



**HAL**  
open science

## A consistent QoS routing strategy for video streaming services in SDN networks

Djamel Eddine Henni, Abdelghani Ghomari, Yassine Hadjadj-Aoul

► **To cite this version:**

Djamel Eddine Henni, Abdelghani Ghomari, Yassine Hadjadj-Aoul. A consistent QoS routing strategy for video streaming services in SDN networks. *International Journal of Communication Systems*, 2019, pp.e4177. 10.1002/dac.4177 . hal-02427661

**HAL Id: hal-02427661**

**<https://inria.hal.science/hal-02427661>**

Submitted on 3 Jan 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A consistent QoS routing strategy for video streaming services in SDN networks

Djamel Eddine HENNI\*\*, Abdelghani GHOMARI\*, Yassine HADJADJ-AOUL<sup>†</sup>

\*RIIR Laboratory, University of Oran 1, Algeria

Email: henni\_djamel@yahoo.fr

<sup>†</sup>Inria, Univ Rennes, CNRS, IRISA, France

Email: yassine.hadjadj-aoul@irisa.fr

## Abstract

The Software Defined Networking (SDN) paradigm proposes to decouple the control plane (decision-making process) and the data plane (packet forwarding) to overcome the limitations of traditional network infrastructures, which are known to be difficult to manage, especially at scale. Although there are previous works focusing on the problem of Quality of Service (QoS) routing in SDN networks, only few solutions have taken into consideration the network consistency, which reflects the adequacy between the decisions made and the decisions that should be taken. Therefore, we propose a network architecture that guarantees the consistency of the decisions to be taken in an SDN network. A consistent QoS routing strategy is, then, introduced in a way to avoid any quality degradation of prioritized traffic, while optimizing resources usage. Thus, we proposed a traffic dispersion heuristic in order to achieve this goal. We compared our approach to several existing framework in terms of best-effort flows average throughput, average video bitrate and video Quality of Experience (QoE). The emulations results, which are performed using the Mininet environment, clearly demonstrate the effectiveness of the proposed approach that outperforms existing frameworks.

## Index Terms

QoS routing, SDN, OpenFlow, Network consistency, Monitoring, Video streaming.

## I. INTRODUCTION

Network industry has been developed, over the last twenty years, on the basis of a model where network devices provide both control functions (i.e. routing) and packets' delivery (i.e. forwarding) in a distributed mode. Although effective, these techniques have proven to be very rigid [1]. Indeed, the slightest change to these devices is very complex to set up and requires a very tedious deployment process, which tends to slow down any evolution.

SDN is an alternative networking paradigm, that is attracting the attention of a growing number of market players. It mainly consists in separating physically the control plane from the data plane, while relocating the network intelligence on a remote entity, called the controller [2]. The latter is responsible for calculating and updating network nodes' forwarding rules, while network switches have the responsibility of packets' forwarding according to these rules. SDN therefore allows us to program the network behavior according to our needs while making the infrastructure layer abstract for network applications and services.

By simplifying considerably the network control while having the ability to support current and future services, SDN generates enthusiasm within both the research community and Networks Operators (NOs). In fact, SDN gives NOs the possibility to integrate new logic without having to modify the infrastructure. Moreover, the centralization of the control plane <sup>1</sup> reduces significantly the complexity of the services, since this prevents us from having the side effects of distributed applications. Thus, Real-time applications like video streaming, for example, can be managed more easily with the emergence of SDN using more advanced QoS routing techniques [3].

For many years, QoS routing has been a major concern in several studies. The main objective of such research is to propose routing algorithms that can provide the best paths to satisfy customer Service Level Agreement (SLA). To achieve this objective, two main tasks must be carried out, the first one is to have a global view of the network topology and keep it up to date, while the second is to find the best paths based on the results of the first task. Therefore, the best routing algorithm supporting customer SLA is the one that optimally performs these two tasks [4].

Despite the major importance of consistency in the networks, only very few studies have addressed this issue. We believe that this is one of the most challenging aspect to make SDN networks a reality in resources constrained networks. In this work, a new Consistent Network Architecture for routing in SDN is proposed to achieve the consistency and to provide better QoE for video streaming over SDN networks. The main contributions of the proposed approach are summarized as follows:

- 1) We conceived a consistent network architecture focusing on QoS routing in SDN.
- 2) We proposed a flow placement strategy, which notably consists in reducing the concentration of prioritized flows, in a same segment, across the entire network topology. This allows protecting prioritized packets from losses, in case of congestion through using traditional QoS enforcement tools, like traffic shaping, priority queuing or any other mechanism.

<sup>1</sup>It is meant by the centralization of the control plane, the logical or the physical centralization.

- 3) We proposed a dynamic flow entry timeout allocation strategy, which significantly improves resources utilization.
- 4) We developed the framework using only open source code.

To the best of our knowledge, this is the first work that introduces the idea of flow dispersion according to their concentration (presence), which avoids using very complex resource allocation mechanisms.

The remainder of this paper is organized as follows. Section 2 reviews some related works in the field of QoS routing in SDN networks. In Section 3, we describe the proposed consistent network architecture, while a consistent routing strategy for adaptive video streaming is detailed in Section 4. Section 5 presents experiments and results' discussion. Finally, we give concluding remarks and potential future work, in Section 6.

## II. BACKGROUND

### A. Related Works

Several studies have been conducted in recent years to address the issue of quality of service support in SDN networks. In [3], a survey on QoS management in OpenFlow-enabled SDN networks is proposed. In this survey, relevant research studies are organized into several categories according to mechanisms such as network monitoring, scheduling/queuing, resource reservation, QoE-aware etc. On the other hand, the authors outlined the major challenges and outstanding issues that need to be addressed to improve the management of QoS in SDN. In [5], a recent and in-depth survey on QoS routing in SDN has been conducted. The authors re-implemented and evaluated all studied algorithms. They classified unicast QoS routing approaches based on routing strategies into several classes, including queuing, Lagrange relaxation and Bellman-Ford based algorithms, as well as those that consider least-delay and least-cost paths. In order to evaluate the studied QoS algorithms, the authors proposed the use of a framework that uses topology type and scalability as well as the delay constraint tightness. They conclude that the performance of each QoS routing algorithm depends on the used scenario. Therefore, the selection of the best algorithm is linked to the scenario and to the considered platform.

