

A New Architecture to Improve Multimedia QoS over Software Defined Networks

Mohammad Reza Parsaei
SDN Laboratory, Department of
Computer Engineering and IT,
Shiraz University of Technology,
Shiraz, Iran

Mahshid Naderi
SDN Laboratory, Department of
Computer Engineering and IT,
Shiraz University of Technology,
Shiraz, Iran

Reza Javidan
SDN Laboratory, Department of
Computer Engineering and IT,
Shiraz University of Technology,
Shiraz, Iran

ABSTRACT

Nowadays operators are forced to struggle with low level configurations and exclusives of network devices vendors to establish and maintain a secure connection and implement complex management policies on these devices. Software Defined Networks (SDN) are a new style of computer networks which provides new facilities and methods for network management and configuration by separating control level and data level. In order to improve the video Quality of Service (QoS) and increase the efficiency of the end-to-end performance, it is necessary to select the best route and implement QoS for videos, whilst there may be different routes between the transmitter and the receiver and due to network's complexity, discover the optimal route would be impossible. Hence the existing solution is a SDN with an overall view. In this paper, At First, QoS definitions are expressed. Second, traffic engineering methods in current networks is discussed. Finally, a new technique to improve QoS over SDN is proposed. Results of performance evaluation showed that the proposed method is better than existing methods.

Keywords

Software Defined Networking, Multimedia, Quality of Service, Traffic Engineering.

1. INTRODUCTION

The ability of provide better service to specific traffic flow by using different technologies is called QoS. QoS requirements includes operation balancing, bandwidth, Delay variation (Jitter), packet loss and service availability. The current design of the Internet architecture is based on the end to end parameters and network participation to operate this task is low. With this architecture, only best effort services that leads to reliability, is provided in network layer. Packet loss percentage, delay and jitter should be considered to transmit interactive videos such as video conference and telemedicine applications [1-2]. QoS parameters should be properly controlled and managed to obtain different level of QoS [3]:

- **Traffic Classification:** Classification of different traffic flow data in different service classes
- **Service Level Agreement:** a contract between the service provider and the consumer for a specific class of service
- **Acceptance of Control and Policy:** manage different defined classes in SLA by service providers and control resources to prevent consumers misuses

- **Active Management of Queue:** manage queues based on SLA

Network infrastructures have to use mechanisms such as integrated services, distinct services and labeling. Peer to peer networks can be used where the receiver's requested QoS is commensurate with the quality of the transmitter's node and its path.

Due to network complexity and dynamics, network management and configuration is a challenge. Generally these networks are composed of switches, routers, firewalls and different types of Middle box with different events which occurs concurrently. Network operators are responsible for network configuration to apply high level policies and its responsiveness to wide range of events that can occur such as traffic relocation or sniffing. Since the implementation of these high-level policies is only implemented in the form of a distributed and low-level configuration, Network configuration is so overwhelming [4].

Nowadays network operators have to implement management policies and tasks with a very limited set of low-level configuration commands in a command-line environment (CLD). On the other hand these policies are not equipped to react against continuous changes in network status. Therefore, in response to these changes, operators must manually modify the network configuration so that the network complies with the changes. Although operators use the external tools to reconfigure the network dynamically, in order to resolve this limitation, if an event occurs, these continuous changes will result in a series of errors in the configuration [5].

SDN has created a hope for more convenient methods for network configuration and management. In this new type of network, the control level (which is responsible for computing the path) is separated from the data level (which is responsible for data transmission) and place in a central server called the controller. By separating control logic from routers and switches and place it in a central controller which is the mastermind of the network, applying management policy, configuration and reconfiguration and process of computer network evolution is easier [6]. Figure 1 shows SDN architecture:

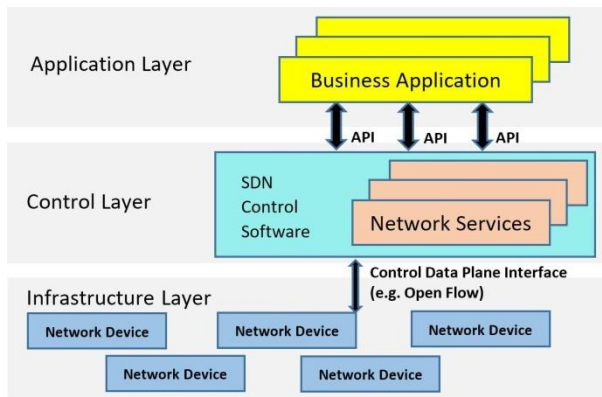


Figure 1: SDN Architecture

OpenFlow (OF) protocol is one of the most important and practical communication protocols, which enables controller to interact with the network switches. This protocol is a standard interface that is used mostly in SDN. OF switches contain one or multiple flow tables with flow entries. Each entry consists of paired rules and actions. The tables are filled by the controller. Each rule consists of fields related to headers data, such as source and destination MAC and IP addresses, port numbers and other necessary information. Each action determines instruction(s) to be executed on the packet in order to match the entry's rule [7].

Separation of data plan and control plan, gives ability to network administrators to make programmable policies and easily manage data plan via the controller. SDN also makes it easy to have a dynamic management, configuration, troubleshooting and even testing new protocols and ideas in the network without troubles [8].

One of the major consequent of Software Defined Network is the separation of three categories: network policies definition, implement these policies on switches and switch traffic from each other. In the other words, if computer network is divided into three levels as management level, control level and data level, the policies are defined at the management level and these policies are applied by the control level to navigational devices and on the basis of that, data level only forwards data. Some major consequences of this separation are here:

- Achieving high flexible networks
- Break the whole controlling problem into little solvable problems
- Simplify creation and definition of new abstract levels in network
- Simplify network management
- Facilitate computer networks evolution and Innovation process

Also integrating Cloud Computing, Wireless Sensor Networks and Internet of Things based on SDN for providing QoS in computer networks is very popular [9-13].

2. RELATED WORK

In this section, some related works on multimedia and video streaming traffic engineering over SDN are surveyed. As mentioned in previous section, SDN is a new paradigm and as a result, there are few works on multimedia traffic engineering over SDN in the literatures.

In the years 1995 to 1997, IETF has proposed IntServ [14-15] to transmit end to end multimedia. The idea behind IntServ on the Internet is to guarantee absolute QoS for each stream. In this mechanism, Application requests a specific service before sending the data is started. The request is granted by using Explicit signaling (Resource Reservation Protocol (RSVP)) and the application starts to send data after receiving network verification. In this method, network information and application requested resources are used to accept control. Classification, policy-making and queuing are also used based on the status of each flow in network. Two major mechanism of this method includes guaranteed services for Real time streams and Controlled load services for none real time flows. Two kinds of messages are used to reserve path between the sender and the receiver in IntServ architecture using RSVP. Route messages are sent from sender to fix a reverse route on the routers and notify the receiver of the route traffic status. RESV messages sent by the receiver is used to request QoS guarantees from middle routers [16]. The disadvantages of this method include high process overhead in routers, the lack of security, data transmission required before sending the original data and do not support packets with short life time.

DiffServ [17] was proposed by IETF to guarantee QoS; a method that can be implemented locally and individually on each router by using packets classification, timing and AQM on routers. A set of different classes of services is created based on queue priority and bandwidth allocation. Packets forwarding rules are defined corresponding to this set. Different sets have different services like delay, jitter, packet loss probability and etc. Granting different services to classes is based on marking packets by Differentiated Service Code point (DSCP). Therefore, the boundary routers send packets to internal area by grouping and marking them, and the core routers of the network forward packets based on their DSCP. There is no need to configure resources reservation and waste time to negotiate between the end parties in this method however this mechanism has some disadvantages such as lack of security and starvation.

MPLS [18] protocol has proposed by IETF either; the boundary routers add a label to every packet header based on Forwarding Equivalence Class (FEC). Packet routing and forwarding is based on this label (which is located between data link layer and network layer) instead of destination address in MPLS network. Each router consists of control and data parts; In the control section: label tables such as (LFIB, FIB, LIB) is written, and using them, labels transmit between routers by Label distribution protocol (LDP). In the data section: Package forward is performed based on MPLS tables. Using this label as an index in each router table helps to find the appropriate route for each packet and there is no need to search in every table. MPLS could be integrated with either IntServ or DiffServ [19]:

- MPLS integrated with IntServ: It is provided by reserving the routes of the provider of a specific service, that leads to the creation of tunnels
- MPLS integrated with DiffServ : If different labels are considered for different service classes, then there is no need to search, so packets forwarding will be done quickly in MPLS network and required resources will be maintained along the path.

