Managed Video Services over Multi-Domain Software Defined Networks

K. Tolga Bagci and A. Murat Tekalp
College of Engineering, Koç University, 34450 Sariyer, Istanbul, Turkey
Email: {kbagci, mtekalp}@ku.edu.tr

***Abstract*—We introduce a framework for provisioning end-to-end (E2E) managed video services over a multi-domain SDN, where different domains may be operated by different network providers. The proposed framework enables efficient dynamic management of network resources for network providers and ability to request the desired level of quality of experience (QoE) for end users. In the proposed fully-distributed E2E service framework, controllers of different domains negotiate with each other for the service level parameters of specific flows. The main contributions of this paper are a framework to provide E2E video services over multi-domain SDN, where functions that manage E2E services can collaborate with functions that manage network resources of respective domains, and a procedure for optimization of service parameters within each domain. The proposed framework and procedure have been verified over a newly developed large-scale multi-domain SDN emulation environment.**

I. Introduction

Supporting traffic differentiation and specialized multimedia services with different service-levels driven by end-user preferences and choice, in addition to the open (best-effort) Internet, will be a valuable feature of future networks.

Over the years, there have been many traffic engineering (TE) proposals to improve network performance and provide quality of service (QoS) over classical IP networks, including integrated services (IntServ), differentiated services (DiffServ), multi-label packet switching (MPLS) [1], application layer traffic optimization (ALTO), and path computation element (PCE) [2] over a single operator network. In order to negotiate service level agreements (SLA) between different network domains, bandwidth broker [3] and other architectures have been proposed [4]. Constrained path computation, a key component of traffic engineering and QoS provisioning, determines the best path that each traffic type should follow given the network state, and provides the route for each label switched path (LSP) that is set up. Typically, these computations are done at the head end of each LSP. PCE aims to separate route computations from signaling of end-to-end (E2E) connections and from actual packet forwarding. However, the plethora of network protocols to be supported and amount of computation needed at each router to enable traffic engineering make current Internet routers too heavy and too expensive for QoS provisioning.

As a promising alternative, OpenFlow represents a vision of software-defined networks (SDN) where network control functions are separated from actual packet forwarding. SDN has made crucial impact on data center networks by allowing automatic network reconfiguration everytime a virtual server has been moved to a different physical machine. It is expected to make a similar impact for service provider networks in the form of multi-domain SDN or software-defined wide-area networks (SD-WAN). A standard SDN controller has visibility of all network resources within a domain which makes traffic engineering within the domain practical. In multi-domain SDN, controllers of different domains need to communicate and negotiate with each other about the service level parameters of a specific service request [5], [6]. In our previous work, we proposed a fully-distributed multi-domain SDN architecture [7] and extended standard SDN controller by adding inter-controller communication and service-level negotiation functions [6], [8]. These functionalities enable dynamic E2E service-level negotiation and traffic engineering over a multi-domain SDN in a scalable manner. The main advantage of implementing traffic engineering and E2E quality of service/experience management using OpenFlow/SDN framework rather than the MPLS fabric is scalability, since in the OpenFlow/SDN framework signalling for flow (label) management needs to be carried out only between controllers rather than all routers along a path.

This paper describes a framework to provide E2E video services over multi-domain SDN and a procedure for optimization of service parameters within each domain. We briefly summarize the distributed multi-domain SDN architecture in Section II. We introduce the proposed E2E video service framework in Section III. We present a multi-domain SDN emulation environment to validate our framework and experimental results in Section IV and conclusions in Section V.

II. A Distributed Multi-Domain SDN Architecture

This section presents a general framework for enabling dynamic, E2E service-level negotiation over a distributed multi-domain SDN. In the proposed framework, each domain controller has complete control of its own intra-domain routing, while it communicates and negotiates with other domain controllers for inter-domain routing with the desired service-level. The proposed inter-domain E2E service-level management model is dynamic. In case the negotiated service-level cannot be fulfilled anymore due to new service requests or link failures, the service parameters are re-negotiated and E2E paths are recomputed in real-time. In order to realize this vision, we propose to extend a standard SDN controller that traditionally manages a single domain with i) multi-domain

978-0-9928-6265-7/16/$31.00

controller-to-controller (C2C) communication extensions and ii) E2E service-level management extensions, which are depicted in Fig.1 in green and blue boxes, respectively. We have implemented these extensions on top of Floodlight [9] controller. We elaborate on these extended functionalities in the following.

