An Overview of Multimedia QoS in SDN-Enabled IP Networks

Amir Hossein Moravejosharieh
Auckland Institute of Studies
amirm@ais.ac.nz

Michael Watts
Auckland Institute of Studies
New Zealand
mjwatts@ieee.org

Kourosh Ahmadi
Auckland Institute of Studies
New Zealand
kourosha@ais.ac.nz

ABSTRACT

Although traditional IP networks have been widely adopted by societies and telecommunication services, there are still numerous concerns about them. It is hard to implement predefined network policies and to reconfigure the network to adaptively and effectively respond to network failures, unforeseen network loads, and unexpected changes. Software-Defined Networking (SDN) is a new networking paradigm designed to resolve traditional IP network shortcomings by breaking the vertical integration of control and data planes. SDN separates the network control logic from underlying routers and switches and introduces the ability to program the network. This offers better Quality of Service (QoS) for different types of media, especially for multimedia traffic. In this paper, we provide an overview of what has been proposed to improve the multimedia QoS in SDN-enabled networks. Additionally, we nominate two approaches that can be deployed by SDN controllers to offer better QoS for various types of media services.

Keywords: Software Defined Networking, Quality of Service, Multimedia traffic, Traditional IP networks

1. INTRODUCTION

In the past decade, using multimedia services on the Internet has increased considerably. Internet Service Providers (ISPs) are in a serious competition to offer multimedia services with reasonably high QoS. However, their attempts have not been entirely successful due to a number of limitations imposed by traditional IP networks: high level of complexity; very difficult to manage; unable to rectify failures; unable to deal with ever-changing state of critical loads such as multimedia and last but not least, unable to respond to unexpected changes such as adaptively complying with newly arisen network QoS requirements. The main cause of the above-mentioned issues is the vertical integration of the control plane (which handles the network traffic) and the data plane (which forwards the traffic according to the forwarding instruction given by the control plane). The control and data plane are tightly bundled together and installed inside of the networking devices. Moreover, such vertical integration has reduced the flexibility of the network infrastructure in terms of scalability and also hindered the innovation and evolution of the IP networks (Benson, Akella, & Maltz, 2009; Shu et al., 2016).

Software-Defined Networking (SDN) is a new emerging networking paradigm that is designed to resolve issues and challenges imposed by the nature of the traditional IP networks (Ramos, Kreutz, & Verissimo, 2015; Casado, Koponen, Shenker, & Tootoonchian, 2012). In SDN, the network's control logic (control plane) is separated from the routers and switches (data plane) that are mainly responsible for forwarding network traffic. This separation has allowed routers and switches to become simple forwarding devices and the network's control logic is now implemented on a logically centralised controller.

The SDN controller receives the requirements from the network applications in the form of high level languages and converts them to low-level commands to be instructed on forwarding devices. This strategy has simplified enforcing new policies, adaptively reconfiguring the QoS required for different media types (e.g. audio, video and data) and making the network more flexible in terms of scalability (Kim & Feamster, 2013; Koponen, Casado, Gude, & Stribling, 2014). The separation of the control plane and the data plane can be accomplished by creating a well-defined programming interface between the network application and SDN controller, and also between SDN controller and forwarding devices. OpenFlow (McKeown et al., 2008; McKeown, 2009) is an example of such interface between the SDN controller (hereafter called controller) and forwarding devices. SDN and OpenFlow framework started as an academic project (McKeown et al., 2008), However, they have received a great deal of attention in the industry over the past few years. Well-known companies such as Google, Facebook and Yahoo have invested in promoting and growing the adoption of SDN through open standard development. For instance, Google has deployed SDN to interconnect its data centres across the globe.

