

# Quality of service provisioning to deliver scalable high efficiency video coding (SHVC) over SDN networks

## Abstract

Nowadays, Internet traffic is dominated by video streaming and delivering these services is a challenging task. Providing end-to-end Quality of Service (QoS), and consequently providing an adequate Quality of Experience (QoE), is essential in the multimedia industry. In this paper, we designed LSDN, an effective SDN-based solution that leverages SDN and SHVC video codec feature to provide a reliable video streaming mechanism. LSDN uses the ability of SDN to dynamically update forwarding tables of routers and switches and transfer different video layers of SHVC over distinct network paths. To evaluate the performance of LSDN, we emulated LSDN over Mininet. Our experimental results show that LSDN significantly improves the performance of delivering high-quality videos in terms of deduction in delay, loss, and jitter. Moreover, we observed that LSDN does not have negative effects on the background traffic.

**Keywords:** SDN, QOE, SHVC learning routing, multimedia

Volume 8 Issue 2 - 2022

**Sina Keshvadi,<sup>1</sup> Bahnam Behnam Faghah<sup>2</sup>**

<sup>1</sup>Thompson Rivers University, Kamloops, BC, Canada

<sup>2</sup>Maynooth University, Maynooth, Ireland

**Correspondence:** Sina Keshvadi, Assistant Professor, Thompson Rivers University, Kamloops, BC, Canada, Email [sina.keshvadi@tru.ca](mailto:sina.keshvadi@tru.ca)

**Received:** August 01, 2022 | **Published:** September 12, 2022

**Abbreviations:** SDN, software-defined networking; QoE, quality of experience; QoS, quality of service; SHVC, scalable high efficiency video coding

## Introduction

Nowadays, Internet streaming has the biggest share in global Internet traffic.<sup>1</sup> Video content delivery is shifting from traditional television broadcasting to Internet-based video on demand (VoD). The number of online video streaming services such as YouTube and Netflix, users, and TV shows is increasing. In addition, emerging and popularity of high-resolution standards such as 4K, 8K, and virtual reality (VR) have placed an even greater demand for Internet performance. Cisco recently predicted that 66% of connected flat-panel TV sets in 2023 will be 4K, and Internet video will present 84% of global Internet traffic.<sup>2</sup> This growth in streaming will be a heavy burden on Internet infrastructure.

On the other side, video coding standards are evolving to provide more efficient storage and transmission solutions for delivering video content.<sup>3</sup> Currently, High Efficiency Video Coding (HEVC),<sup>4</sup> also known as H.265, is the latest video compression standard. It is designed as part of the MPEG-H project and it is a successor to the widely used Advanced Video Coding (AVC, H.264).<sup>4</sup> The scalable extension of the HEVC, namely SHVC,<sup>5</sup> is provided to achieve efficient scalable coding. SHVC provides a mechanism for coding video in multiple layers, where each layer represents a different quality of the same video.<sup>6</sup> such scalable codec encodes video streams in a base layer and one or more enhancement layers. To offer a reasonable guarantee with reasonable quality, the base layer of a video stream could transmit with hard QoS i.e. without any packet loss or delay variation, while the enhancement layers could be transmitted with a soft QoS or best-effort. We should consider this fact that all enhancement layers could be decoded only with reference to the base layer.

Due to the best-effort nature of the Internet, a continuous stream cannot be guaranteed when congestion arises. Streaming multimedia applications require predictable, steady network resources with little

delay variation and no packet loss which cannot be always met by the standard best-effort Internet. Therefore, over the past decade, the Internet Engineering Task Force (IETF) has explored several Quality of Service (QoS) architectures. The Integrated Services (IntServ) model<sup>7</sup> exploits the Resource Reservation Protocol (RSVP) to reserve a path between sender and receiver for end-to-end communication. The Differentiated Services (DiffServ) model<sup>8</sup> is proposed based on the Type of Service (ToS) field in the header of the packets and classifies the incoming flows. Multi-Protocol Label Switching (MPLS)<sup>9</sup> is another solution to provide QoS by labeling techniques to reduce the complex routing table lookups. However, none of these efforts by IETF has been truly successful and globally implemented.<sup>10</sup> This is because QoS architectures such as IntServ and Diffserv are built on top of the current Internet distributed hop-by-hop routing architecture, lacking a broader picture of overall network resources. Although tunnelling with MPLS provides a partial solution, it lacks real-time reconfigurability and adaptivity.<sup>10</sup>

The emerging of Software Defined Networking (SDN)<sup>11</sup> enables us to achieve the diversification in routing. SDN allows the dynamic adjustment of forwarding tables to reroute different flows. This combination of the layered characteristics of SHVC, and SDN's ability to update routing tables provide an opportunity for transferring different video layers over distinct network paths. In this paper, we present LSDN, an SDN-based solution to deliver layered video codecs over SDN. The proposed mechanism monitors the network links and then calculate the best-fit delivery paths for different video layers. It enables the network to guarantee a reliable and continuous video play out for the base layer, and it selects the next best paths to deliver enhancement layers.

