IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 13, 2019, accepted November 30, 2019, date of publication December 13, 2019,
date of current version January 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2959351

Multiclass Queueing Network Modeling and Traffic Flow Analysis for SDN-Enabled Mobile Core Networks With Network Slicing

SANTHOSHA KAMATH[1], SANJAY SINGH[1], (Senior Member, IEEE), AND M. SATHISH KUMAR[2]
[1]Department of Information and Communication Technology, Manipal Institute of Technology, MAHE, Manipal 576104, India
[2]Department of Electronics and Communication Engineering, Manipal Institute of Technology, MAHE, Manipal 576104, India
Corresponding author: Sanjay Singh (sanjay.singh@manipal.edu)

ABSTRACT The back-haul networks of 5G are formed by heterogeneous links which need to handle massive traffic. The service providers are not able to provide good QoS for their users. The technology like Software Defined Networks(SDN) and Network Slicing helps a little for a service provider to providing QoS for multiple links. The service providers face a challenge in the efficient utilization of resources to fulfill the QoS requirement of users to comply with the growth and thereby increasing the revenue. These problems require an accurate traffic model to determine the steady-state of the system. The proposed model uses an architecture that has the combination of two technologies: SDN and network slicing, which empowers an administrator a flexible, programmable network, and the best management of network resources. Heterogeneous application is well managed by creating multiple logical networks called slicing. The slicing can be modeled using multi-class queuing networks. These technologies encourage service providers to fulfill QoS and revenue growth. To leverage the benefits of these technologies in allocating QoS is to identify the performance of the system, which requires a precise model of traffic to decide the steady-state condition of the framework. In this paper, we focus on SDN and slicing in mobile networks and quantify the performance measure considering an in-band OpenFlow architecture for a single node and homogeneous traffic class, which is further extended to the multi-class heterogeneous class queuing model and analyzed. The results obtained help a service provider to monitor the utilization of resources in every node by every class of core network, which in turn helps to allocate the resources precisely to fulfill QoS requirements.

INDEX TERMS Mobile networks, network modeling, network slicing, software defined network, traffic flow analysis.

I. INTRODUCTION

Current mobile backhaul networks are built with a large number of middle-boxes and forwarding devices. Each device is embedded with some intelligence in it to perform some functions. However, despite up-gradation and limited resources of mobile network technology, mobile service providers fall short of promising the customer requirement with expanding data demand and traffic pattern. Customers are using data services more, which are heterogeneous and rarely use the services provided by the service providers like voice, and SMS. Making an informed decision in such a situation demands that the service provider knows the user's QoS requirements and act accordingly. That enforces service provider to enable Over The Top (OTT) [1] services on their network. Providing OTT on the mobile network requires to solve many challenges. These OTT applications will have different transmission characteristics. OTT on mobile networks follows the characteristics of heterogeneous Ultra-Dense Network (UDN) of 5G backhaul networks. Heterogeneous application is well managed by creating multiple logical networks on the top of common physical infrastructure. Therefore network slicing along with SDN will have a great impact on managing the network.

The transmission characteristic for a given UDN varies from user to user in mobile networks. In other words, policies

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Martalo.

from user to user or application to application changes. It is worth noting that mobile traffic depends heavily on the day-time and less in the night and more in urban than in rural. All these make mobile traffic a non-ergodic pattern. The main impact of this is significant and multiple. They result in the migration of subscribers and hence, revenue loss for the service providers. Thus it is necessary to limit and manage the network traffic resources wisely and dynamically. The analytical model that we propose can be viewed at different levels: it can be used at the validation level to detect congestion in the mobile network traffic and at monitoring level, where the model analyzes the traffic flow and detect future congestion. The proposed system needs to be analyzed, which gives answers to the questions like

i) which application is used by the user most?
ii) which part of the system has more traffic? and
iii) which application is consuming what resources the most?

Answers for these questions help the service providers to deploy the policies and allocate the resources.

The above requirement demands an architecture for mobile networks that is flexible to deploy policies and utilize the limited resources to gain maximum benefits. One of the architectures proposed in line with this is as shown in Fig. 3. The proposed architecture uses two technologies: Software Defined Networks (SDN), and Network Slicing. In the next paragraph, we briefly discuss the SDN and network slicing.

SDN is a type of network system architecture that isolates the network into the data plane and control plane. The network devices that forward traffic from the control plane and the software logic that decides how the traffic should be processed through the network. The separation enables a system administrator to control network behavior, from a single high-level control program. OpenFlow [2] is used to communicate between the data and the control plane. Deployments of SDN helps to manage the variety of network problem in real networks.

Unlike traditional networks, SDN based network manages all the resources such as routing, firewall, encryption, gateways, Network Address Translation (NAT), Deep Packet Inspection (DPI) by the controller. OpenFlow based switches transmit the request to the controller, and the controller receives the request and applies the action on those packets.

The communication between the control and data plane in OpenFlow has two architectures: In-band and Out-band architecture, as shown in Fig. 1 and Fig. 2. In Out-band, there is a separate and dedicated connection between every SDN-enabled device and the controller. This type of architecture is followed when the distance between controller and devices is small such as data-centers. Whereas in In-band, both data plane and southbound traffic are sent through the same channel. This type of architecture can be used when the distance between the controller and SDN devices is large such as mobile networks. The applications like mobile networks, In-band architecture suits since it is deployed over a large geographical area.

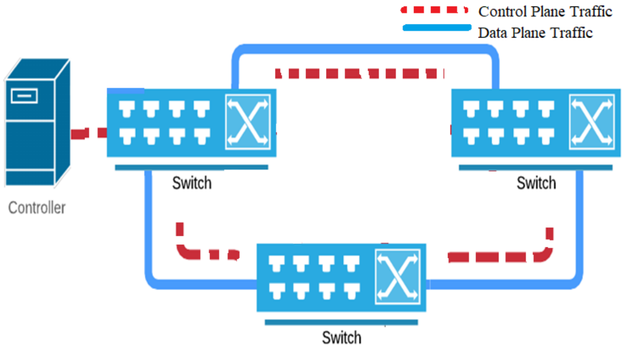


FIGURE 1. In-band SDN controller.

