Enhancing the Quality Level Support for Real-time Multimedia Applications in Software-Defined Networks

Francesco Ongaro†, Eduardo Cerqueira*, Luca Foschini†, Antonio Corradi†, and Mario Gerla‡
† Department of Computer Science and Engineering, University of Bologna, Bologna, Italy
francesco.ongaro@studio.unibo.it, {antonio.corradi, luca.foschini}@unibo.it
* Institute of Technology, Federal University of Pará, Belém, Brazil
cerqueira@ufpa.br
‡ Department of Computer Science, University of California Los Angeles, Los Angeles, USA
gerla@cs.ucla.edu

***Abstract*—Nowadays, the explosive growth of real-time applications that need stringent Quality of Service (QoS) and Quality of Experience (QoE) support, forces network programmers to design network protocols that deliver specified performance guarantees. This paper exploits the use of Software-Defined Networking (SDN) in conjunction with the OpenFlow protocol to differentiate network services with quality level assurance and to respect agreed Service Level Agreements. Initially, we define a Management and Orchestration architecture that allows us to manage the network in a modular way. Then, we provide a seamless integration of the proposed architecture and the SDN standard following the separation between the control and data planes. Finally, we give an Integer Linear Programming formulation of the problem of enhancing QoS and QoE in SDNs in terms of packet loss and delay, taking into account the network constraints and the requirements of real-time applications, i.e., maximum acceptable packet loss and delay rates. Given the optimal solution of the problem, we evaluate the impact and benefits of the proposed scheme by means of the Mininet network emulator.**

***Index Terms*—Software-Defined Networking; OpenFlow; Wireless and Wired Networks; Quality of Service; Quality of Experience; Emulator; Multi-Commodity Flow;**

I. Introduction

In the last few years, there has been a continuous evolution of network services and applications. Unfortunately, the network infrastructure system has been maintained almost in the same shape for decades according to the phenomenon known as ”Internet ossification”. The Software-Defined Networking (SDN) [1] paradigm is one of the best and most attractive solution for enhancing the Internet with more flexibility and adaptability.

SDN allows a logically centralized software program to control the behavior of an entire network by decoupling the routing decision tier from the forwarding layer. The OpenFlow protocol [2] makes the communication between the control plane and the data plane possible. Thus, it allows us to write high-level control programs that specify the behavior of the network components and that can take care of various networking tasks, including resource management procedures.

One of the main important networking problem is the lack of efficient resource management schemes to provide Quality of Service (QoS) and Quality of Experience (QoE) support for real-time applications, especially in networks affected by packet loss and delay. For example, the Voice over IP (VoIP) and the Interactive-Video services are not tolerant to packet loss that should be no more than 1%, especially if the services use compressed codec. The latency is also very important and it should be no more than $150ms$ between two end-points, i.e., one-way latency [3].

The paper addresses the issues above proposing a solution that shows several novel aspects. Firstly, we define a modular and extensible QoS architecture providing a seamless integration between our solution and the standard SDN paradigm. Secondly, by exploiting OpenFlow in SDNs, we show how our proposal can be used for managing differentiated network services for multimedia applications with quality level support in wired and wireless environments.

Since we are also interested in dealing with multi-commodity flows, we present in Section III an Integer Linear Programming (ILP) formulation that considers the well-known Multi-Commodity Flow Problem in conjunction with the Constrained Shortest Path. Specifically, the ILP finds the shortest path between source and destination taking into account both network constraints and service requirements in terms of packet loss and delay for guaranteeing a QoS in SDNs. Moreover, the QoS architecture gives us the possibility to map the optimal solution provided by the mathematical model to various levels of QoS based on a well-known QoE metric called Mean Opinion Score (MOS) [4].

To make the tests possible, we define a multiple path topology composed of a wired and wireless network with also real mobile devices in order to be as close as possible to a real environment. We also test our proposal putting into the simulation different values of packet loss evaluating the impact of that on the user perception (QoE).

Performance evaluation results, collected from the Mininet network emulator [5], confirm the benefits of our resource management architecture in providing QoS/QoE support for differentiated services in SDNs compared to the traditional solution. The simulations also remark that by means of our QoS architecture, it is possible to handle the network con-

978-1-4799-6959-3/15/$31.00 ©2015 IEEE

gestion effects providing guarantees to preserve throughput as much as possible.

The remainder of the paper is structured as follows. Sections II and III present our QoS Management and Orchestration architecture and the mathematical optimization model. The evaluation scenario and the results are given in the Section IV. Then, the Section V reviews some of the related work. The conclusion are left for the last Section VI.

