2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) | 979-8-3503-3667-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/WiMob58348.2023.10187887

Enhancing Video Streaming Quality of Service with Software-Defined Networking and Network Slicing: A Scalable AV1 Approach

1st Sina Keshvadi
dept. Software Engineering
Thompson Rivers University
Kamloops, Canada
skeshvadi@tru.ca

2nd Yogesh Sharma
Faculty of Engineering and Applied Science
University of Regina
Regina, Canada
yogesh.sharma@uregina.ca

***Abstract*—To meet the rising demand for high-quality video content with diverse quality of service (QoS) requirements, advanced video codecs, such as AV1 and network technologies, such as Software-Defined Networking (SDN) and network slicing, have been developed. While benefiting from these advancements, in this work, we propose AV1-SLICE, a QoS-aware network slicing mechanism that efficiently delivers AV1 video streams at varying resolutions and bit-rates over an SDN network. AV1's open-source and scalable coding feature encodes a single video into multiple layers, which AV1-SLICE manages through traffic shaping, rate limiting, and QoS labels. Through experiments conducted in a network emulation environment using 'Mininet' and with real-world Internet streaming, we demonstrate that AV1-SLICE significantly enhances the delivery of high-quality videos by increasing throughput and reducing jitter and packet loss compared to conventional video streaming methods.**

***Index Terms*—AV1 video codec, QoS, SDN, Network Slicing**

I. Introduction

Video streaming has emerged as a primary source of Internet traffic and is expected to continue its growth in the near future [1]. The increasing demand for high-quality video streaming services has led to the development of several video codecs, such as H.264/AVC, H.265/HEVC, VP9, and AV1. However, delivering high-quality video streams over the Internet is a challenging task due to the limited and variable bandwidth, network congestion, and network topology. This causes high latency, choppy video, and poor video quality [1]. Hence, to provide video streaming services over the Internet while addressing the specified challenges, advanced network technologies, such as Software-Defined Networking (SDN) and Network Slicing are proposed [2], [3]. SDN enables the separation of the control plane and data plane in a network, leading to more efficient network management. Whereas, Network Slicing allows the creation of multiple virtualized networks, each tailored to the needs of a particular application or service.

With the aim of enhancing the user experience of video streaming services by employing SDN and Network Slicing technologies, in this study, we introduce AV1-SLICE. It is a Quality of Service (QoS)-aware network slicing mechanism to deliver video streams with low-latency and high reliability over SDN networks. AV1-SLICE leverages the scalable coding feature of the AV1 video codec [4] to efficiently deliver videos at varying resolutions and bit-rates. The scalable coding capability of AV1 enables the encoding of a single video into multiple layers. Each layer contains different levels of resolution and quality. For efficient resource allocation and to ensure timely and consistent delivery of AV1 video traffic, AV1-SLICE employs traffic shaping, rate limiting, and QoS labeling methods. These methods are to allocate adequate resources and priority to the base layer of AV1 video traffic.

To evaluate the effectiveness of AV1-SLICE, we conducted a series of experiments in a network emulation environment using 'Mininet' [5]. The experiments involved the implementation of our proposed solution and the streaming of video content from the real-world Internet. The performance of AV1-SLICE is measured in terms of throughput, packet loss, jitter, and peak signal-to-noise ratio (PSNR). The obtained results demonstrate increased throughput and reduced packet loss and jitter. Which shows that AV1-SLICE significantly improve the delivery of high-quality videos. Additionally, we evaluated our proposed solution with OpenQoS [6], a well-designed SDN-based traffic shaping approach in adaptive streaming. It is observed that unlike OpenQoS which uses inbuilt optimization mechanisms for PSNR management, our solution does not use any optimization mechanism and provides better and consistent PSNR in the presence of background traffic.

The rest of this paper is organized as follows: Section II provides an overview of background concepts and related works. In section III, the proposed solution is described in detail. The experimental setup is explained in section IV, followed by evaluation of results in section V. Section VI discusses the limitations and future work. Finally, the conclusion is presented in section VII.

II. Background and Related Work

A. Background

The efficient delivery of video streams over the Internet meets a lot of challenges due to some unsolved issues such as network's limited global view, per-hop decisions, and limited QoS abilities for flows. Therefore, there is a need for designing a QoS-based routing framework to cope with varying network conditions in multimedia settings [7]. Two main aspects of video streaming i.e., video codec and networking infrastructure have a significant impact on the quality of video streaming.