In [6] Eglimez et al. proposed an application that takes into consideration QoS for adaptive video streaming over OpenFlow networks. The authors considered the dynamic QoS routing as a constrained shortest path (CSP) problem. They used different routing strategy for scalable encoded video base and enhancement layers where the best paths are used for the base layer flows. Their experimental results showed that the proposed framework achieves significant improvement in the overall quality of the scalable video streams. Similarly, the authors, in [7], proposed to differentiate the packets of the base layer from the enhancement layers by treating them separately, as two levels of QoS flows, as in [6], but with a different strategy and different metrics. The authors used delay variation and packet loss as routing metrics and solved the CSP problem using the Lagrange Relaxation based Aggregated Cost (LARAC) algorithm. Thus, if the delay variation constraint is not satisfied, the base layer packets are rerouted on another path. However, in case of lack of bandwidth, only the packets of the enhancement layers that are re-routed. Simulation results showed that, compared with [6], their approach can reduce base layer packets loss rate up to 77.3%. In [8], Mohammadi and Javidan proposed a QoS routing strategy to enhance video surveillance quality over SDN. Although the authors used the LARAC [6] algorithm to calculate optimal paths, they proposed a novel methodology, which consists in calculating the cost of links based on type-2 and type-1 fuzzy logic. This makes the approach particularly suitable for a dedicated video surveillance network, whereas it does not really consider best effort flows. In [9], the authors proposed an SDN architecture, in which Multi Path TCP (MPTCP) is used to guarantee the SLA for clients using video streaming services. In this study, a Mixed Integer Linear Programming (MILP) optimization model is used to determine optimal paths. In order to reduce the computational complexity, the authors designed a heuristic algorithm that runs in polynomial time. The obtained results showed that the proposed approach increased video streaming flow throughput up to 25% compared to a general-purposed bitrate enhancement approach.

In [10], Eglimez et al. proposed a new SDN controller architecture that supports multimedia streaming QoS. The proposed framework consists in updating dynamically the routes according to the total bandwidth utilization. Indeed, the authors considered that a link is congested if its bandwidth utilization (total measured traffic, i.e. multimedia and non multimedia flows) is greater than a predefined threshold. In this case, if all links are congested with best effort flows, all the best paths calculated by the controller will be congested and multimedia streams QoS will not be guaranteed because of the lack of bandwidth. In [11], Slavica et al. presented a new QoS-enabled SDN control framework. The proposed framework uses the available bandwidth amount and queuing techniques when calculating optimal routes instead of standard shortest path routing to provide bandwidth guarantees for prioritized flows while protecting as much as possible best-effort traffic. In this study, even if the authors consider dynamic routing for best-effort flows instead of the classical shortest path based routing, they did not take into consideration video streams concentration at queues level.

Other types of approaches have also been proposed to optimize the functioning or improve the QoS of the flows. In [12], Eglimez et al. proposed extending traditional approaches to multi-domain in order to better manage the scaling of services while managing the end-to-end congestion. The authors considered the calculation of optimal paths per domain (i.e. local minimum) and the sharing of this knowledge between distributed SDN controllers. In [13], given the high cost and energy consumption of The TCAM (Ternary Content Addressable Memory) memories that are used in SDN-enabled switches to store flow forwarding rules, the authors proposed an efficient use of such memories. For this purpose, a study based jointly on traffic

engineering and rule placement is conducted. In [14], Cello et al. presented a solution that takes into account packet loss and queue control at switches level. These tasks are performed by implementing a neural network predictor in the SDN controller. Our previous work [4] differs from the latter in the metric used to re-route flows to ensure the targeted QoS/QoE level. In fact, we used the Delay Constrained Least Cost (DCLC) algorithm in a remote controller to solve the Constrained Shortest Path First (CSPF) problem and improve the QoS/QoE of the received video.

In order to summarize what has been mentioned above, one can note that the frameworks discussed in [10], [8], [9], [6] and [7] consider only the improvement of video streams using different strategies of QoS routing without supporting any mechanism to protect them from heavy best-effort traffic. On the other hand, even if the study discussed in [11] considered this aspect while ensuring the protection at best of this kind of flows, it did not take into consideration real time streams concentration when performing queuing configuration. In fact, sharing the bandwidth with best-effort traffic will certainly degrade the video quality. Thus, a mechanism to protect video streams from best effort traffic is certainly needed, as well as avoiding prioritized (video) flows concentration in the same links at best. One also needs to leverage best effort traffic when there is enough bandwidth. These different aspects are considered in the proposed approach.

### B. Problem description

If we consider, in the following, a network based on an SDN architecture, as shown in Figure 1, and the use of a traditional routing technique such as the Open Shortest Path First (OSPF) protocol [15], which uses the link capacity as a routing metric. In this case, when the host  $H2$  download a video stream from the Video Server (VS), the stream will follow the path  $s4-s1-s8-s2$ . Now, if we suppose that there is a heavy best effort traffic sent from host  $H1$  to host  $H5$ , this flow will be routed on the path  $s1-s8-s2$ . In this case the video stream quality will be degraded because the sub-path  $s1-s8-s2$  is common between the real time and best effort traffic.

Many solutions have been proposed to solve this issue such as those discussed in the related works subsection. Indeed, when we consider these approaches, two major problems arise. The first one is the difficulty to control best-effort flows with a very high flow rate, while the second problem occurs when the controller is forced in some cases to put several prioritized flows in the same link due to the lack of resources in the network, which forces such flows to share the remaining bandwidth at the expense of their quality.

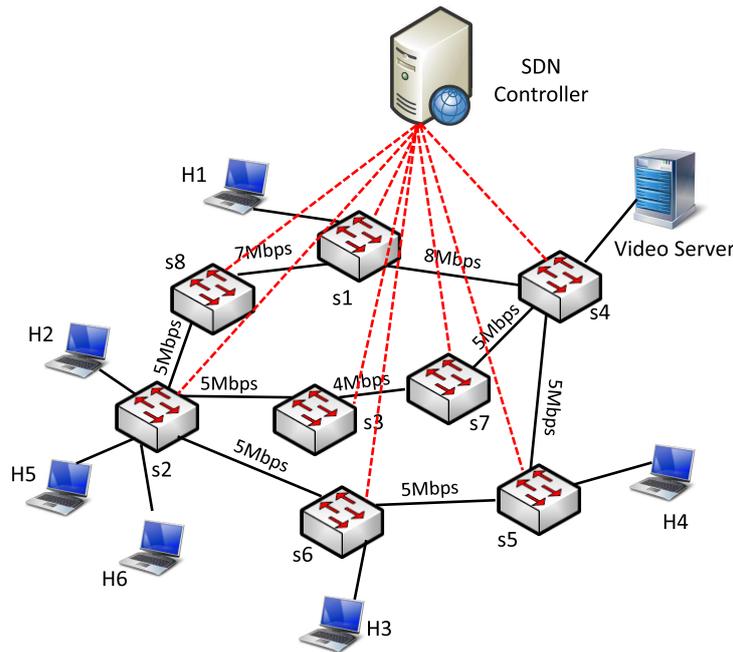


Fig. 1. A network based on SDN architecture

Consequently, we propose a consistent network architecture for routing in SDN based networks that takes into consideration these weaknesses and guarantees prioritized flows QoS while preserving best-effort flows at best.