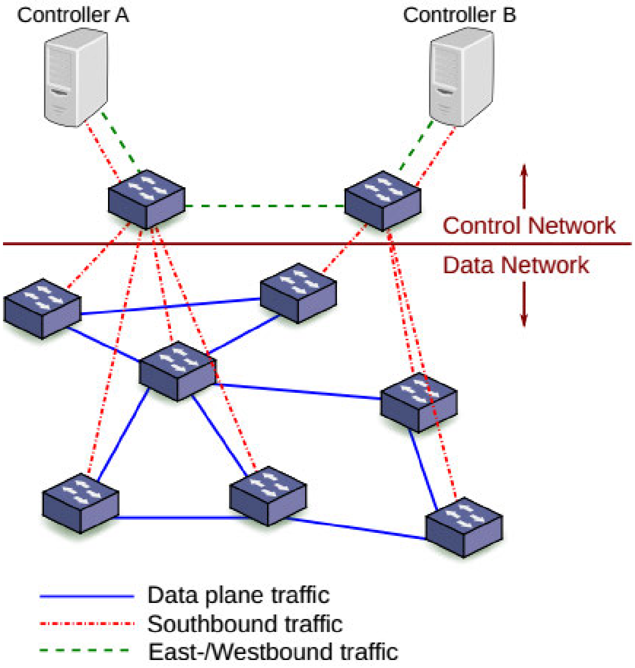


FIGURE 2. Out-band SDN controller.

Network Slicing is an emerging trend, which defines the production network into logical slices [3]. It enables different virtual systems to be made over a standard shared physical framework. The virtual systems are then tweaked to meet the particular needs of applications, services, and devices to the users or administrators.

Users might be able to control their network traffic in the given slice. Slicing enforces strong isolation between each slice, so that action in one slice does not affect another slice. Therefore each slice controls its packet forwarding. There are various reasons one might want to slice the network. One is that the network might have multiple administrative groups, such as different applications. The development of versatile correspondences frameworks towards the Fifth Generation (5G) must meet the testing essentials of Mobile Broadband (MBB) [4] use cases too and furthermore new various use cases related with clients of new market sections and vertical

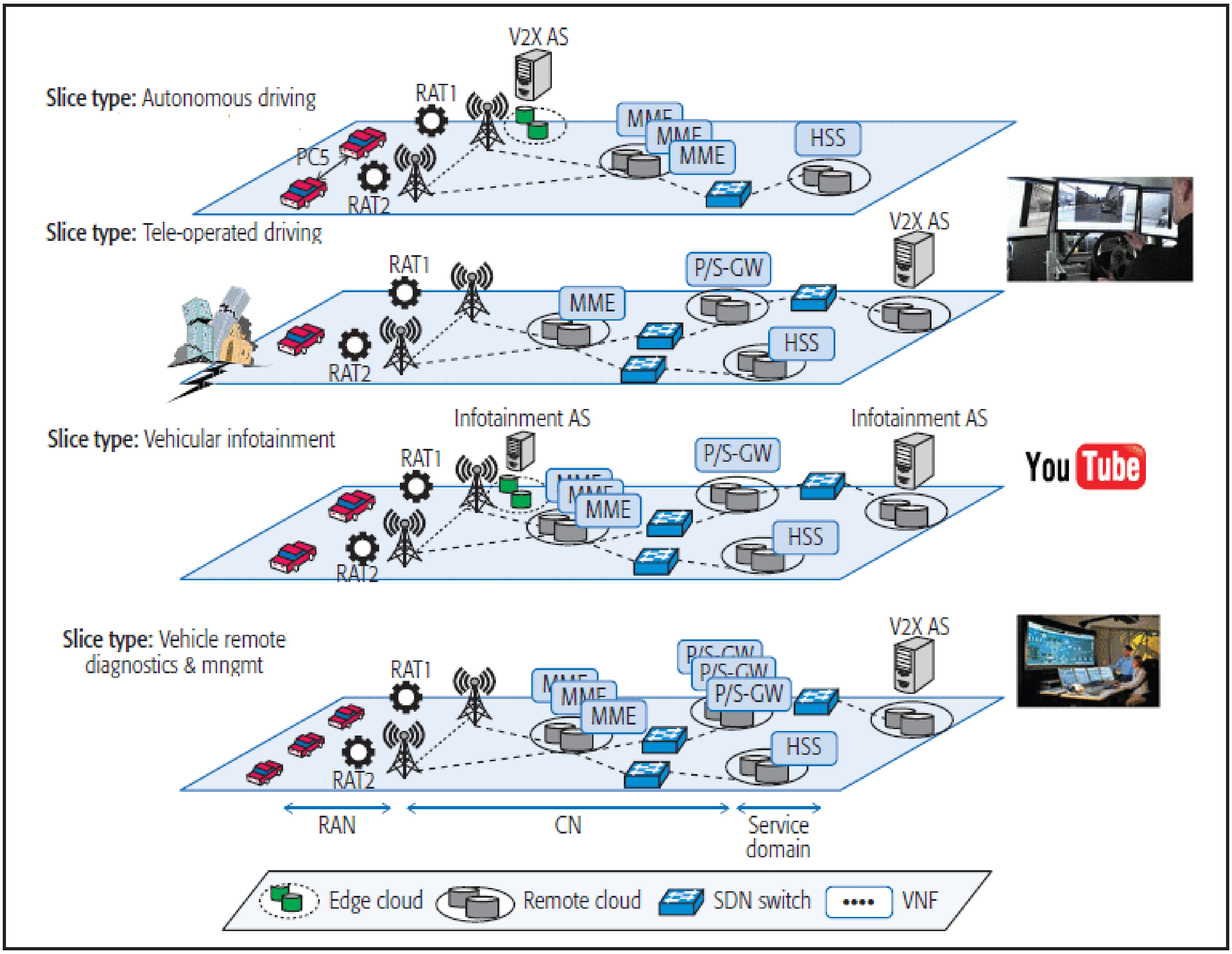


FIGURE 3. Example for network slicing.

enterprises (e.g., e-wellbeing [5], IoT, MBB and so forth) as shown in Fig. 3.

These applications have altogether different transmission qualities. For instance, UDN will commonly have a pervasive number of gadgets. However, every gadget may have low throughput. MBB has about the contrary properties since it will have a lot more modest number of gadgets, yet everyone will transmit or getting extremely high transfer bandwidth content. The objective of network slicing is to segment the physical system at a start to finish level to permit the ideal gathering of traffic, separation from different tenants. Therefore a slice type could be defined from a functional or behavioral perspective. It is therefore anticipated that mobile network operator could deploy a single slice that satisfies the need of multiple vertical, as well as multiple slices of different types that are packaged as a single product targeted on customers who has many and diverse requirements. For example, a vehicle may need simultaneously a high bandwidth for entertainment as well as an ultra-reliable slice for assisted driving. Heterogeneous application is well managed by creating multiple logical networks on the top of common physical infrastructure. Therefore network slicing along with SDN will have a great impact on managing the network.