II. Enhanced QoS Architecture

Our proposed architecture, indicated with a red dashed line in Figure 1, aims to reach an enhanced QoS and QoE in SDNs by means of several functionalities. Our solution provides the wired and wireless SDN environments with QoS and QoE support, by calculating the best path for video streaming and file transfer services according to both the networks constraints and the application requirements. This goal is achieved by continuously monitoring the network status and, incidentally, by exploiting our mathematical optimization model, explained in Section III.

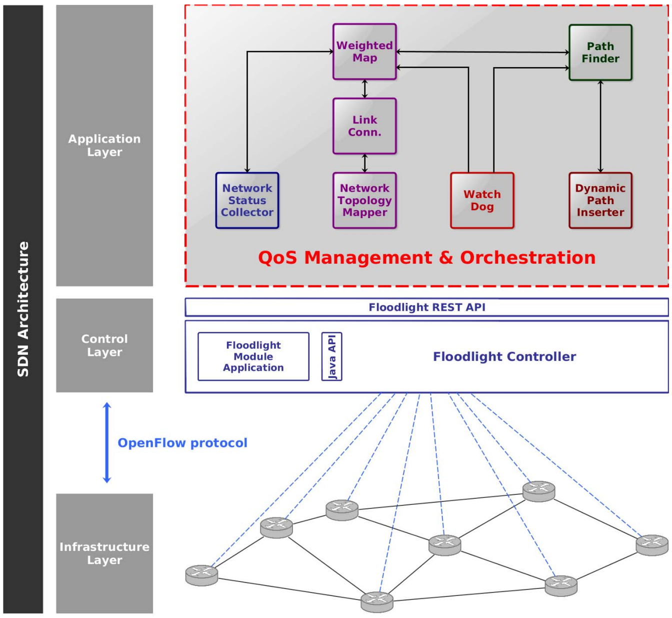


Fig. 1: QoS Management & Orchestration architecture.

We are assuming Floodlight as our SDN controller, but it can be changed for other controllers, such as NOX. The architecture is modular and new modules and functionalities can be added. Specifically, the architecture for enhancing the quality level of real-time multimedia transmission in SDNs is composed of the following logical modules:

a. The ***Network Topology Mapper*** frequently gets information about the network topology, i.e., links and nodes, and maps it into the structure called *Link Connection*.
b. The ***Network Status Collector*** continuously monitors and retrieves information about the network status, e.g., the available bandwidth for each link and it stores them in the *Weighted Map* structure.
c. The ***Dynamic Path Inserter*** injects the flow entries into the switches, through the Floodlight REST API, to set the best path found by the *Path Finder* module. The module uses also the *Link Connection* for getting the ports of the network devices involved in the path. Moreover, the *Dynamic Path Inserter* allows us to configure the paths according to some additional external decisions, e.g., forcing specific paths in such a way to avoid some switches.
d. The ***Path Finder*** implements the mathematical model detailed in Section III. The module is designated to find the best path according to both the application requirements and the network status stored in the *Weighted Map*.
e. The ***Watch Dog*** module is responsible for triggering of the path changing scheme. It continuously analyzes the network information collected by the *Network Status Collector* to decide when path changing is necessary. In case of triggering, the *Watch Dog* directly exploits the *Path Finder* to find the best path and, consequently, it uses the *Dynamic Path Inserter* for putting the flow entries into the switches involved in the new path.
f. The ***QoS Management & Orchestration*** is the "wrapper" module and it uses all the modules for our integrated network management. Specifically, it dynamically analyzes the network status, decides when it is necessary to redefine a new path for a specific flow, finds which is the best path for each flow, and inserts the rules into the switches in the form of flow entries. Moreover, exploiting both the *Dynamic Path Inserter* and the *Path Finder*, we are able to achieve the "make-before-break" approach and, incidentally, offer the best QoS as possible.

III. Multi-Commodity Flow and Constrained Shortest Path Model

The proposed mathematical model, called Multi-Commodity Flow and Constrained Shortest Path (MCFCSP), is suitable to be used in wired and wireless networking scenarios to exploit all potentially available resources. It takes advantage of two well-known problems derived from Operations Research: the Multi-Commodity Flow Problem (MFP) and the Constrained Shortest Path (CSP). Thus, our model allows us to find the shortest path, for each service, according to the given set of constraints. We define the MCFCSP problem by using A Mathematical Programming Language (AMPL) [6] and we propose to use the IBM CPLEX solver [7] to solve the ILP problem.