This mechanism has disadvantages such as different management methods, security overhead, more complexity and cost while propose QoS.

OSPF-TE [20] is an OSPF protocol extended as one of the traffic engineering routing protocols. OSPF-TE is the basic routing protocol which is used by internet routers is OSPF. OSPF is a link state protocol which is created due to requirements such as software or hardware independencies, the need for the dynamic algorithms, the ability to use other cost metrics to determine the optimal route, the need for service quality support, load balancing, the need for hierarchical routing, increased security level, Support for tunnel creation mechanism.

The OSPF protocol features are simplicity, robustness, and scalability, but OSPF may lead to congestion, so it has a low performance in the lack of traffic engineering. Therefore, traffic engineering in OSPF networks is very important, and this can improve network performance. Two important aspects to control the level of traffic engineering in IP / MPLS networks is reducing the flood routing mechanisms and evaluating the cost of routing and signaling protocols processes. In flood reduction mechanisms, the amount of information exchanged by traffic engineering routing protocols is reduced. In node architecture based on traffic engineering, it is assumed that the edge nodes (LER labels edge routers) receive traffic requests, which is a dynamic process. Here the concept of a traffic request is a flow. When a request is sent to an edge node, it is assumed that the request arrives to the RDE (Route Decision Engine), which selects the appropriate path according to the network. RDE collects current topology information and resources using in the network, along with traffic engineering routing protocols.

When the RDE has selected a path for a traffic request, the corresponding LSP will be created using the RSVP-TE protocol, which will take care of accepting node-to-node control and allocating real resources. The LSP is referred to the designated path through which the packets arrive at the destination. OSPF-TE, broadcasts local allocation status changes to all other LSRs by sending a LSU message (link status update). This LSU message contains a specific link-state advertisement (LSA). LSR is referred to routers along the LSP path and forward packets. The LSU message is broadcasted to all LSRs that use the flooding OSPF. In order to avoid flood information, flood reduction mechanisms are implemented for each minimum change. In this traffic engineering process, two problems with the dissemination of information time and incorrect information must be considered. The problem of time propagation relates to the time it takes to disseminate information in the network, and the problem of inaccurate information is related to the reduced information that can be distributed using OSPF-TE.

The original OSPF version does not support QoS routing, so QoS was added to the OSPF and was called QOSPF. QOSPF includes parameters such as bandwidth and delay of each link, and route selection is based on the widest-shortest path algorithm. This algorithm first calculates all of the shortest hop routes using the Dijkstra algorithm [2], then selects the highest bandwidth bottleneck, this protocol is based on two key ideas: 1) Increase dissemination of the link status and topology of the database to include network resource information (such as available bandwidth). 2) Using an Alternate route computation algorithm to calculate paths that consider this resource information.

QOSPF [21] provides two optimizations, which include: 1) Update link status. 2) Route calculation in such a way that find the network constraint.

When QOSPF works without a resource reservation framework (such as RSVP), it uses a statistical method law, this method measures the bandwidth and latency of the links, and in order to estimate the link's ability to find the bandwidth limitation and requested latency is used, therefore, without explicit resource reservations, QOSPF can not guarantee that these restrictions will not be broken. Hence, QOSPF is a QoS routing protocol and the weakness is using the shortest hop path for the first time. From traffic engineering aspect, improve the effective load balancing is not always desirable.

Egilmez [22] proposed a framework to guarantee the amount of delay in multimedia applications. This framework named OpenQoS causes video traffic flows optimally in the network and is received by destination with more quality. The controller created two paths between source and destination: one for common traffic (based on OSPF algorithm) and another for multimedia traffic (based on CSP model [23] and Optimization [24-28]). Simulation results showed that for both actual scenario and simulation one using this model decreased PSNR parameter of receiving video in the destination. Table 1 compares these methods.

