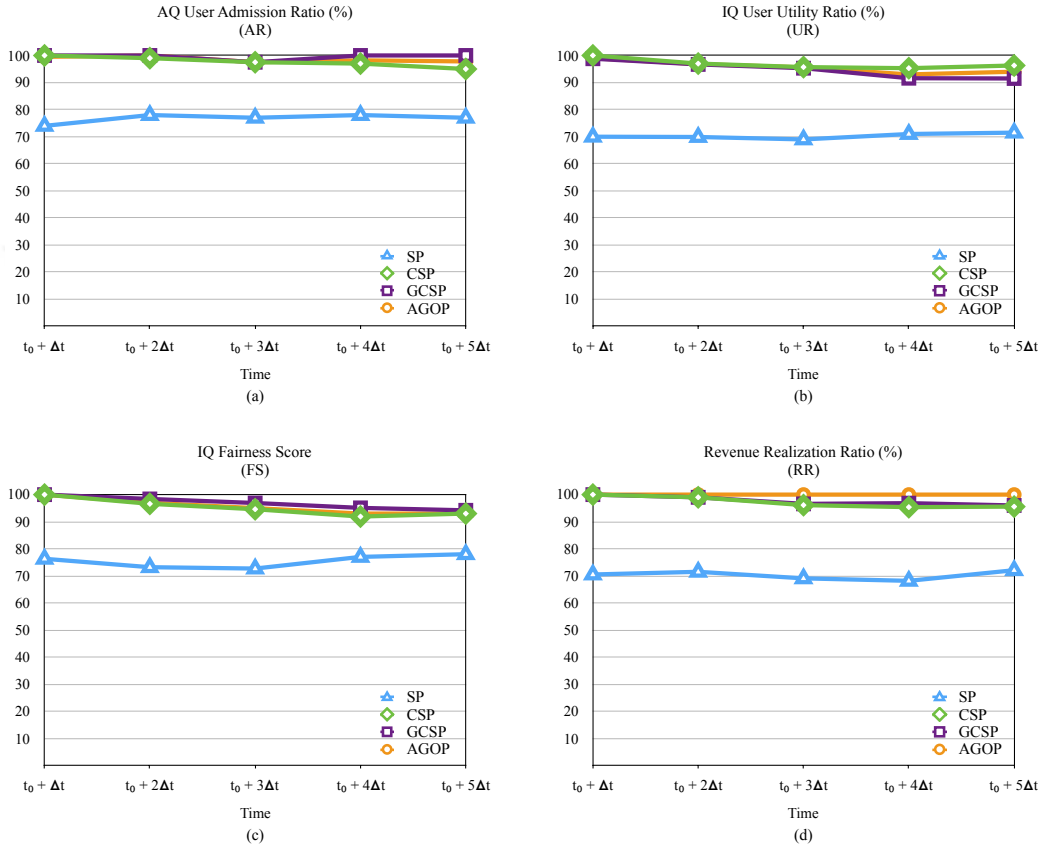

Figure 3.4: Performance results in normal mode of operation, where the rates for the new and expiring service requests are equal: (a) AQ user admission ratio, (b) IQ user utility ratio, (c) IQ fairness score, (d) revenue realization ratio.

3.5.2 Results

In order to implement the AGOP method, the network is geographically partitioned into 5 domains of sizes ranging from 40 to 50 switches [69]. Each domain has up to 4 border switches in total for peering with other neighboring

domains. The per-megabit bandwidth cost for IQ (p_{IQ}) and AQ (p_{AQ}) service levels are set to 1 and 1.8 units, respectively. The numbers of new IQ and AQ service requests that arrive in independent time intervals Δt are given by Poisson distributions with means Λ_{IQ}^{in} and Λ_{AQ}^{in}, respectively. Our use of Poisson arrival process to model temporal characterization of video traffic flows is based on a study by Mori et al. [67] which shows that the flow arrival process of Youtube can be well modelled by the Poisson arrival process. Considering the number of IQ service requests is more likely to be higher than the number of AQ requests, we choose Λ_{IQ}^{in} as $c \cdot p_{AQ}$ and Λ_{AQ}^{in} as $c \cdot p_{IQ}$, where c indicates the volume of the service requests and is set to 10. Moreover, we select the proportion of client device types and video resolution types requested by clients according to the numbers projected for 2021 in the Cisco report [1]. The estimated distribution of the requested video resolutions are 41%, 47%, and 12% for SD, HD, and UHD, respectively, creating different bandwidth requirements for the service requests.

We quantify the performance of the proposed methods using the following metrics:

- AR defines the admission ratio which is calculated as the ratio of the number of AQ requests admitted and total number of AQ requests.

- UR denotes the average utility ratio for IQ requests, which is the average of utility of individual IQ requests (ur_i) defined by the ratio of allocated bandwidth to the requested bandwidth for request i.

- FS is the fairness score among IQ requests. Deviation between ur_i's is a reasonable metric representing how fair the network resources are allocated to different requests. FS is defined by $100 \cdot (1 - \sigma_{ur})$, where σ_{ur} is the standard deviation of ur_i's for all IQ requests. If the score is high, then IQ requests are served with similar utility ratios.

- RR is the revenue realization ratio, which is defined as the ratio of the revenues collected by a particular method to the revenue achieved by the benchmark AGOP method.

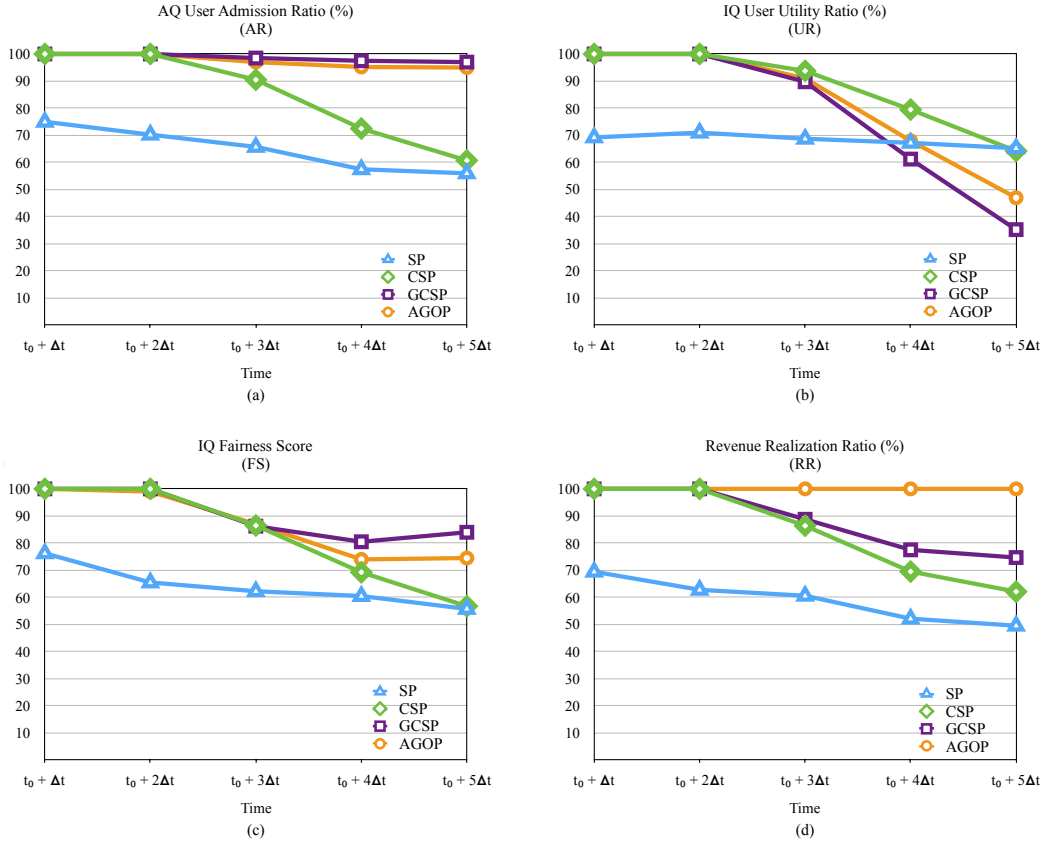

Figure 3.5: Performance results in congested mode of operation, where the rates of new service requests are higher than those of expiring services: (a) AQ user admission ratio, (b) IQ user utility ratio, (c) IQ fairness score, (d) revenue realization ratio.

Figures 3.4 and 3.5 show the performance of SP, CSP, GCSP and AGOP methods using the mean values of above metrics computed over 10 repetitions of each experiment. Each experiment is performed periodically at every Δt sec. starting at a reference time t_0. Cumulative performance values are computed at five consecutive Δt instants between t_0 and $t_0 + 5\Delta t$ for each of the metrics. Note that during each Δt interval, service requests arrive and expire. In the SP and CSP methods, resource allocations are performed immediately as a service request arrives, while the group processing procedures are executed after waiting for Δt=100 ms for GCSP. We note that AGOP requires processing time on the order of seconds approximately (10-20

seconds) but the same requests are used as inputs to different methods for fair comparison. We consider two scenarios, the normal and congested mode of operation.

In the normal mode of operation, the network has sufficient capacity to serve incoming premium service requests at a given time instant, and the video traffic is stable with the rate of arriving and expiring service requests equal, i.e., $\Lambda_{IQ}^{in} + \Lambda_{AQ}^{in} = \Lambda_{IQ}^{out} + \Lambda_{AQ}^{out}$. In the normal mode of operation, the outcomes in Figure 3.4 for each of the methods are stable at different time instants since the rate of new service requests and the rate of service expires are equal. We observe that there is no major performance differences when CSP, GCSP, and AGOP are considered for all metrics. Such performance similarity is mainly attributed to the total network resources that are sufficient to handle all service requests at a given time instant. On the other hand, the performance of SP method is much lower compared to other approaches for all metrics since it cannot utilize the all network resources and the resources along the shortest path routes may not be sufficient to serve all premium requests. Overall, there exist some time instants where the UR for CSP is 3% on the average higher than GCSP and AGOP, while it has 4% less AR performance on the average decreasing the the overall revenue RR negligibly.