We assume a scenario with a network of interconnected nodes where each link has both a capacity, i.e., available bandwidth, and a cost that is computed as the sum of the delay and packet loss. We also take into account multiple flows, related to different services, that should be sent from various sources to distinct destinations.

Our goal is to find the optimal set of routes through the network, for each commodity, with the minimum flow cost subject to some constraints. Specifically, the constraints are the available bandwidth on the links and the maximum acceptable delay and packet loss depending on the service type requirements.

Thus, we define our formal notation for the MCFCSP model. The network is represented by an oriented graph

$G = (N, A)$, where N is the set of nodes and A is the set of arcs between each pair of nodes. The arcs E are bi-directional and they have associated the available bandwidth b_{ij}, the delay d_{ij}, the packet loss $p_{i,j}$, and the cost per unit of flow c_{ij}. Specifically, we compute the cost c_{ij} as follows:

$$c_{ij} = \alpha\, d_{ij} \;+\; \beta\, p_{ij} \quad \forall (i,j) \in A$$

Where α and β are the scale factors and allows us to weight the cost according to the importance of the delay and packet loss for a particular flow. Thus, we can manage these parameters according to the requirements of the type of service.

Now, we provide the ILP formulation for the MCFCSP problem. The optimization objective is to route all the flows in the network along the shortest path, with the minimum cost. The set of different traffic flows to be routed on the graph is represented by k.

Sets:

- Nodes: $n \in N$
- Arcs: $(i,j) \in A$
- Edges: $(i,j) \in A \;\cup\; (j,i) \in A$

Variables:

- $x^k_{ij} \geq 0$: amount of the flow corresponding to the service k routed on the link (i,j).

Parameters:

- $b_{ij} \geq 0$: available bandwidth on the link (i,j);
- $c_{ij} \geq 0$: cost of the link (i,j), computed as $\alpha d_{ij} + \beta p_{ij}$;
- $\alpha \geq 0$: scale factor for the delay;
- $\beta \geq 0$: scale factor for the packet loss;
- $s_k \in N$: source of the flow k;
- $t_k \in N$: destination of the flow k;
- $f_k \geq 0$: amount of the flow k to be sent from s_k to t_k;
- $P^k_{max} \geq 0$: maximum acceptable packet loss;
- $p_{ij} \geq 0$: packet loss on the link (i,j);
- $D^k_{max} \geq 0$: maximum acceptable delay;
- $d_{ij} \geq 0$: delay on the link (i,j);
- $B^k \geq 0$: bandwidth required by the service k;

Objective Function:

$$\min: \sum_{(i,j)\in A} \sum_{k \in K} c_{ij} x^k_{ij} \tag{1}$$

Constraints:

$$\sum_{(i,j)\in A} x^k_{ij} \;-\; \sum_{(j,i)\in A} x^k_{ji} = \begin{cases} f_k & \text{if } i = s_k, \\ -f_k & \text{if } i = t_k, \\ 0 & \text{if } i \neq s_k, t_k \end{cases} \quad \forall i \in N, \forall k \in K \tag{2}$$

$$\sum_{(i,j)\in A} p_{ij} x^k_{ij} \;\leq\; P^k_{max} \quad \forall k \in K \tag{3}$$

$$\sum_{(i,j)\in A} d_{ij} x^k_{ij} \;\leq\; D^k_{max} \quad \forall k \in K \tag{4}$$

$$\sum_{k \in K} B^k x^k_{ij} \;\leq\; b_{ij} \quad \forall (i,j) \in A \tag{5}$$

$$x^k_{ij} \in \{0,1\} \quad \forall (i,j) \in A, \;\; \forall k \in K \tag{6}$$

The objective function, detailed in Equation 1, represents the cost minimization that depends on both the delay and packet loss of the used links. Equation 2 defines the well-known Flow Conservation Law for the flow balancing. The maximum acceptable values for the packet loss and the delay are defined in Equations 3 and 4, respectively. The parameters P^k_{max} and D^k_{max} impose a bound on the packet loss and delay for each service k. The arc capacity constraint is taken into account in Equation 5. It imposes a limit on the available bandwidth for each link considering all the k flows. Finally, Equation 6 defines the variable domain and guarantees that the decision variable is 0 or 1.

About the size of the problem, we notice that the number of variables is $|A||K|$ and the number of constraints is $|N||K| + |A| + |K|$. This problem is *NP-complete* [8], but we can solve it viably for a limited reasonably small network.