The rest of this paper is organized as follows: Section 2 represents an overview of background concepts. Then, the proposed solution is described in section 3. The experimental setup is explained in section 4, followed by evaluation results in section 5. A review of related works is discussed in section 6, and we provide a conclusion briefly in section 7.

## Background

### SHVC

The High-Efficiency Video Coding (HEVC/H.265) as a successor to Advanced Video Coding (AVC/H.264), is the state-of-the-art video coding standard developed by the Joint Collaborative Team.<sup>12</sup> HEVC delivers a 50% bandwidth saving over AVC (H.264) without any loss in the quality than AVC. Although Google introduced the VP9 codec<sup>13</sup> at the same time, HEVC has gained popularity in both academic and multimedia communities and is widely supported by new multimedia devices like televisions, computers and smartphones. SHVC (Scalable High-Efficiency Video Coding), the scalable extension to the H.265 standard, is a new video encoding standard that supports ultra-high-definition video streams.<sup>6</sup> the scalable feature operates by dividing a single video stream into several layers. Layers are hierarchically ordered with the higher layers are dependent on lower layers for decoding. This configuration enables video streams to adapt to receiver devices and network constraints easily and quickly.

### QoS provisioning

QoS is typically defined as an ability of a network to provide the required services for a selected network traffic.<sup>14</sup> The primary QoS parameters including but not limited to throughput, delay, jitter and loss.<sup>7</sup> QoS is mainly implemented in two approaches: hard QoS and soft QoS. The hard QoS method guarantees the QoS requirements of connections and the soft QoS method is not as strict as the hard QoS methods regarding QoS requirements. Different types of QoS have been defined by Internet Engineering Task Force (IETF)<sup>15</sup> to support Quality of Service (QoS).

The *Integrated Services (IntServ)* model exploits the Resource Reservation Protocol (RSVP) to provide end-to-end QoS.<sup>9</sup> In this model, a path between sender and receiver is reserved for end-to-end communication. To provide IntServ, all routers along the traffic path must support it and each router should keep the state of the reservation. Therefore, this model suffers from scalability issues, is not applicable in wide networks like the Internet. To address the scalability concerns, the *Differentiated Services (DiffServ)* model is proposed. This model is based on the Type of Service (ToS) field in the header of the packets and classifies the incoming flows. Since this model treats all packets in the same class identically, it is difficult to provide quantitative QoS to individual flows.<sup>8</sup> *Multi-Protocol Label Switching (MPLS)*<sup>10</sup> is another solution to provide QoS by labeling techniques to reduce the complex routing table lookups. MPLS avoids complex routing table lookups in a routing table and directs packets from one node to the next based on short path labels. It uses a 3-bit Traffic Class (TC) field for QoS priority. One potential drawback of MPLS is that the carrier has to play a role in configuration of the overall network. These pros and cons show that the current QoS solutions are not truly successful at QoS support for service providers and is not fit for Multimedia applications.

### SDN (Software Defined Networking)

SDN is a paradigm that has changed the Internet architecture by decoupling control and data plane<sup>16</sup> This decoupling provides an effective means to provide dynamically reconfigurable QoS requirements, which may be preferable to best-effort routing. Within an SDN network, an external device called "controller" determines the forwarding rules on switches. Open Flow is the first protocol defined for the communication between the controller and the switches.<sup>16</sup> Open Flow enabled switches update their forwarding tables according to the commands from the controller. An overview of the Open Flow architecture is shown in Figure 1.

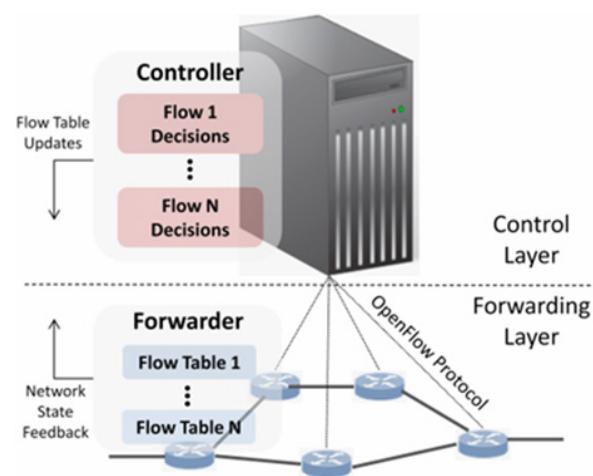


Figure 1 An overview of the OpenFlow structure.<sup>17</sup>

Open Flow offers a new forwarding solution by allowing network to define different routing rules associated with data flows and it uses the controller to make routing decisions.

