A CLASS-BASED ADAPTIVE QOS CONTROL SCHEME ADOPTING OPTIMIZATION TECHNIQUE OVER WLAN SDN ARCHITECTURE

Pacharakit Vanitchasatit and Teerapat Sanguankotchakorn

Telecommunications Field of Study, Asian Institute of Technology, Bangkok, Thailand

ABSTRACT

Recently, Software-Defined Networking (SDN), a network architecture approach that enables the network to be intelligently and centrally controlled by using software applications, has been introduced. Another important issue in the network management context is Quality of Service (QoS).

This work investigates the QoS provisioning of various traffic classes on an SDN-enabled network. We propose and implement the class-based adaptive QoS control scheme on SDN-enabled network for various traffic classes, namely VoIP, Video Streaming and File Transfer. The effectiveness of our proposed scheme is validated by simulation using Mininet and Ryu controller. The procedure to create the simulation platform and all details relevant to all software used are described step-by-step in detail. The main performance evaluation metric is the Maximum Number of Traffic Flows admittable with QoS while Average Throughput, Latency, Jitter, and Packet Loss Rate are maintained at the comparable level of the existing work in the literature called JMABC [11]. Our simulation results are illustrated with 95% confidence interval. According to the simulation results, it is obvious that our proposed class-based adaptive QoS control scheme adopting the optimization technique significantly outperforms the existing similar QoS provision scheme in terms of the maximum number of the high priority traffic flows (VoIP) admittable with QoS while the other evaluation metrics are maintained at the same level.

KEYWORDS

Software-Defined Networking (SDN), Quality-of-Service (QoS).

1. INTRODUCTION

In recent years, the traffic of various applications has been drastically increasing. There is a large number of applications running on the Internet or communication networks. Some of the most popular applications are such as web surfing, file transfer, voice, and streaming video. However, the fundamental structure of the traditional network, namely Internet, computer communication, and Internet Protocol (IP), has a basic fashion to provide the best effort services which are non-reliable [1], [7]. Additionally, the existing networks are required to provide different users' requirements with different fashion of services and resources. Therefore, the mechanism to allocate the resources and satisfy the users/applications' requirements is needed. Quality of Service (QoS) [1,] [3], [6], [7], [12], [16], [17], [19], [20] is one mechanism that can be used in this case. QoS is a set of technologies that work on a network to guarantee its ability to dependably run high-priority applications and traffic under limited network capacity [3], [6], [7], [12]. QoS technologies accomplish this by providing differentiated handling and capacity allocation to specific flows in network traffic under limited network capacity [16]. The QoS mechanism sequences packets and allocates bandwidth by queuing and distributing bandwidth depending on the priority of packets. It manages network resources such as bandwidth by controlling and setting priorities and policies for specific types of data on the network. This solution helps to manage throughput, packet loss, delay, and jitter on network infrastructure. There are two models currently used for QoS provisioning [7]: Integrated Services (IntServ) and Differentiated Service (DiffServ). DiffServ achieves better QoS scalability and is suitable for large-scale networks, while the IntServ provides a tighter QoS mechanism for real-time traffic. IntServ model is one of the solutions for real-time traffic but the main problem with IntServ working properly is all network devices along the traffic path must support it [18].

Recently, Software-Defined Networking (SDN), a network architecture approach that enables the network to be intelligently and centrally controlled by using software applications, has been introduced [3], [4], [8], [10], [12-16], [18], [21]. This technology is an approach to network management that enables dynamic, programmatically efficient network configuration in order to improve network performance and monitoring, making it more like cloud computing than traditional network management [10]. It also helps operators manage the entire network consistently and holistically, regardless of the underlying network technology. Based on the characteristics of SDN architecture that has a centralized controller [2],[4],[10], SDN seems to be appropriate for adoption in the current traditional network, where the complexity as well as the need for QoS provisioning to various applications having different QoS requirements, are challenging issues [2-4,] [6], [8], [12]. To improve the performance and reduce the complexity of the traditional network, SDN has been introduced to centrally control the network by using software applications. This architecture divides the network into a control plane that contains logically centralized controlling components and a data plane that contains only the network devices, which makes the network more intelligent. The centralized controller can perceive a global view of the network and users. The network operators can program their policies or forward rules for their network at any stage in networking [6]. That is, the network operator can control the network resource management in real-time. The new concept of a programmable network is the solution for end-to-end network management. Even though the number of users connecting to the network is increasing, but not all of them need the same amount of network resources. Implementing an adaptive QoS will improve network resource management to have higher efficiency and better performance [3].

In this work, we propose the class-based adaptive QoS control scheme on an SDN-enabled network adopting the optimization technique. The traffic types under consideration consist of VoIP, video streaming, and file transfer. The optimization technique is utilized in order to maximize the number of traffic flows that can be admitted into the network with QoS satisfaction. The simulation is carried out using software called Mininet and Mininet-WiFi [13] to create the network as well as all network elements, while the Phyton-based software called Ryu is used to create the SDN controller [2]. The performance of our proposed scheme is evaluated using the maximum number of traffic flows admittable with QoS as the major metric while the metrics namely, average throughput, latency, jitter, and packet loss rate are observed. The simulation results are compared with the existing QoS provisioning scheme called JMABC [11] and the best effort scheme. According to the simulations results, it is obvious that our proposed adaptive QoS control scheme on SDN can clearly admit the much higher number of traffic flows satisfying QoS requirements than JMABC [11] while the other performance metrics, namely, average throughput, latency, jitter and packet loss rate are maintained at the same level as JMABC [11], but much higher than Best Effort scheme.

The rest of this paper is structured as follows: In section 2, we describe the related works. Section 3 presents the proposed algorithms where Section 4 describes the simulation software used in this work in detail. Section 5 illustrates the simulation procedures, network topology, and simulation parameters. Section 6 shows the simulation results as well as a detailed discussion. Finally, we conclude our work and future work in section 7.

2. RELATED WORK

There are many works in the literature relating to the QoS provisioning scheme in the other networks [17-20] as well as in SDN architecture that has been carried out [2-4],[8],[10],[12],[15],[16],[20]. Here, we highlight some works that are of interest and relevant to our proposed scheme.

In [8], a QoS design called Open QoS was proposed. The OpenQoS is an end-to-end QoS support that uses centralized control capabilities over the networks. The mechanism of OpenQoS is similar to the IntServ model which supports QoS from an end-to-end network point of view. They use dynamic QoS routing where both the QoS requirements and network characteristics are considered. The experiment is based on a standard OpenFlow controller, Floodlight using video streaming traffic over a real OpenFlow test network. The quality difference between streaming with QoS support and without QoS support of a sample test is illustrated. In conclusion, OpenQoS can minimize packet loss and latency of the flows.