3. ARCHITECTURE OF THE PROPOSED METHOD

To establish video QoS and improve end-to-end performance, find the optimal route is required, however there may be different pathways between the sender and the receiver and find the best route may not be possible due to network complexity. Hence the solution is to use a network with general view such as figure 2 and a simple protocol for requesting QoS assurance. SDN using a central controller can be used to have a general view.

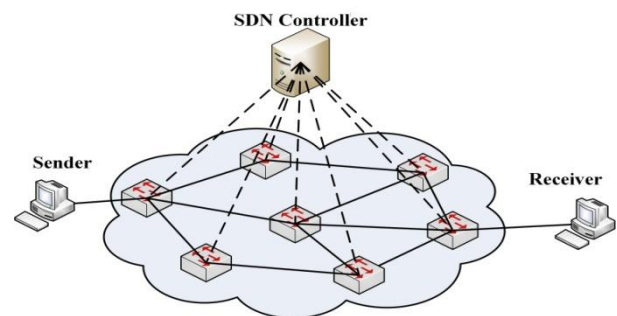


Figure 2: Using a network with general view

In this paper, by using a network traffic classification in terms of quality parameters and their queuing, a method to improve the use of video conferencing is presented. There are two traffic queues based on traffic classifications. Obviously, a queue which contains more sensitive streams (such as video conferencing) has the highest priority and the queue which contains the best-effort streams has the lowest priority. In this paper, it is assumed that the queue with the highest priority only includes video conferencing streams.

When the new stream reaches the switch, it sends the packets to the controller. The designed modules in the controller are intended to accommodate package headers with table entries. If no match is obtained, the queue with the lowest priority is considered and the shortest path is considered according to the Dijkstra algorithm (the default controller algorithm). If matching is obtained, the queue with the highest priority is assigned to it, and K shortest path between the source and the destination, using the Dijkstra algorithm are obtained, and then the delay of the K path is calculated and finally the path

with the least delay is chosen to send the packet. After selecting the appropriate path for video traffic, the rules are announce to all the switches on the selected route.

Since OpenFlow Type 1.3 only supports a limited set of queue management features, changes to this protocol have also been

made to support the switch operation, which includes a token bucket to shape existing traffic. The proposed architecture is presented in Figure 3.

Table 1: Comparison of Traffic Engineering Methods

	Initial configuration and probable changes	implementation	cost	Security	availability	QoS guarantee	Qos and routing distinction
IntServ	Initial reserve, probable changes just in the controller	Improvement over IP but not widespread	Sender and receiver	Depend on SDN security	High in network	Hard	No
DiffServ	Class initial definition, probable changes just in the controller	Improvement over IP but not widespread	Network	Depend on SDN security	High in network	Soft	No
MPLS	Initial labeling just in the route, probable changes just in the controller	Improvement over IP	Network	Depend on SDN security	High in network	Soft	No
OSPF-TE	Add some header	Improvement over IP but not widespread	Network	Depend on SDN security	High in network	Soft	No
QOSPF	Find network limitations, using Dijkstra algorithm	Improvement over IP but not widespread	network	Depend on SDN security	low in network	Soft	No

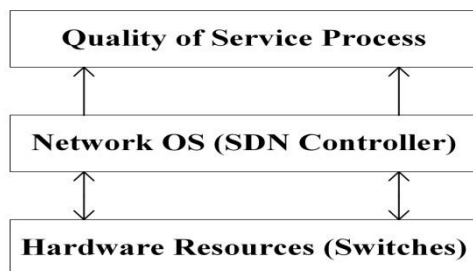


Figure 3: Architecture of the proposed method

The structure of the switch can also be so that the reservation is made by the controller, commands are sent from the controller to the switch according to the requested QoS: create an entry in the switch flow table, create a separate flow table for that service, and create a priority queue to exit the switch. An arrived packet at the switch is stored in the input queue; it is sent to the controller if the type of the packet is policy type, otherwise placed in a particular output queue according to the

flow tables and the existence of an entry for that type of packet. If there is no entry, the packet will be drop.