The choice of codec can have a significant impact on the perceived quality, file size, and bandwidth requirements of the video stream. AV1 [8] is developed by Alliance for Open Media (AOM) to address the challenges of delivering high-quality video streaming with low latency and high efficiency over the internet. It serves as a royalty-free and open-source alternative to Joint Collaborative Team on Video Coding (JCT-VC) codecs, such as H.264/AVC and H.265/HEVC. AV1 boasts several technical advantages over traditional video codecs [8]. Firstly, it employs a more advanced video compression algorithm known as 'wavefront parallel processing' (WPP) [9], which reduces the amount of data required to represent a video by up to 50% compared to H.265/HEVC. This results in a substantial decrease in the bandwidth needed for video streaming, particularly over low-bandwidth mobile networks. Secondly, AV1 incorporates a scalable coding feature using 'Flexible Macroblock Ordering (FMO)' and 'Adaptive Motion Vector Resolution (AMVR)' [8]. This is for efficient delivery of videos at different resolutions and bit-rates based on receiver and network conditions. Additionally, AV1 is designed to be highly efficient in terms of computation and memory utilization and reduces the resources required for encoding and decoding video streams. This is particularly beneficial for devices with limited resources, such as mobile phones, tablets, and IoT devices [8].

Software-Defined Networking (SDN) [10] is a transformative paradigm in computer networks, which aims to increase the flexibility and programmability in network design and resource management. This is achieved by separating the control and data planes and employing a logically centralized controller, communicating with network devices through standard protocols, such as OpenFlow [11]. The OpenFlow architecture is depicted in Figure , which illustrates the separation of the control and data planes in SDN. The controller communicate with the network devices, such as switches, routers, and firewalls, via OpenFlow protocol. Network slicing is another revolutionary technology developed to facilitate the coexistence of Internet services with different QoS requirements on the same infrastructure [12]. It allows the creation of virtual networks with unique characteristics, set of policies, and performance guarantees [13]. This enhances network resource utilization and improves the ability to cater to the diverse requirements of different applications and services.

B. Related Work

In a multimedia setting, the order of some flows may have more priority than the other flows. This situation impacts the QoS and handled by using traffic shaping. Civanlar et. al [14] studied the QoS routing of video streaming over OpenFlow networks. They present an optimization methodology based on linear programming to reduce packet loss and achieve a stable jitter in Scalable Video Coding (SVC) streams. Eglimez [6] presents OpenQoS, an OpenFlow 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 to the QoS-supported paths while other data flows are subject to best-effort routing. Pilar et al. [15] applied SDN in wireless networks to enhance QoS. While OpenFlow is commonly used to configure forwarding elements in wired SDN environments, the authors used the SDN paradigm to dynamically set up wireless networks and improve their performance.

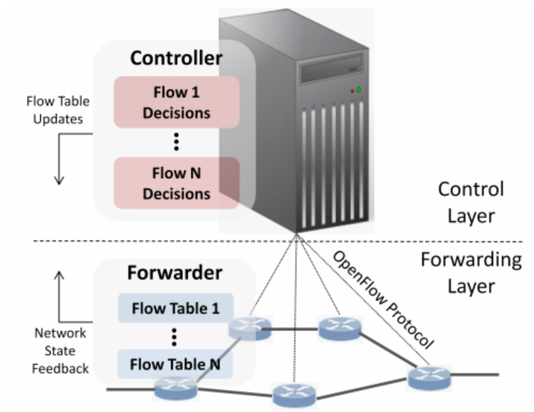


Fig. 1: An overview of an OpenFlow structure [6]

III. AV1-SLICE Architecture

In this section, we introduce AV1-SLICE, a framework for delivering layered video streams over a SDN network. First, we describe its design followed by implementation details.

A. Architecture and Design

Our approach leverages the flexibility and programmability of SDN to establish a reliable path for transmitting the base layer of the AV1 codec, even under adverse network conditions. To achieve this, we have designed and implemented following set of modules within the SDN controller to dynamically allocate network resources and optimize the delivery of video layers.

- **Network Topology Discovery**: This module is responsible for identifying and mapping the physical topology of the network, including devices, links, and their characteristics, such as bandwidth and delay. It employs link-state routing protocols to gather this information and maintains a comprehensive map of the network topology. This information is further utilized to make routing decisions and to establish a reliable path for transmitting the base layer of the AV1 codec.
- **Traffic Monitor**: This module continuously monitors and collect the data on network conditions, such as bandwidth utilization, packet loss, latency and jitter. NetFlow protocol is used to collect the information [16]. The collected

data is then analyzed to identify any network congestion or high latency. This information is then utilized to make real-time adjustments to the network configuration, in order to optimize the performance and ensure a reliable video delivery.

