Int. j. inf. tecnol. (October 2022) 14(6):2941–2950
https://doi.org/10.1007/s41870-022-01053-1

ORIGINAL RESEARCH

LARIC: latency-aware QoS routing for interactive communication in software defined multimedia

P. Suguna[1] · **R. Leela Velusamy**[1]

Received: 11 March 2022 / Accepted: 28 July 2022 / Published online: 10 August 2022

Abstract With the evolution of the Internet world, the online streaming use cases such as Online classes, Video communications, Peer-to-Peer (P2P), Voice over Internet Protocol (VoIP), distance learning, and online interactive gaming are increasing day by day. The Quality of Service in Online streaming is focused on by many researchers. A big challenge is to provide the Quality of Service for the online user without any buffering. Hence, there is need for QoS aware routing for multimedia transmission to handle the delay sensitive packets. This paper is focused to provide latency-aware QoS routing for Interactive multimedia communication in Software Defined Networks (SDN). The open source Iperf networking tool is used to generate the multimedia traffic in a simulated SDN environment and measure the performance of the networks in terms of bandwidth utilization, packet loss, and delay time. All the traffic packets are transferred to the target host using the proposed latency aware path selection controller application.

Keywords Software defined networking · QoS routing · Multipath routing · RYU · Mininet

✉ P. Suguna
psuguna2020@gmail.com

R. Leela Velusamy
leela@nitt.edu

1 Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu 620015, India

1 Introduction

In recent years, Software-Defined Networking is an emerging technology that gives a dynamic, centralized, manageable, and programmable network by separating the control plane from the data plane [1]. It is the new way of designing and deploying applications frequently. Nowadays video communication is very popular than manual (personal communication). Types of video and audio communications are VoIP, Internet Protocol Television (IP TV), P2P, online classes, and video conferences [2]. A large volume of video data is shared globally on the Internet. Inorder to improve the multimedia communications, Many research challenges are essential to improve the resource utilization, Quality of Service [3], Quality of Experience for the future expected behavior. SDN is an ongoing research area to provide the above mentioned services. Designing and managing network services are possible by using the programmable nature of SDN architecture. The various use cases of online streaming of Internet world is shown in Fig.1.

The current internet architecture is requiring QoS parameters such as bandwidth utilization, reduced delay, packet loss to ensure high speed, and reliable data communications. But in the traditional network [4]- [5], there is no active mechanism to find the end-to-end path which satisfies the above QoS parameters. So, recent technology like SDN architecture is used to enhance the traditional QoS requirements for the end-to-end packet transmission. The proposed work implemented using SDN RYU Controller and Mininet Emulator provides a suitable environment to optimize the quality metric such as bandwidth and delay for real time application. The key contributions of this work are

- Network Monitoring

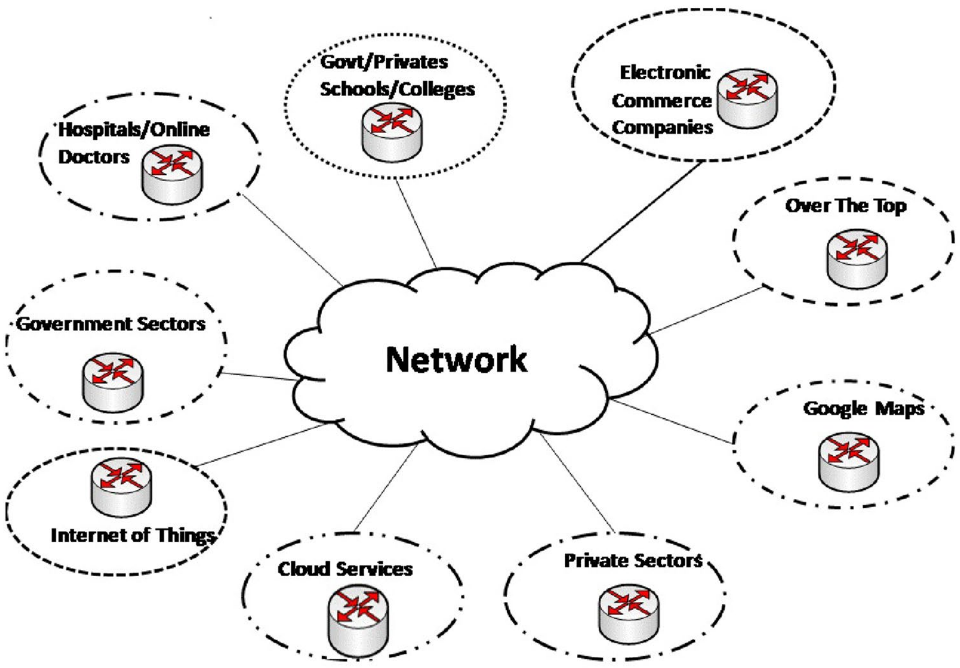


Fig. 1 Online streaming

  - Topology Discovery process using Link Layer Discovery Protocol
  - Converting topology information into Graph using networkx graph library
  - Query all the paths from source switch to destination switch and display all paths
- Calculate the path latency using the latency of each links
- Algorithm to choose the lowest latency path and install the open flows on all the switches
- Simulate the different traffic test using Iperf to analyze the various performance metrics.