### LSDN framework

In this section, we introduce LSDN, a framework for delivering layered video streams over a SDN network. First, we describe its design, and then we discuss its implementation in more details.

### Architectures and design

To meet the required end-to-end QoS, LSDN uses a new prioritization scheme to prioritize forwarding flows. Our proposed mechanism is different from IntServ and DiffServ architectures, since it does not use resource reservation and therefore it does not have negative impacts on other traffics. In the current network routing structure, when a packet arrives at a router, the destination addresses is checked with the routing table and then it is forwarded by predefined rules defined by network operator. Open Flow offers flexible per-flow based routing by defining different types of flows and a set of actions and rules that could be modified on demand. There are several ways to distinguish between different flows. For example, packets with TCP port 80 can be a flow definition or each flow may be determined by using packet header fields such as TOS (Type of Service) field of IPv4 header.<sup>17</sup>

In Open Flow, network devices store flows, their priorities, and actions in flow tables. LSDN exploit Open Flow features to differentiate base layer and enhancement layer[s] of video streaming from other network traffic, and then assigns appropriate actions. LSDN sets priorities to different flows as follows:

- QoS level-1, for the base layer of SHVC video stream, has the highest priority and requires hard guarantee.
- QoS level-i, for enhancement layer i, has highest priority after the QoS level-(i-1) flows, and requires soft guarantee.
- QoS level-n, for other traffics, has lowest priority, and requires best effort guarantee.

LSDN provides hard guarantee of QoS level-1 by selecting the best path to forward base-layer flows, and it selects second best path for QoS level-2. The other non-QoS traffic is considered as best effort. We implement our policies as an extension to the SDN controller.

The controller component of Open Flow is the brain of the network where routing changes are determined. The controller provides access to flow tables, and the rules that tell the network forwarders how to direct traffic flows. In this paper, we only detect two-layers of SHVC, but it can be easily modified to add new routing policies to a new type of service.

## Implementation

We implemented LSDN by adding extension modules to the Open Daylight controller. Open Daylight<sup>19</sup> is a java-based open-source controller that presents a modular programming environment to add new modules on top of it. This feature enables an operator to decide which existing modules to run. Our implementation includes two main QoS provision modules as follow:

**Topology manager:** This module discovers and maintains all paths information through the data received from forwarders. This module has two key functions. First, *Topo\_Discover* function that it is used for discovering the network topology, and second, *Find\_Path* function that it is provided to gather and keep up-to-date a sorted list of all available paths.

**Flow manager:** This module uses information gathered by the Topology Manager to allocate distinct paths to different flows with different priorities. The key function of Flow Manager is *Flow\_Shaper* function that distinguishes different sorts of flows and modifies their action in flow table to enable customized flow forwarding.

From the implement perspective, we use the function and API provided by OpenVSwitch<sup>18</sup> to manipulate flows and flow table. This module checks new incoming flows based on their ToS (Type of Service) bits, and changes the flow's *Output\_Action* in the flow table of the switch to forward them in a corresponding path. When a *PACKET\_IN* message arrives in the controller, the Flow Manager module checks the packet ToS field and determines the QoS level of the packet based on pre-defined ToS fields. The controller sets the associated *Output\_Action* to the flow by receiving the path's information from the Topology Manager. In this way, the next incoming packets with that ToS fields are forwarded based on the new output action.

We evaluate the performance of the proposed approach in the next sections.

## Experimental setup

We evaluate the performance of over proposed solution over SDN by using Mininet<sup>19</sup> emulator. Mininet hosts run standard Linux network software, and its switches support Open Flow for highly flexible custom routing and Software-Defined Networking. Our test bed is shown in Figure 2. We provide three hosts as senders of the base layer, enhancement layer, and background traffic, respectively, and one host as a client. We use two switches to forward flows. Path 1 is designated as the best path between sender and receiver with more bandwidth, and paths 2 and 3 are considered as longer paths.

Three different video qualities are used in three test scenarios. The traffic for all sessions is generated using Iperf.<sup>20</sup> Iperf is a widely used network performance measurement and tuning tool. It supports both TCP and UDP data streams, and it reports bandwidth, delay jitter, and datagram loss. In our experiment, similar to layered video conferencing, UDP is employed as a transport protocol. We generate UDP traffic for generating background traffic and video stream with three different bitrates as shown in Table 1.