A. 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 as it is done today. To this effect, we have proposed reactive and proactive discovery processes [8].

C2C Messaging Manager performs LLDP-like mechanism for discovering neighbor domain controllers and border gateways. Information about the non-neighbor domain controllers are received through Topology and Link Information Sharing messages sent by the controllers of the neighbor domains. The information collected in the discovery process is stored in Controller Information Base (CIB) of each controller where each entry contains controller ID, border gateway switch DPID to access the neighbor domain and status of the controller.

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

B. Data-Plane Monitoring

Each domain controller monitors its own data-plane network and calculates network parameters through a Monitoring Manager (extension to core controller modules). Link bandwidths in use are estimated via per-port/per-queue statistics requested from the switches using periodic *StatisticsRequest* and *StatisticsReply* messages provided by the OpenFlow protocol [10]. The durations between request and reply messages between controller and switches are used for estimating the control-plane delay. We use the method proposed in [11] for measuring the delay at each link between switches and inter-domain links between border gateways of peering domains. Delay variations on these links are also calculated by differences of estimated delays.

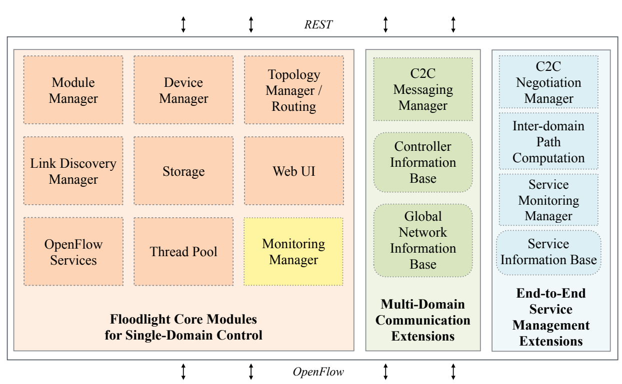

Fig. 1: SDN controller with multi-domain communication and end-to-end service management capabilities

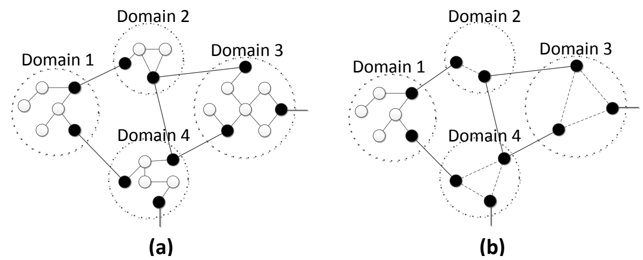

Fig. 2: Sample multi-domain SDN network topology: (a) Complete network topology, (b) Global network topology as seen by controller of domain 1

C. Topology and Link Information Sharing

Each domain controller has full view of the topology for its own domain, but does not have access to the topology of other domains. Sharing some topology information between domain controllers is essential for inter-domain flow management. Only aggregated information will be shared for confidentiality/security reasons as operators wish to hide their physical network topology as well as to minimize inter-domain messaging overhead. Fig.2 illustrates a multi-domain SDN with 4 domains. The complete network topology is presented in Fig.2a, while aggregated model as seen by a particular domain controller is presented in Fig.2b. The unfilled and filled dots stand for intra-domain switches and border gateways, respectively. Links connecting two border gateways are called inter-domain links, while all other links are called intra-domain links. The original network is aggregated by replacing intra-domain links by a set of virtual links between border gateways that are the end points of inter-domain links. The monitored network resource information is also aggregated. We estimate the parameters for virtual links by solving least cost path problem minimizing total delay and hop count. Obviously, network and link information aggregation introduces some imprecision on the global network state information but this is necessary for confidentiality/security and tolerable to compute initial E2E route candidates as described in Section III. After a controller is discovered via the controller discovery process, a *PacketOut* message containing the aggregated topology and virtual link information is sent to the related border gateway which forwards the message to the discovered controller. All controllers share aggregated network information, such that each can store a global network map with parameters, such as available link capacities and delays, in a Global Network Information Base (GNIB) together with Domain Information Base (DIB) that contains full topology and resources for its own domain. Controllers periodically syncs their GNIB to keep the global map current.