IV. Performance Evaluation

This section aims to show the benefits of our proposed solution in assuring QoS and QoE for application in wired and wireless networks, by using the Mininet network emulator. The scenario is composed of both an emulated and a real wired and wireless network topology, as depicted in Figure 2. On the one hand, inside Mininet, there is a video streaming server, a medical server (modeled by a file transfer server), four Loader servers (that run the *Iperf* tool), and six switches OpenFlow-enabled connected in a manner to ensure multiple paths. On the other hand, outside Mininet, there are two real mobile devices, such as smartphones and laptops, that are connected to the emulated network through a Wi-Fi router. Finally, the SDN controller and our QoS Management and Orchestration architecture, that are placed outside Mininet, control the entire network behavior.

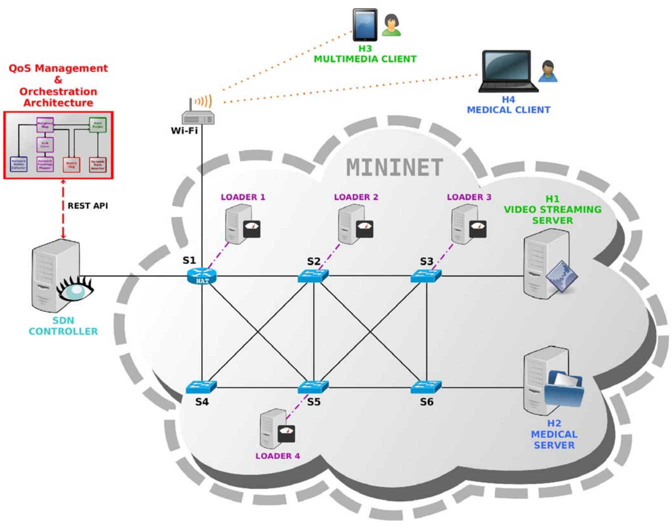


Fig. 2: Mininet hybrid topology.

Since the nodes inside Mininet have private addresses, to make the interaction with the outside network possible, it is necessary to realize a NAT in the switch $S1$, as presented in Figure 2. It is also mandatory to define a set of IPTable rules to forward the traffic from the public to the private network.

For the video streaming flow, on the server side, we use a *VLC media player* [9] as a video streaming server. For the video content, we use the *Big Buck Bunny* open movie [10], encoded in *ogg* format with an HD resolution of $1280x720$ pixel. On the client side, we use the *VLC media player* as a client to receive the network video stream. For the file transfer flow, on the server side, we use a simple FTP server [11] bound on port 21. On the client side, we use the *wget* Linux command to download a general file from the FTP server.

To measure the benefits of our proposal, we will analyze the network behavior during a link congestion to verify the need of a QoS management and the impact of that on the QoE by verifying the respective MOS. Specifically, it is important to monitor the status of our relevant services in terms of throughput, i.e., bandwidth usage.

Starting from the scenario illustrated in Figure 2, we overload the network during the execution of both the video streaming and file transfer services, by means of the Loader servers 1 and 2 that are respectively connected to the switches $S1$ and $S2$. On the one hand, the video streaming throughput is between a Lower Bound (LB) of $2Mbps$ and an Upper Bound (UB) of $3Mbps$ and these values depending on the video characteristics, e.g., resolution, format, and compression. On the other hand, we set the maximum throughput for the file transfer service to $4Mbps$. Figure 3a depicts the throughput trend related to the services.

When the network overloading starts, the video streaming falls below the critical threshold of $2Mbps$ (LB) approximately after 10 seconds, as shown in Figure 3a. Consequently, the video is stuck for lack of sufficient available bandwidth and it impacts negatively on the video streaming service. Since a file transfer generally does not require strict bandwidth constraints, such as the video streaming service, however, the total amount of time required for the transfer proportionally increases with the decreasing of the available bandwidth. Hence, if it was possible to analyze the network status, it would be feasible to define the network paths to avoid the link congestion and, consequently, provide strict bandwidth guarantees to the services.

Now, we report on the network behavior with our architecture to enhance the QoS. We start considering that without awareness of the network available bandwidth, the controller cannot know the overloaded links and, consequently, it is not able to find and deploy the best path. In addition, when it is necessary to deal with multi-commodity flows, it is of paramount importance to consider that the amount of flows can be more than one at the same time.