The preliminary studies are explained in Sect. 2. Section 3, discuss the related work for Quality of Service models, mechanisms of traditional architecture, QoS parameters and limitations. The implementation of the proposed work is discussed in Section 4. The experiments and simulation results are shown in Sect. 5. Finally, Sect. 6 Concludes the work and the importance of continuing the effort was emphasized.

2 Preliminary study

Differentiated services, the significance of software defined networks, and OpenFlow are all explained in this section. A limited amount of research highlights the value of software defined networks in addressing numerous network issues, including those related to quality of service and effective resource [18] use in traditional networks. A few studies claimed that installing a controller is utilized to address load balancing issues in networks [19, 20].

2.1 Differentiated services (Diffserv):

This is a distributed service model [5] where resources are distributed over the routers in the domain. It doesn’t require any resource reservation mechanism and allows hosts to classify packets into different traffic classes. The priorities are marked in each packet using Differentiated Service Code Point (DSCP) for traffic classification. It performs the admission control based on the statistics traffic classes. This proposed work is used to improve the QoS for different types of traffic based on DiffServ.

2.2 Software defined networks and openflow

The SDN is a new way of designing and controlling the network by extracting the networks control from the dataplanes. The logically centrallized controller is used to build and maintain the IP routing table. The SDN OpenFlow protocols are used to communicate between the controller and dataplane. The OpenFlow supports management protocols like Secure Shell (SSH), Network Configuration Protocol (NETCONF), Simple Network Management Protocol (SNMP), and Representational State Transfer Application Program Interface (REST API). There are many OpenFlow controllers like RYU, Floodlight, and ONOS. RYU controller is used in this work, it is a component based framework for SDN which supports Python. In simple, It provides a platform for the programmer to customize, scale, and automate the networks at any size [21, 22].

3 Related work

A study of existing QoS models based using SDN was published recently [3]. Also, QoS models in Wireless multimedia sensor networks reviewed in [23]. Table 1 summarizes the current state of recent QoS Mechanisms. Also, shows the majority of existing techniques failed to satisfy the QoS metrics, and there is still potential for improvement in terms of Quality of Service Routing and Framework. Few research looked at data-center networks where QoS characteristics are taken into account for resource allocation and management [6–8]. Many types of QoS models [9–11, 14] are designed that are typically used for SDN research. Also, few models are retrieved and categorized for recent QoS aware routing [12, 13, 15, 24] in SDN. According to studies, there are a few basic techniques to establishing QoS Models, including changing routing strategies first and subsequently deploying QoS models using machine learning techniques [25, 26] for Network Traffic Classification.

[16, 17, 27] recent research that a breakthrough intelligent and flexible QoS management system that leverages SDN to dynamically reallocate and release bandwidth for QoS management, reducing the overall delay of all network lines. Reviewed the QoS models for Software Defined Network (SDN) in this work. Lonkar et al. [28] evaluated the VoLTE call quality for audio and video. An important issue in the existing solutions is the combination of different QoS metrics. i.e., the consideration of linear combination metrics which fail to satisfy the QoS requirement for a single application service. The suggested approach relies on the all path technique to discover the lowest latency path among accessible options to withstand network delays by handling delay-sensitive packets.

4 Proposed method

This section explains how to use Software Defined Multipath Routing for real-time applications to send several forms of traffic via the SDN network, such as text, audio, video, and so on. The proposed RYU SDN architecture is shown in Fig. 2. The switch manager speeds up service delivery and makes virtual and physical device provisioning more flexible. An OpenFlow Controller connects and configures network devices such as routers, switches, etc., to identify the optimum routing for application traffic using the OpenFlow protocol. By delivering Link Layer Discovery Protocol (LLDP) packets across SDN switch ports in the network, Link Discovery protocols are used to discover links between SDN switches.

The Packet-in Handler function is used to receive packets from the switches and Flow-mod function is used to transfer the received packets to the Controller. The Packet-out message function is used to forward the controller packets to the specified port. The Topology manager module is having information about the global view of the network. Flow rule function is used to place the flow rules to take the forwarding decision once the path was calculated. Latency-aware path selection module is implemented in Algorithm 3 to select the lowest latency path. The proposed approach consists of 2 phases as discussed below:

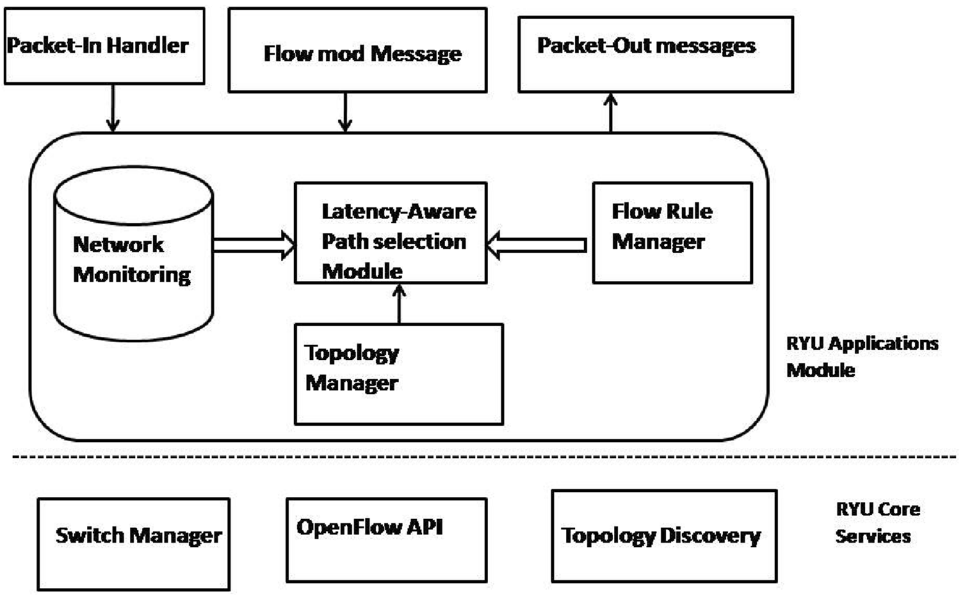


Fig. 2 Proposed controller architecture

Table 1 Related work

Refs.	Methodology	QoS Parameters	Traffics used	Results	Limitations
[6]	Multipath Routing	Loss rate, psnr, and end-to-end delay	Video streaming	Find the optimum path for video streaming to improve QoS routing	No other traffic considered
[7]	Multipath Routing	Server response time, throughput and delay	Different types of traffic	The controller's determination of the best path results in high throughput, as well as reduced transmission delays and server response time	There is no information on traffic types.
[8]	Multipath Routing	Bandwidth and Delay	Video	Find the data center network's quickest sparse route that minimizes packet delays for various sessions	No audio or other sorts of traffic are considered.
[9]	QoS Monitoring	Packet loss, delay and jitter	VoIP	Automatically detect and locate VoIP traffic quality issues	Only a few VoIP calls were tested via UDP.
[10]	Queue Management	Delay	VoIP	Prioritize VoIP packets to lower the number of congested users on the network, which increased the number of network-supported high-quality conversations	To create a more equitable algorithm to stop call drops and preserve good call quality
[11]	SIP Architecture	Delay, jitter and response time	VoIP	Find a load-balancing path that produces high throughput, efficiency and a quick reaction time	To create a mathematical model for SIP that supports encrypted traffic like SIP over TLS and uses delay-optimized routing.
[12]	Resource Reservation	Bandwidth	NIL	Switch routing approach that maximizes connection utilization and reduces the amount of entries in the switches	Suggest a rule insertion approach for TCAM space load balancing
[13]	Multipath Rerouting	Recovery time and packet loss	NIL	Rerouting reduces memory requirements by decreasing the number of backup paths, resulting in fewer extra flow rules on average	There is no information on traffic types
[14]	QoE Aware Model	Delay, packet loss and bandwidth	VoIP	To increase the VoIP quality of experience, use the optimal interface for managing network traffic on multi-homed end-hosts	In wireless networks, only the client side solution is considered (no network based solution)
[15]	Multipath Routing	Delay and bandwidth	Text, Video,Audio	Use the OpenContrail controller to reduce delay time and connection utilization in an OpenStack context to improve QoS	The packetloss and other parameters were not taken into account
[16]	Batch Routing	Delay and bandwidth	Video	A batch routing for video streaming was built in greedy based iteration using centralized architecture to reduce delay and boost network capacity	There is no audio or other sorts of traffic, and there are no other metrics like packet loss
[17]	Resource Reservation	Bandwidth and link availability	VoIP, FTP, and Video Streaming	Design a QoS management strategy for efficient resource allocation in SDN to reduce latency, jitter, and packet loss.	To deal with offloading and link aggregation in order to improve performance

1. Network monitoring
2. Latency measurement-routing

4.1 Network monitoring

Network Monitoring is an important task and helps the operator to manage the network and its Components. Also, it provides the overall network status behavior to the programmer to reduce the operation overheads by taking efficient solutions. Next, quality of service depends on the network monitoring to choose the correct decision.

4.2 Topology discovery

SDN uses a virtual network as a directed graph network G(N, E) in which all the devices are connected through the software. In SDN based OpenFlow monitoring, Link Layer Discovery Protocol (LLDP) [29, 30] packets are used to discover the switches and links. After establishing the topology view, the application can do the following activities such as finding the paths for flows, make the forwarding control decision, also find an alternate path in case of congestion. The controller is generating an LLDP packet every 30 seconds to discover the topology information like add link, remove link, links go up, and link go down. Topology database is maintained to store the LLDP information [31]. Controller using the OpenFlow messages like PACKET _IN message, PACKET _OUT for the topology discovery process.

Topology to Graph: The Networkx is a graph library for networks and used to create a graph as topology, add a node as a switch, and add edge as a link. i.e., networkx receives all of this topological data. Networkx is updated once the topology discovered. All topology information stored in the database. Finally, get all the paths from source switch to destination switch using this Networkx.