Note that providing QoS routing can't be achieved without having a global view of the topology. The need for network monitoring remains, indeed, important and continues to be crucial in order to optimize networks' usage while avoiding serious threats from network failures. Also, the abstraction of resources, by a classical network slicing, as proposed in FlowVisor [16], will certainly help significantly in reducing the management's complexity, but it would have a detrimental effect on resources as it may induce an under-utilization. Table I illustrates a comparison of related works and our proposed framework in terms of QoS routing aspects.

TABLE I  
COMPARISON OF RELATED WORKS WITH OUR PROPOSED FRAMEWORK ON QoS ROUTING.

Solutions (citations)	[9]	[10]	[8]	[7]	[14]	[12]	[6]	[11]	[13]	Proposed Framework
Topology Management	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Route Management	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Route Calculation	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
QoS Routing	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
Cross Traffic Shaping	NOK	NOK	NOK	NOK	NOK	NOK	NOK	OK	NOK	OK
PFPR Based Dispersion	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	OK
Dynamic Timeout Allocation	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	OK

### III. CNA4SDN: A CONSISTENT NETWORK ARCHITECTURE FOR ROUTING IN SOFTWARE DEFINED NETWORKS

#### A. Limitations of the classical notion of controller

The notion of SDN controller as conceived today, essentially around Openflow controllers, is more like a simple transducer or a driver for accessing network resources via various protocols. In fact, the Openflow protocol is used to write rules in the switch's flow tables and the consistency of data is, thus, not guaranteed by the controller. Indeed, the rules are directly implemented by applications that in some cases may have conflicting needs, which could have a detrimental effect on the proper functioning of the network. It is, for example, more effective to avoid concentrating flows requiring reliability on the same path because, in the event of congestion, these flows would suffer from the potential losses.

Controllers should not only be interfaces to the underlying network but should also ensure the consistency of the decisions being taken. This objective could be achieved by enriching the latter with functions that would make it possible to guarantee the wider functioning of the system.

In the next sub-section, we propose an end-to-end architecture to ensure a consistent routing decisions at the network level.

#### B. CNA4SDN description

The consistency problem is generally a consequence of distribution. Thus, in traditional networks, the consistency mainly concerns the abstracted view of the network, which is constructed locally at each node (i.e. router, switch or even a middlebox) using distributed protocols. In SDN networks, the apparent simplicity of the architecture hides a complex reality. In fact, in addition to the consistency problem related to the creation of an abstracted view of the network, we have the consistency in the decisions taken remotely by the controllers.

We address the above-mentioned aspects through the architecture depicted in Figure 2, which targets three types of consistencies: Abstract view (consistent view(s) of the network), Decision making (acting consistently on a view), and Network control (consistent rules).

1) *Abstract view creation*:: By relocating the control functionality in a remote entity, the measurements' accuracy (i.e. network monitoring) of the resources' utilization, becomes even more important in SDN networks. Indeed, an optimal control requires an accurate knowledge, in real-time, of the network. This would be prohibitively expensive in terms of overload, which would affect the network efficiency. In this sense, data processing and analysis is certainly an important step to be performed for a better characterization of the network [17]. This helps in addressing the following dimensions:

- Correcting measurements' error using filtering techniques or predicting tools
- Identifying and diagnosing the origins of networks' failures and the most elements prone to failures
- Analyzing the effectiveness of the network strategies

In a previous study [18], this abstraction layer was achieved using, in particular, a prediction technique that we proposed and that provides a more relevant snapshot of the network status. The latter has also allowed us to reduce the signalling overhead by adjusting the polling frequency. Although the results were very good, in this study we opt for a periodic measurements to remove any bias that might impact our proposed QoS routing strategy .

2) *Decision making*:: Having a consistent view of the network, one could act efficiently on the resources (i.e. resources' allocation). We focus on this aspect in the following section by proposing a consistent routing algorithm, which allow guaranteeing the QoS of prioritized traffic.

3) *Network control*:: Having consistent decisions and network views is not enough to guarantee the network consistency. In fact, the rules' enforcement strategy is a major issue, which should be addressed to avoid misleading behaviors [1]. There are two levels of consistency at the network control level. First, there is the consistency of the actions of the applications attached to the controller. In the case of today controllers, the rules are implemented directly by applications that, in some cases, may have conflicting needs, which could have a detrimental effect on the functioning of the network [19]. This question has been addressed by several authors, in particular through the proposal of a high-level network programming language to ensure a certain level of consistency, especially when it comes to updating the rules in the switches [20]. The second level of

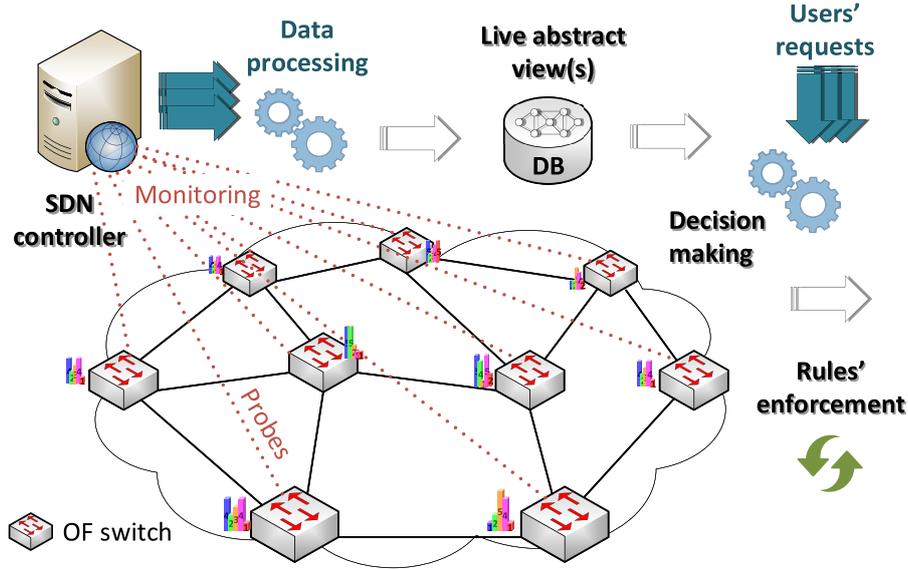


Fig. 2. An Overall Consistent Network Architecture

consistency concerns controllers acting on the same substrate infrastructure. Indeed, the action of one controller can affect the traffic set up by another one or even may cancel its action. In this respect, some proposals in the literature also address this point by proposing, for example, a transactional system, inspired from the domain of distributed databases, for the composition of rules [21].