Recently, we have witnessed significantly large demand for using multimedia in Internet. Statistics show that the usage of multimedia (more particularly video) traffic will take up to 90 percent of the Internet traffic in the next few years (Parsaei, Khalilian, & Javidan, 2016; Parsaei, 2017). Therefore, providing a reasonable level of QoS for multimedia traffic is a must. Numerous QoS architectures and schemes have been proposed and developed to enhance the QoS for different types of traffic such as InServ (Braden, Clark, & Shenker, 1994) and DiffServ (Carlson et al., 1998). However, they were quite unsuccessful when implemented at global scale, mainly due to using distributed routing strategies because of lack of global knowledge of network resources. While in SDN, video streams can be adaptively re-routed based on varying network conditions aiming to offer better QoS for multimedia traffic at the cost of the lowest level of network congestions. In this paper, we provide an overview of numerous algorithms and approaches proposed to improve the QoS of the multimedia services. We also nominate two approaches that can be useful in offering reasonably better QoS level for multimedia services.

This quality assured paper appeared at the 9th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2018) and the 31st Annual Conference of the National Advisory Committee on Computing Qualifications, Wellington, NZ, July 11-13, 2018 as part of ITx 2018.

The remainder of this paper is organised as follows: Section 2 introduces the structure of the SDN-enabled networks. Different methods, algorithms and approaches for improving the multimedia QoS are discussed in Section 3. In Section 4, we propose a few approaches to improve the multimedia QoS SDN-enabled networks. Finally, we conclude our paper in Section 5.

2. SOFTWARE DEFINED NETWORKING STRUCTURE

Generally, software defined networking is designed to provide computer networks with the capability of being programmed. This capability can potentially enhance network traffic measurement and eventually network traffic management. Clearly, better network traffic management would result in offering better QoS for different types of media services e.g. multimedia. Network architecture in SDN is designed in such a way that a network's control logic is entirely separated from network devices. More specifically, data transfer layer is separated from control layer.

In SDN-enabled networks a logical central controller collects the requirements (e.g. QoS, security or resources) from an application that intends to establish a connection to the other end of the network. Based on the collected requirements, the controller instructs network devices (that are now simple forwarding devices) to reconfigure some settings in order to satisfy the application requirements (Lara, Kolasani, & Ramamurthy, 2014; Nunes, Mendonca, Nguyen, Obraczka, & Turletti, 2014; Jarraya, Madi, & Debbabi, 2014). An implementation of SDN is based on implementation of OpenFlow framework (Bianco, 2010) which is widely accepted as an open standard. In OpenFlow framework forwarding devices are simply called OpenFlow switches and all routing decisions, security functions and QoS requirements would be the main responsibility of the OpenFlow controller. Figure 1 presents the main compartments of SDN structure.

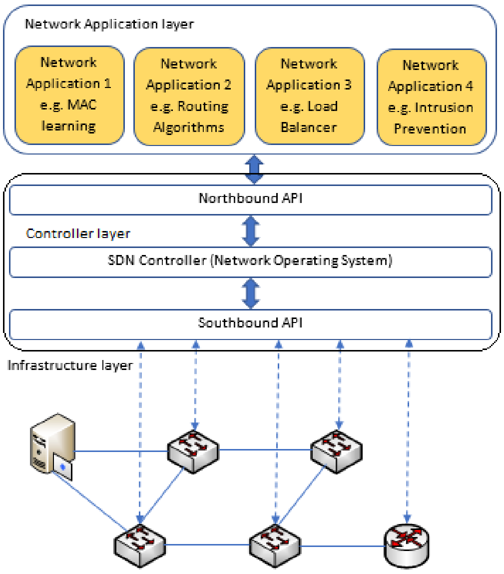

Figure 1. SDN structure

SDN structure is generally composed of three main layers: *Application* layer, *Control* layer and *Infrastructure* layer. The application layer contains application requirements such as seamless routing, resource provision and allocation, network policies, security and QoS requirements. The control layer (consisting of SDN Controller, Northbound and Southbound APIs) mainly consists of the controller which is also referred to as a network operating system and provides a programmatic interface between applications with high level language and forwarding devices that only understand low level commands. Infrastructure layer mainly consists of forwarding devices whose behaviours are instructed and dictated by the controller.