In this mechanism, Due to the presence of a central controller in SDN which has general view, possible changes are applied only in the controller section and the entire network is informed simultaneously, but using SDN does not remove the need of signaling before start sending (which can be resolved by using another method of QoS improvement). Also, the scalability of this mechanism is somewhat improved, due to the stored reservations in the controller (but it does not completely disappear, due to the limited storage capacity in the controller). Although the use of SDN, generally removes expensive routers and relocates them with simple forwarding switches, and reduces costs, but changes in different parts, even in the sender and the receiver causes costs. SDN security is also guaranteed by securing the controller, so there will no longer be resource reservation attacks on the network because the resource request is checked in the controller. Moreover, with SDN, if the reservation request service is not provide, the controller with its comprehensive view of the network can

inform the sender/receiver of its service, which will increase availability. Also, using this method, the routing separation and requested QoS are removed and the route is determined based on QoS.

Also in this architecture we can give a higher priority to a specific traffic, so that the sender creates and sends video packets to the receiver. The packet is sent to the first switch on the network, depending on the fact that there is still no operation for the new flow received in the switch flow table, the packet would be sent to the controller. The controller will specify the type of QoS depends on the content inside the packet (such as, rapid forwarding) and add the corresponding DSCP to the packet, and then record the operation in the flow table of the path's switches. The rest of the switches use the DSCP field to recognize receiving the fast forward packet and apply its corresponding policy; for example, this policy can allow all packets associated with that stream to be queued at the queue with highest rate of sending.

4. SIMULATION AND RESULTS

The performance of the proposed method and baseline is evaluated using the following QoS parameters:

Average end to end delay: Average end to end delay refers to the elapsed time between sending a packet in sender node and receiving the packet in the receiver node.

Packet loss ratio: Packet loss ratio denotes to the fraction of the number of lost packets to the number of sent packets.

In this paper, the proposed architecture is evaluated from QoS point of view in terms of delay and packet loss. In the test-bed network, OpenDaylight controller runs the proposed traffic engineering technique. OpenDaylight is an open source and powerful controller which is used for enabling SDN and network programmability. OpenDaylight supports java programming language, and can be used for controlling and managing networks. The test-bed network is emulated in Mininet. Mininet is an open source SDN and OpenFlow emulator which can create a network of controllers, switches, hosts and links. The bandwidth of all links is set between 10 to 20 Mbps.

Considered topology is like Fig. 1, the network has 7 OpenFlow-based switches and two clients in which they are as sender and receiver of multimedia and UDP (UDP data which generated with hping3 software) traffic. The result of the proposed method in comparison to Dijkstra algorithm is shown in Figure 4 and Figure 5 for delay and packet loss respectively.