Algorithm 1 Function to Calculate Packet Latency

```
1: Function calculate pktlatency(self)
2: for sender in self.latency db do
3:     for receiver in self.latency db[sender] do
4:         no_of_samples=len(self.latency_db[sender][receiver][latency])
5:         total_latency=sum(self.latency_db[sender][receiver][latency])
6:     end for
7: end for
8: if no_of _samples >= 1 then
9:     self.latency_db[sender][receiver][avg_latency]= total_latency/no_of _samples
10: else
11:     self.latency _db[sender][receiver][avg_latency]=0
12: end if
```

Algorithm 2 Function for Path Latency

```
1: total_pathloss = 0
2: total_latency = 0
3: length =len(path)
4: for i = 0 to $(length - 1)$ do
5:     sender = path[i]
6:     receiver = path[i+1]
7: end for
8: avg_latency = self.latency_db[sender][receiver]["avg_latency"]
9: total_latency = total_latency + avg_latency
10: return total_latency
```

Algorithm 3 Pseudocode for Find Shortest Latency Path

```
Input: Graph,Source ports,Destination Ports
Output: Find the lowest latency path Process:
    all _simple_paths()
1:  Get the Switch connected to Source Host
2:  Get the Switch connected to Destination Host
3:  Get the shortest path between the switches
4:  for paths in shortpaths do
5:      path latency=get path latency(shortpaths)
6:      if lowest latency==None then
7:          lowest latency=path latency
8:          continue
9:      else
10:         if path latency<lowest latency then
11:             lowest latency=path latency
12:             lowest path=shortpaths
13:         end if
14:     end if
15: end for
16: shortpaths= lowest path
17: Repeat step 3 to step 16 until the lowest
    latency path is selected.
18: return lowest latency path
```

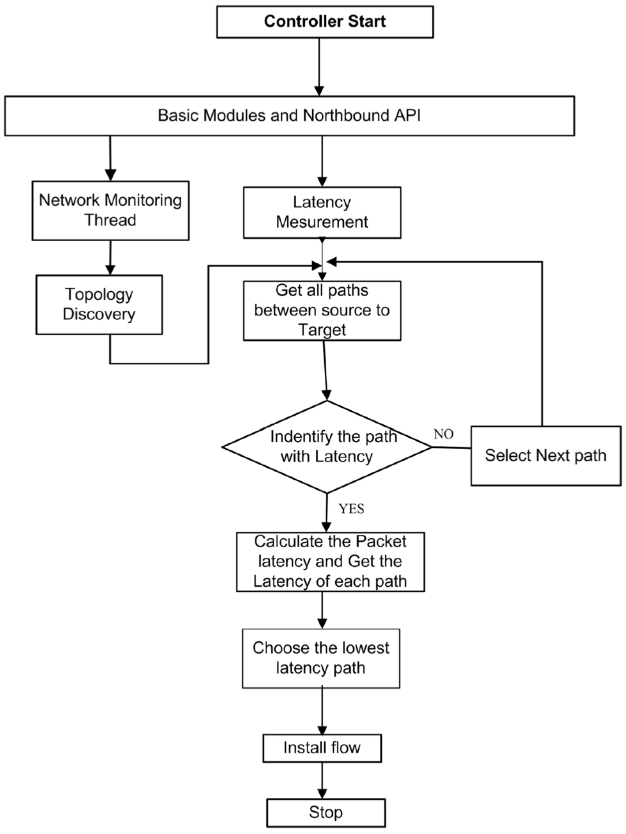

Controller Start
Basic Modules and Northbound API
Network Monitoring Thread
Latency Mesurement
Topology Discovery
Get all paths between source to Target
Indentify the path with Latency
NO
Select Next path
YES
Calculate the Packet latency and Get the Latency of each path
Choose the lowest latency path
Install flow
Stop

Fig. 3 Block diagram of the Proposed Method (LARIC)

4.3 Calculation of latency-aware path in LARIC

This section deals with the design of the proposed latency based multipath QoS routing. The work flow of the proposed framework is shown Fig.3. There are two thread function used in this work. First thread is used to monitor network current information. The Second thread is calculating the latency of each path. The controller monitors the network state continuously so that, it helps to identify the optimal path for each data type. The performance of flow services is analyzed and transmitted in a virtual network through multipath routes. This proposal focuses on all_simple_path provisioning to address the issue of delay sensitive data in real-time applications. In this case, each node needs to have availability of all paths between the source to destination. So the possibility of a path is a direct relation with the nodes reliability, hence the failure of the node causes the nonfunctional path. However, on these terms, this is represented as a real-time problem that requires networks to be more trustworthy in order for packets to be successfully transmitted.

Algorithm 1 Explains the calculation of single packet latency which is calculated between the source host and target host in the topology. Also, the dictionary database is maintained between the hosts to store the updated information for each link in every 10 seconds. The sum of all the packet's latency gives the total latency of packets. The average packet latency is calculated by dividing the total latency by the number of samples which is described in the packet latency function.