In cellular networks such as 5G and Long Term Evolution (LTE), the SDN concept is incorporated into different parts of the core and backhaul network, such as Serving Gateway (S-GW), Packet Gateway (P-GW), Mobility Management Entity (MME), Home Subscriber Service (HSS), etc. However, most of the work related to SDN that we have come across was based solely on experiments and simulations. Once the packet enters into the network, its performance metrics depends on what are the different components on which application need to be processed. Such as S-GW, P-GW, HSS, and MME and its population matrix. The population matrix can be calculated by the observation or by using some distribution. Similarly, many messages are exchanged between a controller and a switch like *Packetin, Flowmode, PortStatus* to deploy the policies. These messages should be treated with high priority to reduce the delay. Therefore logically grouping these components based on applications helps to manage the network easily.

A mathematical model can provide researchers with insights into how an OpenFlow architecture performs according to specific parameters under given circumstances, thus leading to propose efficient algorithms. Therefore, we provide an analytical model based on results from the queuing theory of an In-band OpenFlow-based mobile network.

To the best of our knowledge, there is not much work done in multi-class traffic problems for a mobile network in the literature. However, there are a couple of experimental studies have been done that investigate the switch performance of SDN-enabled networks. Ludwig *et al.* [6] highlighted the performance of the switch factors like flow table size, processing rate, and packet arrival rate. Bosakov and Rizk [7] discussed the SDN switch performance using a network calculus-based approach without considering the feedback. Naous *et al.* [8] discussed the switch performance using the deterministic model. They have determined the performance of the SDN switch under the FPGA platform. Lazaris *et al.* [9] showed the effect of processing time on a high packet arrival rate. They demonstrated that the ability of the SDN switch negatively extends the service time of the switch, which likewise influences the execution of the control plane. The researchers concluded that the location of rules has a significant influence on a flow response time of the switch [10]. It is because of both the equipment and programming based switches having their processing abilities.

Apart from these few works like [11], [12] is carried out by simulation or using some measurement tool. Jarschel *et al.* [13] proposed a general $M/M/1$ based single node modeling for OpenFlow based SDN switch. In this, they considered that processing time is only exponential distribution. They extended their study in [14] for multiple nodes using the $M/M/1$ model. Sood *et al.* [15] proposed a $M/Geo/1$ based mathematical model for SDN switch, where the author assumes that the processing rate is geometric where it searches the policies in the table without prior knowledge and is modeled as geometric distribution. It assumes that complete processing is geometric. Therefore it is not clear whether the processing time should be considered as a geometric or exponential distribution.

Several studies [6]–[11], [13]–[15] carried out for Out-band OpenFlow networks. To the best of our knowledge, there is no work done in determining the performance of SDN-enabled network under slicing for in-band OpenFlow architecture in the literature for the model that consists of both SDN and slicing. Therefore first, we need to clarify the type of queue at a given SDN-enabled switch under In-band architecture, and we extend it to SDN-enabled network slicing for the architecture.

In general, SDN is assumed to be adopted in the backhaul/transport portion rather than the operation of a mobile network. In this paper, we try to link different aspects, including in-band OpenFlow-based SDN, queuing theory, cellular networks, and network slicing aspects [5].

We propose an analytical model for the mobile network that takes advantage of both SDN and network slicing concepts like considered in [5], which helps the network operator to offer the required QoS and also helps to utilize the resources at great extent. The traffic monitoring is one of the considerable challenges in the proposed architecture. The analytical tool can be used to monitor the traffic and utilization of the resources at every node by every slice. Our study is twofold; first, we want to find an adequate analysis model to represent traffic characteristics at mobile networks. Second, we want to validate the analysis model by performing a traffic simulation and determining performance measures of the mobile network. For the time being, we will consider the simulation model as a verification means to validate the analysis model. The analysis model is built with a multiclass queuing theorem where each class represents a slice or application. The main contributions of this paper are summarized as follows:

- This paper identifies the traffic characteristic at SDN-enabled mobile networks under slicing.
- Design of an adequate analysis model of flow-based Internet traffic to represent the traffic characteristic at mobile networks under In-band OpenFlow architecture for single node homogeneous class.
- Applying an analytical model for multi-class heterogeneous applications queuing theory to measure the performance matrices under network slicing for traffic monitoring at every node in an application.

The rest of the paper is organized as follows. Section II gives the traffic characteristics in SDN-enabled mobile network followed by a brief background of multi-class queuing and the proof to show that the type of queue in OpenFlow In-band architecture. Section III explains the application of an analytical model for multi-class under slicing. Section IV discusses the performance evaluation and application of our model to the network dimensioning and management, and Section V concludes this paper.

II. TRAFFIC CHARACTERISTIC IN SDN-ENABLED MOBILE CORE NETWORK USING NETWORK SLICING AND NETWORK MODEL

SDN-enabled devices use OpenFlow protocol for communication [16] between a switch and a controller. An OpenFlow-enabled switch essentially maintains a flow table. This table holds a set of entries, each of which comprises of match fields. The communication between a switch and a controller is bidirectional with a certain type of standard messages [17], such as *Packetin, Port-status, Flow-Removed, hello, echo, and Error messages*. When a packet arrives at a switch, policies for the packet in the flow table is checked and applied. If not found, then *Packetin* message is initiated and forwarded to the controller to get the policies. Similarly, if a user moves from one place to another, then *Port-status* message is initiated. These messages, that is, the messages exchanged between a controller and a switch are need to be considered as high priority messages. We logically divide the communication between the controller and the components as a slice. The logical division helps to manage and maintain the network and resources easily and effectively [5]. This concept is well explained in the slicing of the mobile network, as shown in Fig 3. After deploying the policies, depending upon the packets, it is necessary to process into a different component of core networks such as S-GW, P-GW, or into EPC, etc. We consider all these components as a node in our work, along with the simple forwarding devices. Similarly, in the mobile network, each user has different plans or different requirements. Therefore logically dividing the application on a shared medium helps to deploy the policies that make ease if network slicing is used. We consider each slice into a class. However, packet processing depends on the table size [13] and the policy. It varies from packet to packet of every slice. A mobile network is deployed on a large geographical area, and it is difficult to have a separate connection from every node to a controller. Therefore, In-band architecture is more suitable in such a scenario.