In addition to the above-mentioned layers, there are two interfaces that enable the controller to communicate with the application and infrastructure layers, namely: southbound and northbound interfaces. The southbound interface defines the communication protocol between forwarding devices and the control plane elements and is mainly responsible for interacting between forwarding devices and the controller. The OpenFlow southbound interface, Extensible Messaging and Presence Protocol (XMPP) (Saint-Andre, 2011), One Platform Kit (OnePK, Cisco product) and Path Computation Element Communication Protocol (PCEP) (Le Roux, 2009) are a few examples of southbound interfaces among which, OpenFlow is the most commonly used southbound interface. The northbound interface is an Application Programming Interface (API) for application developers. This interface typically abstracts the low-level command lines which will be used by southbound interface to program forwarding devices. There are a number of proposed northbound interfaces such as Frenetic (Foster et al., 2011), Nettle (Voellmy & Hudak, 2011) and ALTO (Burger & Seedorf, 2009).

In the SDN-enabled network with the above-mentioned structure, forwarding decisions are flow-based rather than destination-based. According to SDN/OpenFlow context, a flow is defined as a sequence of packets between a source-destination pair. All packets of a flow would receive the same treatment in the form of service policies at the forwarding devices (Newman, Minshall, & Lyon, 1998; Gude et al., 2008). In SDN-enabled networks, abstraction of flows provides the opportunity of unifying the behaviour of routers, switches, firewalls and other middleboxes (Jamjoom, Williams, & Sharma, 2014).

To discuss the QoS for multimedia traffic in SDN, we first need to elaborate Traffic Engineering (TE). TE is comprised of two main parts: traffic measurement and traffic management. Traffic measurement mainly focuses on monitoring, measuring and acquiring network status information (e.g. current network topology structure, end-to-end latency, bandwidth utilisation, packet losses, packet counters, traffic metrics and so on) in the SDN-enabled networks. Based on the network status information the SDN controller determines the suitable allocation of the network resources to the end users.

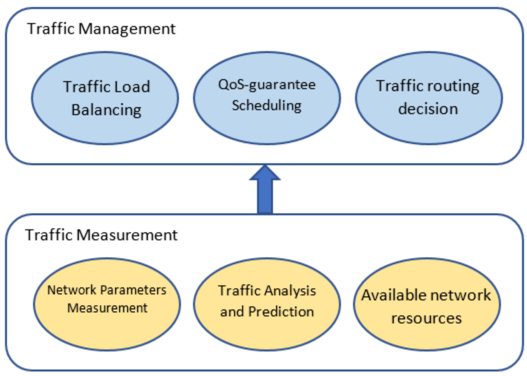

Figure 2. Simplified traffic engineering structure

Traffic management, however, determines how to manage and schedule different types of network traffic based on the information provided by the network traffic measurement aiming to satisfy the application requirements e.g. QoS or security. Figure 2 depicts a TE framework indicating the collaboration of network traffic measurement and network traffic management.

3. LITERATURE REVIEW

In this section, we elaborate on research and studies conducted on QoS in SDN with the focus on QoS for multimedia traffic. We propose to classify such studies into five major groups, namely: video encoding approach; Dijkstra algorithm enhancement; IntServ service model; QoS routing; and resource reservation/allocation. We discuss a number of studies relevant to each group. Please note, such classification may not be limited to the above-mentioned groups. There are other studies that just compared QoS in SDN-enabled network in comparison with traditional IP networks in terms of end-to-end delivery of multimedia traffic (Jimenez, Romero Martínez, Rego, Dilendra, & Lloret, 2015; Bari, Chowdhury, Ahmed, & Boutaba, 2013). Such studies are not considered in our proposed classification.