In the congested mode of operation, the network experiences burst of requests where the rates of new service requests are higher than those of expiring services, i.e., $\Lambda_{IQ}^{in} + \Lambda_{AQ}^{in} > \Lambda_{IQ}^{out} + \Lambda_{AQ}^{out}$, due to an extreme condition such as start of live streaming of an important event or real time communications with people in a location of common interest. We set Λ_{IQ}^{out} to $0.25\Lambda_{IQ}^{in}$ and Λ_{AQ}^{out} to $0.25\Lambda_{AQ}^{in}$. Graphs in Figure 3.5 present the performance results for the congested mode of operation scenario. We observe that for all metrics (except AGOP's RR since it is used as benchmark and considered as the maximum achievable revenue) there is a major performance decrease between time t_0 and $t_0 + 5\Delta t$ for all methods except SP. Since SP cannot utilize the network resources along the alternative paths for the current service requests, there may still exist resources that are not yet utilized and can be allocated for the next incoming requests whose shortest-path routes may vary depend-

ing on the location of the service requesting clients on the network. In this manner, SP's performance can be considered as stable compared to others, however it's AR, UR and FS performances will obviously further decrease some time after $t_0 + 5\Delta t$ as all of the resources on all links gets allocated.

Both AR and UR directly affect RR since higher AR and UR increase the revenue. We observe that RR starts decreasing considerably after time $t_0 + 2\Delta t$ as shown in Figure 3.5-d for CSP and GCSP along with AR (in CSP only) and UR. Based on the experiments, the network utilization, which is calculated as the ratio of the total traffic on the links outgoing from the switch that the VSP is connected to and sum of total capacities of these links, at time $t_0 + 2\Delta t$ corresponds to 94.9% on the average. Furthermore, the load factor for the service requests, which is calculated as the ratio of the total bandwidth required for the incoming requests (minus the bandwidths being released due to the expiring services) and sum of the total link capacities outgoing from the VSP switch, between time $t_0 + 2\Delta t$ and $t_0 + 3\Delta t$ is 8.5% on the average. In such circumstances, NSP cannot fulfill the service requirements over a network with 94.9% utilization. Therefore, AR and UR tend to decrease after $t_0 + 2\Delta t$. Considering the AR performances in Figure 3.5-a, group processing based methods GCSP and AGOP are superior compared to others. At time $t_0 + 5\Delta t$, GCSP achieves 37% higher AR performance compared to CSP. However, as more requests arrive, it is natural that AR starts slightly decreasing even with these methods especially after time $t_0 + 4\Delta t$.

When the network load exceeds a certain point, group processing methods start sacrificing some IQ users and allocate them less bandwidth compared to originally requested ones in order to accommodate more AQ users with higher service fees. As Figure 3.5-b presents, CSP has 29% and 18% higher UR performances at time $t_0 + 5\Delta t$ compared to GCSP and AGOP, respectively, which in turn results in lower AR. Note that the fastest decay in UR performance occurs when GCSP is used. Since AGOP maximizes the revenue and does not explicitly favor AQ users, there may be multiple possibilities of AQ admission and IQ utilization combinations depending on the network and service request states for the same revenue. Therefore, with AGOP method, higher UR and slightly lower AR is achieved compared to

GCSP. Revenue realization RR is above 75% with GCSP. Compared to CSP, up to 13% higher performance is achieved with GCSP method.

Considering fairness among IQ users, FS for CSP, GCSP, and AGOP decrease at a similar rate between time $t_0 + \Delta t$ and $t_0 + 3\Delta t$. After time $t_0 + 3\Delta t$ GCSP and AGOP show stable FS performance, while FS continues to decrease for the CSP method, and the performance gap with GCSP and CSP reaches 29% at time $t_0 + 5\Delta t$. When the network is in congested mode, part of the incoming IQ requests may receive resource allocations with low utility since there is no possibility of re-considering existing services in the sequential methods.

3.6 Conclusions

The main novelty of the work presented in this chapter is a practical method for batch-optimization of resource allocation for a group of premium service requests with the objective of maximizing the NSP revenue in future NSP-managed value-added video services over SDN, making near-real time e2e service provisioning with multiple service levels practical. The state-of-the art in resource allocation is first-come first-serve service provisioning, which we demonstrate is not optimal, in the sense of total NSP revenue, especially in congested mode of operation. We first provide the formulation of the optimization problem for NSP revenue maximization subject to service-level and network constraints, and observe that group processing makes the problem NP-complete. Then, we propose the GCSP method, which is a fast solution achieving on the average 90% revenue realization compared to the approximately optimal solution, while providing fairness among all users with the same service-level. The proposed NSP-managed service architecture and GCSP method are beneficial for all parties involved in the delivery of OTT video services, i.e., the NSP, video service providers (VSP), and users can all benefit from new business models enabled by the next generation technologies.

Chapter 4

Value-Added Video Services over Multi Operator SDN

4.1 Introduction

It is well-known that networked multimedia services need specific throughput and delay requirements in order to provide a sufficient level of quality of experience (QoE) to users. However, the current Internet supports only the best effort service. Prevalent practices to address network performance issues are: i) network service providers (NSP), i.e., operators, resort to over-provisioning, and ii) video service providers (VSP), i.e., over-the-top content providers, rely on content distribution networks (CDN). Both solutions are inefficient and costly. Furthermore, the fees NSPs charge their customers have little correspondence with how much actual resources are used. As a result, content providers and end-users have no incentive to use bandwidth efficiently, which aggravates network congestion. To this effect, NSPs are starting to introduce premium multimedia services with different service-levels as value-added services driven by end-user preferences and choice, in addition to the current best-effort services.

An NSP has full control of traffic within its domain and can manage/allocate resources to offer quality of service (QoS) within its domain. Over the years, there have been many traffic engineering (TE) proposals to improve network

performance and provide QoS over a single operator network, including integrated services (IntServ), differentiated services (DiffServ), multi-label packet switching (MPLS) [3], application layer traffic optimization (ALTO), and path computation element (PCE) [5]. However, these services are not widely deployed, except for closed networks, due to scalability concerns. Recently, software-defined networking (SDN) has emerged as a promising alternative to realize service-aware routing, flow-level quality assurance and efficient dynamic resource management, since an SDN controller can have visibility of all network resources within a domain. Besides network programmability, SDN enables an abstraction of the physical network, such that one physical network can support a variety of services over customized network slices. Recently, several SDN-enabled video services with QoS over a single-operator network have been proposed [7], [8], [9], [10], [11].

It is likely that in a streaming video service the VSP and the client or in a video-conference service two or more peers reside in different NSP networks. This necessitates a cooperation between multiple NSPs to offer a video service with E2E QoS. Today, different NSPs and CDNs typically connect with each other by means of Internet exchange points (IXP), which provide simple connectivity based on topology maps build by the border gateway protocol (BGP). Existing peering and transit interconnection agreements between operators do not provide service-aware routing and management or QoS assurance, and pertain to inter-domain traffic aggregates of multiple elastic and inelastic services. In order to negotiate service level agreements (SLA) between different network domains, bandwidth broker [73] and other architectures [74] have been proposed. SDN-based solutions to Internet exchange points are discussed in [75], [76], [77] [78] and [79]. We elaborate on the details of these solutions that are based on a centralized control architecture in Section 4.2. Yet, dynamic network reconfiguration and service provisioning across multi-operator networks in order to provide e2e managed-quality services per-user or per-flow remains as significant challenges.

Unlike the state of the art centralized frameworks that require a physical exchange infrastructure, this chapter proposes a distributed framework to enable dynamic E2E inter-NSP path optimization and real-time E2E service-

aware flow management in multi-operator SDN. Wide area SDN is now a reality, and we foresee multi-domain wide area SDN operated by multiple NSPs, each managing their own SDN domain. Hence, we propose a distributed framework for cooperation between SDN controller of multiple NSPs, where each controller realizes the functionality of a "virtual" exchange in addition to managing the resources and traffic within its network domain. From a service provider perspective, the proposed framework enables each NSP to offer inter-NSP services with QoS as if the entire network is a single network by directly dealing with each other without relying on a third-party exchange authority, while each NSP retains full control of its own network. From a content-provider or end-user perspective, the proposed framework makes E2E services with different service-levels possible, where source and destination of a service may reside in different domains possibly managed by different operators.

The rest of this chapter is organized as follows: We discuss related works in Section 4.2. We introduce the SDN-enabled distributed "virtual" open exchange architecture in Section 4.3. Section 4.4 provides details of the proposed distributed procedures for multi-domain video service provisioning with E2E QoS. Section 4.5 describes our test environment and presents experimental emulation and simulation results. Conclusions are provided in Section 4.6.

4.2 Related Works and Novelty

This section reviews the state of the art in inter-operator service provisioning with E2E QoS and states the novelty of our work in this chapter. Classical multi-operator services over the Internet are supported as BGP connectivity for single type of traffic based on interconnection agreements between operators, which do not provide service-aware routing or E2E QoS. Scientific problems and economic models to enable inter-operator services with service-aware QoS guarantees have been studied in the context of future Internet [74] and 5G services [79].

An early proposal for economical models and architectures to automatic-

ally establish inter-domain services with E2E QoS was presented by Pouyllau *et al.* [80]. Provisioning inter-operator services with E2E QoS requires some level of cooperation between different NSPs. This cooperation can be based on forming federations among NSPs (similar to airline alliances) or on open market (price competition) models [81].

The problems of inter-NSP path optimization and E2E SLA negotiation are related. Inter-NSP path computation over a predetermined sequence of NSP domains has been addressed by Djarallah *et al.* [82]. Computation of E2E paths over multiple inter-operator routes to satisfy requested QoS constraints more efficiently is discussed in [83]. Groleat *et al.* [84] propose distributed algorithms to manage inter-NSP SLA negotiations based on reinforcement learning techniques. Matos *et al.* [85] presents a model for QoS for inter-domain services that allows inter-domain QoS-aware services to be defined, configured, and adapted in a dynamic and on-demand fashion, among service providers. Rakas *et al.* [86] propose a policy-based model that relies on a centralized approach via a third party agent for E2E service negotiation among multiple domains. Blocq *et al.* [87] pose inter-domain routing as a cooperative game theory problem and propose price of heterogeneity. However, the plethora of protocols to be supported and computation needed at each router to enable E2E QoS provisioning would make current Internet routers too heavy and expensive.