Note that, we mainly focus, in the following section, on the two first modules in order to enforce consistent routing decisions. Also it must be noted that the approach we propose takes the issue of consistency at all levels and can integrate the mechanisms, considered as relevant, that have been proposed in the literature. For example, it is possible to integrate the RAFT algorithm [22], which is used by ONOS [23], to ensure consensus between distributed controllers, which would fit perfectly into the proposed network control layer. On the other hand, the proposed architecture does not position itself as a competitor to existing architectures but rather as a complement or even as features to be integrated. Indeed, this architecture could also interface and even integrate with the Open Network Automation Platform (ONAP) [24].

#### IV. CONSISTENT ROUTING STRATEGY FOR ADAPTIVE VIDEO STREAMING

An optimal routing strategy must consider two major aspects. A first aspect concerns obtaining an accurate knowledge of the network topology state. A second aspect concerns the routing algorithm, which should not only consider the supported services, in terms of SLA, but also the intrinsic variability of resources to allow an optimal use of the network. In this way, we propose to enrich a classical SDN controller architecture with new modules to achieve the objectives introduced above.

##### A. Topology characterization

Network characterization necessarily involves a monitoring phase of network state in terms of flows throughput and bandwidth availability. Different measurement platforms can be considered in this sense. In our approach, we consider using the SDN southbound messaging standard provided by OpenFlow 1.3.

In the following, we provide a brief summary of the considered messages used to infer the different metrics used in the proposed solutions:

- PacketIn message: sent by the switch to the controller when the switch receives a flow that does not match with any rule in its forwarding table (when a new flow arrives on the switch or when flows timeouts are reached).
- FlowMod message: sent from the controller to the switch in order to install the necessary forwarding rules. The controller specify in this message an idle timeout and a hard timeout for a forwarding rule. The idle timeout refers to the inactivity period, after which a forwarding rule (and eventually the associated flows) is evicted from the switch, while the hard timeout refers to the period, after which, the controller modify the forwarding rule according to the routing information.

These parameters clearly influence the optimality of resources management and the selection of the best path when performing routing. Optionally, flow that will not be rerouted even when topology conditions change are configured with a hard time equal to 0.

- FlowRemoved message: sent from the switch to the controller when a flow is evicted when timeouts are reached.
- FlowStatReq: Message sent from the controller to a switch requesting its current set of statistics for flows, ports, etc.
- FlowStatRes: Message sent from a switch to the controller, in reply to a request message. It contains the number of sent and received packets, flow duration ... etc.

In order to perform the proposed prioritized flows dispersion strategy based on their presence ratio, we propose the monitoring of notably the *PFPR* at different links level using equation (1).

$$PFPR^{(k)} = \frac{\sum_{i=1}^n PFT_i^{(k)}}{TLB} \quad (1)$$

where  $PFT_i^{(k)}$  and  $TLB$  represent, respectively, the prioritized flow  $i$  throughput and the total link bandwidth at step  $k$ .  $PFT_i^{(k)}$  value is obtained using equation (2).

$$PFT_i^{(k)} = \frac{B_{PFT_i}^{(k)} - B_{PFT_i}^{(k-1)}}{t^{(k)} - t^{(k-1)}} \quad (2)$$

where  $B_{PFT_i}^{(k)}$  and  $t^{(k)}$  represent, respectively, bytes count of the real time stream  $i$  and the universal time at step  $k$ .

In order to preserve non prioritized flows as much as possible. We place them on paths that have the most available bandwidth. To do so, we have to measure the *LFB* (Link Free Bandwidth) using equation (3).

$$LFB^{(k)} = TLB - \sum_{i=1}^n PFT_i^{(k)} - BT^{(k)} \quad (3)$$

where  $BT^{(k)}$  represents the best-effort traffic throughput at time step  $k$  that is obtained using equation (4).

$$BT^{(k)} = \frac{B_{BT}^{(k)} - B_{BT}^{(k-1)}}{t^{(k)} - t^{(k-1)}} \quad (4)$$

where  $B_{BT}^{(k)}$  represents best-effort traffic bytes count at step  $k$ .

Note that best effort traffic are not considered in the placement of prioritized constrained traffic. Indeed, using simple queues scheduling allows easily in that case protecting such flows from packets loss and delays.

Note that flow differentiation are performed using classical Openflow's specification, where we identify flows according to packet header fields such as the type of service (TOS) field of IPv4 or the traffic class in IPv6, as well as the traffic class field in MPLS. We can also use flow socket addresses information when the source and/or destination IP address/port are known.

## B. Consistent Routing Strategy (CRS)

The main objective of our contribution, is the protection of prioritized traffic, while at the same time preserving at best non-prioritized one. For that, we propose the dispersion of prioritized flows on the  $K$ -shortest paths in order to avoid, at best, any competition between them, which would be detrimental in terms of quality of service (QoS/QoE). In fact, the flows' dispersion, as proposed (see Algorithm 1), guarantees not only reasonable delays for the flows but also their protection in the event of congestion. Indeed, in this case, only non-prioritized flows will suffer from the potential losses.

In our formulation, a network is represented as a directed simple graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of links. In such a topology, the placement of a flow  $f$  within  $G$  pass by the calculation of the  $K$  shortest path using the function *getKShortestPaths()*, the latter is based on the Yen's algorithm [25] that has a time complexity of  $O(K \times n^3)$ . Even if we consider, in the following, the number of hops for the calculation of the shortest paths, other objectives or multi-objective functions could be considered. Moreover, the goal of the proposed algorithm is to find  $K$  loopless but not necessarily completely disjoint paths to connect the source (src) and the destination (dst). Since the disjoint paths number depends on the minimum cut [26], we considered, in the following, a  $K$  at least equal to that quantity. Thus, in the example provided in Figure 3, we considered a  $K$  equal to 4. Once the paths selected, we eliminate those that do not respect the SLA requested by the service using the *getSubCompliantPaths()* function. Of course, it is obvious that none of the paths will be deleted for best effort flows.

Note that the proposed routing solution acts as a connection admission control since no placement is accepted when the services' SLA cannot be met (see lines 5 – 7).