**Table 1** Experiment's bitrates

Experiments streams			
Stream/Bitrate	Low quality	Mid quality	High quality
Base Layer	1 Mbps	3 Mbps	5 Mbps
Enhancement Layer	1 Mbps	3 Mbps	5 Mbps
Background Traffic	5 Mbps	5 Mbps	5 Mbps

The streaming bitrate of the base layer and two enhancement layers are 1 Mbps, 3 Mbps, and 5 Mbps, respectively. These bitrates are approximate of the size of three different qualities of the layered version of the Big Buck Bunny movie. The background traffic is constant in all scenarios with a bit rate of 5 Mbps. We use Wire shark, a well-known network packet analyzer tool, to validate the flow's data size. In this experiment, we evaluate throughput, jitter, and loss as network QoS metrics.

## Evaluation

In this section, we evaluate LSDN through different scenarios. We compare LSDN with the default routing mechanism embedded in Open Daylight controller. Open Daylight uses shortest path routing mechanism as default routing mechanism. We created one scenario for each video quality where the base layer and enhancement layer are sent to the receiver at time zero for one minute duration, while background traffic is generated at time 20s and continue for 40 seconds. In the following parts, we discuss the impact on LSDN on throughput, jitter, and loss rate of different layers of transmitted streams.

### Throughput

Figure 3 shows the time-varying throughput for the base layer, enhancement layer, and background traffic for our scenarios. In the first scenario, the client streamed a low-quality video file. As we see in Figures 3 (a) and (b), the background traffic that is sent to the network at the time 20s does not change the throughput of transmitted layers in a low-quality case. Also, in this case, the default routing mechanism has a better throughput than applying LSDN. Since all packets are transmitted from senders through the shortest path mechanism, there is enough available bandwidth for transferring those flows and it provides better throughput.

As we see in Figures 3 (c) and (e), the shortest path algorithm has difficulties delivering higher quality video streams after injecting the background traffic. However, Figures 3(d) and (f) represent that LSDN provides a stable throughput for the base layer and enhancement layer. In addition, we observed that there is a delay to transmit high-quality video with the shortest path algorithm. Overall, it shows that LSDN has a better performance when we have bandwidth constraints, and it prevents increasing the transmission time of high-quality video streaming.

### Jitter

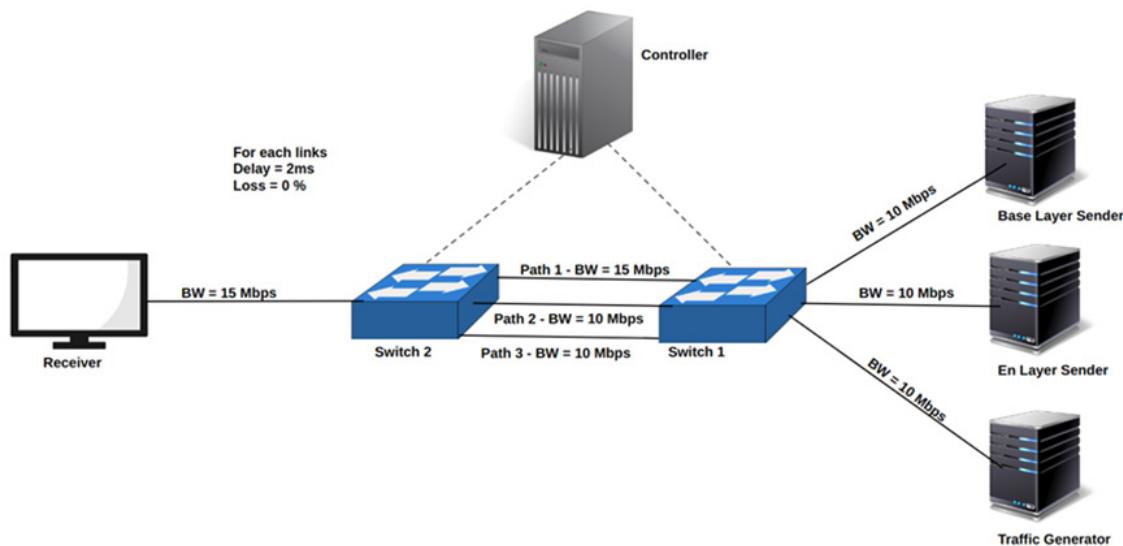
In this section, we evaluate the impact of utilizing LSDN over jitter. Jitter is defined as a variation in the delay of received packets and it is also impacting the quality of experience (QoE) of the streaming experience. In the case of streams with low quality, Figures 4 (a) to (c) show that in low-quality video streaming scenarios, both routing schemes have the same behavior in terms of jitter. But the results are different in mid and high-quality video streaming sessions. As we see in Figures 4 (d) to (i), the video transmissions by using the shortest path solution are suffered in terms of delay variation where the background traffic is generated and all following arrive with a delay. The red lines in Figure 4 show the delay variation when the LSDN scheme is applied. In this scenario, when background traffic