- **Network Slicing**: This module creates the virtual networks each with their own set of policies, Quality of Service (QoS) requirements, and performance guarantees [17]. This enables the efficient use of network resources and improves the ability to meet the diverse needs of different types of applications and services. Based on the network conditions and QoS requirements, this module allocates a reliable network slice for the base layer of AV1 and other slices for the enhancement layers and rest of the traffic. Furthermore, it ensures isolation between slices to prevent the traffic of one slice affecting the traffic of another slice.
- **Main Module**: This module serves as the central hub of the controller. It manages the other modules and makes decisions based on the data collected from the Traffic Monitor module. It uses the shortest path algorithm to compute the best path for the traffic flow based on the network conditions. This module is responsible for setting the routing and forwarding policies to the appropriate network slices based on the network conditions and QoS requirements of each slice.

To meet the end-to-end QoS requirements in video streaming, our proposed mechanism employs a prioritization scheme for flow forwarding. The video stream is divided into multiple layers, arranged in a hierarchical manner, with higher layers dependent on lower layers for decoding. This makes the video stream adaptable to the receiving devices and network conditions, which results in a reliable and low-latency delivery even under variable network conditions.

B. AV1-SLICE Implementation

In this section, we describe the implementation details of our proposed system. We integrated the modules described in Section III-A with OpenDaylight, a widely adopted open-source Java-based SDN controller [18]. OpenDaylight provides a modular programming environment, enabling the operators to selectively run specific modules as per their requirements. We leveraged the functionality and Application Programming Interfaces (APIs) provided by Open vSwitch (OVS) to manipulate flows and flow tables [19]. The Flow Manager module, within the controller, examine new incoming flows based on their Type of Service (ToS) bits [20], and modifies the flow's Output Action in the flow table of the switch to forward the traffic to the appropriate path. This approach ensures that subsequent packets with the same ToS field are forwarded based on the established Output Action.

To emulate the proposed system, we used Mininet, an open-source network emulator that allows to emulate different network devices and creation of virtual networks with various topologies [5]. Mininet supports the OpenFlow and can be used to implement network slicing. First, we created the virtual networks followed by the network slices by using OpenDaylight. Each virtual network was enabled with its own set of policies, QoS requirements, and performance guarantees. The OpenFlow controller was used to configure the switches and routers within the virtual networks to handle the different types of traffic. Mininet's built-in tools were used to monitor, analyze and reconfigure the networks to obtain optimal performance.

IV. Experimental Setup

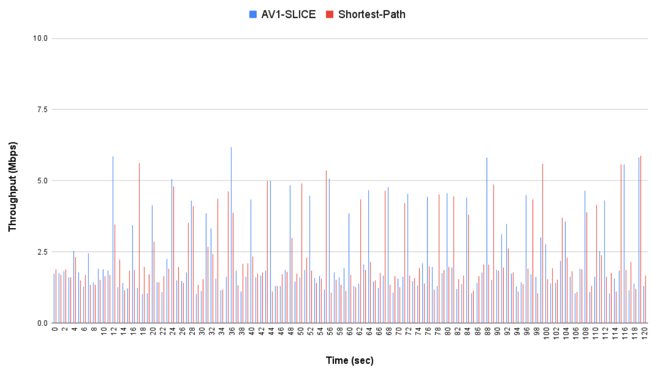
In this study, a Linux-based testbed was established that consisted of a local Mininet network emulator and a video server configured with the Nginx web server on Google Cloud Engine. The video server was deployed on a virtual machine instance with fixed resources, such as CPU, memory, and network bandwidth, to provide a controlled and stable testing environment. The video content was pre-stored on the server, and the AV1 video codec was generated using the FFmpeg tool [21] with raw Big Buck Bunny [22] video files.

The client system was emulated using Iperf3, a widely recognized network performance measurement tool that supports both TCP and UDP data streams [23]. In the experiment, we generated one TCP traffic stream as the background traffic and two video streams consisting of the base and enhancement layer with approximate bit-rates of 1 Mbps and 3 Mbps, respectively for low-quality video streaming; and 2 Mbps and 4 Mbps, respectively for high-quality video. The background traffic was kept constant at 2 Mbps in all scenarios. Wireshark, a network packet analyzing tool, was employed to capture ingress/egress network traffic.