We set the flow bandwidth QoS requirements to $4Mbps$ and $3Mbps$ for the file transfer service and the video streaming, respectively. Moreover, we define a video streaming throughput threshold suitable for triggering the path changing. Specifically, considering the LB of $2Mbps$, we subtract the 10% to avoid borderline throughput values. Hence, the *Watch Dog* module will trigger the changing of the path when the throughput falls below the critical threshold of $1.8Mbps$.

Similarly to the situation in Figure 3a, we test our QoS architecture to check the throughput of the services during a network congestion, as in Figure 3b. The video streaming throughput does not significantly suffer from the network congestion and it falls below the critical threshold only for a short time, as illustrated in Figure 3b. In fact, the video player buffer can overcome this short lack of bandwidth and the result is a good video quality with a very short block or no block at all.

Comparing Figures 3a and 3b, respectively without and with our QoS architecture, it is clear that our solution can maintain the video streaming throughput higher enough not to present any service discontinuity. Since the video streaming service needs a minimum amount of available bandwidth, the network without a resource management mechanism is not able to guarantee the video streaming requirements and, consequently, a good quality.

We also test our proposal putting into the simulation different values of packet loss evaluating the impact of that on the user perception. Incidentally, for each packet loss percentage, we calculate the value of the mathematical model giving to it the input parameters about the network status. Then, the same procedure is repeated using different values of delay. Moreover, we test several combinations of both delay and packet loss to observe the impact of that on the QoE. By means of Equation 1 of the MCFCSP model described in Section III, we find a correlation between the cost given by the objective function and the MOS score levels, as detailed in Table I.

COST FUNCTION	MOS	QUALITY	IMPAIRMENT
< 40	5	Excellent	No block at all
40 - 55	4	Good	No block or sporadic short blocks
56 - 69	3	Fair	A couple of short blocks $(1-2$ s)
70 - 79	2	Poor	Several long blocks $(2-4$ s)
≥ 80	1	Bad	A lot of long blocks $(7-10$ s)

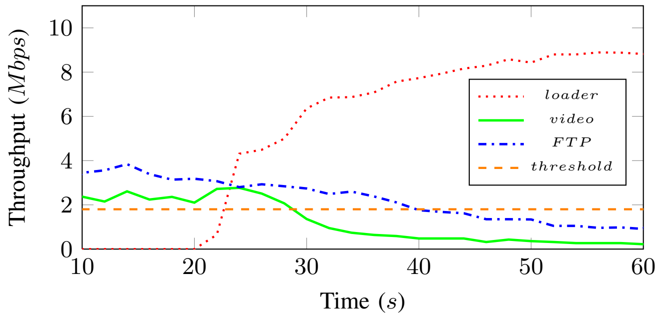
TABLE I: Conversion between our mathematical cost function and the MOS levels related to the video streaming.

Table I reports a connection between the value ranges given by the mathematical model and the MOS levels. For instance, it indicates that, in case of cost values ≥ 80, the QoE becomes very low. Additionally, when the cost function is between 100 and 150 (e.g., in case of 10% and 15% of packet loss, respectively), although it is possible to start the streaming, the video is completely blocked or it is blocked every $2-3$ seconds. The same situation occurs when the total delay is around $250-300ms$. Finally, with a total packet loss of about 18% (cost value ≈ 150), the video streaming does not start at all. The relationships in Table I are very important because they allow us to define a mapping between the mathematical basics and the QoS and QoE in our hybrid environment that can be followed by an adaptive automatable strategy.

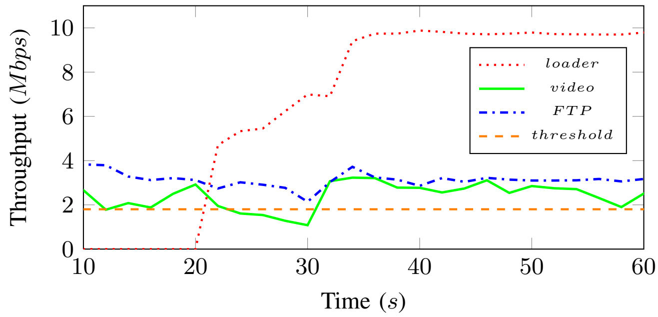
V. Related Work

Several works focus on the problem of network resource management and monitoring to achieve a better resource utilization and improve the QoS/QoE in SDNs. The authors in [12] implement *OpenNetMon* to monitor network parameters. This application, by monitoring per-flow QoS metric, allows to determine on-line whether the end-to-end QoS parameters are satisfactory. The main drawback is that the authors do not suggest how the controller can find the new paths based on the real-time data.