In [3], an adaptive QoS algorithm for data transfers in SDN was proposed. The framework, to adjust the resources by redistributing the bandwidth to make sure that each flow meets the QoS requirement, is implemented. To perform the evaluation, virtual switches, virtual hosts, traffic generators, and one controller was created by using the open-source software called Open vSwitch. The simulation results of their frameworks are compared with the existing QoS methods.In conclusion, their framework can outperform those other scheduling methods meaning that resource redistributing is one of the keys to improving the performance.

In [1], Ahmed, Hamma, and Nasir presented a two-level scheduling algorithm for the WiMax standard. In the first level, the scheduler allocates bandwidth to different types of traffic based on traffic demands and QoS requirements. Then, in the second level, they distribute bandwidth among traffic flows of the same traffic type. The simulation results show that the proposed solution can provide QoS for all of the traffic types that are supported by the standard. In the end, it can be concluded the concept of a two-level scheduler can improve the performance of the network scheduler.

In [11], Leong and Chieng propose a Joint Measurement-based Admission and Bandwidth Control (JMABC) control scheme which consists of three sub-modules, Feedback Monitoring Module which is used for monitoring throughput and sending the feedback, Admission Control Module that is used for traffic policing by comparing throughput of network with their threshold value, and Packets Scheduling Module is used to put traffic flow into QoS queue. Network Simulator Tool (NS2) is the software used to evaluate their proposed scheme. In conclusion, their work illustrated an improvement in terms of throughput and user control.

In [5], Carella, Yamada, Blum, and Luck presented Cross-Layer Orchestrator (CLO) using the information between the Application and the Network Layers to provide a QoS guarantee on the end-to-end network. They simulate various types of peer-to-peer traffic and conference video call traffics with different bandwidth requirements and QoS levels and then measure the bandwidth utilization of each flow and collect data. It can be concluded, by using information from both Application and Network layers, that they can create a scheme that is able to provide all data flow requirements.

In [15], Wang, Lin, and Luo proposed a scheme named the QAMO-SDN. Typically, QAMO is used to achieve QoS in data centers by controlling bandwidth using the reservation method in the OBS layer and Multipath TCP while maintaining throughput performance. The typical QAMO was designed for traditional networks and did not support SDN architecture, however, the proposed method in [15] enhances QAMO-SDN algorithm to make it compatible with SDN architecture and improve the performance as well. The simulation was developed using C++. As a result, QAMO-SDN can perform slightly better than QAMO and the traditional TCP network. In conclusion, their traditional network scheduler scheme can be implemented in SDN architecture and also can improve performance.

In [21], Kuribayashi S. proposed the scheme to use SDN together with NFV (Network Functions Virtualization) paradigms to dynamically shape the traffic flow. This scheme can select the optimal communication flows to be shaped, and the optimal shaping points dynamically. It was shown that the net cost can be reduced significantly by using this proposed scheme.

The concept of QoS has been proposed not only in SDN but also in various types of networks [17],[19],[20]. For example, in [17], the QoS in the protocol called OLSR in Mobile Ad Hoc Network was investigated under two types of traffic. This work proposed the parameters called weighted connectivity index to look for the next node to forward the packets to. The effectiveness was shown by simulation using NS2. It was concluded that the proposed scheme can provide better performance in terms of throughput, average end-to-end delay, packet delivery ratio, overhead and power consumption than the traditional OLSR. In [19], the generalized multi-constrained path QoS routing algorithm for mobile ad hoc network (G_MCP) was proposed. The weighted connectivity index and nonlinear cost function were used. The simulation was carried out using NS2 adopting OLSR as routing protocol. The simulation results illustrated that the proposed algorithm provided superior performance in terms of throughput, packet delivery ratio delay and success ratio than the traditional OLSR. While in [20], the general multi-constrained QoS routing using weighted metrics (G_MQW) was proposed. The nonlinear cost functions and relaxed Dijkstra's algorithm were adopted. The general mathematical closed-form was derived. The effectiveness of the proposed algorithm was confirmed by simulation using Matlab and Waxman network topology. According to the simulation results, the G_MQW provided better performance in terms of success ratio than most of the existing algorithms.

3. PROPOSED ADAPTIVE QOS CONTROL SCHEME

3.1. Overview of Adaptive QoS Control Scheme

In this section, the proposed adaptive QoS control scheme on SDN architecture is described and implemented. To practically implement our proposed scheme, we divide the control scheme into two subsections. The first one is the class-based QoS control module used to calculate the distribution of network resources for each traffic class, and the other is the QoS queue management module used for traffic policing and scheduling. Figure 1 illustrates the conceptual model of our proposed class-based adaptive QoS control scheme on SDN architecture.

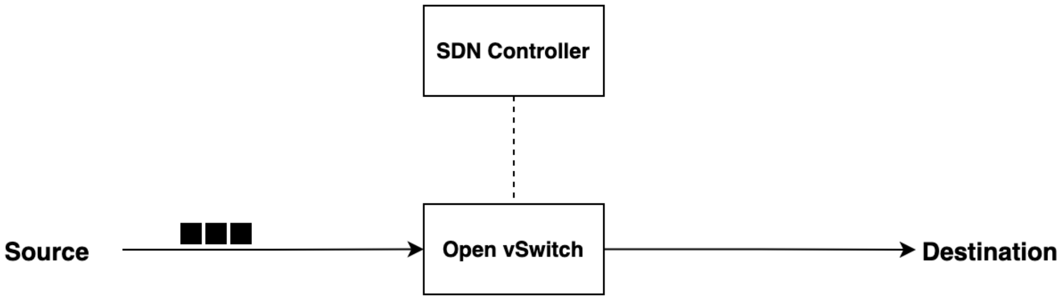

Figure 1. Adaptive QoS Control Scheme on SDN Architecture

From figure1, when the network is on, the SDN controller will collect information about the network, then run a QoS control algorithm and send a new policy rule to Open-vSwitch. The scheduling is performed when a new connection flow request had arrived at Open-vSwitch. The detailed workflow of this algorithm is shown as pseudo-code below.