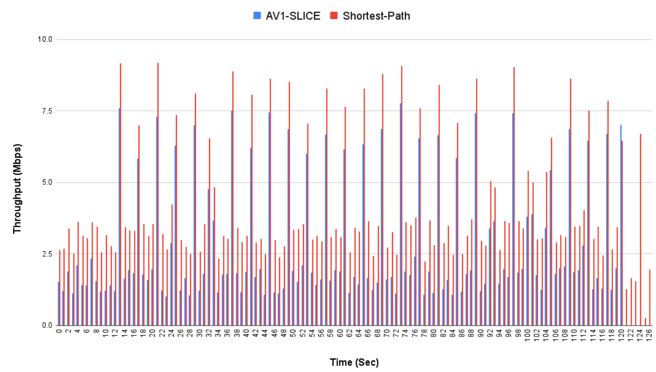
The performance of AV1-SLICE is evaluated by measuring various network parameters, such as throughput, packet loss, and jitter, during the video streaming process. The results are analyzed to assess the efficacy of AV1-SLICE in optimizing QoS for video streaming. The experiment were repeated three times to ensure the reliable and reproducible results. More configuration used in this study for Nginx server, video server, and the experiment can be found in the project's GitHub repository[1].

V. Evaluation

This section evaluates the AV1-SLICE architecture. Our evaluation compares the performance of AV1-SLICE with the default routing mechanism employed by the OpenDaylight controller, which implements a shortest path algorithm. To assess the impact of AV1-SLICE on network performance, we designed experiments that simulate different video quality levels. Each experiment consisted of the transmission of the base layer and one enhancement layer of the video content for a duration of two minutes, with background traffic being generated and sustained during the video streaming. The results and analysis of our experiments are presented in the subsequent sections.



(a) Low-Quality



(b) High-Quality

Fig. 2: Throughput - AV1-SLICE vs Shortest Path

A. Throughput

Figure 2 represents the observed throughput characterizing the video streaming using a browser, which fetches video file chunks to fill the buffer, plays the video, and then fetches the next chunk of data. It is observed that in the low-quality scenario where SDN network architecture had enough capacity to handle the video and background traffic, both solutions provided almost similar throughput (Figure 2(a)). This shows that the performance of default shortest path routing mechanism in the OpenDayLight controller and AVI-SLICE is similar for low-quality scenarios. However, Figure 2(b) demonstrates that the shortest path algorithm struggles to deliver high-quality video streams in the presence of background traffic. Whereas, AV1-SLICE could maintain a stable throughput for both the base layer and enhancement layer. Additionally, we noticed a slight increase in transmission time for high-quality video when using the shortest path algorithm. These results suggest that AV1-SLICE provides better performance in bandwidth-constrained scenarios and significantly reduces the transmission time for high-quality video streaming.

B. Jitter

This section assesses the impact of our proposed solution on jitter, which is defined as the variation in the delay of received packets. This variation can negatively affect the quality of experience (QoE) of video streaming [24]. Our results in Figure 3 demonstrate that AV1-SLICE reduces the jitter of video streams compared to the default routing mechanism in the OpenDayLight controller. This reduction ensures smoother and less delayed transmission of video streams. In low-quality video streaming scenarios (Figure 3(a)), both routing solutions exhibit similar jitter behavior. However, for high-quality video streaming sessions (Figure 3(b)), AV1-SLICE provides a stable jitter performance than the default shortest path routing mechanism. These results suggest that for high-quality video streaming, the SDN-based solution provides more favorable performance in terms of jitter compared to the default routing solution.