Algorithm 2 Describes calculation of the total path latency by adding the sum of all the link latency between source to destination. The packet latency is used to find the path latency between source to destination. Path latency is calculated by adding the average packet latency with total latency.

Algorithm 3 illustrate the lowest latency path. From the networkx library, it uses the shortest simple path function for calculating all_ simple_paths between source to destination host. Choose the lowest latency path to add flow to the corresponding switch.

The following steps are implemented to calculate the lowest latency path in this module.

- SDN Controller using special ping packet measure the latency of each link.

- Special IP & MAC (1.1.1.1 , 11:11:11:11:11:11) is used as source & destination IP and macs.
- In this ping packet, datapath id (originated switch id), generation timestamp, a sequence number are included. On reception of this packet, the receiver datapath id, receiving timestamp will be noted by the controller.
- Latency is identified by differentiating the receiving timestamp and sending timestamp.
- This ping packet is sent to all switches by SDN Controllers at regular intervals. As a result, the latency is updated every 10 seconds.
- Install the OpenFlow flows on all the switches through chosen path.

The routing parameters like bandwidth, delay time are calculated between source to destination using different types of dynamic testing and monitoring tools [6, 7]. This topology is using all_simple_paths Networkx to generate a list of all paths between source to target. By managing delay sensitive packets, the generated pathways make the delay tolerant.

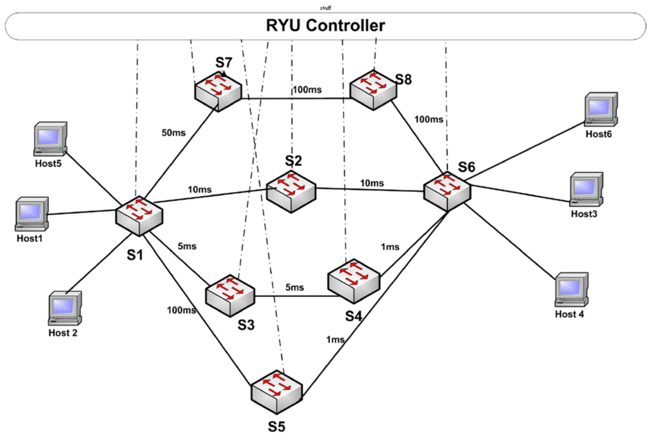
5 Experiment setup and performance results

This experiment was conducted on the Linux host of Ubuntu 18.04.5 Bionic Beaver Operating System. The proposed approach was experimented in the SDN environment using the RYU Controller, OpenVswitch and the Mininet version 2.2 [32]. A network topology is designed with 8 switches and 6 hosts connected to the SDN Ryu controller. Fig. 4. shows the configuration of the simulation. Host addresses are like 192.168.1.1, 192.168.1.2. etc, 8 switches configured with different bandwidths and delay in this simulation. The objective of the proposed module finds the optimal paths that ensure the QoS for multimedia applications by decreasing the delay in an SDN environment. The outcomes of the experiment were compared with the other existing applications like as l4 SDN switch application [33], the single path routing application [15], and Efficient Resource allocation (ERA) [17]. The outcomes are shown in Figs. 5 and 6 . The proposed methodology clearly has a shorter delay time than existing QoS methods since it solves the latency problem in multimedia transmission.

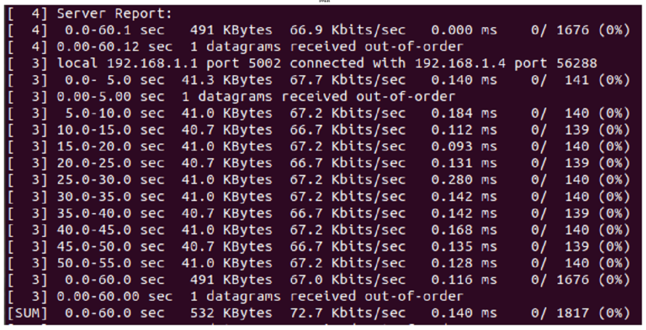
5.1 Throughput analysis for transport layer traffic

This section describes the network performance of the proposed LARIC model. In this section, TCP flows are added to test the transport layer traffic experiments by utilizing the emulated topology and SDN switch application. Finally, the results are compared to those obtained using the LARIC application. Iperf is a open source network traffic generation tool to perform different traffic tests and generally works in server-client mode that gives more accurate results. The emulation is done by sending the 10Mbps of TCP traffic between the assumed server (Host4) and client (Host1) to measure the bandwidth for every 5ms of interval. The TCP flows pass through the switches s1,s3,s4,s6 to reach the destination with maximum link utilization and the results of emulations are shown in Fig. 5. The DumpFlow and Dumpport messages are used to examine the traffic of OpenFlow switches in both directions in order to regulate the flow and monitor the QoS parameters such as Latency, jitter, and packet loss.