Contrary to traditional approaches that would place priority and non-priority flows only according to the available bandwidth, we propose different approaches for these flows. In fact, we propose to place prioritized traffic according to the *PFPR*, which reflects the presence ratio of such traffic in each link. The highest value for a path is taken into consideration because the link

**Algorithm 1** CRS Algorithm

---

[1] **Input:** Flow  $f$  to place, topology  $G$ , hard timeout  $HT$ ,  $K$  **Output:** Placement's proposition, hard timeout  $Paths = getKShortestPaths(f.src, f.dst)$   $cPaths = getSubCompliantPaths(Paths, f.SLA)$   $cPaths = \{\Phi\}$  **return** **Null**  $f.Prioritized$   $p$  **in**  $cPaths$   $p.PFPR = \mathbf{max}$   $getTopologyInfo(G, p, metric = PFPR)$   $p^* = \mathbf{parg\ min}$   $\{p.PFPR, \forall p \in cPaths\}$   $f.hard\_timeout = random(HT)$  **return**  $p^*$ ,  $f.hard\_timeout$   $p$  **in**  $cPaths$   $p.LFB = \mathbf{min}$   $getTopologyInfo(G, p, metric = LFB)$   $p^* = \mathbf{parg\ max}$   $\{p.LFB, \forall p \in cPaths\}$   $f.hard\_timeout = random(HT)$  **return**  $p^*$ ,  $f.hard\_timeout$

---

TABLE II  
TABLE OF ACRONYMS

Acronyms	Definition
$SLA$	Service Level Agreement
$PFPR$	Prioritized Flows Presence Ratio
$TLB$	Total Link Bandwidth
$PFT_i^{(k)}$	Prioritized Flow $i$ Throughput at step $k$
$B_{PFT_i}^{(k)}$	Bytes Count of the Real Time Stream $i$ at step $k$
$LFB^{(k)}$	Link Free Bandwidth at step $k$
$BT^{(k)}$	Best-Effort Throughput at step $k$
$B_{BT}^{(k)}$	Best-Effort Traffic Bytes Count at step $k$
$HT$	Hard Timeout
$t$	The universal Time
$P_{s,d}$	The set of $k$ Shortest Paths between the source $s$ and the destination $d$
$l_i^p$	The link $i$ of the path $p$ , $p \in P_{s,d}$

that has the highest  $PFPR$  is the one that is forwarding a lot of priority flows, so, in order to ensure an optimized dispersion of such flows and reduce their packet loss ratio in case of congestion, we select the path  $p^*$  that has the lowest  $PFPR$ . Accordingly, we select the path  $p \in P$  minimizing the maximum  $PFPR$  within its links, see equation (5) below.

$$p^* = \mathbf{parg\ min} \{ \mathbf{max} PFPR(l_i^p) \}, \quad \forall p \in P_{s,d} \quad (5)$$

where  $l_i^p$ ,  $P$ ,  $PFPR$  represent, respectively, the link  $i$  of the path  $p$ , the set of  $k$  shortest paths between the source  $s$  and the destination  $d$ , and a function returning the  $PFPR$  of the link, which is returned by the  $getTopologyInfo()$  function in the algorithm.

The placement of non prioritized flows is based on the link free bandwidth in order to preserve non-priority flows as much as possible. This time, the lowest value is taken into consideration because the link that has the lowest  $LFB$  represents the bottleneck of the path, so, in order to optimize bandwidth utilization while reducing the packet loss ratio, we select the path that has the highest  $LFB$ . Thus, the following rule is applied for the selection of the best path  $p^*$ , see equation (6) below :

$$p^* = \mathbf{parg\ max} \{ \mathbf{min} LFB(l_i^p) \}, \quad \forall p \in P_{s,d} \quad (6)$$

where the function  $LFB$  returns the available bandwidth of a link. Table II summarizes some acronyms that we used in this section.

### C. Dynamic timeout allocation strategy

In order to ensure rerouting operation over the best chosen path, we must assign a different timeout value for each flow. In fact, using the same hard-time value (i.e. time spent in the flow table) for different flows may imply a simultaneous rerouting operation by the controller, and therefore an inconsistency of the decision. Indeed, when the controller performs monitoring operations, it sends FlowStatReq messages periodically (every second in most of the cases) in order to have a more or less accurate picture of reality<sup>2</sup>. Thus, when the controller performs rerouting operations, there is a risk that prioritized flows will share a number of links because their rerouting operation has been done in the same time, or within the monitoring frequency period. To avoid this weakness and optimize bandwidth utilization, we propose the use of a dynamic timeout allocation strategy for different flows to avoid their rerouting at the same time based on a random function as described in algorithm 1.

Having described our proposal, we now direct our attention on evaluating its performance using realistic use cases.

<sup>2</sup>Note that the Internet traffic has a distribution corresponding to a poisson law, for sub-second time scales [27].

## V. EXPERIMENTS AND RESULTS DISCUSSION

### A. Experiments setup

In this section, the performance of the proposed routing strategy is evaluated under the Mininet emulation environment [28]. All the proposed modules were programmed using the Python programming language and integrated within the RYU controller [29]. We analyzed the proposed solution using the Abilene topology, depicted in Figure 3, which is composed of 12 switches, 11 terminals and two video servers, this topology was obtained from the Zoo Topology framework [30], and was converted, using [31], into usable Mininet topology with its real parameters (number of switches, hosts, links' delay, bandwidth ... etc.). The emulation environment was done on a machine doted with an Intel Core i5-6500-3.20 Ghz CPU, 8GB of RAM and Linux OS. Table III summarizes the links' delay of the used topology. The link delays in Zoo topology are obtained using geographical location of the nodes and by assuming the speed of light propagation in an optical medium. Note that, for the experiments purpose, we had to consider a connected graph, we, thus, added the switch s12 and some links with a latency of  $1ms$ .

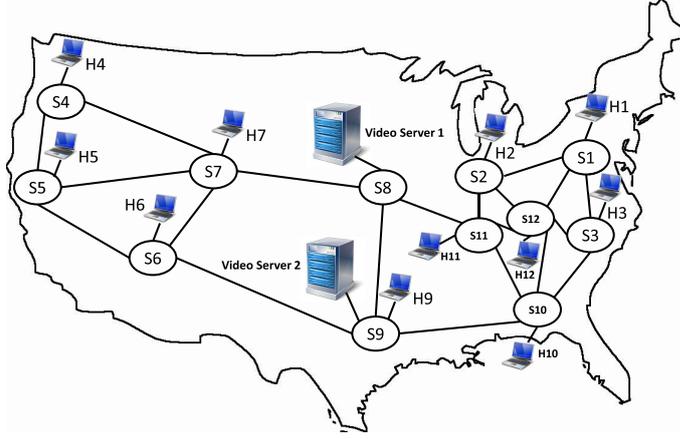


Fig. 3. The Abilene Topology

TABLE III  
ABILENE TOPOLOGY LINKS' DELAY

Link	Delay (ms)
NewYork (S1)- Chicago (S2)	0.806374975652
NewYork (S1) - WashingtonDC (S3)	0.605826192092
Chicago (S2) - Indianapolis (S11)	1.34462717203
WashingtonDC (S3) - Atlanta (S10)	0.557636936322
Seattle (S4) - Sunnyvale (S5)	1.28837123738
Seattle (S4) - Denver (S7)	1.11169346865
Sunnyvale (S5) - LosAngeles (S6)	0.590813628707
Denver (S7) - LosAngeles (S6)	1
Sunnyvale (S5) - Denver (S7)	0.997327682281
LosAngeles (S5) - Houston (S8)	1.20160833263
Denver (S7) - KansasCity (S8)	0.223328790403
KansasCity (S8) - Houston (S9)	1.71325092726
KansasCity (S8) - Indianapolis (S11)	0.240899959477
Houston (S9) - Atlanta (S10)	1.34344500256
Atlanta (S10) - Indianapolis (S11)	0.544962634977
Chicago (S2) - Virginia (S12)	1
WashingtonDC (S3) - Virginia (S12)	1
NewYork (S1) - Virginia (S12)	1
Atlanta (S10) - Virginia (S12)	1
Indianapolis (S11) - Virginia (S12)	1