ALGORITHM ADAPTIVE QOS CONTROL SCHEME

BEGIN
Part 1 - Class-based QoS Control Module
INPUT Network Topology, Number of Traffic Class (C),QoS Requirements
FOR $i = 1$ **TO** C
b_i = min_bandwidth (i) // Finding Minimum bandwidth of traffic class i
x_i = max_number (i) // Finding Maximum number of users for traffic class i
ENDFOR
Part 2 - QoS Queue Management Module
DEFINE w_i , q_i = empty queue for traffic class i
REPEAT
IF New Flow Request Arrives **THEN**
INPUT F = New Traffic Flow Request
c = getTrafficClass (F)
IF size(q_i) < x_i**THEN**
q_i.enqueue(F) // Every flow in $q_i$are admitted to network and will be remove automatically, if connection ended
ELSE
w_i.enqueue(F) // Waiting for service
ENDIF
ELSE
FOR $i = 1$ **TO** C
IF q_i is **NOT** full **THEN**
F = $w_i.de$queue()
q_i.enqueue(F)
ENDIF
ENDFOR
ENDIF
UNTIL No New Traffic Request **AND** w_i , $q_i$is **EMPTY**
END

3.2. Class-based QoS Control Module with Optimization Technique

The role of the proposed class-based QoS control module is to setup the QoS policy rule to define the treatment of network flow, which can be described step-by-step as follows:

Step 1: SDN controller collects information about network topology and QoS requirements.

Step 2: Calculate the minimum bandwidth requirement of different types of traffic based on traffic type and QoS requirements. The calculation of each traffic type depends on its traffic class. For example, in voice traffic, the minimum bandwidth requirement will be calculated by using audio codec and sampling rate.

Step 3: Calculate the maximum number of traffic flows for each traffic type by using Integer Linear Programming (ILP) optimization. To formulate ILP problem, the decision variables, objective function, and constraints are defined as follows:

maximize $$\sum_{i \in C} x_i$$

subject to

$$\sum_{i \in C} (b_i * x_i) \leq B$$

$$x_i \leq N, \forall i \in C$$

$$x_i \geq x_{i+1}, \forall i \in C$$

where N is the number of users in network
B is total system bandwidth of network
C is the number of traffic classes.
b_i is the minimum bandwidth requirement for each traffic class i^{th}.
x_i is the maximum number of flows of traffic class
i^{th} satisfying QoS requirement admittable into the network.

Here, we adopt the optimization technique as shown above to maximize the number of flows of each traffic class that can be admitted into the network by having bandwidth limitation constraints.

3.3. QoS Queue Management Module

In order to implement our proposed class-based adaptive QoS control scheme in SDN architecture, the procedure for our QoS management module needs to be defined.

Step 1: SDN controller collects information about the network such as network topology, traffic classes, QoS requirements, then run a class-based QoS control module to find the minimum bandwidth based on QoS requirement and maximum number of traffic flows for each traffic class that can be admitted.

Step 2: SDN controller sends a message to Open-vSwitch to define a new rule for traffic scheduling.

Step 3: Open-vSwitch defines new QoS queue based on traffic class. Each queue must contain minimum bandwidth requirement and maximum number of traffic flows that can be admitted into the network. Each QoS queue will be divided into two parts; the active queue for an active flow and waiting queue for the flow that cannot be admitted under current circumstance.

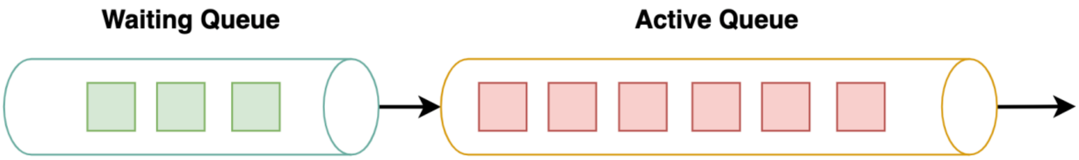

Figure 2. The conceptual model of QoS Queue

Step 4: Waiting for a new flow request arrives. Then, Open-vSwitch checks the traffic class of that requesting flow, classifies and sends it to QoS queue based on traffic class. Open-vSwitch will put the flow into the active queue, if the active queue is available. However, Open-vSwitch will put it into waiting queue to receive the service later on, if the current number of flows in the active queue has already reached the maximum value.

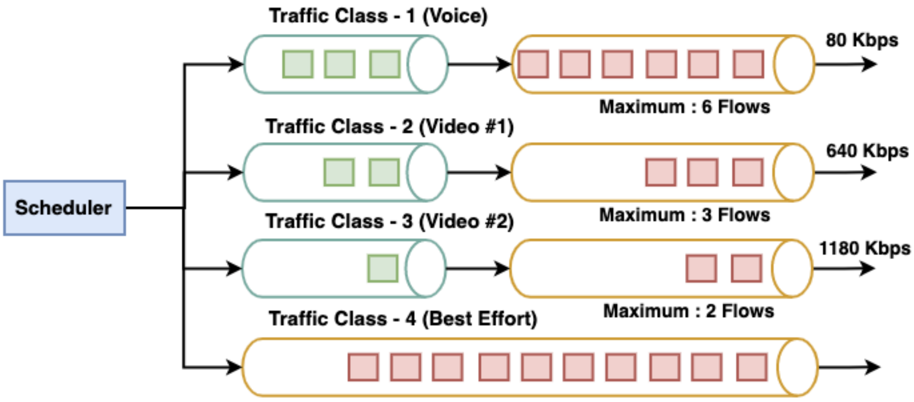

Figure 3. Example of QoS management module

Step 5: Checking whether any flows in the active queues end their connections, Open-vSwitch, then, has to admit the new traffic request from waiting queue to active queue.

4. Network Simulation Software

In this section, we describe the software used for simulation in our proposed scheme. The software used mainly in this work consists of two software; Mininet, a well-known network emulator for SDN architecture and SDN controller, and Ryu, a phyton-based opensource controller, which is compatible with the Mininet emulator.

4.1. Mininet

Mininet is a network emulator that has been developed by Mininet Core Team, [13]. It can create a network that consists of virtual hosts, switches, controllers, and links. Mininet is used for development and testing network simulations with OpenFlow protocol on SDN architecture.

4.2. Mininet-WiFi

Mininet-WiFi, [9] is an extension of the Mininet emulator where the functionalities of wireless station components such as WiFi stations and access points are added. The Mininet-WiFi has been developed based on standard Mininet code by adding or modifying classes and scripts. This means that Mininet-WiFi supports all standard Mininet components and can work along with the SDN controller being supported by standard Mininet.

4.3. Ryu Controller

Ryu is a python-based opensource SDN controller. Ryu provides a well-defined API for developers to manage network components and applications. Ryu supports various OpenFlow protocols which are used in the Mininet emulator.

4.4. IPerf

IPerf is a cross-platform network measurement tool that can generate traffic with various parameters. IPerf also has server and client functionality where we can create data flows to measure the performance from the end-to-end network.