Recently, SDN-enabled centralized exchange architectures that leverage on the capabilities of SDN have been proposed. The software-defined Internet exchange point (SDX) is a centralized infrastructure that enables participating NSPs to write their own dynamic policies and support application specific peering [75], [76]. Control exchange points (CXP) are centralized exchanges, where the participating NSPs advertise partial paths (called pathlets) within their domain with specific properties, and the CXP dynamically stitches them together to orchestra E2E QoS paths [88], [77] [78]. 5GEx is a proposal to enable 5G verticals by designing an exchange framework that is capable of orchestration of both network and cloud resources over multiple technological and administrative domains [79].

Considering wide-area SDN, distributed frameworks for cooperation between

controllers of multiple SDN domains, possibly operated by different NSPs, were independently proposed by us [61, 89] and others [90]. In our previous work, we proposed both centralized and fully-distributed multi-domain SDN architectures [61], and extended the standard SDN controller by adding inter-controller communication and SLA negotiation functions [13, 14, 89] to realize the distributed architecture. The work in this chapter extends our prior work to reformulate multi-operator video services with dynamic E2E service-level negotiation and traffic engineering over a multi-domain SDN in a scalable manner.

There are two types of content-distribution networks today, closed networks that offer some level of QoS guarantees (e.g., IPTV networks) and over-the-top (OTT) services over the open Internet with best effort service. With the advent of SDN network slicing and cooperation between multiple NSPs, it is now possible to offer these two services (best-effort and value-added services) as inter-operator services with multiple service levels on the same physical network [74], 5G-Ex [79].

To summarize, the main novelties of the proposed *distributed* (virtual) autonomous NSP exchange architecture are:

- it is based on cooperation between SDN controllers of different NSPs (as opposed to being a centralized exchange that relies on a third party physical infrastructure),
- it is based on the open market cooperation model with dynamic inter-domain path pricing (as opposed to static pre-negotiated pricing) which should lead to lower prices due to competition among multiple QoS path providers,
- it supports inter-operator services with multiple E2E service levels for different value-added services on the same physical network.

4.3 A Framework for Cooperation of SDN Controllers for a Distributed Open Exchange

A standard SDN controller, whose functionalities are depicted in orange boxes in Fig. 4.1, has full visibility and control of its data-plane network. For example, it hosts a Monitoring Manager, which monitors vital link parameters, such as bandwidth and delay via per-port or per-queue statistics requested from switches using periodic *StatisticsRequest* and *StatisticsReply* messages provided by the OpenFlow protocol [91]. It also hosts a Routing Manager, which manages traffic within its own data-plane based on estimated network parameters and updates flow tables at its switches accordingly.

In order to realize the vision of a distributed open exchange, where each controller performs both intra-domain and inter-NSP dynamic, service-aware path computation with E2E QoS over multi-operator SDN, we propose extensions to a standard (single domain) SDN controller. These extensions are i) inter-NSP controller-to-controller (C2C) communication extension, ii) aggregated topology computation and global network information base extension, and iii) inter-operator E2E path computation functions for value-added services, which are depicted in Fig. 4.1 in green and blue boxes, respectively. We elaborate on the first two below. The third is discussed in Section 4.4.

4.3.1 Controller to Controller Communication

As a first step, controllers of different domains need to automatically discover/authenticate each other, without a need for manual configuration. To this effect, we proposed reactive and proactive discovery processes [13]. In the proactive discovery process, C2C Messaging Manager, depicted in Fig. 4.1, sends LLDP-like periodic discovery messages from all ports of its border gateway switches to discover neighbor domain controllers and border gateways. If inter-domain connection(s) exist, response message(s) with neighbor domain controller ID is received. Discovery messages are sent periodically to check

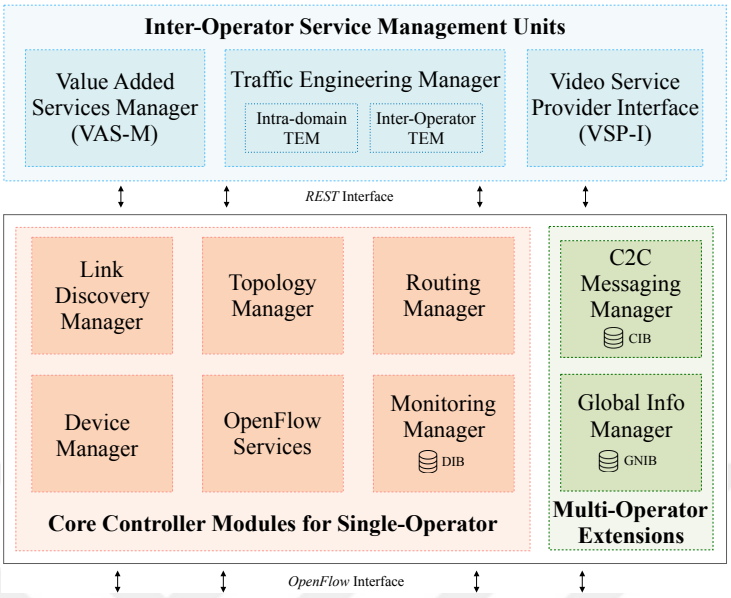

Figure 4.1: An extended SDN controller with inter-controller communication and inter-operator service management functions.

current status of pool of discovered controllers. Discovery messages also contain a field for border gateway ID of source domain sending the message, which is used for discovery of border gateways of destination domains. Information about the non-neighbor domain controllers are received indirectly through the controllers of the neighbor domains. The information collected in the discovery process is stored in the Controller Information Base (CIB) of each controller, where each entry contains controller ID, border gateway switch ID to access the neighbor domain, and the status of the controller.

C2C Messaging Manager is also responsible to broadcast periodic aggregated topology and available link status with price information to other controllers in the CIB (as discussed in Section 4.3.2) as well as managing link bidding and reservation messages between controllers (as discussed in Section 4.4).

C2C messaging may be out-of-band (i.e., over a separate control plane network) or in-band using OpenFlow compatible data plane messages. In-band messaging option does not create scalability problems since messaging is only between domain controllers through border gateways and not all

switches.

4.3.2 Topology Aggregation and Virtual Path Advertising

Each controller has full view of the topology for its own network along with link parameters (e.g., available bandwidth, delay, jitter, etc.) gathered by the monitoring manager, but does not have access to the local information of other NSPs. Sharing some limited topology and network state information between controllers is essential for inter-operator service provisioning. Only aggregated information will be shared for confidentiality/security reasons as operators hide their physical network topology and parameters. Sharing aggregated state information also helps minimizing inter-domain messaging overhead.

The aggregated topology consists of all border gateways of a NSP and all available "virtual links" between these border gateways (hiding the actual physical path information) with their estimated bandwidths, delays, and prices. The concepts of topology aggregation and "virtual links" are demonstrated in Fig. 4.2. Fig. 4.2(a) shows an actual network topology with physical links as a directed graph, where the unfilled and filled dots stand for intra-domain switches and border gateways, respectively. The aggregated graph with only the border gateways and multiple directed virtual links connecting them, where each virtual link represents a different physical path between border switches is depicted in Fig. 4.2(b). Clearly, aggregation introduces some imprecision on the global network state information but this is necessary for confidentiality/security reasons and is tolerable to compute E2E route candidates. The aggregation of network topology, i.e., computing all virtual links and their parameters, is performed using Algorithm 7.

Let $G(S, L)$ be a directed weighted graph representing of the actual topology of a NSP network, where S and L represent sets of switches and links, respectively. Let S^* be the subset of S including only the border switches and BP be the set consisting of all pairs of border switches in S^*. Algorithm 7 is iterative and runs as long as there exists unassigned resources along a dir-

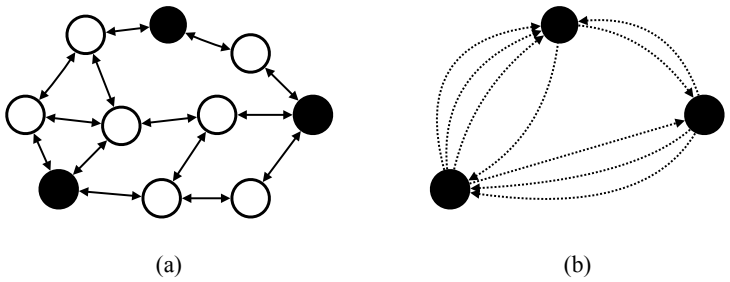

Figure 4.2: Topology aggregation: (a) Actual topology of a NSP network with directed physical links between all switches, (b) Aggregated topology with multiple advertised virtual links between border gateway switches (denoted by black nodes).

ected path p_n connecting border pair $bp_n \in BP$ (Line 3). For each border pair bp_n (Line 4), we find the current shortest path p_n^* for the current border switch pair bp_n (Line 5). If no path is found, we remove bp_n from BP (Line 14). If p_n^* exists, we compute the end-to-end available bandwidth b_n, which is the minimum available bandwidth of all links l along p_n^* (Line 7) and create a directed virtual link with bandwidth b_n connecting the switches bp_n and add it to bp_n's virtual bandwidth set in B_V (Line 8). Accordingly, the total delay d_n of the path p_n^* is calculated as the sum of delays on each link l along p_n^* (Line 9) and added to bp_n's virtual delay set in D_V (Line 10). Then, b_n value is deducted from the available bandwidths b_l of links l along the path (Line 12) and remove any link l if it has no more available bandwidth (Line 14). Note that there might be multiple options for the aggregated representation depending on the order of bp_n's $\in BP$ (Line 4).