### B. Traffic pattern

The traffic pattern that we used is composed of best effort Poisson TCP flows that was generated using the MGEN tool [32] and real time Dynamic Adaptive Streaming over HTTP (DASH)-based video flows, which is available at a variety of bit rates

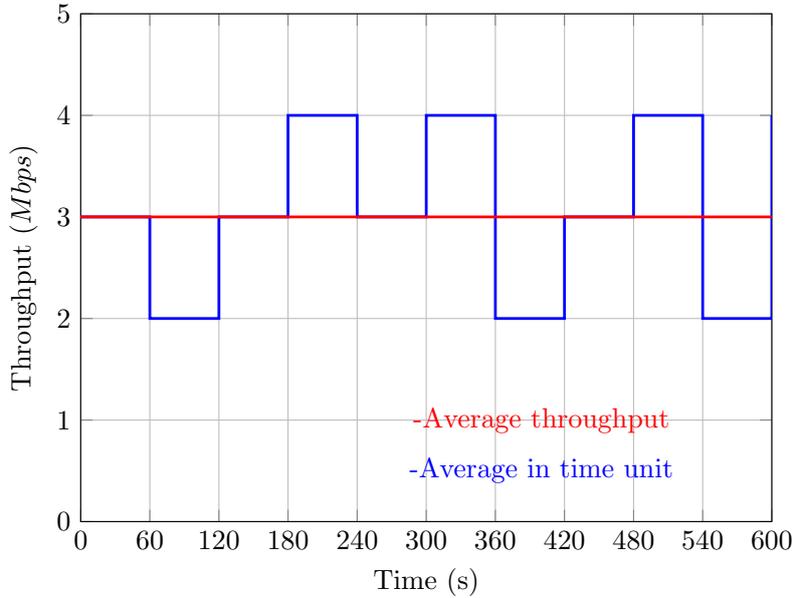


Fig. 4. Best-effort traffic throughput

and qualities (128 Kbps, 256 Kbps, 512 Kbps, 1024 Kbps, 2048 Kbps and 4096 Kbps). The video content which is streamed by video servers (VS1 and VS2) is the Big Buck Bunny animation [33]. The choice of hosts that participate in the emulation has been made in such a way as to allow their generated traffic to cross the entire network topology. Table IV summarizes the used traffic pattern, we note that BT denote best-effort traffic.

TABLE IV  
TRAFFIC PATTERN

Hosts	Traffic Type
H5 to H1	BT (Fig. 4)
H5 to H12	BT (Fig. 4)
H10 to H4	BT (Fig. 4)
H10 to H11	BT (Fig. 4)
VS1 to H3	DASH
VS2 to H2	DASH
VS1 to H6	DASH
VS2 to H7	DASH

Figure 4 represents the best-effort average throughput in each time slot (blue plot) while the red plot represents its average throughout the entire emulation time.

In order to evaluate and validate the performance of the proposed CRS, we compare it to three benchmarks, the traditional single shortest path based routing (SSPR) [15], the link free bandwidth based multi-path routing (LFBMR)<sup>3</sup>, and OpenQoS [10]. To that purpose, we measure, in the following section, for each strategy, the obtained best-effort traffic average throughput and video streaming average bitrate, as well as the obtained video QoE in terms of Mean Opinion Score (MOS), the latter is obtained using a subjective tool that was developed by one of the authors of the paper and that was published in [34].

### C. Obtained results

a) **Best-effort traffic average throughput.**: Figure 5 illustrates the obtained results in terms of best-effort traffic average throughput using the compared strategies. When comparing these approaches, one can notice that LFBMR strategy allows having the highest average throughput. This is mainly the consequence of the routing strategy, which selects the best placement for flows without necessarily considering video streams. Among the other remaining techniques, the proposed routing strategy (CRS) presents the highest average throughput, as it considers first a strategical placement of video streams then the placement of best effort traffic by considering the link free bandwidth. OpenQoS and SSPR have almost the same performance regarding best effort flows because both strategies consider only traditional shortest paths for this kind of flows, with a slight performance of OpenQoS, because the latter consider dynamic routing of prioritized flows, which improves bandwidth management, to the benefit of best effort flows.

<sup>3</sup>We note that in order to implement the LFBMR benchmark, we considered only the placement of non prioritized flows approach (based on the LFB strategy) for both video and best effort flows.

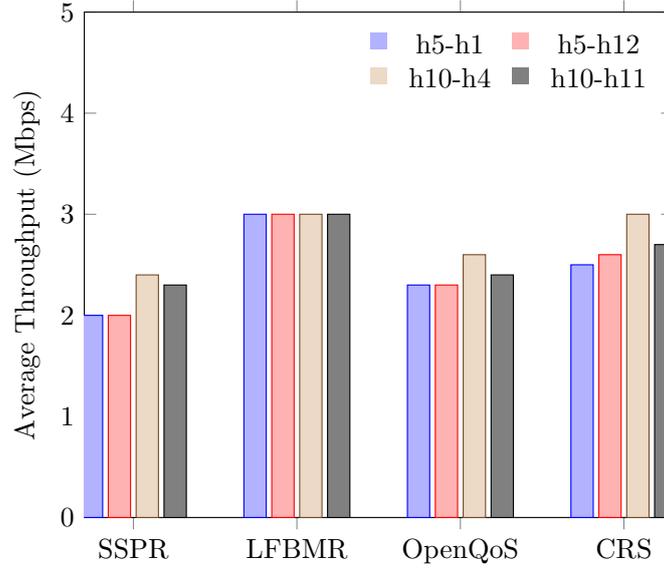


Fig. 5. Obtained Best-Effort Average Throughput Using Different Routing Strategies

*b) Video streams' downloading bitrate and QoE.:* It is clear, from Figures 6 and 7, that the proposed CRS performs better results than the other benchmarks in terms of video streaming bitrate and QoE because the LFBMR approach does not take into consideration the flows' type and performs routing indifferently for video streaming flows and best-effort flows. Indeed, in this approach only the link free bandwidth is considered for assigning links' cost, which allows best-effort streams to dominate bandwidth utilization at the expense of a poor video streams quality (due to quality adaption of video streams). This also explains the increase in the flow rate of best-effort flows, as mentioned earlier. The OpenQoS approach performs better than the LFBMR one due to its routing strategy that consider prioritized flows by placing them at best on no congested paths while considering traditional shortest path for best-effort flows. On the other hand, CRS performs better than OpenQoS because it considers first, prioritized flows presence ratio at link level when performing routing, which avoid concentration of such flows and therefore reduce their packets loss risk in case of congestion, secondly, CRS perform best-effort traffic shaping and dynamic timeout allocation, which significantly improves bandwidth management. Also, the OpenQoS strategy considers only bandwidth availability for prioritized flows without caring about their concentration ratio.