Packets that enter and exit a device may fall under packet-based, and flow or stream-based processing. Packet-based, or stateless processing treats packets discretely. Each packet is assessed individually for treatment. Packet level models for high-speed links are difficult to calibrate, because of the high level of multiplexing of many flows whose behavior is strongly influenced by the transport protocol and by the application. Besides, monitoring the traffic at the packet level becomes critical at the optical carrier (OC-192) and above link speeds. Packet-level models are useful in estimating parameters such as queue sizes at buffers and throughput. They have been vigorously used to study the stability of congestion control. Stream-based packet handling treats related packets in the same fashion. This model is used in the backbone network [18]. Furthermore, the flow level traffic is easy to monitor using tools like NetFlow [19]. In SDN, enabled network policies for the packet belong to a slice are the same; therefore, we consider flow-based processing.

Traffic Analysis (TA) of the above system can be modeled using queuing models (QM) [20]. QM is reasonable for analyzing frameworks where some constrained resources are utilized to play out a specific task. The resources are generally referred to as middle-boxes or nodes and the task as packets. Every packet has a service prerequisite, and after accepting it departs from the system. Kendall's annotation [20] is used to indicate the properties of a queue. Usually, the entries (arrival) of packets are stochastic, and the service prerequisites are accepted to pursue some likelihood probability distribution, and thus, the queue procedure is arbitrary (random) in nature. Queuing discipline decides how the capacity is shared among the packets at a given time; for instance, the First-Come-First-Serve (FCFS) discipline serves the packets in their arrival order. Processor Sharing (PS) shares the processor capacity among multiple class or arrival link. For example, the capacity of the server is shared between every packet equally or based on scheduling algorithms among the class. Frequently processor sharing refers to Egalitarian Processor Sharing (EPS).

In SDN-enabled network with slicing, we have different logical slices. Each slice has a packet belonging to different users or tenants. Each user or application carries packets in its slice. Packets belonging to different tenants or users are considered as classes, and each tenant may use different applications. It means user choice (of service plans) keeps on changing from time to time; therefore, packet-level modeling is suitable in this scenario. Similarly, the multi-tenants gives the motivation for the multi-class model. Here, we use the multiclass model to analyze the key factors: throughput, delay, and utilization statistics.

There are two types of multiclass models [21]: i) Open class, in which packets arrive from the external world into the system and exit out of the system, and ii) Closed class, in which packets are generated inside the network and die within the network. One can have a mixed model where both open and closed classes are considered.

The appropriate analytical model for SDN-enabled under slicing is shown in Fig. 4. In In-Band, if the SDN device does not find the policies, it forwards a packet to the controller using *Packetin* message as a new packet through other nodes. Furthermore, the controller deploys the policies using *Flow* message. Therefore there is a new flow generated if policies are not found for the incoming packet; we model this as a change of class. First, we evaluate the homogeneous network that is a single class, and then we extend our work to multi-class queuing of the In-band OpenFlow network.

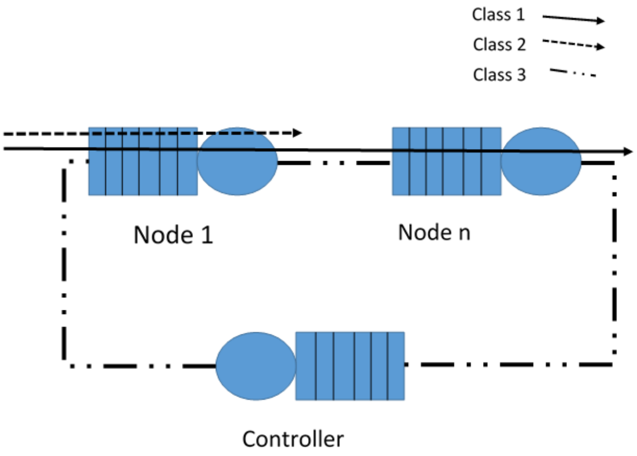


FIGURE 4. Multi-Class Model for In-band OpenFlow architecture.

Therefore we begin with the arrival distribution and then processing distribution at a node for SDN-enabled networks for a single class that is arrival and processing rate at every node because of using an in-band architecture, and then we extend it to multiple nodes with the multi-class network.

A. DETERMINING QUEUE IN AN SDN-ENABLED SWITCH FOR IN-BAND OPENFLOW ARCHITECTURE FOR HOMOGENEOUS NETWORK

The arrival and processing distribution at a node depends on two possibilities:

- availability of policies, and
- non-availability of policies.

We discuss these possibilities and conclude the type of queue that suits both the possibilities.

1) ARRIVAL DISTRIBUTION AT IN-BAND OPENFLOW NODE AND HETEROGENEOUS NETWORK

The packet arrives from outside network follows the Poisson distribution with arrival rate λ_1 at node 1, λ_2 at node 2, and λ_n at node n respectively. It is done based on the following assumptions:

- Flow arrivals follow a homogeneous Poisson process of finite rate λ. Poisson might be the right model if we consider recent findings by [22], [23] about the process of flow arrivals in the backbone of the Internet, where a large number of flows are multiplexed. It is shown in [23] that a Weibull distribution very well approximates the distribution of flow inter-arrival times with a shape parameter smaller than 1, and that as the traffic intensity increases, flow inter-arrival times become independent, whereas the Weibull shape parameter gets close to 1. Thus, the flow arrival process tends to be in good agreement with a Poisson process. The well-known results explain its limit on the superposition of marked point processes. The Poisson property is also known to apply to aggregates at the session level. Note that since our model does not depend on a particular definition of flow, one can group packets into sessions that have Poisson arrivals, and apply the model at the session level.
- Flow rate functions are independent of each other and identically distributed. The assumption on the independence of flow rate functions is based on the following facts:
  i) The link we consider is a backbone link kept under-utilized by engineering rules. It does not, therefore, experience congestion, and so it does not introduce dependency among the flow rate functions.
  ii) The flows sharing this link have a large number of different sources and destinations and use many different routes before being multiplexed on the backbone link. The assumption of identical distribution can be relaxed by introducing multiple classes (based on the transport protocol, flow size, or any other metric).