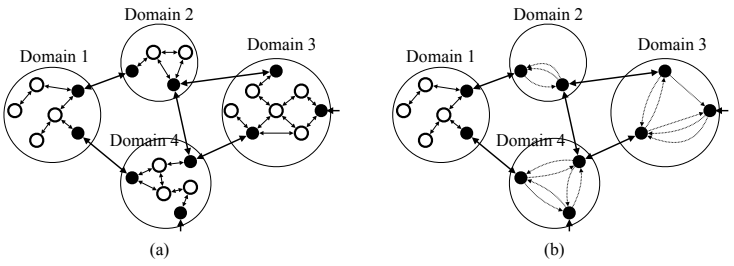

Figure 4.3: SDN with four NSPs: (a) Complete network topology (not visible to anyone), (b) Global network topology as seen by NSP 1 (sees only aggregated topology of other NSP networks)

Algorithm 7: NSP Topology Aggregation

```
1  Input: G and BP;
2  Output: B_V, D_V;
3  while any p_n in G exists for any bp_n ∈ BP do
4      foreach border pair bp_n ∈ BP do
5          Find shortest path p*_n in G;
6          if p*_n exists then
7              Compute the b_n of p*_n;
8              Add b_n to bp_n's virtual bandwidth set in B_V;
9              Compute the d_n of p*_n;
10             Add d_n to bp_n's virtual delay set in D_V;
11             foreach link l along the p*_n do
12                 b_l = b_l − b_n;
13                 if b_l = 0 then
14                     Remove l from G;
15         else
16             Remove bp_n from BP;
```

Controllers communicate to share the aggregated network information using *PacketOut* messages, which are sent to the related border gateway which forwards the message to the target neighbor controller. Each domain controller keeps a database called Global Network Information Base (GNIB) which stores a global network map with parameters such as available bandwidths and delays for i) virtual links of all domains that are computed by the Algorithm 7, and ii) direct inter-operator links. We note that information on non-neighboring domains are indirectly gathered by neighbor domains through GNIB sharing and synchronizing with the Topology and Link Information Sharing C2C message [13]. Controllers periodically sync their GNIB to keep their global network map current. In addition, each controller keeps the full topology and resource information for its own network in its Domain Information Base (DIB). Using DIB and GNIB, each domain has a particular view of the global network state. Fig. 4.3 illustrates the global network view of an operator for a multi-domain SDN with four domains. The complete network topology, which is not seen by any NSP, is shown in

Fig.4.3a, while the view of the global network as seen by the controller of NSP 1 is presented in Fig.4.3b. Once an operator constructs its own view of the global network state, it can process E2E service requests for its own clients using the dynamic E2E resource provisioning procedure described in Section 4.4.

4.4 Distributed Dynamic End-to-End Service Provisioning over Multi-Operator SDN

The proposed multi-operator value-added services with service-level differentiation is motivated by economical factors. NSPs would like to configure such services dynamically under their own control to better monetize their resources by offering E2E hard or soft QoS to its customers (end-users or third-party video service providers). They also would like to dynamically better manage their own network resources. We discuss open market model and possible path pricing models that drive the distributed open exchange in Section 4.4.1.

In the proposed distributed open exchange framework, controller of each NSP performs inter-operator path optimization, which is discussed in Section 4.4.3, based on its most recent view of the global aggregated network view with virtual links advertised by other operators. Each NSP also has complete control of its own intra-domain routing, i.e., selecting which virtual paths to advertise, how to price them, and the actual physical paths that they correspond to, as discussed in Section 4.4.4.

4.4.1 Multiple Service-Levels and NSP Revenue Model

It has been foreseen that service-levels other than the best-effort (BE), such as improved quality (IQ) and assured quality (AQ) levels, will be available in future networks for multimedia services [74]. In the AQ service (hard

QoS), users will be guaranteed to receive a desired throughput and/or delay subject to admission control. The IQ service provides soft QoS, where IQ users shall be given higher priority to receive higher quality services as long as network resources are available. In order to provide such value-added services, different network operators must dynamically negotiate per flow E2E service level agreements (SLA) among each other.

We propose a distributed open exchange framework, where controllers of different domains cooperate with each other to automatically provision per flow E2E QoS. The open exchange framework is based on an open market model with dynamic price competition between different network operators. Each operator has its own revenue model and advertises its prices for virtual QoS paths across its domain accordingly. Different dynamic demand-based price models that are available in other industries, e.g., airline seat price models [92], can be applied to dynamic pricing of QoS paths for value-added services.

4.4.2 Example Multi-Operator Value-Added Service Scenario

This section presents an example value-added video streaming service scenario, depicted in Fig. 4.4, in order to demonstrate the interaction between an end user, video service provider, and multiple NSPs involved in provisioning an E2E service.

The information flow in our example scenario of setting up a premium video streaming service is as follows: i) A user, who belongs to NSP 1, requests a specific video content from the OTT VSP, which resides in NSP 4. ii) The VSP sends the video parameters, such as resolution and bitrate, as well as client information, such as IP address, to the controller of NSP 4 (its own domain). iii) The controller of NSP 4 passes these parameters to the controller of NSP 1 (user's domain). Note that inter-NSP flow of such control messages is realized between the VSP-I units of controllers via the C2C Messaging Manager. Any inter-domain path can be used for control messaging, e.g., in Fig. 4.4, controllers of either NSP 2 or NSP 3 can be a transit hub

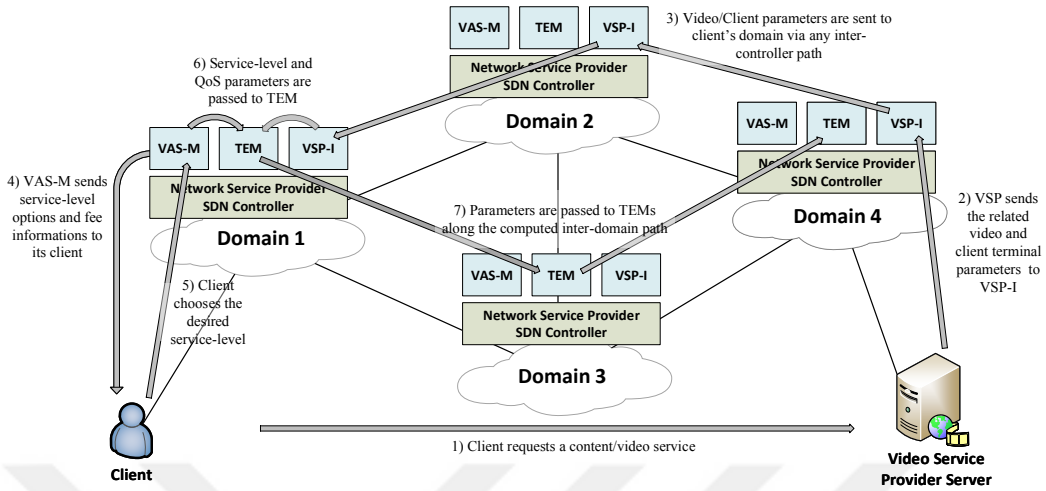

Figure 4.4: Information flow between the user, VSP, and controllers of four NSPs for requesting and setting up an E2E value-added video service with a user-specified service-level across a multi-operator network.

when sending a message from controller of NSP 4 to controller of NSP 1. iv) Once the controller of client's NSP receives video parameters, it notifies VAS-M to contact the user and offer value-added services with service-level options and pricing information. v) Client selects the desired service-level according to his/her QoE expectations and pricing information. vi) VAS-M transfers the user selected service-level information and video parameters to the TEM. Note that TEM of each NSP can access the network state of its own domain (stored in DIB) that is monitored by the controller, as well as the global network state (stored in GNIB) that is periodically synchronized. vii) The Inter-TEM of NSP 1 decides the inter-operator path as described in Section 4.4.3. viii) The service-level parameters and entry/exit border-gateways of transit NSPs are passed to TEM's of NSPs along the path using C2C messaging. ix) Finally, intra-TEM of each NSP computes the path within the specified border gateways of their domains given the service parameters and sends this information to NSP 1, which stitches them together.

4.4.3 Inter-Operator Path Optimization (Inter-TEM)

The inter-TEM unit of each NSP finds the best "virtual" inter-operator path between the source and destination domains, as stated by step vii) in Section 4.4.2, based on the global aggregated network map, e.g., that shown in Fig. 4.3b. Recall that this map advertises virtual paths between border gateways in each domain provided by operators with properties such as bandwidth, delay and price. The aim of inter-operator path optimization is to determine which transit NSP domains to use to go from the source NSP to the destination NSP assuming that there are more than one alternatives, and which border gateways to use to enter and/or exit each NSP domain, assuming that each NSP network has more than one gateway.

This is a constrained shortest path problem which determines the cheapest path from the source NSP to the destination NSP that satisfies the QoS parameters and user specified service level parameters. The solution is obtained by searching the best E2E virtual path over the aggregated global view of alternative "virtual" paths that is stored in the GNIB. The advertised virtual link/path information includes the QoS parameters of each virtual path and associated per-megabit service prices for each value-added service levels. This problem is solved using Algorithm 8 by the inter-TEM function of the controller of the NSP where the user resides.