is generated, there is a slight increase in jitter for the base layer and enhancement layer and they keep a lower jitter than the default routing algorithm. In this experiment, when it comes to high-quality video streaming, the SDN-based solution provides better performance in terms of jitter than the default routing solution.

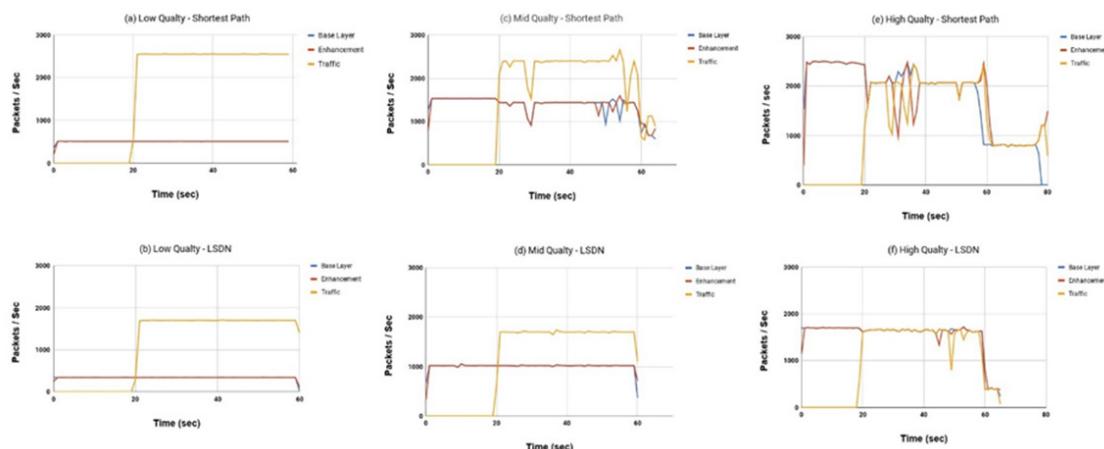
### Packet loss

In this part, we evaluate the impact of the packet proposed algorithm on packet loss ratio. As it is shown in Figure 5 (a), in the

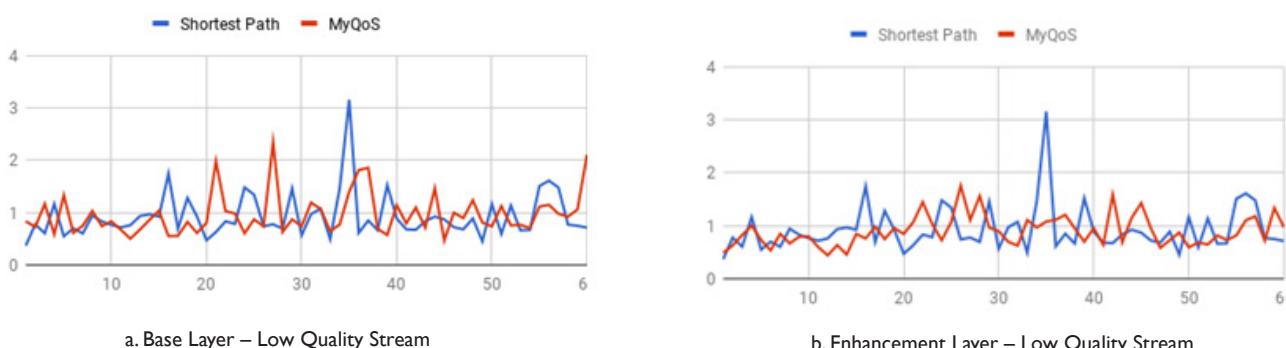
low-quality streaming scenario, there is a negligible packet loss of background traffic when we use LSDN and shortest path algorithms. This shows that our experiment test bed losses a few packets because of its links' capacity. As we see in Figure 5 (b) and (c), there is a significant packet loss when we use shortest path routing. As we expected, LSDN significantly keep packet loss low when the client is streaming high-quality videos. In addition, the results show that LSDN does not impact outstandingly on packet loss of background traffic.