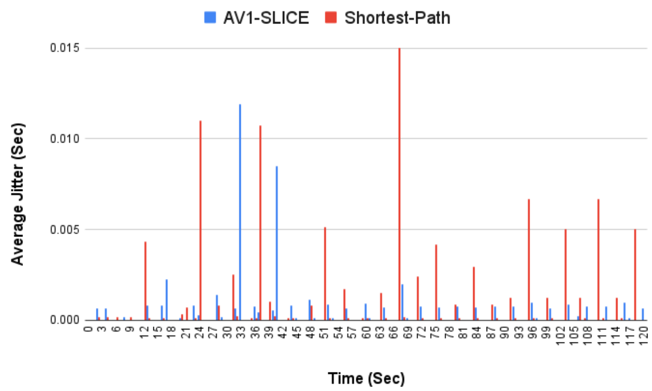
C. Packet Loss

This section evaluates the effectiveness of the AV1-SLICE algorithm in reducing the packet loss ratio. We treated retransmitted packets as lost packets because late-arriving packets in video streaming are not used for display. As depicted in Figure 4(a), in the low quality video streaming scenario, both the AV1-SLICE and shortest path algorithms showed a negligible packet loss ratio due to background traffic. This suggests that the experiment testbed only experienced a limited number of lost packets. This can be attributed to the limited capacity of its network links. However, for high-quality videos (Figure 4(b)), AV1-SLICE clearly outperforms the shortest path routing algorithm by showing significantly lower packet loss ratio. This is because the ability of AV1-SLICE to dynamically adjust the routing of the video streams based on network conditions. Which ensures a more efficient use of the available bandwidth and minimizes the chance of congestion causing packet loss.

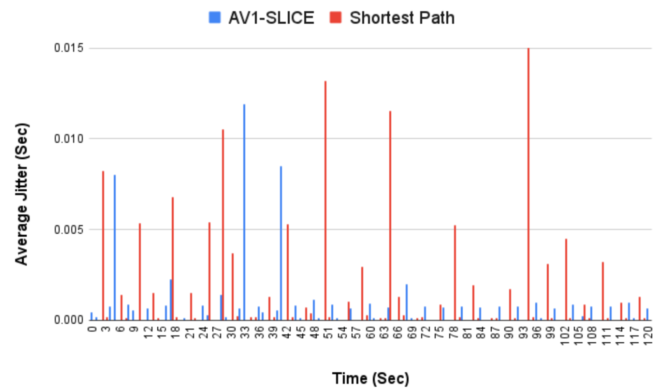
D. PSNR

We conducted a comparative study between AV1-SLICE and OpenQoS [6]. We used the 'in to tree' video from Xiph.org [25] for the experiment. The raw video of the test sequence was obtained and ffmpeg was used to generate a video with the same resolution as the original. We looped the video to create a 40-second video stream at the rate of 1800 kbps. Since we employed TCP for video streaming, we compared our work to HTTP-based Adaptive Streaming described in [6]. We set up the same network configuration as detailed in the paper. In this experiment, we evaluated the quality of the video stream using the Peak Signal-to-Noise Ratio (PSNR) metric and compared it with the results obtained in [6]. PSNR measures the difference between the original and streamed video by calculating the ratio of the peak signal power to the noise power introduced by compression. The higher the PSNR value, the better the quality of the video.