III. End-to-End Video Service Framework

This section proposes a dynamic E2E video flow management framework based on the distributed multi-domain SDN architecture introduced in Section II. We support three levels of service: i) assured quality (resource reservation), ii) best-effort-plus (BE+), and iii) best-effort (BE). When a user (USR) requests a video service from a video content provider (VCP), it triggers the following procedure, which is summarized in Fig.3: 1) The controller of the network provider that USR receives service from (C_1 in Fig.3) prompts the USR with the desired service-level options. 2) The VCP specifies the QoS parameters (e.g., minimum/maximum bitrate and delay) for the service requested, which differ for UHD or HD video streaming or Real Time Communication (RTC) service (in Fig.3, this QoS specification message first goes to C_3, and then C_3 to C_1). 3) The controller of the USR's domain decides on the traffic class, and initiates a negotiation process for improved service levels. No negotiation is conducted for best effort services. The controller of the USR's (source of the request) domain: i) initially computes a number of candidate paths from border gateways of the VSP's domain to its domain based on its current aggregated global network map stored in GNIB, ii) sends messages to controllers along the candidate paths to request service bids, iii) compares received bids (available service parameters and price), calculates the optimum virtual path fixing only entry and exit border gateways for each domain, and notifies the controllers along the chosen path. 4) Controllers of each domain along the chosen path finally decides for the actual physical routes to be followed in their respective domains given the entry and exit border gateways. The final physical route is obtained by concatenation of the routes provided by respective domain controllers. 5) The Service Monitoring Module in the controller of the USR's domain monitors whether the agreed service parameters are satisfied for each service. A re-negotiation process is initiated if the service agreement cannot be fulfilled by one of the domains at any time. We now discuss the details of these steps.

A. Inter-Domain Path Calculation

Given the aggregated global network map, stored in GNIB, with costs, e.g., delay, of virtual links, the controller in the source domain decides for a short list of best inter-domain E2E paths. This problem can be posed as a Constrained Least Cost (CLC) problem. An aggregate global network model is presented by a simple graph $G_g^a(N_g^a, A_g^a)$, where N_g^a is the set of border gateways and A_g^a is the set of all virtual links connecting the border gateways, so that link (i, j) is an ordered pair, outgoing from node i and incoming to node j. Let $R_v(s,t)$ denote all virtual paths (subsets of A_g^a) from source node s to a border node t in the destination domain. For any E2E virtual path $r \in R_v(s,t)$, we define the total cost $f_C(r)$ and E2E constraint $f_E(r)$ using suitable measures. The CLC problem aims to find

$$r_a^* = \arg\min_r \{f_c(r) | r \in R_v(s,t), f_E(r) \leq E_{max}\}$$

i.e., a path r_a^* over the aggregated graph that minimizes the cost $f_C(r)$ subject to constraint $f_E(r)$ to be less than or equal to E_{max}. We use link delays as cost $f_C(r)$ and a certain value for E2E delay variation as constraint E_{max}. We solve multiple instances of this problem, each time randomly removing some links on previously calculated inter-domain paths of the global network, to find alternate inter-domain path candidates.