Algorithm 8 takes the requested bandwidth b_i^{req} for user i, source and destination node ids src_i, dst_i, requested service-level $Service\_Level_i$, global aggregated view of network resources stored in GNIB, and bandwidth step size δ as inputs. The algorithm returns the bandwidth b_i^{alloc} that can be allocated to user i, the best path p_i^*, the price f_i, and the IDs of NSPs along p_i^* together with their entry and exit border gateways. The algorithm iteratively searches for b_i^{alloc}, which is initialized as b_i^{req} (line 3). It first creates a directed weighted multi-graph G representation of the global network view from GNIB (line 5), then removes all links with available bandwidth less than b_i^{alloc} (line 6-8). Then, it retrieves from GNIB the set BP of all entry/exit border-gateway pairs bp_n along with the corresponding set L of virtual links l_n connecting each pair n of border gateways (line 9). For each border gateway

Algorithm 8: Inter-Operator Virtual Path Computation

Input: b_i^{req}, src_i, dst_i, $Service\_Level_i$, $GNIB$, δ
Output: b_i^{alloc}, p_i^*, f_i, $(C, (s_{src}, d_{dst}))$
$b_i^{alloc} = b_i^{req}$;
do
    Create multi-graph G from $GNIB$;
    foreach *link* l *with bandwidth* b_l *in* G **do**
        if $b_l < b_i^{alloc}$ **then**
            Remove l from G;
    Get (BP, L) map from $GNIB$;
    foreach *border pair* bp_n *in* BP **do**
        Find the cheapest link l_{bp_n} in link set l_n;
        Remove l_{bp_n} from link set l_n;
        Remove set l_n from G;
    Find the least-cost path $p_i^* \leftarrow$ Dijkstra(G, src_i, dst_i);
    if p_i^* *exists* **then**
        $f_i = \sum_{l \in p_i^*} f(l)$;
        Construct $(C, (s_{src}, d_{dst}))$ map from p_i^*;
    else
        if *(*$Service\_Level_i = AQ$*)* **then**
            check links in l_n other than the cheapest
            otherwise, $b_i^{alloc} = 0$;
        if *(*$Service\_Level_i = IQ$*)* **then**
            $b_i^{alloc} = max(0, b_i^{alloc} - \delta)$;
            if $b_i^{alloc} = 0$ **then**
                break;
while p_i^* *does not exist*;

pair bp_n, the cheapest link for the desired service level is kept and other virtual links are removed from the graph G (line 10-13). At this point, the graph G is not a multi-graph anymore and contains at most a single directed link between border switches in a particular domain. Setting the service-level prices at each virtual link as the cost metric, the cheapest path p_i^* is computed as the least-cost path from src_i to dst_i in G (Line 14). We note that the cost for inter-operator links between border gateways of different NSPs are set equal to zero. Since particular links were removed from G (line 6-13), it is possible that an E2E path p_i^* does not exist. If p_i^* exists, the bandwidth b_i^{alloc} and total price f_i of the path p_i^* (sum of fees for each link along the path) are stored as outputs (Line 16). It also constructs the map containing the entry and/or exit border gateway ID's (s_{src}, s_{dst}) for each NSP along the inter-domain path to notify respective NSP controllers in C (line 17). If p_i^* does not exist and the requested service-level is AQ, then we search a feasible E2E path over virtual links in l_n other than the cheapest ones. If no such E2E path that satisfies the bandwidth constraint exists, then inter-TEM is not able to provide the requested service and the service request is denied (line 19-21). On the other hand, if the requested service-level is IQ, then inter-TEM tries to allocate a lower bandwidth so it decrements b_i^{alloc} by a step size δ (e.g., 100 Kbps), which is an input to the algorithm (line 23). Having an updated desired bandwidth value, the next iteration of the algorithm is performed after constructing a new graph G (line 5-13). Once the inter-domain path is computed, VAS-M communicates with the controllers in C and informs them about the allocated bandwidth, and entry and/or exit switches so that each NSP can perform intra-domain routing.

The controller of the NSP of user's domain monitors whether the advertised SLA are delivered during the duration of the service. The process is dynamic, i.e., in case the advertised SLA cannot be fulfilled by one or more NSPs due to link failures or other reasons, the inter-NSP path is recomputed in real-time, where messaging with other controllers are performed through the C2C Messaging Manager.

4.4.4 Intra-Domain Traffic Engineering (Intra-TEM)

The intra-TEM unit of each NSP is responsible for finding the best "physical" path between the source and destination switches in their own data-plane network, as stated by step ix) in Section 4.4.2, based on the actual network state information that is monitored by the SDN controller. Each NSP chooses its preferred approach for intra-domain path computation, resource allocation, queue management, traffic policing, etc. The algorithms used by each NSP may be proprietary and are not exposed to other operators. In the following, we present an example algorithm for intra-TEM of an NSP.

Algorithm 9: Intra-domain Physical Path Computation

```
1 Input: b_i^alloc, src_i, dst_i, DIB;
2 Output: p_i;
3 Create graph G from DIB;
4 foreach link l with bandwidth b_l in G do
5     if b_l < b_i^alloc then
6         Remove l from G;
7 Find the shortest path p_i^* ← Dijkstra(G, src_i, dst_i);
```

Finding a path with bandwidth b_i^{alloc} between the source and destination switches (src_i and dst_i) is again a constrained shortest path problem. Since the SDN controller of each NSP has full visibility of network traffic and switch queue states within its domain (stored in DIB), the intra-TEM of each NSP can compute a physical network slice between its source and destination switches with the desired bandwidth b_i^{alloc} using Algorithm 9, which first creates a directed graph G using information in the DIB (line 3), and discards links with insufficient bandwidth from G (line 4-6). Then, it runs the shortest path algorithm to compute the path with the minimum number of hops (line 7). For transit NSP, the source and destination switches are both border switches. For the source and destination NSP, the source and destination switches are those the client or the VSP server is connected, respectively.

The E2E network slice then consists of concatenation of sub-slices reserved by each NSP along the E2E path p_i^*. Note that path computation at each NSP is dynamic in the sense that it is repeated if the current path no

longer satisfies the advertised SLA in a dynamic network environment. Once each NSP computes the intra-domain path and reserves requested resources, it sends a C2C message notifying the controller of the NSP where the user resides that its slice is ready [13].

Although the inter-TEM computes an E2E virtual path based on the availability information advertised by each NSP with QoS parameters stored in the GNIB, there may be some unlikely cases where the intra-TEM of a particular NSP could not allocate a physical path with the advertised parameters due to synchronization delay between the DIB of an NSP and GNIB of another NSP. In such a case, the NSP that experiences lack of resources notifies the NSP of the user so that NSP of the user restarts its inter-TEM process with an updated GNIB.

4.5 Evaluation

We describe our test environment in Section 4.5.1. Our experimental results are presented in Section 4.5.2.

4.5.1 Test Environment

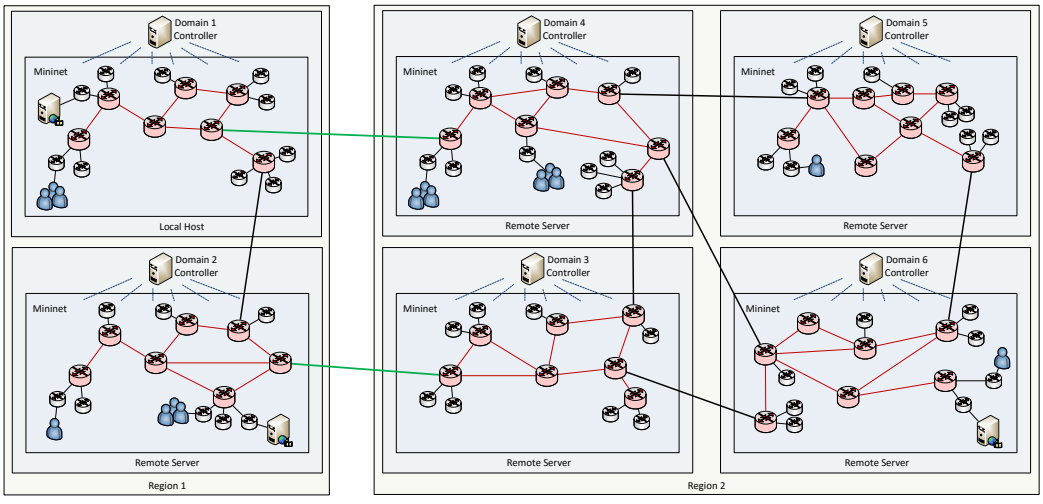

Figure 4.5: Test environment with six SDN Operators

We provide experimental results to validate the proposed architecture in the Mininet environment using OVS switches and Floodlight SDN controllers [49] for each network operator domain. Communications between the controllers of multiple NSP domains is realized using the messaging protocol that was developed in our earlier work [13]. The multi-domain topology model is generated in the same way as in the GT-ITM tool [69]. Each NSP data-plane network has 50 switches, where the average connectivity degree is 3 and links are configured at 1 Gbps. The data plane network is emulated by using the Mininet Cluster edition 2.2.0 [93], which allows us to emulate the nodes and links for each NSP network on different servers. Each controller runs on the server where its data-plane network is emulated. An exemplary multi-domain emulation environment with six NSP domains is depicted in Fig. 4.5. The intra-domain links are virtual Ethernet links, whereas the inter-domain links connecting different remote servers are SSH links. The backbone switches of operator networks and stub switches are shown with red and gray colors, respectively. The latency of all links are modeled as uniform random variables, where inter-domain links have latency between 20-40 ms and intra-domain links have latency between 5-10 ms.

Video service requests are simulated at random according to a Poisson process model, which has been shown to well model video requests from YouTube [94]. The parameter λ of the Poisson process is set to 5 for each domain, simulating arrival of 5 new requests per 200 ms intervals. Client device types and video resolution requested by clients are modeled according to numbers projected for 2021 in the Cisco report [1]. According to this report, the distribution of requested video resolutions are taken as 41%, 47%, and 12% for SD, HD, and UHD, respectively, creating different bandwidth requirements for the service requests. We use YouTube suggested bitrates for encoding videos at these resolutions, which range between 2 Mbps and 18 Mbps. We set the ratio of IQ service level requests to AQ service level requests equal to 2 assuming that more clients choose the cheaper IQ service level, where the per-megabit value added service charges are set as 10 units for IQ and 20 units for AQ service level. Value-added service requests are generated independently in each domain for a duration of 10 seconds creating

a total of 1500 service requests across our multi-domain test environment. We also create random background traffic at different levels on the data plane network using custom scripts running at Mininet hosts.

4.5.2 Experimental Results

We present experimental results under two subsections: 1) to demonstrate the effect of random delays in GNIB synchronization on the proposed distributed exchange framework and 2) to demonstrate the effect of open competitive pricing vs. pre-negotiated fixed pricing on the operator revenues.