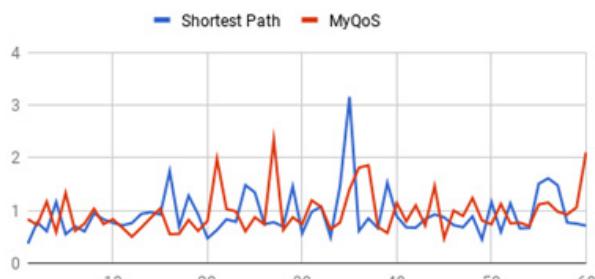


**Figure 2** Our testbed topology.

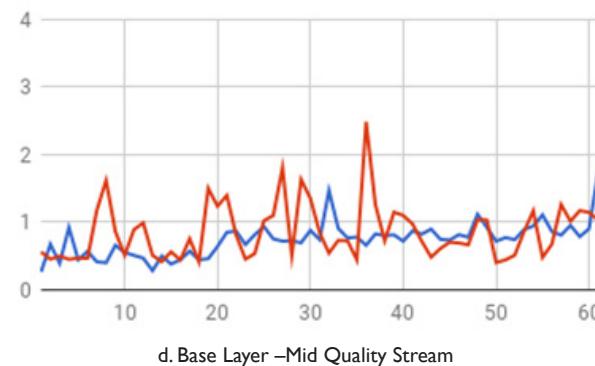


**Figure 3** Observed throughput.

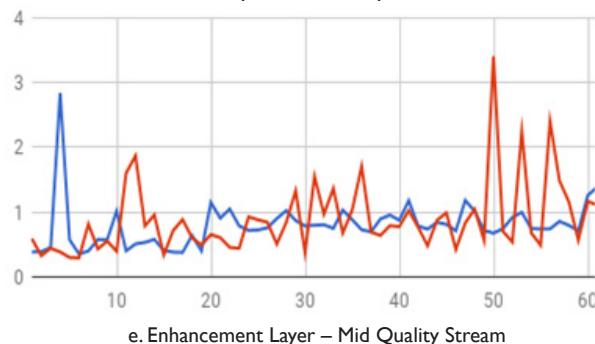




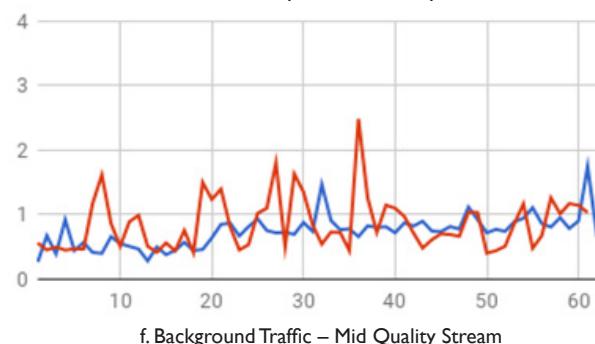
c. Background Traffic – Low Quality Stream



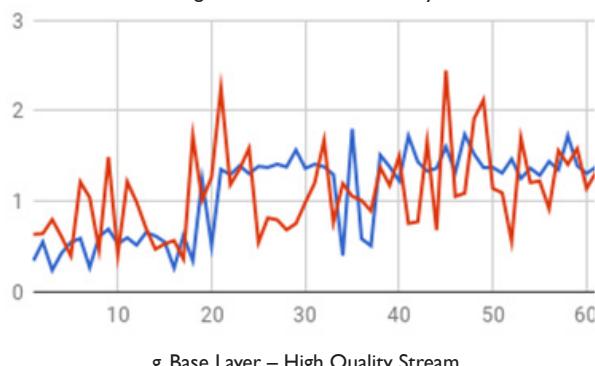
d. Base Layer – Mid Quality Stream



e. Enhancement Layer – Mid Quality Stream



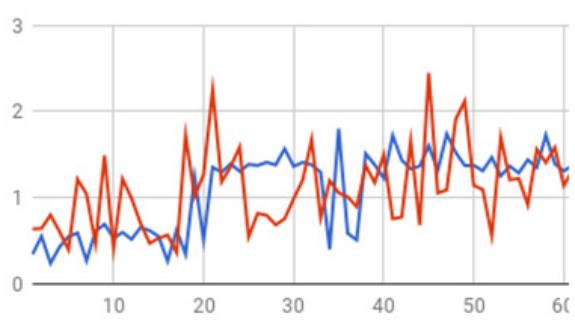
f. Background Traffic – Mid Quality Stream