3.1 Video Encoding Approach

A QoS-guaranteed OpenFlow architecture with scalable video encoding has been proposed in (Civanlar et al., 2010; Thenmozhi, Preethi, Sadhana, Shruthi, & Yuvarani, 2017). In this approach, the controller performs two concurrent routings. In the first routing, the controller analyses the quality of service of different routes via transmission of scalable video encoding traffic. In the second routing which is also called "best effort routing", the controller attempts to find the shortest path to the destination. Based on the acquired network status information the best path is determined. However, their obtained results also revealed that the first routing significantly impacts on other traffic such as best effort traffic and caused considerable amount of packet losses. Scalable video encoding traffic has also been used in (Egilmez, Gorkemli, Tekalp, & Civanlar, 2011) in which authors have created two scenarios to analyse commonly-used routing algorithms for multimedia traffic. In the first scenario, routing of the scalable video encoding traffic is identified as a QoS traffic and packet losses are not considered. While in the second scenario, the best effort traffic is considered as the QoS traffic and the probability of packet losses has also been taken into consideration. Their obtained results have shown that the first scenario has improved the QoS of the multimedia by 14% while the second scenario has improved the QoS for multimedia by almost 7% compared to the first scenario.

3.2 Dijkstra Algorithm Enhancement

Dijkstra algorithm and its enhancements which find the shortest path between a source and a destination seems to have improved the multimedia QoS in SDN. Laga et al. (2014) prioritised the different layers of scalable video encoding streams along with modifying the Dijkstra algorithm to the benefit of QoS routing for multimedia traffic. In their approach the weight of each link is calculated based on specific network status information i.e. link delay and the remaining bandwidth link. This method is implemented in the beacon packet transmitted by the controller. Their obtained results indicate that the delay experienced in the main layer has been reduced by 72% as well as 56% for the secondary layers (Laga et al., 2014). Karl et al. (2013) have also changed the metric model in Dijkstra algorithm aiming to experience better QoS for multimedia traffic such as packet loss, delay and per link bandwidth. In their proposed method OpenFlow switches are programmed to store the number of forwarded packets which would eventually determine the number of packet losses. To determine the delay, the controller transmits UDP packets to a target switch via OpenFlow switch controller and calculates the elapsed time. Their obtained results show that their proposed Dijkstra model (implemented on NOX OpenFlow controller) outperforms the conventional Dijkstra algorithm (Karl, Gruen, & Herfet, 2013).

3.3 IntServ Service Model

IntServ architecture (performed in the traditional IP networks) has been evaluated in (Owens II & Durresi, 2015; Wallner & Cannistra, 2013) to determine its suitability for SDN-enabled networks. In this study Video over SDN (VSDN) is designed and proposed as a new QoS architecture for SDN-enabled networks. VSDN has highlighted the shortcoming of IntServ architecture with the ultimate goal of improving the overall QoS for multimedia traffic. In their proposed method the network has been divided into different multiple slices. The controller allocates different slices to different application traffic. Note that network engineers are to define high-level slices. Therefore, the SDN controller can simply reserve the required resource for an application traffic. The performance of VSDN is evaluated using sensitive and insensitive traffic. The obtained results highlighted considerable packet loss for insensitive traffic while the amount of packet loss for the sensitive traffic was negligible.

3.4 Quality of Service Routing

Egilmez et al. (2014) have proposed a novel OpenFlow controller called "OpenQoS" specifically to improve the end-to-end delivery of QoS for multimedia traffic. According to OpenQoS, QoS routing is performed where routes of multimedia traffic are dynamically optimised to satisfy the QoS requirements for that traffic (Egilmez, Dane, Bagci, & Tekalp, 2012). The performance of OpenQoS has been evaluated using an experimental test-bed and compared with the HTTP-based multi-bitrate adaptive streaming. Their obtained results show that OpenQoS guarantees seamless delivery of multimedia traffic with a minimum adverse impact on end-user experience. The idea of QoS routing has also been used in (Egilmez & Tekalp, 2014) where a large scale SDN-enabled network is divided into multiple subdomains and each subdomain is controlled by a separate controller. More specifically, the controller of a subdomain performs the QoS routing individually and shares its views with other controllers to enable inter-domain QoS routing with reduced adverse effects. In distributed scheme, network status information is collected along with deploying an optimisation framework for flow-based QoS. Finally, a communication mechanism among controllers is employed for scalable QoS routing. Their obtained experimental results promise that the proposed distributed approach can be implemented on global scale. "ARVS" (Yu, Wang, & Hsu, 2015) is another adaptive re-routing approach for video streaming with QoS support over SDN-enabled networks. Two different types of packets (i.e. base layer packet and QoS enhancement layer packet) are used to mitigate the congestion occurrence in the shortest path. Their proposed approach has reduced packet loss rate up to 77%.