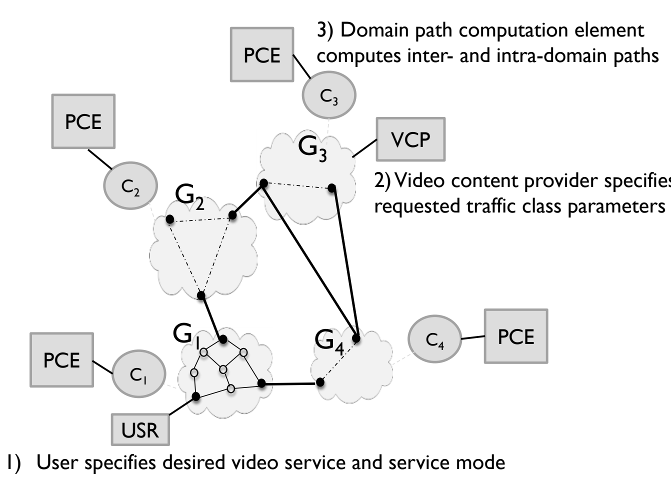

Fig. 3: E2E video service setup between user (USR) and video content provider (VCP).

B. Inter-Controller SLA Negotiation

Once inter-domain path candidates are determined, negotiation for the requested service with domain controllers along each candidate path is performed. The inter-controller SLA negotiation is a recursive messaging process. If controller for domain A wishes to send messages to controllers of domains B, C, and D for a desired inter-domain route A-B-C-D, then controller A sends a message to only controller B. If controller B cannot respond positively, then it sends a negative reply to controller A and no further messages are exchanged. Otherwise, controller B sends a request message to next domain controller C. The messaging process continues until the message reaches the destination controller D. If the destination controller replies positively, then positive reply messages back track from D to C, C to B, and B to A; hence all controllers along the path have accepted a particular SLA. As response messages backtrack from destination domain to source domain, each controller adds its own response field to the message. If an agreement cannot be reached for the first candidate path, then the process is repeated for the second candidate path. The proposed recursive messaging scheme is efficient in terms of total messages exchanged between controllers to reach an SLA agreement. Information for each requested or approved services (e.g., service-level, requested parameters/constraints, service id, etc.) are stored in Service Information Base (SIB) and used for optimizing routes dynamically.

C. Optimization of Service Parameters within a Domain

Once an agreement on the requested service parameters is reached, controllers of each domain along the chosen path are notified, and they allocate resources within their domain considering the service related constraints (e.g., minimum bandwidth). Domains containing server or client calculate path between server/client node and their border gateway nodes, while transit domains compute path between entry and exit border gateways.

Resource allocation procedure should consider already existing video service flows when deciding the route for new incoming requests. Consequently, the intra-domain path computation unit performs resource allocation process for all switches within its own domain for each service flow to optimize path computation subject to QoS constraints. As OpenFlow support queuing actions, queue-based QoS optimization approaches are possible. In our recent work [12], we provide queue allocation optimization method for adaptive video streaming for multiple users with multiple service-levels within a single domain, where switch ports are configured with a fixed number of queues at particular capacities. In our optimization model, service-levels are associated with different per-bandwidth subscription plans for its customers and ISP tries to maximize its revenue by satisfying video parameter constraints of service requesting clients. Intra-domain path computation unit solves the optimization problem, and decides a bandwidth allocated over a sequence of queues forming a path within its domain. The final route is obtained by stitching the final paths computed by each domain controller.

D. Service Monitoring and Re-Negotiation

Each domain controller has a Service Monitoring Manager module that periodically tracks its active services in the SIB to check whether negotiated SLAs are delivered within some tolerance limits. For each service, it keeps throughput and delay statistics and compares them with the agreed service parameters. There exists two potential reasons for an agreed SLA may not be fulfilled: 1) Although assured quality services are subject to admission control and guaranteed by resource reservation, there is no admission control for intermediate quality services and intermediate level SLA may not be fulfilled during periods of increased demand. 2) There may be a link break down in one of the domains; hence, even assured quality SLAs cannot be met. When either of these conditions are detected, a re-negotiation process is initiated.

IV. System Verification and Evaluation

A. Emulation Environment

We verify the proposed multi-domain service-level aware managed video services framework over a new large-scale multi-domain SDN emulation environment that we developed.