The amount of bandwidth used out of the total allocated is referred to as throughput. The network's throughput is measured in megabits per second (Mbps). The performance of the proposed LARIC is found to be more acceptable based



(a) Experiments Topology



```
[  4] Server Report:
[  4]  0.0-60.1 sec   491 KBytes  66.9 Kbits/sec   0.000 ms    0/ 1676 (0%)
[  4] 0.00-60.12 sec  1 datagrams received out-of-order
[  3] local 192.168.1.1 port 5002 connected with 192.168.1.4 port 56288
[  3]  0.0- 5.0 sec  41.3 KBytes  67.7 Kbits/sec   0.140 ms    0/  141 (0%)
[  3] 0.00-5.00 sec  1 datagrams received out-of-order
[  3]  5.0-10.0 sec  41.0 KBytes  67.2 Kbits/sec   0.184 ms    0/  140 (0%)
[  3] 10.0-15.0 sec  40.7 KBytes  66.7 Kbits/sec   0.112 ms    0/  139 (0%)
[  3] 15.0-20.0 sec  41.0 KBytes  67.2 Kbits/sec   0.093 ms    0/  140 (0%)
[  3] 20.0-25.0 sec  40.7 KBytes  66.7 Kbits/sec   0.131 ms    0/  139 (0%)
[  3] 25.0-30.0 sec  41.0 KBytes  67.2 Kbits/sec   0.280 ms    0/  140 (0%)
[  3] 30.0-35.0 sec  41.0 KBytes  67.2 Kbits/sec   0.142 ms    0/  140 (0%)
[  3] 35.0-40.0 sec  40.7 KBytes  66.7 Kbits/sec   0.142 ms    0/  139 (0%)
[  3] 40.0-45.0 sec  41.0 KBytes  67.2 Kbits/sec   0.168 ms    0/  140 (0%)
[  3] 45.0-50.0 sec  40.7 KBytes  66.7 Kbits/sec   0.135 ms    0/  139 (0%)
[  3] 50.0-55.0 sec  41.0 KBytes  67.2 Kbits/sec   0.128 ms    0/  140 (0%)
[  3]  0.0-60.0 sec   491 KBytes  67.0 Kbits/sec   0.116 ms    0/ 1676 (0%)
[  3] 0.00-60.00 sec  1 datagrams received out-of-order
[SUM]  0.0-60.0 sec   532 KBytes  72.7 Kbits/sec   0.140 ms    0/ 1817 (0%)
```

(b) Statistics results of simulation of G7-11 call

Fig. 4 Topology and statistics results

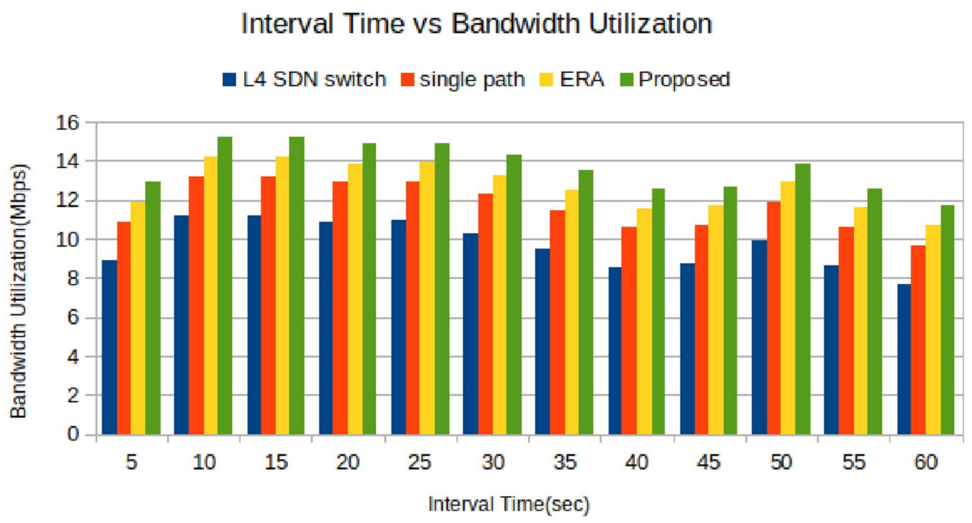
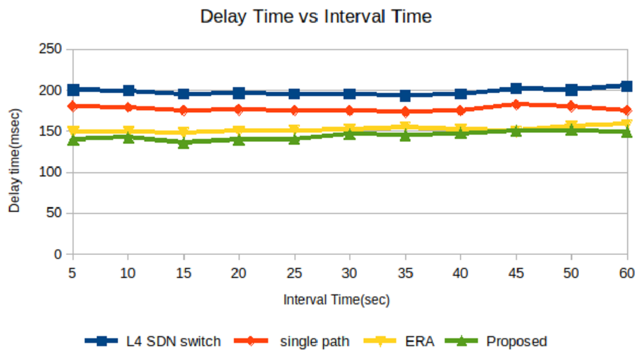
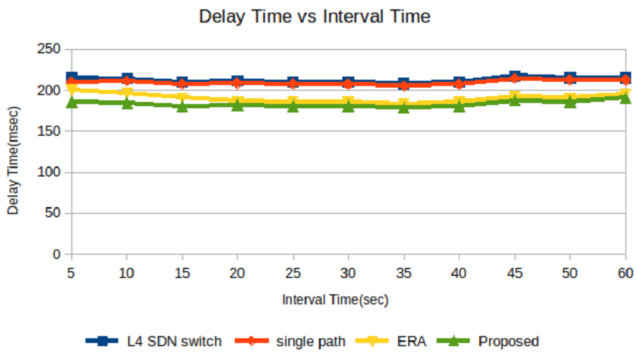