3.5 Resource Reservation/Allocation

Bandwidth reservation/allocation is another approach to guarantee the QoS for multimedia traffic. Authors in (Tomovic, Prasad, & Radusinovic, 2014) have proposed an SDN controller that flexibly and automatically instructs network devices to change their configuration in order to satisfy the multimedia QoS requirements. To accomplish this the controller collects the network status information and performs smart traffic management followed by prioritising the network traffic. Their experimental results show significant performance improvement of the "best-effort" service model compared to the best-effort service model run in traditional IP networks. "Multimedia Gateway" (Diorio & Timóteo, 2016) is another approach to examine, identify and classify the traffic according to its service type (e.g. video, audio or data) and forward each traffic towards a destination according to some pre-defined rules such as different bandwidth allocation and priority. Another example of resource allocation is the study conducted in (Yilmaz, Tekalp, & Unluturk, 2015) where network traffic is constantly monitored, and video streams are send to the servers with lower loads. This load-balancing (Zeeshan, Guru, Alam, Professor, & MVIT, 2016) approach is accomplished via re-routing video streams whenever one server becomes overloaded with video traffic. It has been proved that load-balancing could be considered as a helpful solution to improve the QoS of the multimedia traffic. Authors in (Habibi Gharakheili et al., 2017) have proposed segregation of link speeds i.e. fast-lane and slow-lane for allocating to different media types. They have argued that allocating fast-lane to sensitive traffic such as video traffic and slow-lane to web page loads could significantly improve the end-to-end delivery of video traffic. “FlowQoS” (Seddiki et al., 2015) is an SDN framework that has been designed to allocate bandwidth to various types of applications. A flow classifier and an SDN-based rate limiter DNS-based classifier has been designed to identify different applications that run over common web ports. The other classifier performs lightweight packet inspection to classify non-HTTP traffic flows. Its performance has been evaluated and their obtained results show its advantages in the presence of different competing traffic.

4. ANALYSIS AND DISCUSSION

SDN paradigm has provided the opportunity of centrally managing and controlling the QoS of different service types e.g. multimedia services. Although SDN approach has offered a lot in terms of improving the QoS for multimedia traffic, one could argue that the QoS improvement can be experienced when there are relatively small number of end-users demanding reasonable level of multimedia QoS. Once the number of end-users becomes larger, approaches such as bandwidth reservation/allocation, QoS routing and video encoding approaches are highly likely to fail. This is mainly due to inevitable hardware limitations such as limited available bandwidth. Additionally, sending traffic (related to different layers in video encoding approach) with different treatment and behaviour will increase the network overhead as the number of end-users becomes larger. Therefore, there is a tangible need to design and develop an approach in which such limitations are resolved or at least effectively dealt with. In this section we nominate two ideas that might be helpful to resolve such issues. To the best of our knowledge, the combination of the following nominated ideas with SDN concept aiming to improve the QoS in an SDN-enabled network, has not been proposed anywhere.

4.1 Fuzzy Logic Approach

Fuzzy logic is a type of logic that is much closer to the nature of human thinking as well as human natural language than the traditional logical system. In fuzzy logic control system, an algorithm converts a set of linguistic control strategy – based on expert knowledge – into an automatic control strategy. There must be a mechanism to prioritise multimedia services based on their required level of QoS. This prioritisation can lead to allocating different multimedia services to different paths offering different levels of QoS for multimedia services. Using fuzzy logic approach along with constantly monitoring network status information can better assist SDN controllers to envisage suitable paths for multimedia service based on different QoS requirements. In our future work, we plan to implement fuzzy control logic on the SDN controller and evaluate its performance compared to an ordinary SDN controller in terms of response time. The implementation of fuzzy control logic would simply allow the SDN controller to prioritise different network flows based on different requirements such as dedicated bandwidth, certain threshold for packet loss rate, creation of network overhead. This would be very beneficial when a large number of users demand relatively higher QoS, simultaneously.