5. NETWORK TOPOLOGY AND SIMULATION PARAMETERS

In this section, we illustrate all details regarding our simulation, namely system configuration, network topology, simulation parameters, etc. Here, all our simulations are carried out on a virtual network where the data plane is simulated on Mininet-WiFi emulator by creating virtual hosts, Open-vSwitches, and SDN controller which support OpenFlow protocol, while the control planeis carried out on Ryu SDN controller.

5.1. System Configuration

To set up the network simulation and network environment, it is necessary to install many software tools. The details of software tools of each element/system used in this simulation are shown in Table 1 as follows:

Table 1. System Configuration

Systems	Details
Operating System	Lubuntu 20.04
SDN Controller	Ryu 4.3.4
Switches	Open-vSwitch 1.3
Network Emulator	Mininet 2.2.2
Processor and Memory	3.6 GHz with 16 GB memory

5.2. Network Model and Environment

In this paper, we consider mainly on wireless local area network (WLAN). The topology of network under consideration consists of multiple wireless stations, one access point, one Open-vSwitch, one SDN controller, and one virtual server, as shown in figure 4.

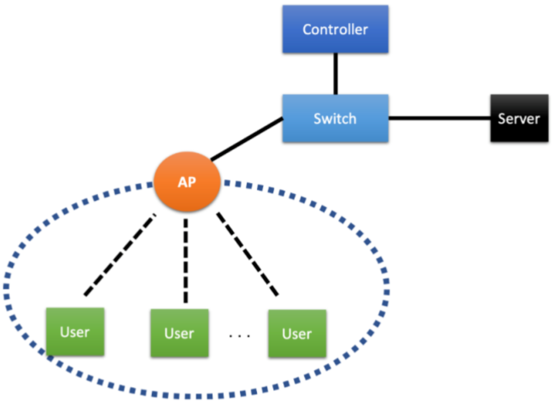

Figure 4. The Network Model under consideration

The parameters of virtual network environments for our simulation are illustrated in Table 2 as follows:

Table 2. Network Environment Parameters

Environments	Values
Total System Bandwidth	5 Mbps
Simulation Time	200 Seconds
Experimental Area	1000x1000 square meters
Position of User Wireless Stations	Uniform Distribution $U\{-300,300\}$
User Arrival Time	Poisson Distribution (1\λ = 10 s)
User Service Time	Exponential Distribution (1\λ = 80 s)
Noise Distribution	Gaussian Distribution – $N(0,1)$
Propagation Model	Log-Distance Path Loss - $\gamma = 3$
Number of Iterations	10 Times

5.3. Network Traffic Classes

In this work, three traffic classes are considered. The details of each traffic classes are shown in Table 3 as follows:

Table 3. Characteristics of each Network Traffic Classes

Traffic Class	Descriptions
Voice over IP	G.711 coded VoIP with 20 ms sampling rate payload
Video Streaming	H.264 coded Video, 704x480 resolution, 5.5 KB frame size, and 15 FPS
Files Transfer	High Speed Download

In our simulation, the concept of traffic priority is also introduced. The priority of traffic classes is shown in descending order as follows:

1) Voice over IP (VoIP)
2) Video Streaming
3) Files Transfer

We assume that each user's wireless stations randomly request a flow connection to Open-vSwitch according to network environment parameters shown in Table 2.

5.4. Performance Evaluation Metrics

The performance evaluation metrics used in this work consist of the major ones, which is the main contribution of this work; the Maximum Number of Admittable Traffic Flows with QoS, and the minor ones; Average Throughput, Average Latency, Average Jitter, and Average Packet Loss Rate. The maximum number of admittable traffic flows with QoS is adopted as the main evaluation metric since our objective is to efficiently allocate the limited resource to the applications that need QoS guarantee while providing no or less impact to the applications that QoS deems unnecessary. All performance evaluation metrics and their definitions are given in Table 4 as follows:

Table 4. Performance Evaluations Metrics

Major Performance Evaluation Metric	**Descriptions**
Max. Number of Admittable Traffic Flows	The maximum number of traffic flows admittable with QoSguaranteed for each class of traffic.
Minor Performance Evaluation Metric	**Descriptions**
Average Throughput	The total amount of data which is successfully transferred fromsource to destination under some certain period of time.
Average Latency	The time delay of traffic from source to destination under some certain period of time.
Average Jitter	The variation in the time between data packets arrivals under some certain period of time.
Average Packet Loss Rate	The percentage of packets that are unable to reach destination under some certain period of time.

The QoS requirements for each class of traffics; Voice over IP, Video Streaming, and File transfer are adopted from Cisco guidelines, [7] and are shown in Table 5 as follows:

Table 5. QoS requirements of Traffic Flows under Consideration

Traffic type	**Bandwidth Requirement**	**Average Jitter**	**Average Latency**	**Packet Loss Rate**
Voice over IP	17~106 Kbps per call depending on sampling rate	≤ 30 ms	< 150 ms	< 1%
Video Streaming	depending on the encoding and the rate of the video stream.	None	< 4 seconds	< 2%
File Transfer	20-25% of total bandwidth (recommendation)	None	None	None

6. SIMULATION PROCEDURES, RESULTS AND DISCUSSION

In this section, the simulation results based on the performance evaluation metrics are shown. Here, we compare the results achieved from our proposed class-based adaptive QoS control scheme with the other QoS provision schemes, namely JMABC [11] and the best effort scheme. All results are plotted using mean values with 95% confidence interval.

6.1. Simulation Procedures

The procedure to run software in our simulation is described step-by-step as follows:

6.1.1. Running Mininet Emulator and Ryu Controller

The first step of the simulation is to run the Mininet emulator along with Ryu SDN controller to create the network. The network topology of this simulation consists of 1 virtual server, 1 access point, 1 Open-vSwitch, 1 SDN controller, and 20 devices which are randomly distributed inside the network area under consideration.

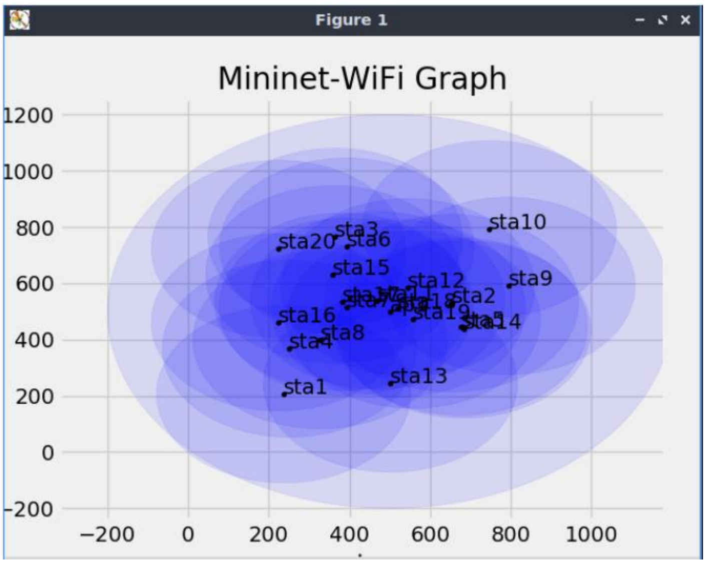

Figure 5. Network Topology showing devices distribution created by Mininet

6.1.2. Running Class-based Adaptive QoS Control Algorithm

Next, we run the software to evaluate the proposed class-based adaptive QoS control algorithm using three traffic classes shown in Table 3. When the minimum bandwidth requirement of each traffic class is identified, the maximum number of flows satisfying QoS requirements can be achieved as shown in Table 6 as follows:

Table 6. Results from Class-based Adaptive QoS Control Algorithm

Traffic Class	Minimum Bandwidth Requirement	Maximum Number of Flows
Voice over IP	64 Kbps per flow	10
Video Streaming	640 Kbps per flow	3
Files Transfer	940 Kbps	No limit

6.2. Simulation Results and Discussion

In this section, all simulation results based on the performance evaluation metrics are illustrated. The results achieved from our proposed algorithm are compared with the existing QoS provision in the literature called Joint Measurement-based Admission and Bandwidth Control (JMABC)[11] and Best Effort scheme.

6.2.1. The Major Performance Evaluation Metric

• The Maximum Number of Admittable Traffic Flows with QoS

The comparison of the maximum number of admittable traffic flows with QoS is shown in Fig. 6 as follows:

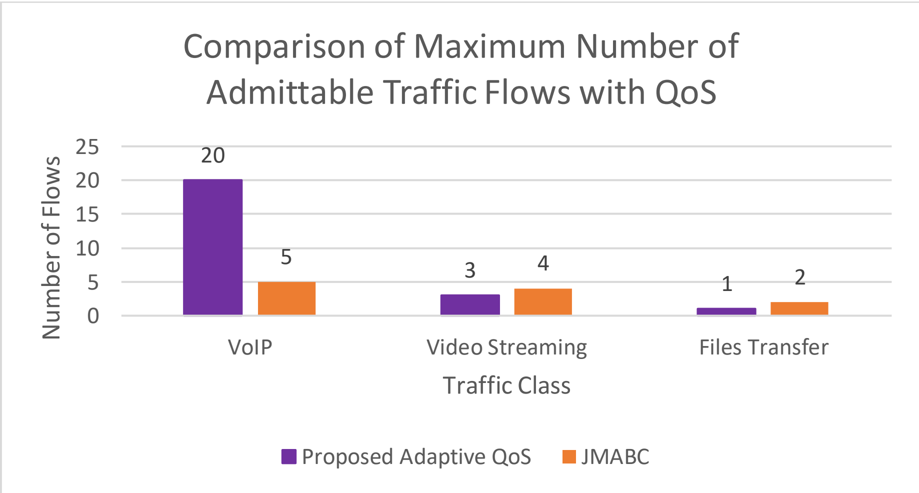

Figure 6. Comparison of Maximum Number of Admittable Traffic Flows with QoS

In this work, our main concern is to enhance the performance of the network by increasing the number of traffic flows admittable with QoS guaranteed using the optimization technique. Our proposed class-based adaptive QoS control scheme is simulated and compared with the existing work called JMABC [11] based on various traffic flows; VoIP, Video Streaming, and File Transfer. In our simulation scenarios, the highest priority is given to VoIP traffic, while the Video Streaming and File Transfer traffic are given the second and the third priority (Best Effort), respectively. Based on our simulation results, it is obvious that our proposed algorithm can admitdrastically the higher number of high priority traffic classes than JMABC [11], while the second priority as well as the best effort traffic are maintained almost at the same level under the same resources and environment. This is due to the priority given to VoIP and the optimization technique adopted to efficiently allocate the resources to the traffic.

6.2.2. Minor Performance Evaluation Metrics

• Average Throughput

Here, the average throughput of our proposed algorithm comparing to JMABC [11] and the best effort scheme of all traffic classes are illustrated.

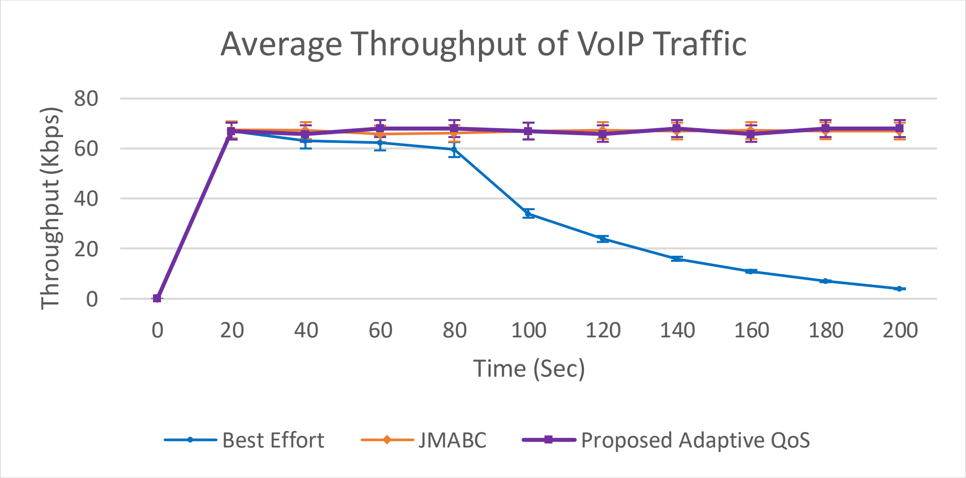

Figure 7. Average Throughput of Voice Over IP Traffic under various QoS provision schemes

Figure 7 illustrates the average throughput of VoIP traffic of all algorithms. In the beginning, the average throughput of all QoS control schemes is directly proportional to the input traffic. Once the throughput is increased beyond the minimum requirement (64 kbps), or a new traffic flow requests for the connection with a QoS guarantee, our proposed algorithm and JMABC [11] try to maintain the existing connections and if possible, admit the new connections satisfying QoS, while the best effort scheme could not maintain the connection. A new traffic flow has an impact on network performance as we can see that the throughput of VoIP traffic decreases rapidly in the case of the best effort scheme, but it has no effect when the QoS control scheme is adopted.