In our network, we assume p is the probability that a node does not find the policy for the packet that it has received. A packet that does not find the policy and the other traffic is forwarded according to its class to the next node. Besides, we assume that there are n nodes in the system between a switch and a controller. The controller deploys the policies for the request that comes from a node. The single-node open network model for in-band is shown in Fig. 5. We also assume l as the probability that packets are dropped from the path, that is, packets are sent to trans-coders or firewall for further processing. We consider it a leakage. If the event of no policies is discovered in the table, the packet is fed back to the same node after processing at the controller. The external arrival rate at the controller is λ_n. Switches and the controller processes each packet at a rate of μ_s and μ_c, respectively. Let Λ be the arrival rate at any given node after the feedback, which is the sum of external arrival and the packet that is fed back to the node by the controller. Let $\rho = \mu/\lambda$ is the intensity of load on a node. Applying the traffic rate Equation to Fig. 5, we get

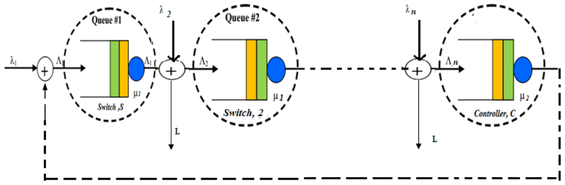


FIGURE 5. Open network model for an in-band architecture.

$$\Lambda_1 = \lambda_1 + p\Lambda_1 \tag{1}$$

At node 2

$$\Lambda_2 = \lambda_2 + p\Lambda_1 - l\Lambda_1 \tag{2}$$

Expecting no packet is lost at the controller, intermediate switches and also we assume that the only traffic that are sent to the controller for not finding policy will return back to the intermediate device, then we can rewrite the Equation (1) as

$$\Lambda_1 = \lambda_1 + p\Lambda_1 = \frac{\lambda_1}{1-p} \tag{3}$$

For simplicity at all the node we assume arrival rate is same that is λ, processing rate μ, and leakage l is same. Then we can rewrite the Equation (3) and (2) as

$$\Lambda_1 = \frac{\lambda}{1-p} \tag{4}$$

$$\Lambda_2 = \lambda + p\Lambda_1 - l\Lambda_1 \tag{5}$$

Similarly for the n^{th} node

$$\Lambda_n = \lambda + p\Lambda_{n-1} - l\Lambda_{n-1} \tag{6}$$

From Equation (3), one can make out that the total arrival input traffic is not Poisson, however bursty, that is, a packet may pass at most two times through the queue depending on the p value. There exist no analytical results for such networks in which inter-arrival and service times are dependent. The Kleinrock assumption [24] proposes that consolidating traffic streams on a transmission line has the impact of re-establishing the autonomy of arrival and service times. It implies that the service time of a packet is autonomously recovered each time the message is added back to the queue, which in turn implies that the same message is considered to have different service times each time it enters the queue. It eliminates arrival or service time correlation. With this approach, we can approximate that the arrival distribution is Poisson with a mean of Λ. Note that it is valid only if one uses Kleinrock's approach.

2) PROCESSING DISTRIBUTION AT SINGLE NODE

To determine the type of queue packet processing distribution at a node is also one of the vital features. Processing the packet follows two steps: i) searching the policy and ii) applying the policy. Searching policy follows geometric distribution as mentioned in [15] (denoted by variable A), and applying that policy follows an exponential distribution, as mentioned in [13] (denoted by variable B), and the combined effect of this is discussed in [25]. According to Santhosh [25], geometric distribution effect can be seen only when the load is low; otherwise, it follows an exponential distribution. Similarly, geometric progression leads to exponential at infinity. That means when the number of policies increases, the searching time leads to exponential. Therefore, there are two independent exponential distributions, one after another. Overall it gives an exponential response. The challenge here is that in [25], the authors considered Out-band architecture, but here the request needs to go through the n number of nodes; therefore, the total delay needs to be explored.

From Fig. 5 we infer that the total delay depends on the load on the other nodes also. If there are n nodes in a path, and we consider $\rho_k = \lambda_k/\mu_k$ as the load on k^{th} node, where $k \in n$, we explore the processing time at a node for two cases that is with leakage and without leakage.

i) *Without Leakage at a Node*: Total load in a path can be written as

$$\rho_t = \rho_1 + \rho_2 + \ldots + \rho_n \tag{7}$$

The Equation (7) can be rewritten in terms of arrival rate and processing rate as follows.

$$\rho_t = \frac{\Lambda_1}{\mu_1} + \frac{\Lambda_2}{\mu_2} + \ldots + \frac{\Lambda_n}{\mu_n} \tag{8}$$

Substituting (2), (3) and assumption made in (4) we get

$$\rho_t = \frac{\lambda_1}{\mu_1} + \frac{\lambda_1 + \lambda_2}{\mu_2} + \frac{\lambda_1 + \lambda_2 + \lambda_3}{\mu_3} + \ldots \\ \ldots + \frac{\lambda_1 + \lambda_2 + \lambda_3 \cdots + \lambda_n}{\mu_n} \tag{9}$$

If we assume same processing time for all the nodes that is $\mu_1 = \mu_2 = \ldots \mu_n = \mu$, then Equation (9) can be written as

$$\rho_t = \frac{n\lambda_1 + (n-1)\lambda_2 + \cdots + \lambda_n}{\mu} \tag{10}$$

ii) *With Leakage of l at a Node*: If we consider l leakage, that is, the packets exit that node after processing.

$$\rho_t = \frac{n\lambda_1 + (n-1)\lambda_2 + \cdots + \lambda_n}{\mu} \\ - l\left(\frac{n\lambda_1 + (n-1)\lambda_2 + \cdots + \lambda_n}{\mu}\right) \tag{11}$$

From Little's law [26] the length of the queue L_q is given by

$$L_q = \frac{\rho_t^2}{1 - \rho_t} \tag{12}$$

Equation (10) reveals that as the number of nodes increases the n^{th} node will have more load which is sum of arrival rate from its previous node with small portion of leakage at every node. On using Equation (11) in (12) it is straightforward to verify that the overall queue length increases exponentially when the load on any node in the path approaches 1. And as the number of nodes increases in the path, the waiting time increases. Similarly if the probability of availability of policies p is less then the model behaves as $M/H/m$ that is hyper-exponential distribution which means processing two times at m switches and once at the controller else the model behaves as $M/M/m$.