4.2 Ant Colony Optimisation

A colony of ants performs beneficial tasks such as finding the shortest path to the source of food as well as sharing such information with other ants through depositing pheromones. Their behaviour has inspired researchers to use ants’ collective intelligence and transform it into optimisation techniques in computer networking. Although Ant Colony Optimisation (ACO) has been considered in QoS routing in traditional IP networks, there are still numerous shortcomings in terms of routing overheads and adaptivity (Sim & Sun, 2003; Ramos et al., 2015; Krishna, Saritha, Vedha, Bhiwal, & Chawla, 2012; Wang, Shi, Ge, & Yu, 2009). In SDN-enabled network, a controller can employ the concept of centrally managing a network along with ant colony optimisation techniques to instruct routers and switches in the network to dynamically find the best path for forwarding packets. Clearly, the need for QoS routing can be eliminated and middleboxes in the network (ants) can effectively and adaptively re-route network traffic towards the destination (source of food). In other words, enabling SDN controller to implement and instruct ACO techniques on data plane in an SDN-enabled network, could significantly improve load-balancing in the network as data plane could contribute in gathering network status information and statistics and deliver them to the SDN controller for further analysis to find the best alternative path. The evaporation of pheromones – in our case, obsolete statistical information – would help the SDN controller to instruct data plane to obtain new network status and statistical information. In our future work, we also plan to enable the SDN controller to apply ACO on data plane aiming to improve the load-balancing. Clearly, this approach may incur extra burden on a group of switches or routers in the network that are selected by the SDN controller and are responsible for performing parallel search over several paths (computational threads) based on a given set of requirements. The collective behavior emerging from the interaction of the different search threads can be extremely beneficial in finding a best alternative path.

5. CONCLUSION

Multimedia services are considered as critical applications that require close attention in terms of end-to-end delivering of QoS. In this paper, we provided an overview of multimedia QoS in SDN-enabled networks. Over the past decade, research communities have proposed various mechanisms and approaches to improve the QoS for multimedia services. However, there are still some issues related to offering an acceptable level of QoS for multimedia traffic in a large-scale network environment e.g. available bandwidth limitations, and imposing network overhead. We have nominated two new approaches, namely: fuzzy logic control system and ant colony optimisation which can be helpful to offer better QoS level for multimedia services.

For the future work, we plan to conduct research on end-to-end delivery of QoS for multimedia services in wireless environment. Clearly, mobility (mobile users) intrinsically would incur a number of limitations e.g. handover and bandwidth constraints.

6. REFERENCES

Bari, M. F., Chowdhury, S. R., Ahmed, R., & Boutaba, R. (2013). *PolicyCop: An autonomic QoS Policy Enforcement Framework for Software Defined Networks.* Paper presented at the Future Networks and Services (SDN4FNS), 2013 IEEE SDN For.

Benson, T., Akella, A., & Maltz, D. A. (2009). *Unraveling the Complexity of Network Management.* Paper presented at the NSDI.

Braden, R., Clark, D., & Shenker, S. (1994). *Integrated Services in the Internet Architecture: an Overview* (2070-1721). Retrieved from *Internet Engineering Task Force (IETF), RFC 1633*

Burger, E. W., & Seedorf, J. (2009). Application-layer traffic optimization (ALTO) problem statement. *Internet Engineering Task Force (IETF), RFC 5693*.

Carlson, M., Weiss, W., Blake, S., Wang, Z., Black, D., & Davies, E. (1998). *An Architecture for Differentiated Services*.

Casado, M., Koponen, T., Shenker, S., & Tootoonchian, A. (2012). *Fabric: A Retrospective on Evolving SDN.* Paper presented at the Proceedings of the first workshop on Hot topics in software defined networks.