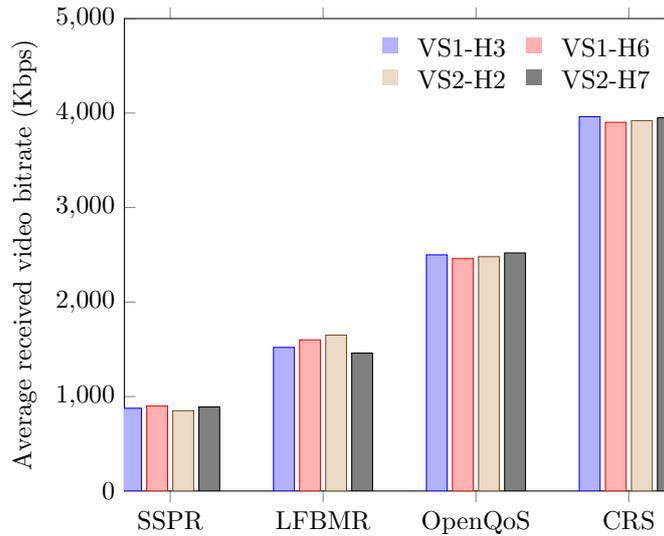


Fig. 6. Average Received Video Bitrate using Different Routing Strategies

The SSPR based routing performs worst in terms of video streams bitrate and QoE, since this approach considers only one

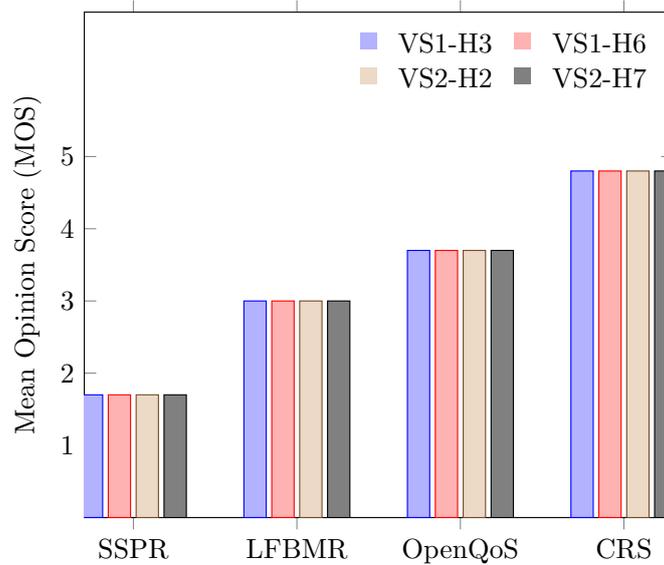


Fig. 7. Obtained Video QoE using Different Routing Strategies

path when performing routing based on the total link bandwidth without taking into consideration other routing constraints, which is good for best-effort flows because of their bandwidth domination at the expense of a poor video streaming quality.

## VI. CONCLUSIONS AND PERSPECTIVES

In this paper, a consistent network architecture for QoS routing in SDN is proposed to enhance video streaming quality and best-effort flows throughput. Our proposed network architecture aims to have a consistent view of the network, consistent decisions and consistent rules' enforcement strategy. In order to implement such architecture, we first monitored the network state using the SDN southbound messaging standard provided by the OpenFlow protocol. Then, we have proposed to reduce, when possible, the concentration of video streams in the links in order to protect them from packets loss in case of congestion while maximizing their QoE. In order to evaluate our proposed approach, we compare it to three benchmarks, the traditional single shortest path based routing (SSPR), the link free bandwidth based multi-path routing (LFBMR), and OpenQoS [10]. Simulation results demonstrated that the proposed traffic engineering framework that considers prioritized flows presence ratio, best effort traffic shaping and dynamic timeout allocation outperforms existing approaches when considering video streaming and best-effort traffic at the same time.

As part of our future work, we plan to generalize the use of the proposed network architecture to resolve other routing constraints.