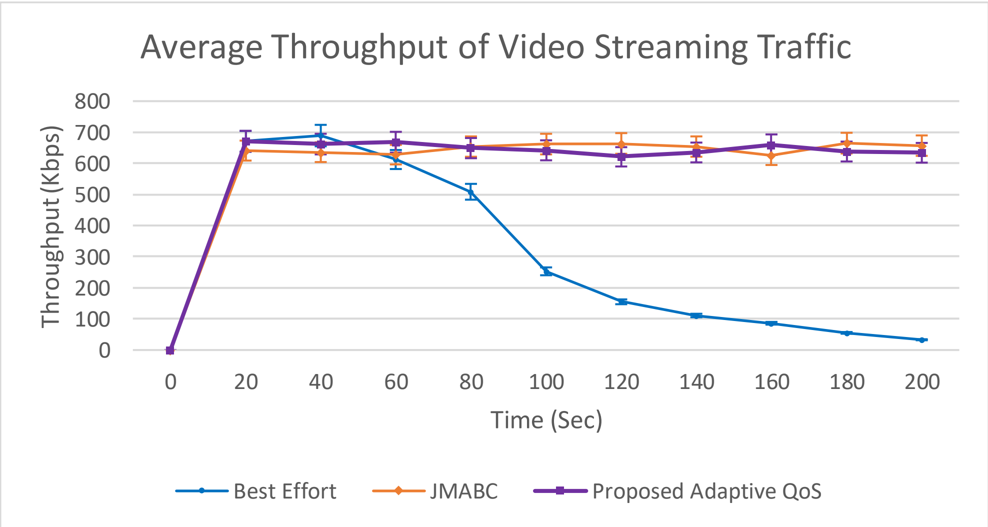

Figure 8. Average Throughput of Video Streaming Traffic under various QoS provision schemes

Similarly, figure 8 illustrates the average throughput of Video Streaming traffic of all algorithms. From figure 8, which is similar to VoIP traffic, when new traffic arrives, it has no impact on the performance of Video Streaming traffic. In addition, the Video Streaming traffic that has QoS control scheme can achieve minimum bandwidth requirement (640 kbps).

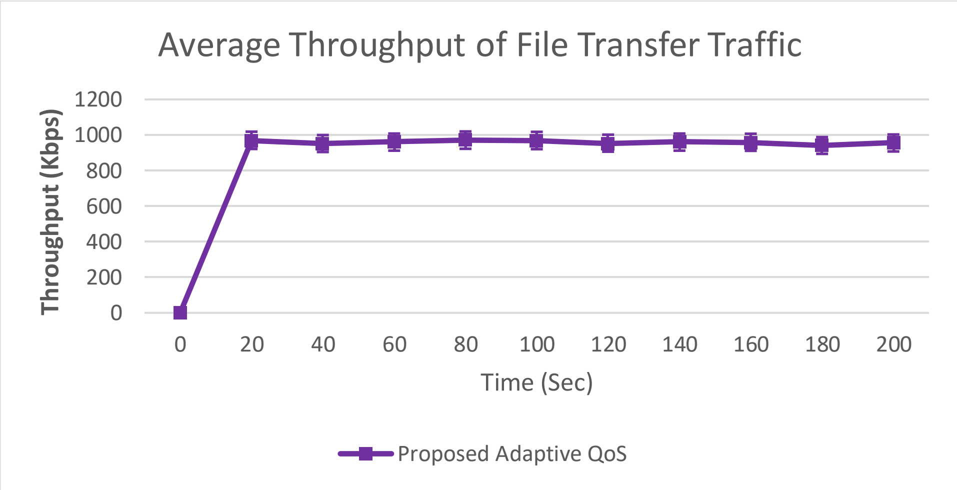

Figure 9. Average Throughput of File Transfer Traffic of our proposed Adaptive QoS provision scheme

Figure 9 shows the average throughput of File Transfer traffic which is classified as the best effort traffic. It is obvious that our proposed class-based adaptive QoS control scheme can achieve the average throughput of the File Transfer traffic nearly 1 Mbps which is approximately 20% of total system bandwidth even though our proposed adaptive QoS provision scheme has already admitted a large number of high priority traffics. This is because the optimization technique introduced in our proposed method can enhance efficiently the resource allocation.

- **Average Latency**

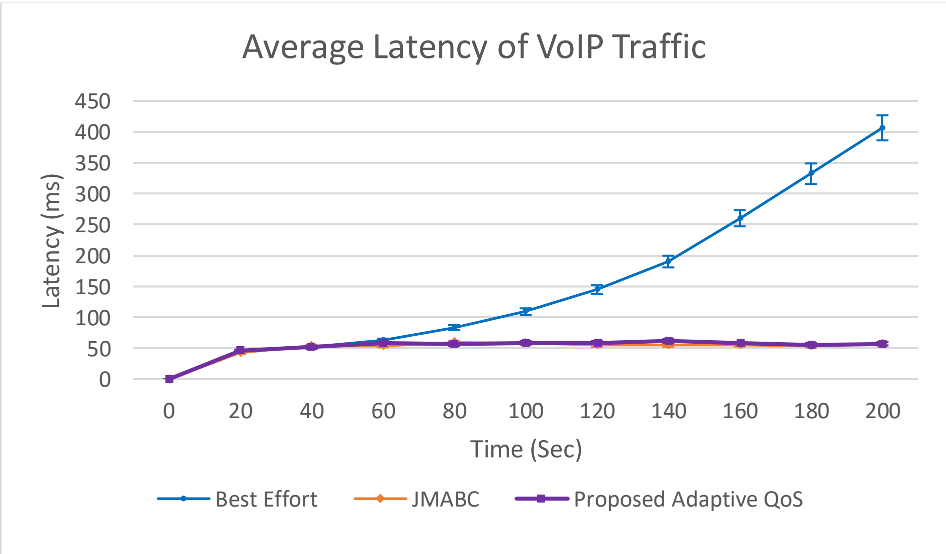

Figure 10. Average Latency of Voice over IP Traffic under various QoS provision schemes

Typically, latency is the most concerned parameter in Voice over IP traffic. The average latency achieved from our proposed algorithm is measured and compared with JMABC [11] and the best effort scheme, as shown in figure 10. Our proposed algorithm as well as JMABC [11] can achieve the recommended latency from Cisco which is lower than 150 ms throughout the time duration of the experiment. The rationale here is similar to the cases of throughput of VoIP.