g. Base Layer – High Quality Stream

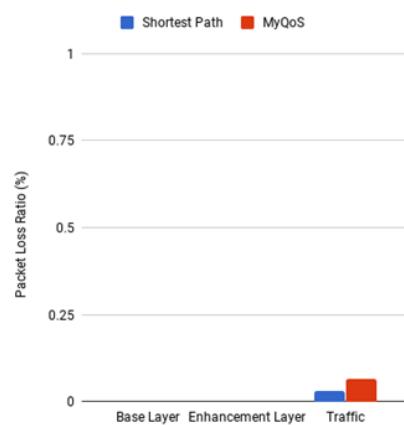


h. Enhancement Layer – High Quality Stream

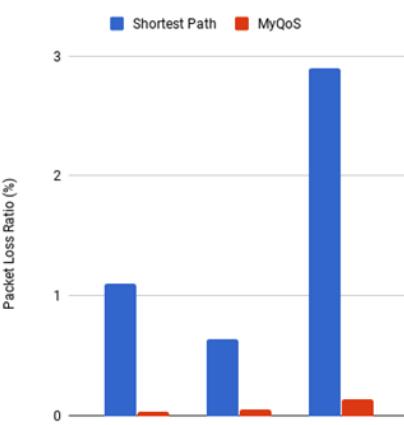


i. Background Traffic – High Quality Stream

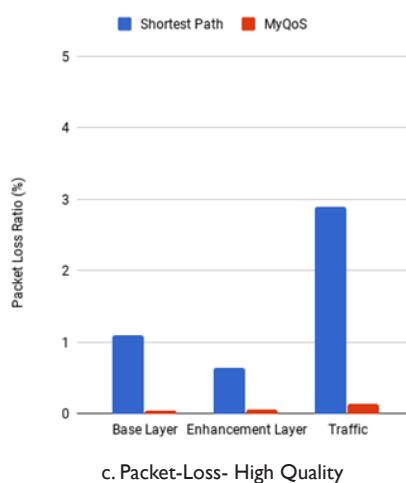
Figure 4 Observed Jitter.



a. Packet-Loss- Low Quality



b. Packet-Loss - Mid Quality



**Figure 5** Packet loss ratio.

## Related work

In recent years, the provision of quality of service (QoS) for multimedia applications such as video conferencing, distance learning, and interactive gaming are becoming prevalent.<sup>17</sup> The efficient delivery of video streams over the Internet meets a lot of challenges due to some unsolved issues such as the network's limited global view, per-hop decisions, and limited QoS abilities for flows<sup>21</sup>. Indeed, there is a need for designing a QoS-based routing framework to cope with varying network conditions in multimedia settings. The SDN is considered a prospective architecture for overcoming these routing problems.

In a multimedia setting, the order of some flows may have more priority than the other flows. This situation has an impact on the QoS, and some studies used traffic shaping concepts to face this situation. Civanlar et al.<sup>22</sup> studies the QoS routing of video streaming over Open Flow networks. They present a formula based on linear programming to reduce packet loss and hold jitter stable in Scalable Video Coding (SVC) streams. The idea is to keep one table of the best paths that are calculated by this formula for the base layer and devote the other typical paths to another type of traffic. HiQoS presents an SDN-based ECMP (Equal Cost Multipath Routing) algorithm to find multiple paths between server and receiver along with using queuing mechanisms to provide bandwidth guarantee for different classes of traffic.<sup>23</sup> It firstly determines different types of traffic and then devotes different bandwidth guarantees to them through queuing mechanisms on the Open Flow switches. The routing is done by a multi-path routing component that finds multiple paths between server and receiver which meet certain QoS constraints and calculates the optimal path for each flow by real-time monitoring of the network state.

Classification of flows and providing service for a different class of flows is exploited for different routing treatments in multimedia networks. Eglimez et al.,<sup>24</sup> presented OpenQoS, an Open Flow controller for video streaming with QoS support. They classify the incoming multimedia flows by checking the packet header field. The multimedia flows are dynamically routed on the QoS-supported paths while other data flows are subject to best-effort routing. In an optimization study, ARVS (Adaptive Routing Video Streaming) for adaptive routing of video packets is introduced.<sup>25</sup> In ARVS, if the shortest path does not satisfy the delay variation constraint, the base layer packets have the priority to be rerouted to a calculated feasible path based on the available bandwidth of this path, and the enhancement layer packets will stay on the shortest path.