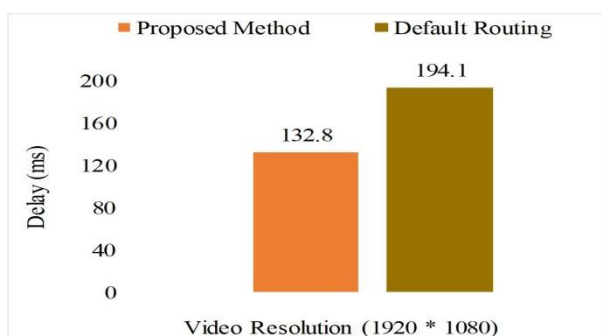


Figure 4: Result of Average End to End Delay

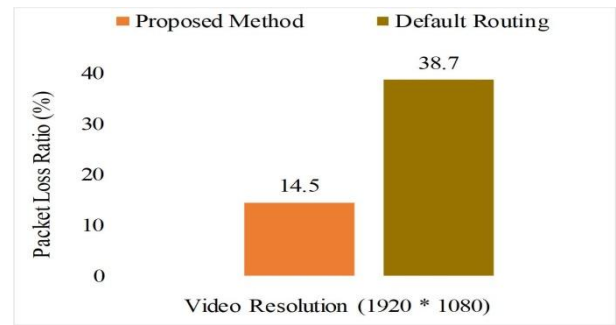


Fig 5: Result of Packet Loss Ratio

5. CONCLUSION

The use of online video is on the rise; in the transmission of multimedia content, there is a need for predictable and stable resources, with little latency and no missing packets, in interactive multimedia programs such as video conferencing, the delay is required, and in the video stream, the network status must be fixed, so the jitter should be within a certain range. To solve the problem of network features, applications, and customer needs, you should not use the new protocol or its extension because it takes a long time to create it (5-10 years), and during this time There may be a need for change; many organizations may also create various protocols and a standard protocol would not be created, while the use of SDN without the need for special hardware is possible. On the other hand, the move to SDN's implementation is increasing day today as Amazon and Facebook companies have announced that their current network will change to this architecture. The idea of establishing QoS in SDN is at the network level; each of the methods mentioned above can also be used to implement QoS on these networks. One of the most important advantages of using SDN is the presence of a comprehensive view of the controller, so the policies and controls of these resources, including time, bandwidth, etc., are simpler and more efficient and unified at the same time. Moreover, the infrastructure available in this type of network is flexible, so some services are guaranteed by QoS and some others without it to apply a balance between QoS and cost. Therefore, in the OpenFlow-based SDN proposed method, it is possible to solve the problem of QoS implementation in the Internet by defining routing rules that are proportional to the data being sent, which can increase QoS guarantee.

For future work, optimization algorithms such as PSO, ACO and Genetic algorithm can be used to find the optimal path. It is also possible to provide dynamic queuing methods and compare them with each other or with static queues for a variety of multimedia applications.

6. REFERENCES

- [1] Parsaei, M. R., Javidan, R., Fatemifar, A. and Einavipour, S. "Providing Multimedia QoS Methods over Software Defined Networks: A Comprehensive Review," International Journal of Computer Applications, vol. 168, no. 9, pp. 55-59, 2017.
- [2] Parsaei, M. R., Mohammadi, R. and Javidan, R. "A new adaptive traffic engineering method for telesurgery using ACO algorithm over Software Defined Networks," European Research in Telemedicine/La Recherche Européenne en Télémedecine, vol. 6, no. 3-4, pp. 173-180, 2017.
- [3] Parsaei, M. R., Sobouti, M. J., Khayami, S. R. and Javidan, R. "Network Traffic Classification using