III. ANALYTICAL MODEL FOR MULTICLASS

In multiple class queuing models, we assume that there are C classes of requests, where $C = \{c_i | i = 1, 2, \ldots, n\}$. In our case we assume three classes. Our first slice is the messages that are exchanged between controller and the switches like *Packetin, Flow-mode*. The remaining two classes are the messages that are entered into the core network from outside of given applications. Each request from class c spends on average time S_{ck} at node k. For open models, we use λ_{ck} to denote the arrival rates, where λ_{ck} is the external arrival rate of class c requests at center k, which may be a node or switch. The transition probability matrix, P for multiple class networks is a $C \times K \times C \times K$ matrix with $P_{ij} = P_{i_{c_l} j_{c_m}}$ where $i, j \in K$ and $l, m \in C$. The P_{ij} is the probability that a packet of the c_l^{th} class at the i^{th} node is transferred to the c_m^{th} class at the j^{th} node (routing probability) [27].

The traffic Equation that is the set of the expected number of visits e_{ck} of class c for node k is given by [28]

$$e_{ck} = \lambda_{ck} + \sum_{\substack{c=1 \\ l,m=1}}^{C} \sum_{\substack{k=1 \\ i,j=1}}^{K} \lambda_{ck} P_{i_{c_l} j_{c_m}} \tag{13}$$

For the class 1 $P_{i_{c_l} j_{c_m}} \in \{0, 1\}$ which means if the policies available then it is either 0 or 1. For other class it depends on processing required at other nodes like S-GW, P-GW, MME etc.

In Equation (13) C corresponds to different slice categories, λ_{ck} is an arrival rate of a class-c packet at node k in the system, which describes an exogenous arrival rate and the arrivals directed from different nodes. It is equivalent to the departure rate of class c packets from switch k if there should be stability.

Let ρ_{ck} be the traffic intensity of the class c at node k. The performance parameter can be calculated for both PS and

FCFS systems using following equations [29]

$$\rho_k = \sum_{c=1}^{C} \rho_{ck}, \quad \text{where } \rho_{ck} = \frac{e_{ck}\lambda_{ck}}{\mu_{ck}}, \tag{14}$$

where ρ_k is the traffic intensity at node k, and μ_{ck} is the processing rate for packet of class c at node k. Utilization U_{ck} of k^{th} node by c^{th} class is given by

$$U_{ck} = \lambda_c D_{ck} \tag{15}$$

where D_{ck} is the demand for class c at node k, and defined as

$$D_{ck} = e_{ck}.S_{ck}. \tag{16}$$

The utilization U_k of node k is given by

$$U_k = \sum_{k=1}^{K} U_{ck}. \tag{17}$$

Residence time of the packet of class c at node k is given by

$$R_{ck} = \frac{D_{ck}}{1-U_k}. \tag{18}$$

The average queue length of class c at node k is given by

$$Q_{ck} = \frac{U_{ck}}{1-U_k}. \tag{19}$$

The system response time for class c is given by

$$R_c = \sum_{k=1}^{K} R_{ck}. \tag{20}$$

These equations are calculated based on assumption that packet arrives from outside network follows the Poisson distribution with arrival rate λ_{ck}.

IV. PERFORMANCE EVALUATION AND DISCUSSION

A. EVALUATION OF SINGLE NODE SINGLE CLASS QUEUE

We have tested the effect of using In-band communication and multi-class using MATLAB. For In-band, we have connected $M/M/m$ nodes in series and increased the arrival rate at every node from 0.1 to 0.9 using Poisson variate. The processing time is used by generating an exponential variate with a mean of 9.8 μs. According to [30] the average processing time of a 1500 byte packet takes 9.8μs at the switch because Maximum Transferable Unit (MTU) value in LTE core network is 1428 bytes [31]. Later we have increased the number of the node to three to test the effect of waiting time with the number of nodes. The simulation results are shown in Fig. 6 and 7 prove that waiting time increases exponentially at a shallow arrival rate as the number of nodes in the path increases; the result validates Equation (11). Similarly, without loss of generality, we conclude that nodes in series form a $M/M/m$ model with different mean. The new arrival depends on arrival at other nodes as well as leakage at every node.

The impact of policy and delay time for a generic application to meet the minimum service requirement has been reported in our earlier work [25]. It is necessary to treat the policy with high priority to reduce the delay for the controller request packet. One more observation is that if the controller is close to the node, delay in loading the policies into the node reduces. Therefore the hierarchical architecture for the placement of the controller will be a good choice in the use cases like mobile networks or delay-sensitive applications. Therefore we consider multi-class based modeling for In-Band architecture.

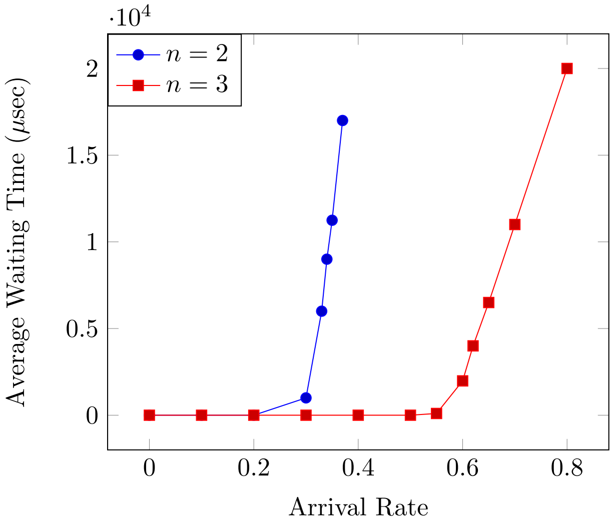


FIGURE 6. Relationship between average waiting time with the number of nodes in series.

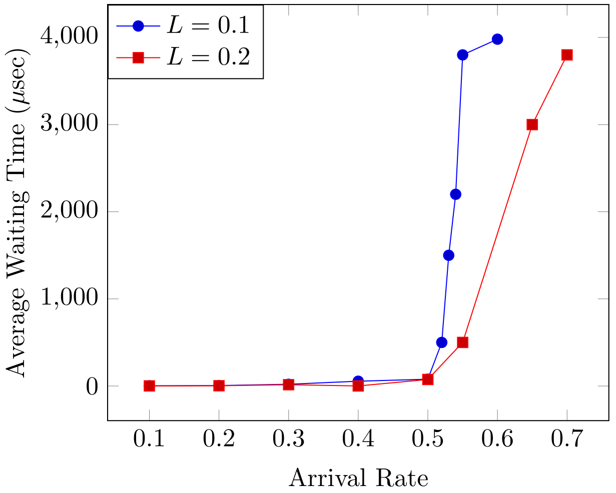


FIGURE 7. Relationship between average waiting time with the leakage factor in series.