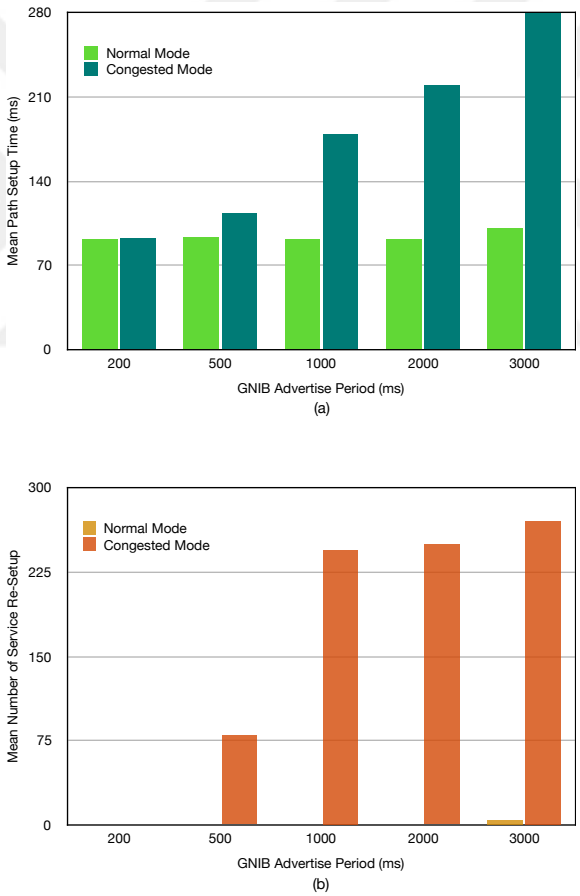

Figure 4.6: Impact of GNIB update period on: (a) Service path setup time, and (b) Number of service path re-computations, when all domains are in normal mode of operation vs. some domains are in congested mode of operation.

4.5.2.1 Effect of GNIB Update Period

This subsection discusses the effect of GNIB advertisement period on the multi-operator E2E service setup time performance. The mean E2E path setup times under different GNIB advertisement periods are presented in Fig. 4.6-a for two cases: i) when all domains are in normal mode of operation, and ii) when some domains are in congested mode of operation, i.e., when some domains are not able to serve all incoming service requests.

Higher GNIB advertisement period means all NSPs may not have an up-to-date view of global network state at a given time. For example, while inter-TEM of a NSP computes an inter-domain path using info from its current GNIB, the network state in other domains may change. This results in computation of inaccurate E2E paths. In Fig. 4.6-a, the service-setup times are around 90 ms in the normal mode of operation, and are not affected by higher GNIB update periods. This can be attributed to availability of ample network resources in the normal mode of operation, which avoids potential problems, i.e., erronous computation of inter-domain paths with insufficient resources. On the other hand, when domains operate in congestion mode, we observe that service setup times increase with higher GNIB advertisement periods. This is mainly because having up-to-date advertisement of available paths is more important when at least one NSP experiences congestion. We observe that after an inter-domain path is calculated and service request messages are transferred to the controllers along the computed path, some domains are not able to provide the requested paths since they are no longer available. In such a case, the inter-TEM re-computes alternative paths, which generally have higher number of hops from the source to destination, using an up-to-date GNIB. The mean number of path re-computations across all domains for varying GNIB advertisement periods are shown in Fig. 4.6-b. In our experiments, the service setup times may increase up to 300 ms due to path recomputations.

We can avoid or reduce the number of path re-computations by more frequent GNIB updates. On the other hand, low GNIB update periods causes higher C2C messaging overhead. Note that there is no service re-setup in the

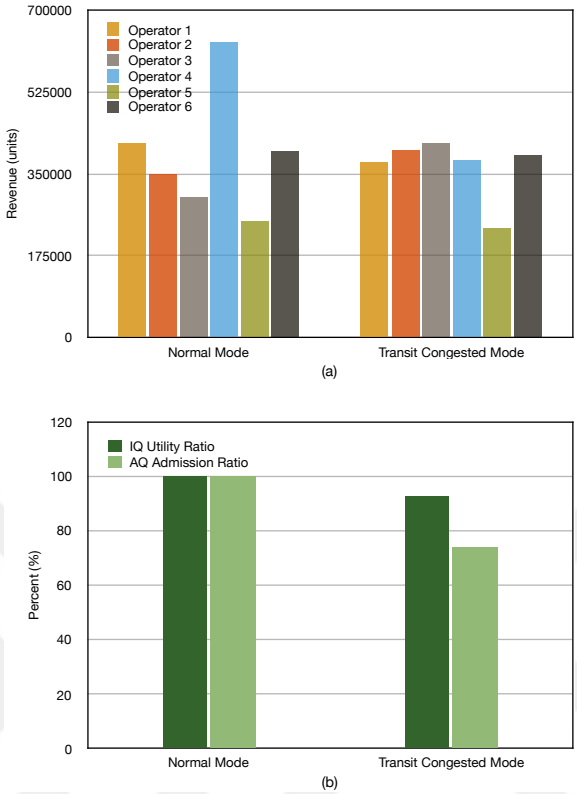

Figure 4.7: Effect of congestion on operator revenue under static price model: (a) Operator revenues when all domains except domain 4 are in normal mode of operation and domain 4 experiences congestion, (b) AQ admission ratio and IQ utility ratio when all domains except domain 4 are in normal mode of operation and domain 4 experiences congestion.

normal mode of operation, while up to 250 requests (i.e., 17% of all requests across multi-domain) are reprocessed as presented in Fig. 4.6-b.

4.5.2.2 Effect of Price Model on Revenue - Static vs. Dynamic Pricing

This subsection studies the effect of static vs. dynamic price model on NSP revenues. Fig. 4.7a shows operator revenues and Fig. 4.7b depicts AQ admission ratio (percentage of admitted AQ users) and IQ utility ratio (average ratio of allocated and requested bandwidths) and compares them when all domains are in normal mode of operation vs. when one domain (domain 4) is in congestion. Note that we keep GNIB advertisement period fixed at 2

seconds in this experiment and all operators adopt the same pricing model, where per-megabit bandwidth price for IQ and AQ service-levels are set to 10 and 20 units, respectively. In Fig. 4.7a, in the normal mode of operation, although each domain generates service requests at the same rate, the revenue of operator 4 is much higher than other operators, while revenues of operators 1, 2 and 6 are comparable, and operator 5 receives the lowest revenue. This uneven distribution of revenue can be explained by the central location of NSP 4, as can be seen in Fig.4.5, where service requests originating from other NSPs use NSP 4 most frequently for transit domain. However, as the background traffic increases in domain 4, revenue of NSP 4 decreases considerably, as transit traffic is dynamically diverted to domains 2 and 3. Hence, we observe that both the traffic and revenues of NSP 2 and 3 increases as revenue of NSP 4 decreases. However, we note that the overall revenue for all value added services across all domains decreases, where IQ utility ratio and AQ admission ratio decreases by up to 25%. Since AQ service level provides all of requested bandwidth or nothing, the revenue decrease due to AQ services is steeper compared to that of IQ service level, as some IQ requests are served with lower bandwidths compared to the requested bandwidth.

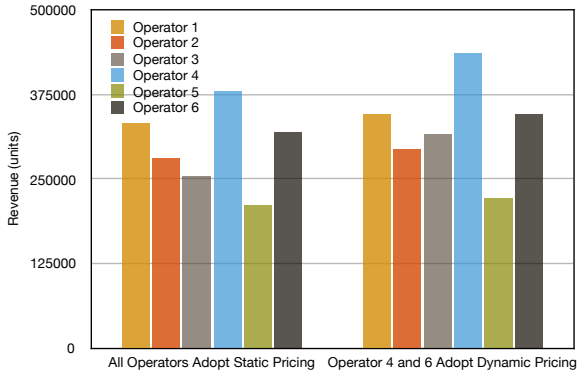

Figure 4.8: Effect of pricing model on operator revenues: (a) Static pricing model, (b) Dynamic pricing model.

In order to avoid revenue losses due to congestion, operators may choose to adopt a dynamic price model, where the price of bandwidth increases with network utilization. We compare static vs. dynamic pricing, where we adopt discrete price model based on the network load factor, that is, per-

megabit price multipliers at each link (for each value added service level) is set equal to 1x, 1.5x, 3x for network utilization states between 0%-50%, 50%-80%, and 80%-100%, respectively. In these experiments, operators 1, 2, 3, and 6 adopt static price model, while operators 4 and 5 adopt the above dynamic price model. Note that other dynamic price models, e.g., airline seat price model, can also be adopted. Fig. 4.8 presents revenue per operator comparing static and dynamic price models. We observe that operator 4 increases its revenue considerably by adopting the dynamic price model. On the other hand, revenue increase in operator 5 is small. This is mainly due to other operator's inter-TEM modules does not use it as transit because of inter-domain location of operator 5's network and increased resource prices of operator 5 shared through GNIB advertisement. In addition to these, we observe revenue increases at operators 2 and 3 even they do not adopt dynamic pricing. Such situation is mainly attributed to increased resource prices of operator 4 that affect inter-domain routes and shifts some of them to operator 2 and 3.

4.6 Conclusions

We propose an SDN-enabled distributed cooperative inter-operator resource management framework to accommodate in-elastic traffic according to varying throughput and delay requirements of applications or flows to support multi-operator video services with E2E QoS. The proposed distributed open exchange framework is scalable with the number of operators and service demand, since signaling between operators for flow management needs to be carried out only between controllers rather than all routers along a path.

The proposed distributed open exchange framework is more advantageous than a centralized physical exchange infrastructure since it gives full autonomy to the operators and allows operators to cooperate based on an open market price competition model. Operators are in complete control of their network resources and can enforce their own policies of choice and pricing model in their collaborations with other operators. Our experimental results clearly demonstrate that operators can increase their revenues by ad-

opting a dynamic pricing policy enabled by the proposed open exchange, where the price of bandwidth varies with network utilization levels.