## REFERENCES

- [1] D. Kreutz, F. Ramos, P. Verissimo, C. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
- [2] B. Astuto, A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials journal*, vol. 16, no. 3, pp. 1617–1634, 2014.
- [3] M. Karakus and A. Durresi, "Quality of service (qos) in software defined networking (sdn): A survey," *Elsevier, Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017.
- [4] D. Henni, A. Ghomari, and Y. Hadjadj-Aoul, "Videoconferencing over openflow networks: An optimization framework for qos routing," ser. Communication and Information Technology (CIT) Conference, Liverpool UK, October 26–28, 2015.
- [5] J. W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer, "Unicast qos routing algorithms for sdn: A comprehensive survey and performance evaluation," *IEEE Communications Surveys & Tutorials journal*, vol. 20, no. 1, pp. 388–415, 2018.
- [6] H. Eglimez, T. Murat, and S. Civanlar, "An optimization framework for qos-enabled adaptive video streaming over openflow networks," *IEEE Communications Letters*, vol. 15, no. 3, pp. 711–715, 2013.
- [7] T. Yu, K. Wang, and Y. Hsu, "Adaptive routing for video streaming with qos support over sdn networks," ser. International Conference On Information Networking (ICOIN), Cambodia Cambodia, January 12–14, 2015.
- [8] R. Mohammadi and R. Javidan, "An adaptive type-2 fuzzy traffic engineering method for video surveillance systems over software defined networks," *Springer Multimedia Tools and Application Journal*, vol. 76, no. 22, pp. 23 627–23 642, 2016.
- [9] H. Korhan, K. Reza Shokri, C. Cihat, and S. Muge, "Towards qos-aware routing for dash utilizing mptcp over sdn," ser. IEEE Conference On Network Function Virtualization and Software Defined Networks (NFV-SDN), Berlin Germany, November 6–8, 2017.
- [10] H. Eglimez, S. Dane, K. Bagci, and A. Murat Tekalp, "Openqos: An openflow controller design for multimedia delivery with end-to-end quality of service over software-defined networks," ser. Signal and Information Processing Association Annual Summit Conference (APSIPA ASC), Asia-Pacific Hollywood CA USA, December 3–6, 2012.
- [11] T. Slavica, P. Neeli, and R. Igor, "Sdn control framework for qos provisioning," ser. 22nd Telecommunications Forum (TELFOR), Belgrade Serbia, November 25–27, 2014.
- [12] H. Eglimez and A. Murat Tekalp, "Distributed qos architectures for multimedia streaming over software defined networks," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1597–1609, 2014.
- [13] H. Huang, S. Guo, P. Li, B. Ye, and I. Stojmenovic, "Joint optimization of rule placement and traffic engineering for qos provisioning in software defined network," *IEEE Transactions on Computers journal*, vol. 64, no. 12, pp. 3488–3499, 2015.
- [14] M. Cello, M. Marchese, and M. Mongelli, "On the qos estimation in an openflow network: The packet loss case," *IEEE Communications Letters*, vol. 20, no. 3, pp. 554–557, 2015.
- [15] <https://tools.ietf.org/html/rfc2328>. Accessed June 25, 2019.
- [16] R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Flowvisor: A network virtualization layer," Deutsche Telekom Inc. R&D Lab, Stanford University, Nicira Networks, Tech. Rep. Technical Report OPENFLOW-TR-2009-1, 2009.
- [17] G. Phillipa, J. Navendu, and N. Nachiappan, "Understanding network failures in data centers: Measurement, analysis, and implications," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 350–361, 2011.
- [18] D. Henni, Y. Hadjadj-Aoul, and A. Ghomari, "Probe-sdn: a smart monitoring framework for sdn-based networks," ser. Global Information Infrastructure and Networking Symposium (GIIS), Porto Portugal, October 19–21, 2016.
- [19] P. Perešini, M. Kuzniar, N. Vasić, M. Canini, and D. Kostü, "Consistent packet processing for openflow," ser. The Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hong Kong China, August 16–16, 2013.
- [20] N. e. a. Foster, "Languages for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 128–134, 2013.
- [21] M. Canini, P. Kuznetsov, D. Levin, and S. Schmid, "Software transactional networking: concurrent and consistent policy composition," ser. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (HotSDN '13), Hong Kong China, August 16–16, 2013.
- [22] E. Sakic and W. Kellerer, "Response time and availability study of raft consensus in distributed sdn control plane," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 304–318, 2018.
- [23] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz O'Connor, P. Radoslavov, M. Snow, and G. Parulkar, "Onos: towards an open, distributed sdn os," ser. Proceedings of the third workshop on Hot topics in software defined networking (HotSDN '14), Chicago Illinois USA, August 22-22, 2014.
- [24] F. Slim, F. Guillemin, A. Gravey, and Y. Hadjadj-Aoul, "Towards a dynamic adaptive placement of virtual network functions under onap," ser. IEEE Conference On Network Function Virtualization and Software Defined Networks (NFV-SDN), Berlin Germany, November 6–8, 2017.
- [25] J. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, pp. 712–716, 1971.
- [26] J. Yi, H. Adnane, S. David, and B. Parrein, "Multipath optimized link state routing for mobile ad hoc networks," *Elsevier, Ad Hoc Networks Journal*, vol. 9, no. 1, pp. 28–47, 2011.
- [27] V. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Communications Magazine*, vol. 32, no. 3, pp. 70–81, 1994.
- [28] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," ser. Hotnets-IX Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Monterey California, October 20–21, 2010.
- [29] <https://osrg.github.io/ryu/>. Accessed June 15, 2019.
- [30] <http://www.topology-zoo.org>. Accessed June 20, 2019.
- [31] <https://github.com/sjas/assessing-mininet>. Accessed June 25, 2019.
- [32] <http://www.nrl.navy.mil/itd/ncs/products/mgen/>. Accessed June 20, 2019.
- [33] <http://www.bigbuckbunny.org>. Accessed June 23, 2019.
- [34] K. Singh, Y. Hadjadj-Aoul, and G. Rubino, "Quality of experience estimation for adaptive http/tcp video streaming using h.264/avc," ser. 9th Annual IEEE Consumer Communications and Networking Conference - Multimedia and Entertainment Networking and Services, CCNC, Las Vegas NV USA, January 14–17, 2012.

## AUTHOR BIOGRAPHY



**Djamel Eddine HENNI.** is a Ph.D student in computer science at Oran 1 Ahmed Benbella University in Algeria. He is a member and researcher in the RIIR laboratory. He received a B.Sc. and Master degrees in telecommunications engineering with high honours from the National Institut of Telecommunications and Information Technology of Oran, Algeria, in 2008 and 2012 respectively. He is also a network administrator at Sonatrach company since 2008. His main research interests concern the fields of network function virtualization, congestion control protocols and QoS provisioning, and software defined networking.



**Abdelghani GHOMARI.** is currently working as an associate professor at Oran 1 Ahmed Benbella University, Algeria, where he is also a member of the RIIR Laboratory. He received his B.Sc. In Computer Science from Oran1 University, in 1990. Dr. GHOMARI received his Master degree, and the PhD in Computer Science from Oran1 University, in 1996 and 2007, respectively. He was involved in several national projects and his main research interests include multimedia documents, multimedia mining and video indexing and retrieval, multimedia networks. Dr. GHOMARI has been on the technical program committee of different International Conferences. His work on multimedia indexing and multimedia networks has led to several technical papers in journals and international conference proceedings.



**Yassine HADJADJ-AOUL.** is currently working as an associate professor at the University of Rennes 1, France, where he is also a member of the IRISA Laboratory and the INRIA project-team Dionysos. He received a B.Sc. In computer engineering with high honors from Mohamed Boudiaf University, Oran, Algeria, in 1999. Dr. Hadjadj received his Master's and Ph.D. degrees in computer science from the University of Versailles, France, in 2002 and 2007, respectively. He was an assistant professor at the University of Versailles from 2005 to 2007, where he was involved in several national and European projects such as NMS, IST-ATHENA, and IST-IMOSAN. He was also a post-doctoral fellow at the University of Lille 1 and a research fellow, under the EUFP6 EIF Marie Curie Action, at the National University of Dublin (UCD), where he was involved in the DOM'COM and IST-CARMEN projects, which aim at developing mixed Wi-Fi/WiMAX wireless mesh networks to support carrier grade services. His main research interests concern the fields of wireless networking, multimedia streaming architectures and protocols, congestion control protocols and QoS provisioning, and satellite and space communications. Dr. Hadjadj has been on the technical program committee of different IEEE conferences, including Globecom, ICC, VTC, PIMRC and IWCMC. His work on multimedia and wireless communications has led to more than 40 technical papers in journals and international conference proceedings.