We adopt the well-known transit-stub (domain) topology model [13], which supports a hierarchy that is similar to real inter-networks. We assume each domain has a single controller. Each operator network is composed of backbone switches and stub domains connected to them. Stub domains, which are leaves of the backbone network, represent access networks or enterprise/local area networks such as campus networks. We assume that each stub domain has a single gateway switch. Since location has significant importance in direct connectivity of switches, we use exponential random distribution model (according to geographical distance) to determine connectivity among switches [13]. The bandwidth and delay of the links between switches are determined taking the hierarchical structure of transit-stub topology model into account where inter-domain link delay and queue capacities is up to 10 times greater than that of intra-domain. Once the multi-domain topology model is generated, resulting links and switches are created using Mininet Cluster edition 2.2.0 [14] allowing us to distribute nodes and links between several remote servers. Smaller model is depicted in Fig.4. Note that intra-domain links are virtual Ethernet links, whereas the inter-domain links are SSH links.

Background TCP traffic is emulated with iPerf [15] where particular Mininet hosts are running specific scripts based on its designated role in the network e.g., server or client. Each client receives data at a particular bitrate (assumed to have Poisson distribution with 4 Mbps mean) and duration (uniformly distributed between 20 and 40 seconds) from a server that is chosen randomly among all servers. There exists a sleeping duration (uniformly distributed between 5 and 10 seconds) for each client between consecutive connections to another server.

We consider the delay between control and data plane in an emulation environment where controller(s) and switches reside in a common physical machine or in the same local area network. Therefore, delay (with normal distribution with 50 ms mean and 5 ms variance) between controller and switches is emulated using NetEm [16] by applying a delay to loopback Ethernet interfaces of each machine which carries controller and switches of particular domains.

B. Results

In the experimental scenario, 2 DASH clients, which are located in Domain 1 in Fig.4, with BE and BE+ service-levels are requesting segments of an HD video content (TearsOfSteel encoded at 4 adaptation levels ranging from 3 Mbps to 10 Mbps) from the host running an HTTP server located in Domain 6.

Based on the inter-domain path calculations and negotiations, video flows of the BE client and BE+ client initially pass through Domain 4 and Domain 3, respectively. As we intentionally increase the congestion in Domain 3 around 210$^{\text{th}}$ second, Domain 1 controller re-negotiates the service of BE+ client and signals the new inter-domain path passing over Domain 4 which is nearly congested. Fig.5 shows the change of inter-domain path for BE+ client. Within Domain 4, we note that BE+ client receives higher quality video segments compared to BE client as it has better service-level. The variations in the received throughput are due to client side adaptation implementation.

V. Conclusions

From a service provider perspective, the proposed service framework enables efficient and flexible management of resources over a multi-domain service provider network. Standard SDN applications perform network functions such as flow routing, QoS provisioning, load balancing, security policy enforcement within a single domain. The proposed multi-domain extensions and service-level negotiation between controllers of its sub-networks will make these functionality available over its entire network.