Chapter 5

Conclusion

In this thesis we have investigated managed video service architectures over future networks which offer (i) OTT video service providers (VSP) the same benefits as IPTV so that they can offer their clients premium content with some level of QoE, (ii) users the choice of service-level that they are willing to pay for, and (iii) network service providers (NSP) additional revenues from value-added video services. Software defined networking (SDN) will make e2e resource computation and QoS provisioning more practical and feasible. Hence, it will be possible to offer NSP-managed video services to individual clients or VSP-managed streaming services to a group of clients over network slices with reserved capacity.

We observe that reserving network resources alone is not sufficient to i) minimize video-quality variability per client, ii) maximize video-quality fairness among heterogeneous clients, and iii) maximize total goodput over the reserved slice capacity, because of the nature of the TCP congestion control mechanisms. Hence, we propose and implement SDN-assisted architectures for collaboration between NSP, VSP, and video clients to estimate a fair-share rate for each client, so that clients can set their TCP receive-window size and DASH video adaptation rate consistently in order to avoid unnecessary packet losses and TCP goodput fluctuations. Experiments show that the proposed managed service architectures provide less frequent video quality variations per client in smaller amounts and almost no freezes compared

to competitive DASH streaming contributing towards a higher QoE.

With the vision of open multi-service inter-networking, including different service-levels, service-level awareness of users, and associated business models, NSPs can offer value-added video services over SDN and maximize its revenues. We demonstrate that resource allocation with first-come first-serve provisioning is not optimal when NSP revenues are considered. Experiments show that our fast batch processing heuristic achieves on the average 90% revenue realization compared to the approximately optimal solution, while providing fairness among all users with the same service-level.

Wide area SDN is now a reality, and we foresee multi-domain wide area SDN operated by multiple NSPs, each managing their own SDN domain. Our proposed distributed e2e resource calculation framework enables operators to cooperate based on an open market price model where each NSP has full autonomy. Experimental results clearly demonstrate that NSPs can increase their revenues by adopting a dynamic pricing policy enabled by the proposed open exchange, where the price of bandwidth varies with network utilization levels.

Overall, the proposed managed service architectures are beneficial for all parties involved in the delivery of OTT video services, i.e., the NSP, VSP, and users can all benefit from new business models enabled by the next generation technologies.

References

[1] "The Zettabyte Era: Trends and Analysis, Cisco Whitepaper," June 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.pdf

[2] NETWORLD2020-Whitepaper, "Service level awareness and open multi-service internetworking - principles and potentials of an evolved internet ecosystem," February 2016.

[3] C. Srinivasan, A. Viswanathan, and T. Nadeau, "Multiprotocol label switching (MPLS) label switching router (LSR) management information base (MIB)," RFC 3813, June 2004.

[4] R. Alimi, R. Penno, Y. Yang, S. Kiesel, S. Previdi, W. Roome, S. Shalunov, and R. Woundy, "Application-layer traffic optimization (ALTO) protocol," RFC 7285, September 2014.

[5] J. Vasseur and J. L. Roux, "Path computation element (PCE) communication protocol (PCEP)," RFC 5440, March 2009.

[6] "Software-defined networking: The new norm for networks," *Open Networking Fundation (ONF) White Paper*, 2012.

[7] K. Bagci, K. Sahin, and A. M. Tekalp, "Compete or collaborate: Architectures for collaborative DASH video over future networks," *IEEE Trans. on Multimedia*, vol. 19, no. 10, pp. 2152–2165, Oct. 2017.

[8] K. E. Sahin, K. T. Bagci, and A. M. Tekalp, "Distributed-collaborative managed DASH video services," in *Int. Conf. on Network and Service Management (CNSM), Tokyo, Japan*, Nov. 2017.

[9] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming," *IEEE Trans. on Multimedia*, vol. 19, no. 10, pp. 2136–2151, Oct. 2017.

[10] K. Bagci and A. M. Tekalp, "Dynamic resource allocation by batch-optimization for value-added video services over SDN," *accepted for publication in IEEE Trans. on Multimedia*, 2018.

[11] R. A. Kirmizioglu, B. C. Kaya, and A. M. Tekalp, "Multi-party WebRTC videoconferencing using scalable VP9 video: From best-effort over-the-top to managed value-added services," in *IEEE Int. Conf. on Multimedia and Expo (ICME), San Diego, CA, USA*, July 2018.

[12] K. T. Bagci, K. E. Sahin, and A. M. Tekalp, "Queue-allocation optimization for adaptive video streaming over software defined networks with multiple service-levels," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 1519–1523.

[13] K. T. Bagci, S. Yilmaz, K. E. Sahin, and A. M. Tekalp, "Dynamic end-to-end service-level negotiation over multi-domain software defined networks," in *IEEE Int. Conf. on Communications and Electronics (ICCE)*, 2016.

[14] K. T. Bagci and A. M. Tekalp, "Managed video services over multi-domain software-defined networks," in *Proc. of EUSIPCO, Budapest, Hungary*, Aug. 2016.

[15] K. Bagci and A. M. Tekalp, "Sdn-enabled distributed open exchange: dynamic qos-path optimization in multi-operator services," *submitted for publication in IEEE Journal on Selected Areas in Communications*, 2018.

[16] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "Acm proc. of int. workshop on network and operating system support for digital audio and video," 2012, pp. 9–14.

[17] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos, "The algorithmic aspects of network slicing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 112–119, 2017.

[18] T. Stockhammer, P. Fröjdh, I. Sodagar, and S. Rhyu, "Information technologympeg systems technologiespart 6: Dynamic adaptive streaming over http (dash)," *ISO/IEC, MPEG Draft International Standard*, 2011.

[19] R. Kuschnig, I. Kofler, and H. Hellwagner, "An evaluation of tcp-based rate-control algorithms for adaptive internet streaming of h. 264/svc," in

Proceedings of the first annual ACM SIGMM conference on Multimedia systems. ACM, 2010, pp. 157–168.

[20] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal processing: Image communication*, vol. 19, no. 2, pp. 121–132, 2004.

[21] Y. Chen, K. Wu, and Q. Zhang, "From qos to qoe: A tutorial on video quality assessment," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1126–1165, 2015.

[22] R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the quality of experience of http video streaming," in *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*. IEEE, 2011, pp. 485–492.

[23] O. Oyman and S. Singh, "Quality of experience for http adaptive streaming services," *IEEE Communications Magazine*, vol. 50, no. 4, 2012.

[24] C. Timmerer, M. Maiero, B. Rainer, S. Petscharnig, D. Weinberger, C. Mueller, and S. Lederer, "Quality of experience of adaptive http streaming in real-world environments," *IEEE Comsoc MTC E-Letter*, vol. 10, no. 3, 2015.

[25] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[26] B. Rainer, S. Petscharnig, C. Timmerer, and H. Hellwagner, "Statistically indifferent quality variation: An approach for reducing multimedia distribution cost for adaptive video streaming services," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 849–860, 2017.

[27] M. Barbera, A. Lombardo, C. Panarello, and G. Schembra, "Active window management: an efficient gateway mechanism for tcp traffic control," in *IEEE International Conference on Communications (ICC)*, 2007, pp. 6141–6148.

[28] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 1, pp. 326–340, 2014.

[29] C. Zhou, C.-W. Lin, and Z. Guo, "mdash: A markov decision-based rate adaptation approach for dynamic http streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 738–751, 2016.

[30] Y. Sun, C. Lee, R. Berry, and A. Haddad, "A load-adaptive ack pacer for tcp traffic control," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 40, no. 3, 2002, pp. 1620–1621.

[31] A. Sideris, E. Markakis, A. Trigonis, G. Alexiou, E. Pallis, and C. Skianis, "Qoe fairness, http adaptive streaming and idvb-t: The awm approach," in *IEEE International Conference on Telecommunications and Multimedia (TEMU)*, 2014, pp. 29–33.

[32] A. Mansy, M. Fayed, and M. Ammar, "Network-layer fairness for adaptive video streams," in *IEEE IFIP Networking Conference, 2015*, 2015, pp. 1–9.

[33] R. Houdaille and S. Gouache, "Shaping http adaptive streams for a better user experience," in *Proceedings of the ACM 3rd Multimedia Systems Conference*, 2012, pp. 1–9.

[34] O. N. Fundation, "Software-defined networking: The new norm for networks," *ONF White Paper*, vol. 2, pp. 2–6, 2012.

[35] M. Karakus and A. Durresi, "Quality of service (qos) in software defined networking (sdn): A survey," *Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017.

[36] D. Palma, J. Gonçalves, B. Sousa, L. Cordeiro, P. Simoes, S. Sharma, and D. Staessens, "The queuepusher: Enabling queue management in openflow," in *IEEE European Workshop on Software Defined Networks (EWSDN)*, 2014, pp. 125–126.

[37] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide qoe fairness using openflow-assisted adaptive video streaming," in *Proceedings of the ACM SIGCOMM workshop on Future human-centric multimedia networking*, 2013, pp. 15–20.

[38] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards qoe-aware video streaming using sdn," in *IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 1317–1322.

[39] S. Ramakrishnan, X. Zhu, F. Chan, and K. Kambhatla, "Sdn based qoe optimization for http-based adaptive video streaming," in *IEEE International Symposium on Multimedia (ISM)*, 2015, pp. 120–123.

[40] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "Qoe-driven rate adaptation heuristic for fair adaptive video streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 2, p. 28, 2016.

[41] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering stable high-quality video: An sdn architecture with dash assisting network elements," in *Proceedings of the ACM International Conference on Multimedia Systems*, 2016, p. 4.

[42] M. Mu, M. Broadbent, A. Farshad, N. Hart, D. Hutchison, Q. Ni, and N. Race, "A scalable user fairness model for adaptive video streaming over sdn-assisted future networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2168–2184, 2016.

[43] A. Bentaleb, A. C. Begen, and R. Zimmermann, "Sdndash: Improving qoe of http adaptive streaming using software defined networking," in *Proceedings of the ACM Conference on Multimedia*. ACM, 2016, pp. 1296–1305.