[1] https://github.com/Keshvadi/AV1-SLICE

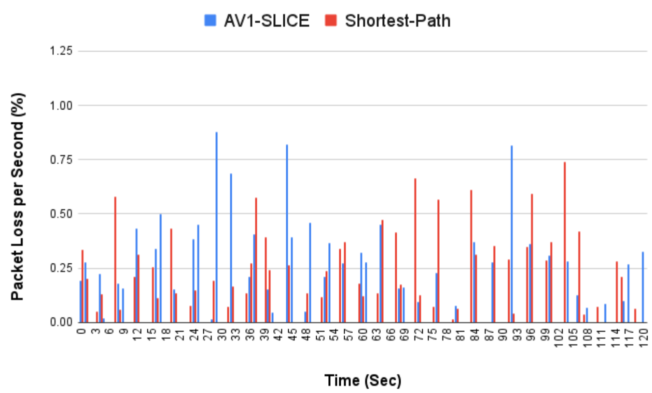


(a) Low-Quality

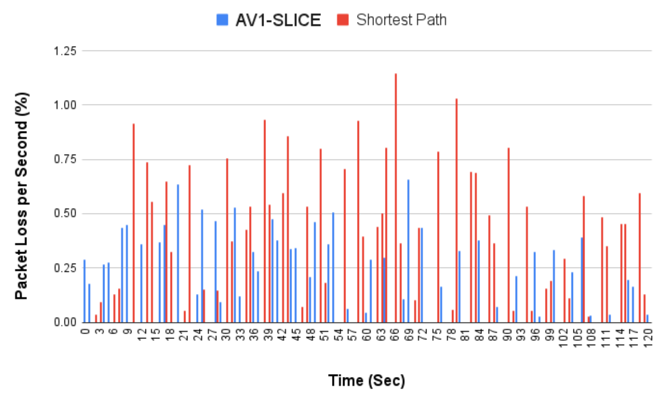


(b) High-Quality

Fig. 3: Jitter - AV1-SLICE vs Shortest Path



(a) Low-Quality



(b) High-Quality

Fig. 4: Packet-Loss - AV1-SLICE vs Shortest Path

Similar to OpenQoS, we introduced the same video in 450 kbps as background traffic in the network from the 10th to the 20th second of video streaming. The video was streamed for 40 seconds. The PSNR results for OpenQoS and our work are shown in Figure 5(a) and Figure 5(b), respectively. Both solutions improved the PSNR in the presence of background traffic. AV1-SLICE showed greater consistency, with only one sharp change in PSNR. It is shown that AV1-SLICE has a better PSNR than the default routing method, even in the absence of background traffic. This is because of two different paths used by AV1-SLICE to send different video layers to the destination which further improved the sending bit-rate.

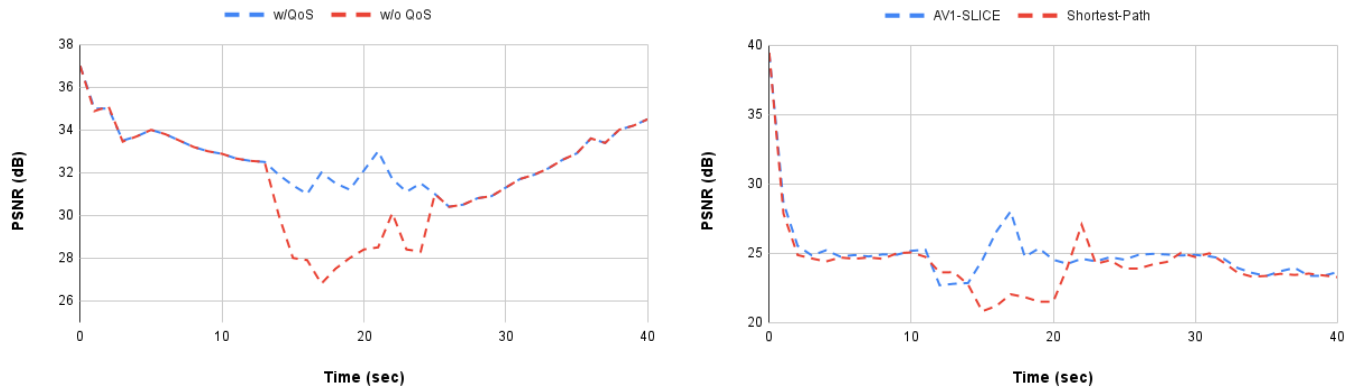
VI. Discussion and Limitations

The obtained results demonstrated that AV1-SLICE algorithm outperforms the default routing mechanism in the OpenDayLight controller. It provides significant improvements in network performance by increasing the throughput and decreasing the jitter and packet loss rate.

It is important to note that our study is limited in the scope. The testbed setup used in this study was limited in size considering a small number of nodes and limited video quality. Further studies are needed to evaluate the scalability of AV1-SLICE in large-scale networks and under high-quality video conditions. Additionally, the AV1-SLICE algorithm and its modules are relatively complex, and more research is required to simplify the implementation while maintaining its performance. These limitations highlight the need for further research to fully evaluate the scalability, complexity, and performance of AV1-SLICE in real-world network scenarios.

VII. Conclusion and Future Work

In this paper, we proposed AV1-SLICE, a software defined networking (SDN)-based architecture to improve quality of services (QoS) in delivering high-quality video streams. The combination of the layered characteristics of AV1 video codec and the virtual slicing features of SDN provide a more reliable QoS in comparison to the traditional network QoS solutions. The obtained evaluation results highlighted the potential of



(a) OpenQoS (b) AV1-SLICE

Fig. 5: PSNR - AV1-SLICE vs OpenQoS [6]

AV1-SLICE as an effective solution for improving the quality of experience for users of video streaming services. AVI-SLICE algorithm outperformed the default routing mechanism in OpenDayLight controller significantly by increasing throughput and decreasing jitter and packet loss rate.

In future, further research is required to evaluate the scalability, complexity, and performance of AV1-SLICE in real-world network scenarios. Additionally, the compatibility of AVI-SLICE with different network architectures and protocols needs to be evaluated besides exploring its potential applications for other types of network services.