Civanlar, S., Parlakisik, M., Tekalp, A. M., Gorkemli, B., Kaytaz, B., & Onem, E. (2010). *A Qos-enabled Openflow Environment for Scalable Video Streaming.* Paper presented at the GLOBECOM Workshops (GC Wkshps), 2010 IEEE.

Diorio, R. F., & Timóteo, V. S. (2016). *Multimedia Content Delivery in OpenFlow SDN: An Approach Based on a Multimedia Gateway.* Paper presented at the Computational Science and Computational Intelligence (CSCI), 2016 International Conference on.

Egilmez, H. E., Dane, S. T., Bagci, K. T., & Tekalp, A. M. (2012). *OpenQoS: An OpenFlow Controller Design For Multimedia Delivery With End-to-end Quality of Service Over Software-Defined Networks.* Paper presented at the Signal & Information processing association annual summit and conference (APSIPA ASC), 2012 Asia-Pacific.

Egilmez, H. E., Gorkemli, B., Tekalp, A. M., & Civanlar, S. (2011). *Scalable Video Streaming Over OpenFlow Networks: An Optimization Framework For QoS Routing.* Paper presented at the Image Processing (ICIP), 2011 18th IEEE International Conference on.

Egilmez, H. E., & Tekalp, A. M. (2014). Distributed QoS architectures for multimedia streaming over software defined networks. *IEEE Transactions on Multimedia, 16*(6), 1597-1609.

Foster, N., Harrison, R., Freedman, M. J., Monsanto, C., Rexford, J., Story, A., & Walker, D. (2011). Frenetic: A network programming language. *ACM Sigplan Notices, 46*(9), 279-291.

Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., & Shenker, S. (2008). NOX: Towards An Operating System For Networks. *ACM SIGCOMM Computer Communication Review, 38*(3), 105-110.

Habibi Gharakheili, H., Sivaraman, V., Moors, T., Vishwanath, A., Matthews, J., & Russell, C. (2017). Enabling Fast and Slow Lanes for Content Providers Using Software Defined Networking. *IEEE/ACM Transactions on Networking (TON), 25*(3), 1373-1385.

Jamjoom, H., Williams, D., & Sharma, U. (2014). *Don't Call Them Middleboxes, Call Them Middlepipes.* Paper presented at the Proceedings of the third workshop on Hot topics in software defined networking.

Jarraya, Y., Madi, T., & Debbabi, M. (2014). A Survey and a Layered Taxonomy of Software-defined Networking. *IEEE communications surveys & tutorials, 16*(4), 1955-1980.

Jimenez, J. M., Romero Martínez, J. O., Rego, A., Dilendra, A., & Lloret, J. (2015). *Study of Multimedia Delivery Over Software Defined Networks.* Paper presented at the Network Protocols and Algorithms.

Karl, M., Gruen, J., & Herfet, T. (2013). *Multimedia Optimized Routing in OpenFlow Networks.* Paper presented at the Networks (ICON), 2013 19th IEEE International Conference on.

Kim, H., & Feamster, N. (2013). Improving Network Management With Software Defined Networking. *IEEE Communications Magazine, 51*(2), 114-119.

Koponen, T., Casado, M., Gude, N., & Stribling, J. (2014). Distributed Control Platform for Large-scale Production Networks: Google Patents.

Krishna, P. V., Saritha, V., Vedha, G., Bhiwal, A., & Chawla, A. S. (2012). Quality-of-service-enabled Ant Colony-based Multipath Routing for Mobile Ad hoc Networks. *IET communications, 6*(1), 76-83.

Laga, S., Van Cleemput, T., Van Raemdonck, F., Vanhoutte, F., Bouten, N., Claeys, M., & De Turck, F. (2014). *OptimizingSscalable Video Delivery Through OpenFlow Layer-based Routing.* Paper presented at the Network Operations and Management Symposium (NOMS), 2014 IEEE.

Lara, A., Kolasani, A., & Ramamurthy, B. (2014). Network Innovation Using Openflow: A Survey. *IEEE communications surveys & tutorials, 16*(1), 493-512.