Fig. 5 Throughput of SDN-based application



(a) Delay time result for VoIP



(b) Delay time result for Video

Fig. 6 Delay time of SDN-based application

on the experimental results and demonstrated in Fig. 5. The bandwidth consumption of the network in the suggested technique was improved due to the quality metrics while sending the TCP traffic. LARIC outperforms l4 SDN switch and other existing works [15, 17] because it considers both all simple lowest latency paths, as well as link delay and bandwidth quality measurements, and only chooses QoS-rich paths. In comparison to l4 SDN switch, shortest path, and ERA, the proposed solution enhanced average bandwidth utilization by 4.00%, 2.0%, and 1.0%, respectively.

5.2 Delay time analysis for application layer traffic

In this section Video streaming and VoIP flows are tested by giving types of service value for audio and video packets. For both traffic the packets arrive in the reporting interval of every 5 seconds. If any variations in the packet arrival time, lead to packet loss, network congestion and degrade the quality of the transmission of video and audio packets. The end-to-end delay is the amount of time it takes for a packet to travel across a network. The end-to-end latency is expressed in milliseconds. For example: In this section, the VoIP applications are simulated in a single call. For 60 seconds generate the stream of UDP packets to test the connection between VoIP endpoints. i.e., By simulating G711 (Codec bit rate) call between two endpoints with RTP stream of 30ms frame size, the bandwidth of 67kBits, buffer length of 300 bytes, Characteristics of VoIP, and reporting interval of 5 seconds. Fig. 4. shows the resulting statistics of simulated G711 calls between VoIP endpoints. This single stream simulation test result shows that 532 KBytes of data transfered in 0.140ms jitter with an average bandwidth of 72.7Kbits per second.

The average end-to-end delay as a function of the number of time intervals employed is shown in Fig.6a and b. Because SDN switch does not incorporate quality measurements when calculating the simple and lowest latency path, it causes more delay than single path and LARIC. The graph shows that SDN switch takes VoIP and video traffic into account in the same way. However, single path and

LARIC approach handles these two applications in different ways. LARIC, on the other hand, accomplishes this with minimal overhead at the networking devices. Despite the fact that LARIC does it at the expense of controller overhead in computing QoS-aware latency pathways, it is only algorithmic complexity and so meets the criteria. The graph also shows that as the size of the network grows, so does the latency. This could be owing to the fact that as the number of nodes in the network grows, the path length grows as well. Although l4 SDN switch application both for audio and video giving similar results. LARIC resulted in 4% improvement with respect to single path application, 5% with respect to ERA and 19% with respect to SDN l4 switch.

6 Conclusion and future work

In this work, latency-aware all-path QoS routing for interactive multimedia communication in SDN has been proposed. The proposed work is used to reduce the packet delay for different types of traffic in the OpenFlow SDN environment. As the number of different types of traffic grows, so does the communication overhead required to determine the best QoS solution. This framework strengthens a QoS algorithm and counts the number of reliable multimedia information transfers that travel via the SDN framework.

Low latency network communications were the emphasis of the proposed LARIC. Among all the paths, it finds the one with the shortest latency. As a result, network communications speeds up. In the future, the proposed work will be enhanced with more QoS criteria that will be applicable to future networks. i.e.,It will be utilized in future 5G networks [4, 34] for V2X communications and low latency applications. Low latency enables real-time interactivity in faraway regions like Telesurgery and Virtual reality gaming.