References

[1] S. Keshvadi and C. Williamson, "An empirical measurement study of free live streaming services," in *Passive and Active Measurement: 22nd International Conference, PAM 2021, Virtual Event, March 29–April 1, 2021, Proceedings 22*. Springer, 2021, pp. 111–127.

[2] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[3] Y. Sharma, D. Bhamare, N. Sastry, B. Javadi, and R. Buyya, "Sla management in intent-driven service management systems: A taxonomy and future directions," *arXiv preprint arXiv:2208.01218*, 2022.

[4] Y. Chen, D. Mukherjee, J. Han, A. Grange, Y. Xu, S. Parker, C. Chen, H. Su, U. Joshi, C.-H. Chiang *et al.*, "An overview of coding tools in av1: The first video codec from the alliance for open media," *APSIPA Transactions on Signal and Information Processing*, vol. 9, p. e6, 2020.

[5] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010, pp. 1–6.

[6] 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 *2012 Asia Pacific signal and information processing association annual summit and conference*. IEEE, 2012, pp. 1–8.

[7] Z. Wu, Z. Lu, P. C. Hung, S.-C. Huang, Y. Tong, and Z. Wang, "Qamec: A qos-driven iovs application optimizing deployment scheme in multimedia edge clouds," *Future Generation Computer Systems*, vol. 92, pp. 17–28, 2019.

[8] J. Han, B. Li, D. Mukherjee, C.-H. Chiang, A. Grange, C. Chen, H. Su, S. Parker, S. Deng, U. Joshi *et al.*, "A technical overview of av1," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1435–1462, 2021.

[9] Y. Zhao and J. Wen, "Wavefront parallel processing for av1 encoder," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 101–105.

[10] N. Feamster, J. Rexford, and E. Zegura, "The road to sdn: an intellectual history of programmable networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 87–98, 2014.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.

[12] D. Scano, L. Valcarenghi, K. Kondepu, P. Castoldi, and A. Giorgetti, "Network slicing in sdn networks," in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2020, pp. 1–4.

[13] Y. Sharma, M. G. Khan, J. Taheri, and A. Kassler, "Performance benchmarking of virtualized network functions to correlate key performance metrics with system activity," in *2020 11th International Conference on Network of the Future (NoF)*. IEEE, 2020, pp. 73–81.

[14] M. O. Elbasheer, A. Aldegheishem, J. Lloret, and N. Alrajeh, "A qos-based routing algorithm over software defined networks," *Journal of Network and Computer Applications*, vol. 194, p. 103215, 2021.

[15] P. Manzanares-Lopez, J. Malgosa-Sanahuja, and J. P. Muñoz-Gea, "A software-defined networking framework to provide dynamic qos management in ieee 802.11 networks," *Sensors*, vol. 18, no. 7, p. 2247, 2018.

[16] B. Claise, "Cisco systems netflow services export version 9," Tech. Rep., 2004.

[17] W.-K. Chen, Y.-F. Liu, Y.-H. Dai, and Z.-Q. Luo, "Optimal qos-aware network slicing for service-oriented networks with flexible routing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 5288–5292.

[18] J. Medved, R. Varga, A. Tkacik, and K. Gray, "Opendaylight: Towards a model-driven sdn controller architecture," in *IEEE international symposium on a world of wireless, mobile and multimedia networks 2014*. IEEE, 2014, pp. 1–6.

[19] B. Pfaff, J. Pettit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar *et al.*, "The design and implementation of open vswitch," in *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, 2015, pp. 117–130.

[20] P. Almquist, "Type of service in the internet protocol suite," Tech. Rep., 1992.

[21] S. Tomar, "Converting video formats with ffmpeg," *Linux journal*, vol. 2006, no. 146, p. 10, 2006.

[22] T. Roosendaal, "Big buck bunny," in *ACM SIGGRAPH ASIA 2008 computer animation festival*, 2008, pp. 62–62.

[23] M. Mortimer, "iperf3 documentation," 2018.

[24] M. Claypool and J. Tanner, "The effects of jitter on the peceptual quality of video," in *Seventh ACM international conference on Multimedia (Part 2)*, 1999, pp. 115–118.

[25] T. van Rozendaal, J. Brehmer, Y. Zhang, R. Pourreza, and T. S. Cohen, "Instance-adaptive video compression: Improving neural codecs by training on the test set," *arXiv preprint arXiv:2111.10302*, 2021.