Le Roux, J. (2009). Path Computation Element (PCE) Communication Protocol (PCEP). *Internet Engineering Task Force (IETF), RFC 5440.*

McKeown, N. (2009). Software-defined Networking. *INFOCOM keynote talk, 17*(2), 30-32.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., . . . Turner, J. (2008). OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review, 38*(2), 69-74.

Newman, P., Minshall, G., & Lyon, T. L. (1998). IP Switching—ATM under IP. *IEEE/ACM Transactions on Networking (TON), 6*(2), 117-129.

Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K., & Turletti, T. (2014). A Survey of Software-defined Networking: Past, present, and Future of Programmable Networks. *IEEE communications surveys & tutorials, 16*(3), 1617-1634.

Owens II, H., & Durresi, A. (2015). Video Over Software-defined Networking (vsdn). *Computer Networks, 92*, 341-356.

Parsaei, M. R., Khalilian, S. H., & Javidan, R. (2016). A Comparative Study on Fault Tolerance Methods in IP Networks Versus Software Defined Networks. *International Academic Journal of Science and Engineering, 3*(4), 146-154.

Ramos, F. M., Kreutz, D., & Verissimo, P. (2015). Software-defined Networks: On the Road To the Softwarization of Networking. *Cutter IT journal*.

Saint-Andre, P. (2011). Extensible Messaging and Presence Protocol (XMPP): Core. *Internet Engineering Task Force (IETF), RFC 6120.*

Seddiki, M. S., Shahbaz, M., Donovan, S., Grover, S., Park, M., Feamster, N., & Song, Y.-Q. (2015). *Flowqos: Per-flow*

Quality of Service for Broadband Access Networks. Retrieved from Georgia Institute of Technology.

Shu, Z., Wan, J., Lin, J., Wang, S., Li, D., Rho, S., & Yang, C. (2016). Traffic Engineering in Software-defined Networking: Measurement and Management. *IEEE Access, 4,* 3246-3256.

Sim, K. M., & Sun, W. H. (2003). Ant Colony Optimization for Routing and Load-balancing: Survey and New Directions. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 33*(5), 560-572.

Thenmozhi, R., Preethi, T., Sadhana, S., Shruthi, V., & Yuvarani, B. (2017). Integrating Multimedia Services Over Software Defined Networking. *International Research Journal of Engineering and Technology (IRJET), Vol 4(3)*.

Tomovic, S., Prasad, N., & Radusinovic, I. (2014). *SDN Control Framework for QoS Provisioning.* Paper presented at the Telecommunications Forum Telfor (TELFOR), 2014 22nd.

Voellmy, A., & Hudak, P. (2011). *Nettle: Taking the Ssting Out of Programming Network Routers.* Paper presented at the International Symposium on Practical Aspects of Declarative Languages.

Wallner, R., & Cannistra, R. (2013). An SDN Approach: Quality of Service Using Big Switch's Floodlight Open-source Controller. *Proceedings of the Asia-Pacific Advanced Network, 35*, 14-19.

Wang, H., Shi, Z., Ge, A., & Yu, C. (2009). An Optimized Ant Colony Algorithm Based on the Gradual Changing Orientation Factor for Multi-constraint QoS Routing. *Computer Communications, 32*(4), 586-593.

Yilmaz, S., Tekalp, A. M., & Unluturk, B. D. (2015). *Video Streaming Over Software Defined Networks with Server Load Balancing*. Paper presented at the Computing, Networking and Communications (ICNC), 2015 International Conference on.

Yu, T.-F., Wang, K., & Hsu, Y.-H. (2015). *Adaptive Routing for Video Streaming with QoS Support Over SDN Networks.* Paper presented at the Information Networking (ICOIN), 2015 International Conference on.

Zeeshan, G. A., Guru, R. S., Alam, I., Professor, M., & MVIT, B. (2016). Video Streaming Over Software Defined Networks with Server Load Balancing. *International Journal of Engineering Science, 2768*.