References

1. Kreutz D, Ramos FM (2014) Software-defined networking: a comprehensive survey. Proc IEEE 103(1):14–76
2. Karl M, Gruen J, Herfet T (2013) Multimedia optimized routing in openflow networks. In: 2013 19th IEEE International Conference on Networks (ICON). IEEE, pp. 1–6
3. Karakus M, Durresi A (2017) Quality of service (qos) in software defined networking (sdn): a survey. J Netw Comput Appl 80:200–218
4. 5Gnetworks, "Lowlatencyusecases," https://www.telit.com/blog/5g-low-latency applications. Accessed 7 Jan 2022
5. Services D, "Traditional qos," https://datatracker.ietf.org/doc/html/rfc2474. Accessed 28 Dec 2021
6. Thi TM, Huynh T, Hwang W-J (2015) Qos-enabled streaming of multiple description coded video over openflow-based networks. Nonlinear Theory Appl IEICE 6(2):144–159
7. Yan J, Zhang H, Shuai Q, Liu B, Guo X (2015) Hiqos: an sdn-based multipath qos solution. China Commun 12(5):123–133
8. Liu Y, Niu D, Li B (2016) Delay-optimized video traffic routing in software-defined interdatacenter networks. IEEE Trans Multimedia 18(5):865–878
9. Thorpe C, Hava A, Langlois J, Dumas A, Olariu C (2016) imos: Enabling voip qos monitoring at intermediate nodes in an openflow sdn. In: 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW). IEEE, pp. 76–81
10. Olariu C, Zuber M, Thorpe C (2017) Delay-based priority queueing for voip over software defined networks. In: (2017) IFIP/IEEE Symposium on Integrated Network and Service Management (IM). IEEE. 652–655
11. Montazerolghaem A, Moghaddam MHY, Leon-Garcia A (2017) Opensip: toward software-defined sip networking. IEEE Trans Netw Serv Manage 15(1):184–199
12. Zhang SQ, Zhang Q, Tizghadam A, Park B, Bannazadeh H, Boutaba R, Leon-Garcia A (2017) Tcam space-efficient routing in a software defined network. Comput Netw 125:26–40
13. Senthilkumaran N, Thangarjan R, Nivetha S (2019) Memory and load-aware traffic rerouting (mltr) in openflow-based sdn. In: 2019 TEQIP III Sponsored International Conference on Microwave Integrated Circuits, Photonics and Wireless Networks (IMICPW). IEEE, pp. 409–413
14. Al-Najjar A, Layeghy S, Portmann M, Indulska J (2018) Enhancing quality of experience of voip traffic in sdn based end-hosts. In: 28th International Telecommunication Networks and Applications Conference (ITNAC). IEEE 2018:1–8
15. Venkatesh K, Srinivas L, Krishnan MM, Shanthini A (2019) Qos improvisation of delay sensitive communication using sdn based multipath routing for medical applications. Futur Gener Comput Syst 93:256–265
16. Saadatpour M, Shabanian T, Behdadfar M, Osgouei AG (2021) Qos improvement in sdn using centralized routing based on feedback," In: 2021 International Conference on Information Networking (ICOIN). IEEE, pp. 132–136
17. Al-Harbi A, Bahnasse A, Louhab FE, Talea M (2021) Towards an efficient resource allocation based on software-defined networking approach. Comput Electr Eng 92:107066
18. Bensalah F, El Hamzaoui M, Bahnasse A et al (2018) Behavior study of sip on ip multimedia subsystem architecture mpls as transport layer. Int J Inf Technol 10(2):113–121
19. Banerjee A, Hussain D (2022) Exprl: experience and prediction based load balancing strategy for multi-controller software defined networks. Int J Inform Technol 14:1–15
20. Bhowmik CD, Gayen T (2022) A technique for topography aware dynamic controller placement in sdn. Int J Inform Technol 14:1–13
21. Keti F, Askar S (2015) Emulation of software defined networks using mininet in different simulation environments. In: 2015 6th International Conference on Intelligent Systems, Modelling and Simulation. IEEE, pp. 205–210
22. DeCusatis C, Carranza A, Delgado-Caceres J (2016) Modeling software defined networks using mininet. In: Proc. 2nd Int Conf Comput Inf Sci Technol Ottawa, Canada, no. 133, pp. 1–6
23. Chiwariro R et al (2022) Quality of service aware routing protocols in wireless multimedia sensor networks: survey. Int J Inform Technol 14:1–12
24. Henni DE, Ghomari A, Hadjadj-Aoul Y (2020) A consistent qos routing strategy for video streaming services in sdn networks. Int J Commun Syst 33(10):e4177
25. Wang P, Lin S-C, Luo M (2016) A framework for qos-aware traffic classification using semi-supervised machine learning in sdns. In: IEEE international conference on services computing (SCC). IEEE 2016:760–765
26. Sendra S, Rego A, Lloret J, Jimenez JM, Romero O (2017) Including artificial intelligence in a routing protocol using software defined networks. In: 017 IEEE International Conference

on Communications Workshops (ICC Workshops). IEEE, pp. 670–674

27. Jia W-K, Chou Y-Y, Chen Y-C (2020) Qos improvement of voip over sdn. In: IEEE 17th Annual Consumer Communications & Networking Conference (CCNC). IEEE 2020:1–6
28. Lonkar SA, Reddy K (2022) Analysis of audio and video quality of voice over lte (volte) call. Int J Inform Technol 14:1–14
29. Popic S, Vuleta M, Cvjetkovic P, Todorović BM (2020) Secure topology detection in software-defined networking with network configuration protocol and link layer discovery protocol. In: International Symposium on Industrial Electronics and Applications (INDEL). IEEE 2020:1–5
30. Zacharis A, Margariti SV, Stergiou E, Angelis C (2021) Performance evaluation of topology discovery protocols in software defined networks. In: 2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). IEEE, pp. 135–140
31. Wazirali R, Ahmad R, Alhiyari S (2021) Sdn-openflow topology discovery: an overview of performance issues. Appl Sci 11(15):6999
32. Kaur K, Singh J, Ghumman NS (2014) Mininet as software defined networking testing platform. In: International Conference on Communication, Computing & Systems (ICCCS), pp. 139–42
33. OpenFlownetwork, "Ryuapplications," https://ryu-sdn.org. Accessed 27 Jan 2022
34. Kaur K, Kumar S, Baliyan A (2020) 5g: a new era of wireless communication. Int J Inf Technol 12(2):619–624