B. EVALUATION OF MULTI-CLASS NETWORK MODEL

For the multi-class processing we consider result obtained in the section IV-A for modeling. From Equation (13) to Equation (20) one can make out that to calculate the performance parameters one need arrival distribution, processing distribution, and the transition probability matrix P. In the following section we explain how these parameters are adopted in our model.

1) ARRIVAL RATE AT MULTI-CLASS NETWORKS

The arrival rate at multi-class defines the number of packets arriving at the node k of class c. We use λ_{ck}, that is, the arriving rate, which is according to the Poisson arrival. We increase the arrival rate from 1 to N. We generate a Poisson variate depicting the Poisson arrival with a mean sample taken between 1 to N.

2) PROCESSING RATE AT MULTI-CLASS NETWORKS

The service rate matrix S, where S_{ck} is the time taken to process the packet of class c at node k. We assume the service rate of every single server is state-independent. The flow is Poisson arrival, and processing is FIFO, as discussed in section IV-A. We also assume that there is a single queue at the node instead of assuming a separate queue per line card. We consider average processing time at a switch for 1500 bytes of the packet is 9.8 μs, and processing time at controller μ_c is 240 μs [30] This value is close to the LAN. We generate exponential variate with $\mu_k = 9.8$ μs to depict the exponential service time. The output matrix S for the switch will have the values in μs, as shown in Equation (21). Where S is the class-node wise service matrix, and rows represent the given class, and column represents nodes.

$$S = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccc} k_1 & k_2 & k_3 & k_4 \end{array} \\ \begin{bmatrix} 2.33 & 19.10 & 7.70 & 18.90 \\ 11.13 & 5.30 & 12.50 & 9.10 \\ 9.80 & 9.00 & 13.40 & 12.40 \end{bmatrix} \end{array} \tag{21}$$

3) TRANSITION PROBABILITY MATRIX IN MULTI-CLASS NETWORKS

Routing probability matrix P is a $C \times K \times C \times K$ matrix. Transition probability depends on two main concepts they are i) Class switching probability and ii) Traffic matrix. The following section explains how they are computed in our model.

CLASS SWITCHING PROBABILITY

Class switching probability is a matrix of size $C \times C$, as shown below. It is the probability of switching a class after processing at a node. In our model (see Fig. 4), we consider that class c_3 carries the controller packets, that is, if a node does not find its policies it changes its class from any class to c_3 like *Packetin, Port-status, Flow-removed* etc. Switching from other classes is not considered in this simulation; however, one can consider this for priority-based scheduling in reality. The c_1, c_2 are application slice classes where packets arrives at the controller. The change of the classes is done according to the probability say p, where p is the probability that a switch finds policies in its table. Therefore change of class from $c_1 \rightarrow c_2$ and $c_2 \rightarrow c_1$ is 0. Wamser *et al.* [32] has shown that the probability of a new flow being observed is 0.16 at the switch. However, if the traffic is using a virtualized network like slicing, then the probability varies from $p = 0.2$ to $p = 0.5$. Similarly, $p = 1$ indicates that the controller handles all the traffic. We calculate the traffic parameters for different values of p. The class switching probability matrix C for $p = 0.9$ is given in Equation (23), and q is the probability of availability of class 2 policies in the switch.

$$C = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{ccc} c_1 & c_2 & c_3 \end{array} \\ \begin{bmatrix} p & 0 & 1-p \\ 0 & q & 1-q \\ 1-p & 1-q & 1-p+1-q \end{bmatrix} \end{array} \tag{22}$$

$$C = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{ccc} c_1 & c_2 & c_3 \end{array} \\ \begin{bmatrix} 0.9 & 0 & 0.1 \\ 0 & 0.9 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array} \tag{23}$$

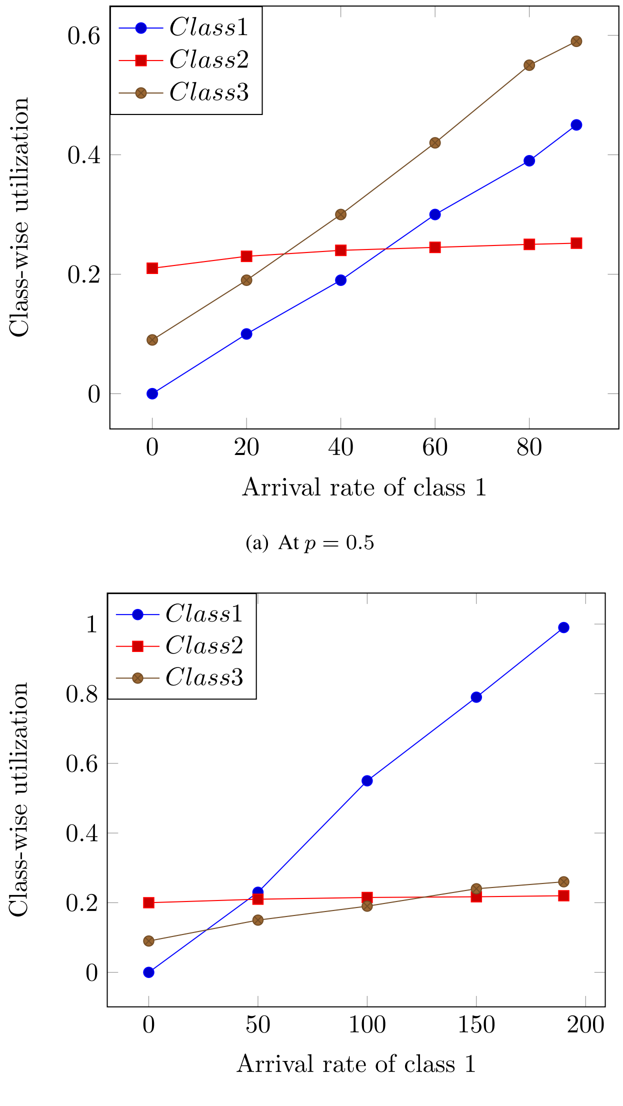
TRAFFIC MATRIX

The traffic matrix is the population exchanges between every pair of network nodes. In SDN-enabled slicing we assume that data transmitted between nodes follows a uniform distribution, except to the same node that is $M_{i,j} = 0$ when $i = j$, where $M_{i,j}$ is the amount of traffic movement from i^{th} node to the j^{th} node. For example, once the packet enters into the core network, S-GSN has to decide whether to send the packet to EPC and to which S-GW it should forward, which P-GW it should use to connect to the Internet. The sample of traffic matrix M is as shown in Equation (24).

$$M = \begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ k_4 \end{array} \begin{array}{c} \begin{array}{cccc} k_1 & k_2 & k_3 & k_4 \end{array} \\ \begin{bmatrix} 0.0000 & 0.2837 & 0.4540 & 0.2627 \\ 0.3414 & 0.0000 & 0.3073 & 0.3513 \\ 0.2617 & 0.2808 & 0.0000 & 0.4573 \\ 0.2068 & 0.4489 & 0.34501 & 0.0000 \end{bmatrix} \end{array} \tag{24}$$

By multiplying traffic matrix and class switching probability one can calculate P for the Equation 13. Using Equations (14) to (20) and matrices C, M, we can calculate the performance matrices of SDN-enabled network with slicing. The results obtained are shown in Fig. 8 to 14.