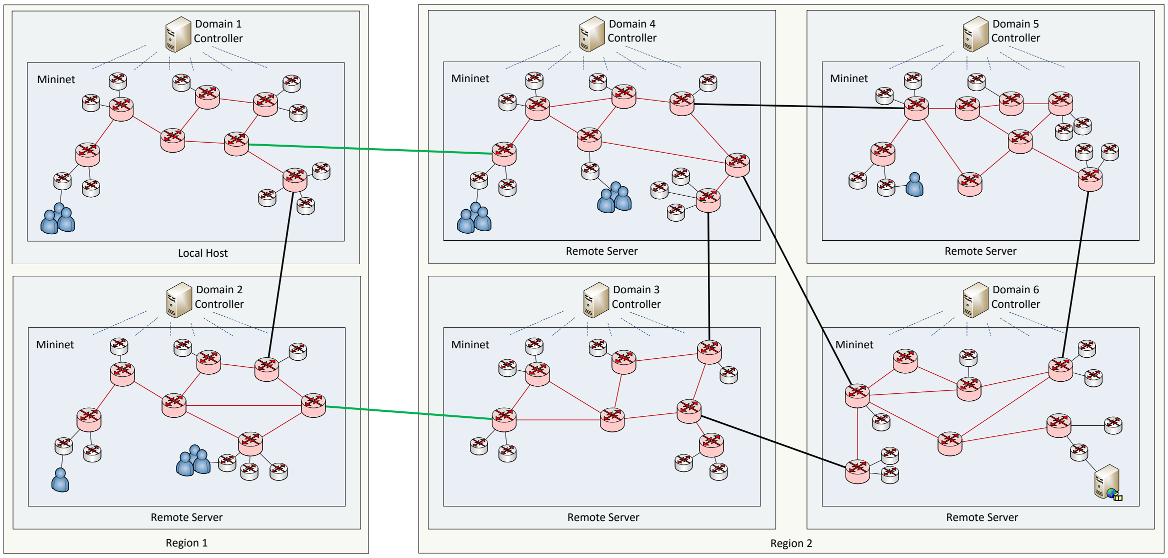

Fig. 4: Exemplary emulation environment with six SDN domains over two regions

From a content-provider or end-user perspective, the proposed service 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 authorities.

Our Mininet Cluster based emulation environment enables us to conduct large-scale tests by distributing controllers, nodes and links over several remote servers. The emulation results show that i) the system is able to perform Controller-to-Controller communication, which is an essential element of a fully-distributed multi-domain SDN architecture, ii) the system is highly scalable and allows fast provisioning of new service requests with E2E QoS across multiple domains, and iii) the system is able to re-route the inter-domain traffic dynamically in case negotiated SLA cannot be delivered.

In the future, we foresee multi-domain wide area SDN with multiple internet service providers, and content-providers, each managing their own SDN domain, but cooperate with each other for E2E path calculations and routing decisions when crossing domains. Furthermore, specialized services with multiple E2E service levels will be enabled on such multi-domain wide area SDN. This paper shows the feasibility of these concepts.

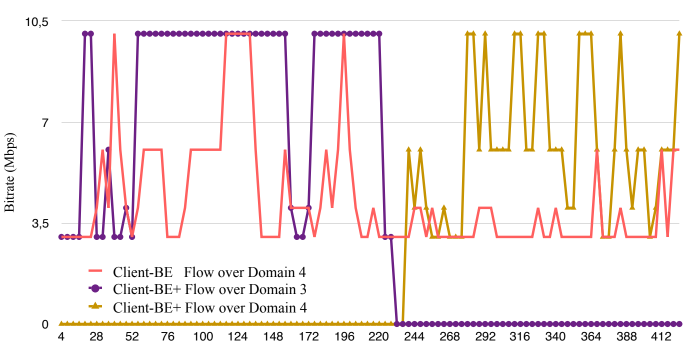

Fig. 5: Bitrates for received video at different MPEG-DASH clients with different service-levels.

ACKNOWLEDGMENT

This work has been funded by TUBITAK Project 113E254 and 115E299. A. Murat Tekalp also acknowledges support from Turkish Academy of Sciences (TUBA).

REFERENCES

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

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

[3] 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, 2008.

[4] T. Groléat and H. Pouyllau, "Distributed inter-domain sla negotiation using reinforcement learning," in *IFIP/IEEE Int. Symposium on Integrated Network Management (IM)*, 2011, pp. 33–40.

[5] 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.

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

[7] H. E. Egilmez *et al.*, "Distributed QoS architectures for multimedia streaming over software defined networks," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1597–1609, 2014.

[8] 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 submitted.

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

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

[11] K. Phemius and M. Bouet, "Monitoring latency with openflow," in *IEEE Int. Conf. on Network and Service Man. (CNSM)*, 2013, pp. 122–125.

[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 Int. Conf. on Image Processing (ICIP)*, 2016 submitted.

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

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

[15] "iPerf – The network bandwidth measurement tool," https://iperf.fr/.

[16] "NetEm," http://www.linuxfoundation.org/collaborate/workgroups/networking/netem/, accessed: 2016-02-23.