Another way to measure network parameters is to use an analytical model. Specifically, [13] presents a model based on the queuing theory. It is tested on a switch OpenFlow-enabled in conjunction with an SDN controller by means of a simulation: authors assume that the OpenFlow architecture



(a) Video streaming and file transfer throughputs during a permanent link congestion.



(b) Video streaming and file transfer throughputs during a permanent link congestion in a network managed by the QoS architecture.

Fig. 3: Comparison of the service throughputs during a permanent link congestion with and without the QoS architecture.

can be viewed as a feedback oriented queuing system model divided into two systems. The main advantage of using an analytical model is that it can provide results in less time than using a simulation. However, the paper does not exploit the fact that measurements can be used to improve the QoS.

OpenQoS [14] is an interesting proposal for multimedia delivery with end-to-end QoS support. The routes of the multimedia traffic are optimized dynamically to respect the QoS requirements, such as packet loss and latency. The paper suggests a dynamic differentiated QoS routing for QoS flows, based on the CSP problem, while other flows remain on their shortest path. However, the paper does not consider the Multi-Commodity Flow Problem that is important especially when the SDN controller has to deal with different flow types.

Another optimization problem for QoS flow routing is suggested in [15]. It presents an architecture suitable for supporting QoS flows by generating routes into flow tables for QoS traffic separately from best effort flow tables. The paper also assumes a complete lossless QoS traffic pattern to route the flows. However, in a real scenario it is not very simple to have a lossless traffic, especially in a wireless environment.

The proposal [16] describes an OpenFlow-assisted QoE Fairness Framework (QFF) to improve the QoE of multiple clients in wired networks. By exploiting the SDNs and OpenFlow, the paper suggests a way to optimize the QoE for all video streaming devices in a network, considering also the device and network requirements. The results show that QFF provides network stability and optimizes video streaming QoE among different devices in a network. However, the proposed solution takes into account only a single OpenFlow-enable switch in a wired environment without multi-path diversity.

VI. Conclusion

This paper analyzes and solves the problem of managing differentiated services guaranteeing the quality level requirements for real-time multimedia applications in SDNs. Moreover, we achieve a better utilization of the network resources by means of the MCFCSP model. The results show that with our QoS architecture, that continuously ascertains the network status and allocates new paths if necessary, it is possible to avoid the link congestion effects, or at least strongly reduce them.

Incidentally, by keeping the service throughputs as high as possible, our architecture is able to enhance the quality of the service and of experience. Furthermore, we are able to map the results given by our MCFCSP model into a MOS scale to forecast an opinion score from the client point of view. As a future plan, we intend to exploit our architecture to manage a seamless vertical handover between different network technologies, such as Wi-Fi, WiMAX, and LTE, controlling the QoS in many differentiated situations, so to be able grant always the best results.

References

[1] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," Open Networking Foundation, Palo Alto, CA, USA, White paper, Apr. 2012.
[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.
[3] T. Szigeti and C. Hattingh, *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs.* Cisco Press, 2004.
[4] I. Rec, "P. 800: Methods for subjective determination of transmission quality," *International Telecommunication Union, Geneva*, 1996.
[5] B. Lantz, B. Heller, and N. Mckeown, "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks," in *In ACM SIGCOMM HotNets Workshop*, 2010.
[6] "AMPL, A Mathematical Programming Language," http://www.ampl.com/.
[7] "IBM CPLEX Optimizer," http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.
[8] R. M. Karp, "Reducibility Among Combinatorial Problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.
[9] "VLC Media Player," http://www.videolan.org/vlc/.
[10] "Big Buck Bunny movie," http://www.bigbuckbunny.org/.
[11] "Apache FtpServer," http://mina.apache.org/ftpserver-project/.
[12] N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers, "OpenNetMon: Network monitoring in OpenFlow Software-Defined Networks," in *NOMS*, 2014, pp. 1–8.
[13] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an OpenFlow architecture," in *Teletraffic Congress (ITC), 2011 23rd International*, Sept 2011, pp. 1–7.
[14] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks," in *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific.* IEEE, Dec. 2012, pp. 1–8.
[15] S. Civanlar, M. Parlakisik, A. Tekalp, B. Gorkemli, B. Kaytaz, and E. Onem, "A QoS-enabled OpenFlow environment for Scalable Video streaming," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, Dec 2010, pp. 351–356.
[16] 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 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking*, ser. FhMN '13. New York, NY, USA: ACM, 2013, pp. 15–20.