- Machine Learning Techniques over Software Defined Networks,” *International journal of Advanced Computer Science and Applications*, vol. 8, no. 7, pp. 220-225, 2017.
- [4] Rowshanrad, S., Parsaei, M. R. and Keshtgari, M. “Implementing NDN using SDN: a review on methods and applications,” *IJUM Engineering Journal*, vol. 17, no. 2, pp. 11-20, 2017.
- [5] Parsaei, M. R., Khalilian, S. H. and Javidan, R. “A Comparative Study on Fault Tolerance Methods in IP Networks versus Software Defined Networks,” *International Academic Journal of Science and Engineering*, vol. 3, no. 4, pp. 146-154, 2016.
- [6] Masoudi, R. and Ghaffari, A. “Software defined networks: A survey,” *International Journal of Network and Computer Applications*, vol. 67, pp. 1-25, 2017.
- [7] Li, W., Meng, W. and Kwok, L. F. “A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures,” *International Journal of Network and Computer Applications*, vol. 68, pp. 126-139, 2016.
- [8] Karakus, M. and Duresi, A. “Quality of Service (QoS) in Software Defined Networking (SDN): A survey,” *International Journal of Network and Computer Applications*, vol. 80, pp. 200-218, 2017.
- [9] Taheri, R. and Parsaei, M. R. “Elearning Framework Based On Cloud Computing,” *Journal of Selcuk University Natural and Applied Science*, pp. 272-278, 2015.
- [10] Parnian, A. R., Parsaei, M. R., Javidan, R. and Mohammadi, R. “Smart Objects Presence Facilitation in the Internet of Things,” *International Journal of Computer Applications*, vol. 168, no. 4, pp. 25-31, 2017.
- [11] Parsaei, M. R., Parnian, A. R., Rostami, S. M. and Javidan, R. “RPL load balancing in Internet of Things,” *IJUM Engineering Journal*, vol. 18, no. 2, pp. 137-150, 2017.
- [12] Taheri, R., Parsaei, M. R. and Javidan, R. “A New Method for Optimizing Energy Consumption in Wireless Sensor Networks Using Enhanced LEACH Protocol,” *Journal of Engineering and Applied Sciences*, vol. 100, no. 3, pp. 576-581, 2016.
- [13] Parsaei, M. R. and Parnian, A. R. “IPv6 based routing in building automation network,” *International Conference on Knowledge-Based Engineering and Innovation (KBEL)*, pp. 1025-1031, 2015.
- [14] Kurose, J. F. and ROSS K. W. “Computer Networking: a top-down approach featuring the Internet,” pp. 640-646, 2004.
- [15] Braden, R., Clark, D. and Shenker, S. “Integrated services in the Internet architecture: an overview,” *RFC 1633*, 1994.
- [16] Owens, II H. and Duresi, A. “Video over software-defined networking (vsdn),” *Computer Networks*, vol. 92, pp. 341-356, 2015.
- [17] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W. “An architecture for differentiated services,” *RFC 2475*, 1998.
- [18] Das, S., Sharafat, A. R., Parulkar, G. and McKeown, N. “MPLS with a simple OPEN control plane,” In *Optical Fiber Communication Conference and Exposition (OFC/NFOEC)*, pp. 1-3, 2011.
- [19] Harju, J. and Kivimaki, P. “Co-operation and comparison of DiffServ and IntServ: performance measurements,” *25th Conference on Local Computer Networks (LCN)*, pp. 177-186, 2000.
- [20] Bi, J., Leng, X. and Wu, J. “OSPF-TE Scalability Study based on Internal Performance Measurement,” *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICN/ICONS/MCL)*, pp. 92-92, 2006.
- [21] Daxin, Z. and Danlin, C. “Research and simulation of distributed QoS routing algorithm,” *International Conference on Broadband Network & Multimedia Technology*, pp. 24-27, 2009.
- [22] Egilmez, H. E., Dane, S. T., Bagci, K. T. and Tekalp, A. M. “OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software Defined Networks,” *Conference on Signal and Information Processing Association*, pp. 1-8, 2012.
- [23] Parsaei, M. R., Javidan, R., Kargar, N. S. and Nik, H.S. “On the global stability of an epidemic model of computer viruses,” *Theory in Biosciences*, vol. 136, no. 3-4, pp. 169-178, 2017.
- [24] Gao, W., Sarlak, V., Parsaei, M. R. and Ferdosi, M. “Combination of fuzzy based on a meta-heuristic algorithm to predict electricity price in an electricity markets,” *Chemical Engineering Research and Design*, pp. 1-25, 2017. Doi: 10.1016/j.cherd.2017.09.021.
- [25] Parsaei, M. R., Javidan, R. and Sobouti, M. J. “Optimization of Fuzzy Rules for Online Fraud Detection with the Use of Developed Genetic Algorithm and Fuzzy Operators,” *Asian Journal of Information Technology*, vol. 15, no. 11, pp. 1856-1864, 2016.
- [26] Parsa, S. S., Sourizaei, M., Dehshibi, M. M., Shateri, R. E. and Parsaei, M. R. “Coarse-grained correspondence-based ancient Sasanian coin classification by fusion of local features and sparse representation-based classifier,” *Multimedia Tools and Applications*, vol. 76, no. 14, pp. 15535-15560, 2017.
- [27] Komijani, H., Parsaei, M. R., Khajeh, E., Golkar, M. J. and Zarrabi, H. “EEG classification using recurrent adaptive neuro-fuzzy network based on time-series prediction,” *Neural Computing and Applications*, pp.1-12, 2017. Doi: 10.1007/s00521-017-3213-3.
- [28] Nabaie, A., Hamian, M., Parsaei, M. R., Safdari, R., Samad-Soltani, T., Zarrabi, H. and Ghassemi, A. “Topologies and performance of intelligent algorithms: a comprehensive review,” *Artificial Intelligence Review*, vol. 49, no. 1, pp. 79-103, 2018.