## Conclusion

By increasing new multimedia applications that generate diverse flows and require different treatments, there has been a continuing debate on how to evolve the Internet to provide QoS for multimedia traffic. Providing adequate QoS requires some requirements that are not easy to meet with today's networking structure. In this paper, we proposed LSDN, an SDN-based architecture to support QoS to improve delivering high-quality video streams. The combination of the layered characteristics of the SHVC video codec and the features of SDN to dynamically update the forwarding tables, provide a more reliable QoS in comparison to the traditional network QoS solutions. LSDN uses a per-flow-routing approach to route layers with different priorities. LSDN relies on two modules: The flow Manager module that is responsible for detecting QoS-flows, it receives topology information from the Topology Manager module to reserve the best routing path for the base layer and the next best one to the next enhancement layer.

We evaluated the performance of LSDN over the Mininet emulator. Inspection of our experimental results shows that LSDN provides better performance in delivering high-quality layered video streams in terms of throughput, loss, and jitter. Unlike other QoS architectures, LSDN minimized the negative effects (such as packet loss and latency) on background traffic.

## Acknowledgments

The authors thank the IRATJ reviewers for their feedback and suggestions.

## Conflicts of interest

Authors declare that there is no conflict of interest.

## References

1. Keshvadi S, Williamson C. *MoVIE: A Measurement tool for mobile video streaming on smartphones*. In Proceedings of the ACM/SPEC International Conference on Performance Engineering: Edmonton, Canada; 2020. 230–237 p.
2. Cisco Annual Internet Report (2018–2023) White Paper; 2020.
3. Zhang T, Mao S. An overview of emerging video coding standards. *GetMobile: Mobile Computing and Communications*. 2019;22(4):13–20.
4. Ohm J, Sullivan G, Han W, et al. Overview of the high efficiency video coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*. 2012;22(12):1649–1668.
5. Boyce JM, Ye Y, Chen J, et al. Overview of SHVC: Scalable extensions of the high efficiency video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*. 2015;26(1):20–34.
6. Ye Y, Yong H, Wang Y, et al. SHVC: the Scalable extensions of hevc, and its applications. *ZTE Communications*. 2019;14(1):24–41.
7. Blake S, Black D, Carlson M, et al. Rfc2475: An architecture for differentiated service; 1998.
8. Braden R, Clark D, Shenker S. RFC1633: Integrated services in the internet architecture: an overview; 1994.
9. Rosen E, Viswanathan A, Callon R. Multiprotocol label switching architecture; 2001.
10. Karakus M, Durresi A. Quality of service (QOS) in software defined networking (SDN): a survey. *Journal of Network and Computer Applications*. 2017;80:200–218.

11. Kreutz D, Ramos F, Verissimo P, et al. Software-defined networking: a comprehensive survey. *Proceedings of the IEEE*. 2014;103(1):14–76.
12. Marzuki I, Ma J, Ahn Y, et al. A context-adaptive fast intra coding algorithm of high-efficiency video coding (HEVC). *Journal of Real-Time Image Processing*. 2019;16(4):883–899.
13. Mukherjee D, Bankoski J, Grange A, et al. The latest open-source video codec VP9 – an overview and preliminary results. Picture Coding Symposium (PCS): San Jose, CA, USA; 2013. 390–393 p.
14. Shenker S, Partridge C, Guerin R. Specification of guaranteed quality of service. 1997;199 (7):RFC 2212.
15. Internet Engineering Task Force.
16. Li W, Meng W, Kwok L. A survey on OpenFlow-based software defined networks: security challenges and countermeasures. *Journal of Network and Computer Applications*. 2016;68:126–139.
17. Karamchati S, Rawat S, Varma V. A novel architecture to enhance quality of service in IP networks. *IEEE International Conference on Information Networking (ICOIN)*. 2017;616–621.
18. <https://www.openvswitch.org>
19. <http://mininet.org/>
20. <https://iperf.fr>
21. Keshvadi S, Williamson C. An empirical measurement study of free live streaming services. In: *International Conference on Passive and Active Network Measurement*. 2021;111–127.
22. Civanlar S, Parlakisik M, Tekalp A, et al. A qos-enabled openflow environment for scalable video streaming. *IEEE GLOBECOM Workshops (GC Wkshps)*. 2010;351–356.
23. Jinyao Y, Hailong Z, Qianjun S, et al. Hiqos: an SDN-based multipath qos solution. *China Commun*. 2015;12(5):123–133.
24. Egilmez H, Tekalp A. Distributed qos architectures for multimedia streaming over software defined networks. *IEEE Trans. Multimed*. 2014;1597–1609.
25. Yu TF, Wang K, Hsu YH. *Adaptive routing for video streaming with QOS support over SDN networks*. In: 2015 International Conference on Information Networking (ICOIN); 2015. 318–323 p.