[44] E. Liotou, G. Tseliou, K. Samdanis, D. Tsolkas, F. Adelantado, and C. Verikoukis, "An sdn qoe-service for dynamically enhancing the performance of ott applications," in *IEEE International Workshop on Quality of Multimedia Experience (QoMEX)*, 2015, pp. 1–2.

[45] "OF-Config 1.2." [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf

[46] B. Pfaff and B. Davie, "The open vSwitch database management protocol," 2013.

[47] "Open vSwitch: An Open Virtual Switch," http://openvswitch.org, accessed: 2016-01-20.

[48] "Mininet overview," http://mininet.org/overview/, accessed: 2015-10-05.

[49] "Floodlight Controller," http://www.projectfloodlight.org/floodlight.

[50] F. Kuipers, P. Van Mieghem, T. Korkmaz, and M. Krunz, "An overview of constraint-based path selection algorithms for QoS routing," *IEEE Communications Magazine, 40 (12)*, 2002.

[51] G. Xue, W. Zhang, J. Tang, and K. Thulasiraman, "Polynomial time approximation algorithms for multi-constrained qos routing," *IEEE/ACM Trans. on Networking (ToN)*, vol. 16, no. 3, pp. 656–669, 2008.

[52] S. Chen, M. Song, and S. Sahni, "Two techniques for fast computation of constrained shortest paths," *IEEE/ACM Trans. on Networking (TON)*, vol. 16, no. 1, pp. 105–115, 2008.

[53] S. Zhang, F. R. Yu, and V. C. Leung, "Joint connection admission control and routing in IEEE 802.16-based mesh networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, 2010.

[54] S. Zhu, S. Li, and X. Chen, "Design QoS-aware multi-path provisioning strategies for efficient cloud-assisted SVC video streaming to heterogeneous clients," *IEEE Trans. on Multimedia*, vol. 15, no. 4, pp. 758–768, 2013.

[55] Y. Wen, X. Zhu, J. J. Rodrigues, and C. W. Chen, "Cloud mobile media: Reflections and outlook," *IEEE Trans. on Multimedia*, vol. 16, no. 4, pp. 885–902, 2014.

[56] B. Cheng, J. Yang, S. Wang, and J. Chen, "Adaptive video transmission control system based on reinforcement learning approach over heterogeneous networks," *IEEE Trans. on Automation Science and Engineering*, vol. 12, no. 3, pp. 1104–1113, July 2015.

[57] J. Wu, B. Cheng, M. Wang, and J. Chen, "Energy-efficient bandwidth aggregation for delay-constrained video over heterogeneous wireless networks," *IEEE J. on Selected Areas in Communications*, vol. 35, no. 1, pp. 30–49, Jan. 2017.

[58] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," *Computer Networks*, vol. 71, pp. 1–30, 2014.

[59] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, "An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks," *IEEE Trans. on Multimedia*, vol. 15, no. 3, pp. 710–715, 2013.

[60] H. Huang, P. Li, S. Guo, and B. Ye, "The joint optimization of rules allocation and traffic engineering in software defined network," in *IEEE Int. Symposium of Quality of Service (IWQoS)*, 2014, pp. 141–146.

[61] H. E. Egilmez and A. M. Tekalp, "Distributed QoS architectures for multimedia streaming over software defined networks," *IEEE Trans. on Multimedia*, vol. 16, no. 6, pp. 1597–1609, 2014.

[62] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming," *IEEE Trans. on Multimedia*, vol. 19, no. 10, pp. 2136–2151, Oct. 2017.

[63] J. Yang, K. Zhu, Y. Ran, W. Cai, and E. Yang, "Joint admission control and routing via approximate dynamic programming for streaming video over software-defined networking," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 619–631, 2017.

[64] T. Flach, P. Papageorge, A. Terzis, L. Pedrosa, Y. Cheng, T. Karim, E. Katz-Bassett, and R. Govindan, "An internet-wide analysis of traffic policing," in *Proceedings of ACM SIGCOMM*, 2016, pp. 468–482.

[65] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using OpenFlow-assisted adaptive video streaming," in *Proc. of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*. ACM, 2013, pp. 15–20.

[66] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," *IEEE/ACM Trans. on Networking*, vol. 14, no. 1, pp. 133–146, Feb. 2006.

[67] T. Mori, R. Kawahara, H. Hasegawa, and S. Shimogawa, "Characterizing traffic flows originating from large-scale video sharing services," in *Proc. of. Int. Workshop on Traffic Monitoring and Analysis (TMA)*, 2010, pp. 17–31.

[68] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. IEEE INFOCOM*, vol. 2, Mar 1996, pp. 594–602.

[69] K. Calvert, M. Doar, and E. Zegura, "Modeling Internet topology," *IEEE Communications Magazine*, vol. 35, no. 6, pp. 160–163, June 1997.

[70] "Mininet Overview," http://mininet.org/overview/, accessed: 2016-01-20.

[71] D. Palma, J. Goncalves, B. Sousa, L. Cordeiro, P. Simoes, S. Sharma, and D. Staessens, "The QueuePusher: Enabling queue management in OpenFlow," in *Third European Workshop on Software Defined Networks (EWSDN)*. IEEE, 2014, pp. 125–126.

[72] G. D. Corporation, "General Algebraic Modeling System (GAMS)," http://www.gams.com.

[73] C. Bouras and K. Stamos, "An efficient architecture for bandwidth brokers in DiffServ networks," *Int. Journal of Network Management*, vol. 18, no. 1, pp. 27–46, Jan./Feb. 2008.

[74] N. L. Sauze (ed.), "ETICS Whitepaper," https://www.ict-etics.eu/news/view/article/etics-final-white-paper.html, 2013, accessed: 2018-02-17.

[75] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "SDX: A software defined internet exchange," in *ACM SIGCOMM, Chicago, IL.*, Aug. 2014.

[76] A. Gupta, R. MacDavid, R. Birkner, M. Canini, N. Feamster, J. Rexford, and L. Vanbever, "iSDX: An industrial-scale software defined internet exchange point," in *USENIX NSDI, Santa Clara, CA.*, March 2016.

[77] V. Kotronis, A. Gamperli, and X. Dimitropoulos, "Routing centralization across domains via sdn: A model and emulation framework for bgp evolution," *Computer Networks*, vol. 92, pp. 227–239, 2015.

[78] V. Kotronis, R. Klti, M. Rost, P. Georgopoulos, B. Ager, S. Schmid, and X. Dimitropoulos, "Stitching inter-domain paths over ixps."

[79] G. Biczk, M. Dramitinos, L. Toka, P. Heegaard, and H. Lonsethagen, "Manufactured by software: Sdn-enabled multi-operator composite services with the 5g exchange," vol. 55, pp. 80–86, 04 2017.

[80] H. Pouyllau, R. Douville, N. B. Djarallah, and N. L. Sauze, "Economic and technical propositions for inter-domain services," *Bell Labs Technical Journal*, vol. 14, no. 1, pp. 185–202, 2009. [Online]. Available: https://doi.org/10.1002/bltj.20362

[81] H. Pouyllau and R. Douville, "End-to-end qos negotiation in network federations," in *IEEE/IFIP Network Operations and Management Symposium (NOMS), Osaka, Japan*, April 2010.

[82] N. B. Djarallah, H. Pouyllau, S. Lahoud, and B. Cousin, "Multi-constrained path computation for inter-domain QoS-capable services," *Int. Jour. CNDS*, vol. 12, no. 4, pp. 420–441, 2014.

[83] N. B. Djarallah, N. L. Sauze, H. Pouyllau, S. Lahoud, and B. Cousin, "Distributed E2E QoS-based path computation algorithm over multiple inter-domain routes," in *IEEE Int. Conf. on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 2011, pp. 169–176.

[84] T. Groleat and H. Pouyllau, "Distributed learning algorithms for inter-NSP SLA negotiation management," *IEEE Trans. on Network and Service Management*, vol. 9, no. 4, pp. 433–445, Dec. 2012.

[85] F. M. Matos, A. V. Matos, P. Simoes, and E. Monteiro, "Provisioning of inter-domain QoS-aware services," *Jour. of Computer Science and Technology*, vol. 30, no. 2, pp. 404–420, March 2015.

[86] S. B. Rakas and M. Stojanovic, "A policy-based approach to E2E service negotiation via third party agent," in *24th Telecommunications Forum (TELFOR), Belgrade, Serbia*, Nov. 2016, pp. 929–932.

[87] G. Blocq and A. Orda, "How good is bargained routing?, volume=24, number=6, pages=3493–3507, year=2016, month=Dec." *IEEE/ACM Transactions on Networking*.

[88] V. Kotronis, X. Dimitropoulos, R. Klöti, B. Ager, P. Georgopoulos, and S. Schmid, "Control Exchange Points: Providing QoS-enabled end-to-end services via SDN-based Inter-domain routing orchestration," in *Open Networking Summit (ONS*. Santa Clara, CA: USENIX, 2014. [Online]. Available: https://www.usenix.org/conference/ons2014/technical-sessions/presentation/kotronis

[89] S. Civanlar, E. Lokman, B. Kaytaz, and A. M. Tekalp, "Distributed management of service-enabled flow-paths across multiple SDN domains," in *Networks and Communications (EuCNC), 2015 European Conference on*. IEEE, 2015, pp. 360–364.

[90] K. Phemius, M. Bouet, and J. Leguay, "Disco: Distributed multi-domain SDN controllers," in *IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–4.

[91] "Open Networking Foundation," https://www.opennetworking.org/.

[92] W. Fan, J. Wang, and W. Ip, "Multi-leg seat inventory control based on EMSU and virtual bucket," *Int. Jour. of Engineering Business Management*, vol. 6, pp. 9–13, 2014.

[93] "Mininet Cluster Edition prototype," https://github.com/mininet/mininet/wiki/Cluster-Edition-Prototype.

[94] T. Mori, R. Kawahara, H. Hasegawa, and S. Shimogawa, "Characterizing traffic flows originating from large-scale video sharing services," in *Int. Conf. on Traffic Monitoring and Analysis, Zurich*, April 2010.