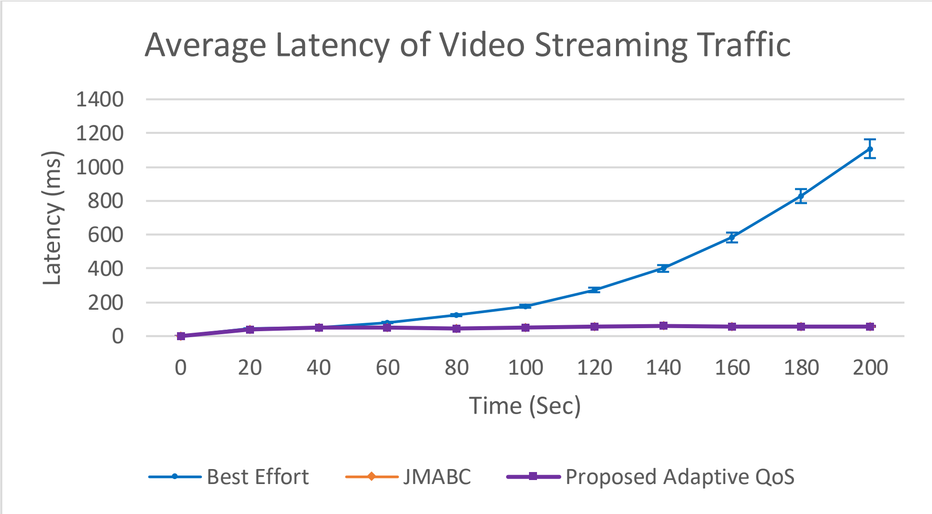

Figure 11. Average Latency of Video Streaming Traffic under various QoS provision schemes

Similarly, the average latency of Video Streaming traffic in figure 11 also follows the similar trend as VoIP traffic shown in figure 10.

- **Average Jitter**

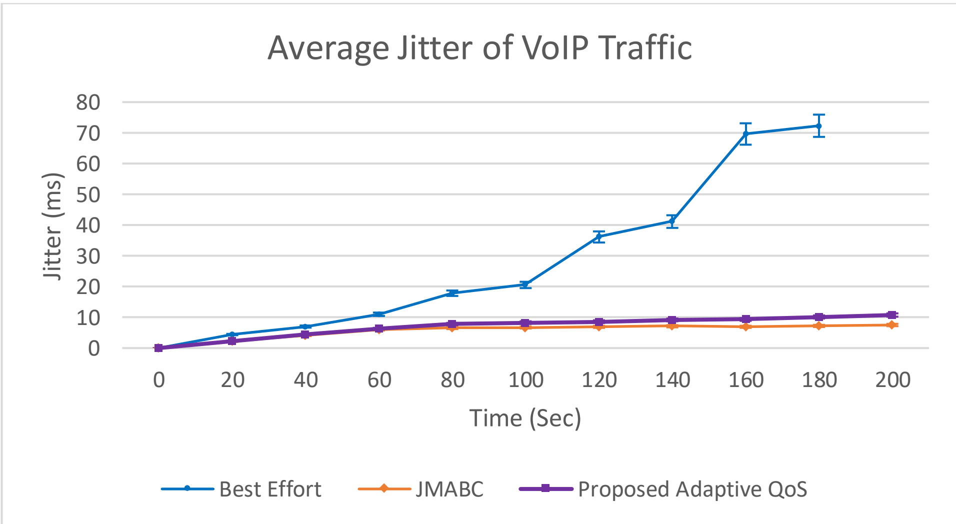

Figure 12. Average Jitter of VoIP Traffic under various QoS provision schemes

Figure 12 displays the average jitter of VoIP traffic of all algorithms. In figure 12, it is apparent that our proposed adaptive QoS control algorithm provides a lower average jitter than the minimum requirement (30 ms) which is almost the same as JMABC [11], while the Best Effort scheme cannot satisfy the QoS requirement.

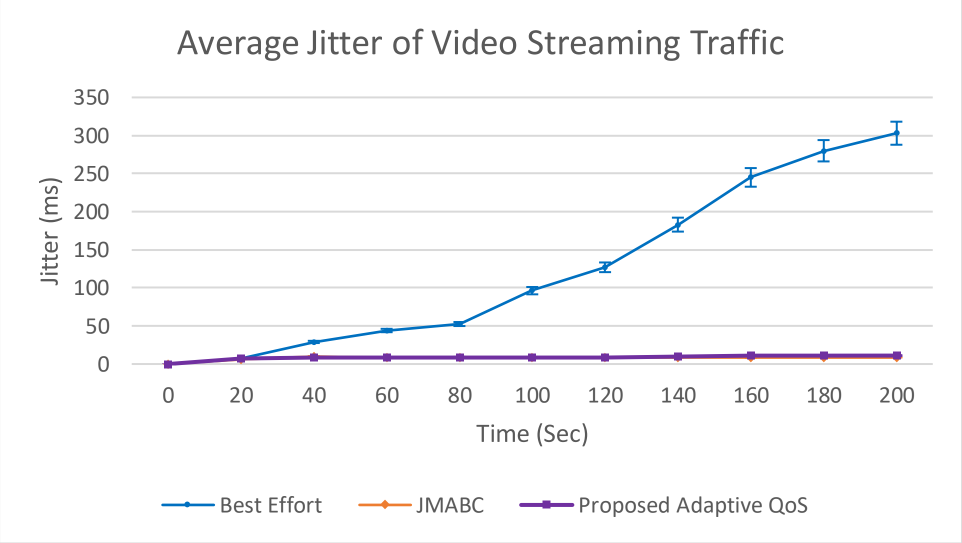

Figure 13. Average Jitter of Video Streaming Traffic under various QoS provision schemes

Figure 13 illustrates the average jitter of Video Streaming traffic of all algorithms. Similar to figure 12, it is apparent that our proposed adaptive QoS control algorithm and JMABC [11] provide a very lower jitter even there is no rigorous jitter requirement for video streaming traffic.

• Average Packet Loss Rate

Both of our proposed adaptive QoS control algorithm and JMABC [11] provide nearly 0% packet loss throughout the time duration of the experiment. Only Best Effort traffic has very high packet loss rate. Therefore, the results of this part are skipped.

7. Conclusions

This paper had demonstrated the integration of Quality of Service (QoS) provisioning and Software-Defined Network (SDN) by implementing Quality of Service in terms of Integrated Service on Software-Defined Network. All details regarding the software as well as implementation were described.

The class-based adaptive QoS control algorithm proposed in this work was the combination of class-based QoS control algorithm adopting optimization technique and QoS queue management in SDN. This adaptive control scheme had offered the method to provide a QoS guarantee to each traffic class and, additionally, had admitted the maximum number of traffic flows satisfying QoS requirements into the network. We had shown the effectiveness of our proposed algorithm by simulation using Mininet and Ryu controller and had compared the results with the existing work called Joint Measurement-based Admission and Bandwidth Control (JMABC) and the best Effort scheme.

It was obvious from the simulation results that our proposed algorithm could provide a QoS guarantee to all traffic under consideration. Additionally, due to the optimization technique adopted in our proposal, it was clear that our proposed algorithm could provide the much higher maximum number of admittable traffic flows (approximately 300% higher than the existing work (JMABC) in the case of VoIP traffic which had the highest priority and almost the same for Video Streaming traffic which had the second priority) satisfying QoS requirements under the same environment as JMABC. Finally, our proposed algorithm could maintain almost equivalent performance to JMABC in terms of average throughput, latency, jitter, and packet loss rate.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ACKNOWLEDGEMENTS

The success of this project is only possible because of the support, collaboration and trust of my advisor and institutions that has helped me to make them come true. Thanks all.

REFERENCES

[1] Ahmed Z., Hamma, S., & Nasir, Z. (2019) "An optimal bandwidth allocation algorithm for improving QoS in WiMAX", *Multimedia Tools and Applications, Springer Verlag*, Vol. 78, June 2019, pp.25937-25976.

[2] Amiri, E., Alizadeh, E., & Rezvani, M. H. (2020) "Controller selection in software defined networks using best-worst multi-criteria decision-making", *Bulletin of Electrical Engineering and Informatics*, Vol. 9, No.4, pp.1506–1517.

[3] Boley, J. M., Jung, E.-S., & Kettimuthu, R. (2016) "Adaptive QoS for Data Transfers Using Software-Defined Networking", *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Bangalore, India, November 6th-9th, 2016

[4] Braun, W., & Menth, M. (2014) "Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices". *Future Internet*, Vol. 6, pp.302–336.

[5] Carella, G., Yamada, J., Blum, N., & Luck, C. (2015) "Cross-layer service to network orchestration", *2015 IEEE International Conference on Communications (ICC)*, London UK, June 8th-12th, 2015

[6] Chen, J., Lv, T., & Zheng, H. (2004) "Cross-layer design for QoS wireless communications", 2004 *IEEE International Symposium on Circuits and Systems*, Vancouver, BC, Canada, May 23rd-26th, 2004.

[7] Cisco. (2006) "Implementing Quality of Service Over Cisco MPLS VPNs", *Selecting MPLS VPN Services*.

[8] Egilmez, H. E., Dane, S. T., Bagci, K. T., & Tekalpe, A. M. (2012) "OpenQoS: An open-flow controller design for multimedia delivery with end-to-end quality of service over software-defined networks", *Proceedings of the 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, Hollywood, CA, USA, December 3rd-6th, 2012.

[9] Fontes, R. R., Afzal, S., Brito, S. H. B., Santos, M., Rothenberg, C. E. (2015) "Mininet-WiFi: Emulating Software-Defined Wireless Networks", 2015 11th International Conference on Network and Service Management (CNSM), Barcelona, Spain, November 9th-13th, 2015.

[10] Kreutz, D., Ramos, F. M. V., Ver´ıssimo, P. E., Rothenberg, C. E., & Azodol, S. (2015) "Software-defined networking: A comprehensive survey", *Proceedings of the IEEE*, Vol.103, Issue 1, January 2015, pp.14-76.

[11] Leong, P. W., & Chieng, D. (2014) "A Joint Measurement-Based Admission and Bandwidth Control QoS Provisioning Scheme for WLANS", *International Conference on Frontiers of Communications, Networks and Applications (ICFCNA 2014)*, Kuala Lumpur, Malaysia, November 3rd-5th, 2014.

[12] Letswamotse, B. B., Modieginyane, K. M., & Malekian, R. (2016) "SDN based QoS provision in WSN technologies", *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, George, South Africa, September 4th-7th, 2016.

[13] Mininet Core Team. (2017) "Mininet: Rapid Prototyping for Software Defined Networks*", Open Networking Foundation (2015), Openflow switch specification, Open Networking Foundation(2016), SDN architecture overview*.

[14] Shome, P., Yan, M., Najafabad, S. M., Mastronarde, N., & Sprintson, A. (2015) "Crossflow: A cross-layer architecture for SDR using SDN principles" *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, San Francisco, CA, USA, November 18th-21st, 2015.

[15] Tariq, S., & Bassiouni, M. (2015) "Qamo-SDN: QoS-aware multipath TCP for software definedoptical networks", *12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, USA, January 9th-12th, 2015.

[16] Wang, P., Lin, S.-C., & Luo, M. (2016) "A Framework for QoS-aware Traffic Classification Using Semi-Supervised Machine Learning in SDNs', *2016 IEEE International Conference on Services Computing*, Vol. 6, San Francisco, CA, USA, June 27th- July 2nd, 2016, pp. 760–765.

[17] Sanguankotchakorn T., Wijayasekara S.K. andSugino N. (2018) "Performance of OLSR MANET adopting Cross-layer Approach under CBR and VBR Traffic Environment", *International Journal of Computer Networks and Communications (IJCNC)*, Vol. 10, No.6, November 2018.

[18] You, Z., Cheng, G., Wang, Y., Chen, P., & Chen, S. (2019) "Cross-layer and SDN based routing scheme for P2P communication in vehicular ad-hoc networks", *Applied Sciences*, Vol.9, Issue 22, November 2019.

[19] Kunavut K. and Sanguankotchakorn T. (2012) "Generalized Multi-constrained Path (G_MCP) QoS Routing Algorithm for Mobile Ad Hoc Networks", Journal of Communications, Vol.7, No.3, March 2012, pp.246-257.

[20] Poungmanee M. and Sanguankotchakorn T. (2020) "General Multi-constrained QoS Routing using Nonlinear Functions and Weighted Metrics", *Journal of Communications*, Vol.15, No.6, June 2020, pp.480-495.

[21] Kuribayashi S. (2021) "Dynamic Shaping Method using SDN and NFV Paradigms", *International Journal of Computer Networks and Communications (IJCNC)*, Vol.13, No.2 March 2021.

AUTHORS

Pacharakit Vanitchasatit was born in Loei, Thailand on April 14, 1997. He received B. Eng in computer engineering from Khon Kaen University, Thailand in 2019 and M. Eng in Telecommunications from Asian Institute of Technology, Bangkok, Thailand in 2021.

Teerapat Sanguankotchakorn was born in Bangkok, Thailand on December 8, 1965. He received the B. Eng in Electrical Engineering from Chulalongkorn University, Thailand in 1987, M. Eng and D. Eng in Information Processing from Tokyo Institute of Technology, Japan in 1990 and 1993, respectively. In 1993, he joined Telecommunication and Information Systems Research Laboratory at Sony Corporation, Japan where he holds two patents on Signal Compression. Since October 1998, he has been with Asian Institute of Technology where he is currently an Associate Professor at Telecommunications Field of Study, School of Engineering and Technology. Dr. Sanguankotchakorn is a Senior member of IEEE and member of IEICE, Japan.