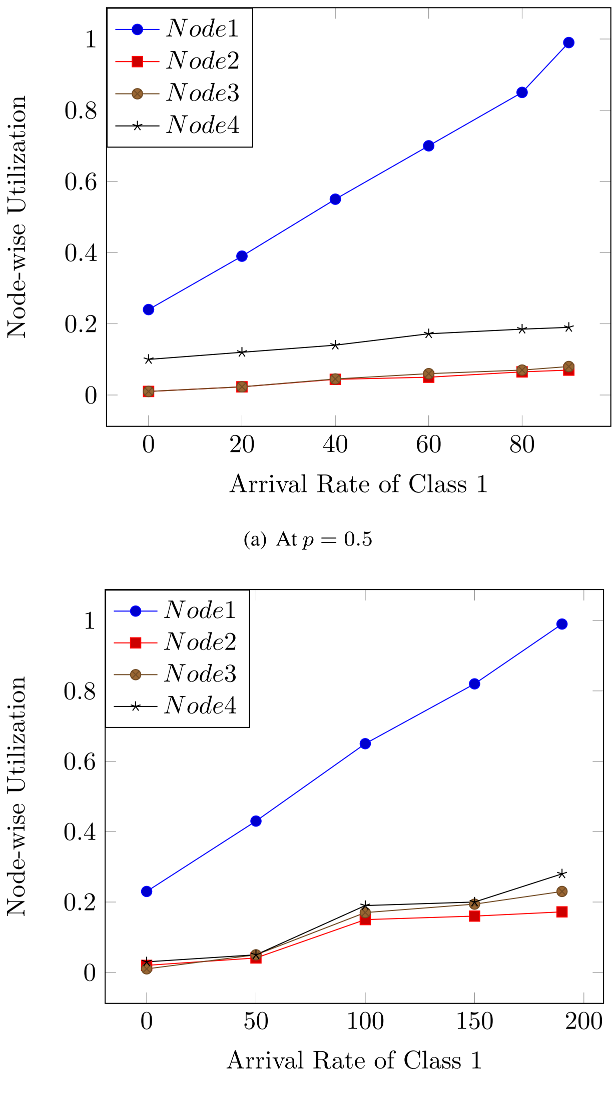
The graphs are shown in Fig.8 to 14 is the effect of availability of the policies on utilization, response time, throughput and the mean queue length of both the node and class at different probability p of class switching and arrival of class1 by keeping class 2 constant. From figure 9 the utilization of nodes increases linearly as per the Equation (15). It is because the demand for a given class in a node remains fixed, and the arrival rate increases linearly. The utilization, particularly at node 1 reaches maximum at around λ_{ck} of 80-90 arrivals per second for $p = 0.5$ and it takes around 170-180 arrivals per second for $p = 0.9$. This observation indicates that the availability of policies in the node supports more and more arrival of the packet for a given processing rate and probability matrix. In Fig. 8, the utilization of class 1



(a) At $p = 0.5$

(b) At $p = 0.9$

FIGURE 8. **Average class-wise utilization in the network at different values of *p*.**



(a) At $p = 0.5$

(b) At $p = 0.9$

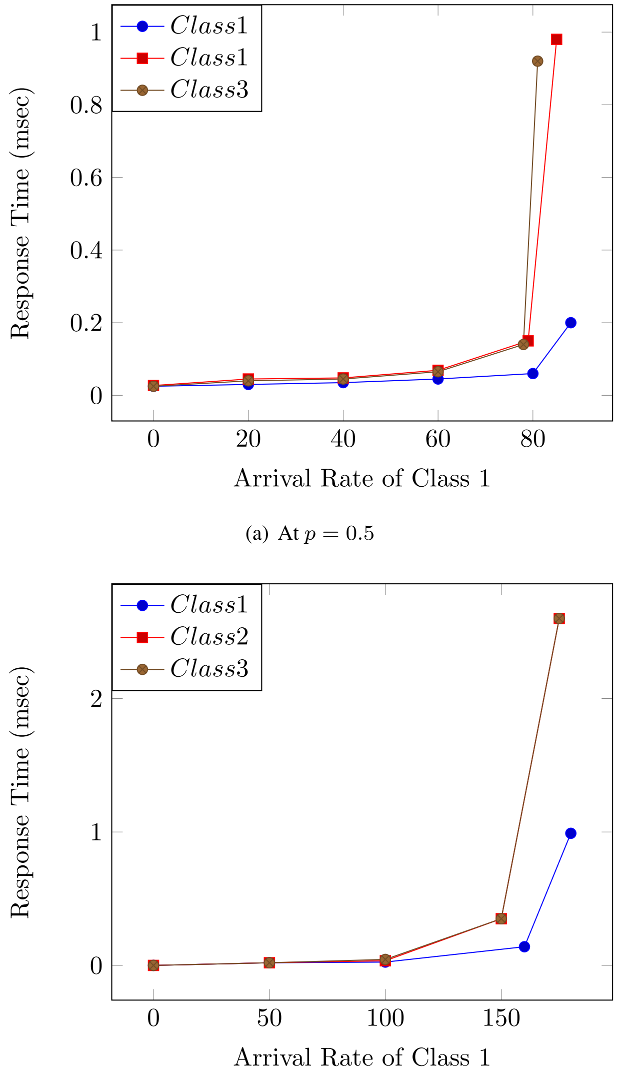
FIGURE 9. **Average node-wise utilization.**

is less when a node depends more on the controller for the policies. That is because a given node need not process the packet again. Moreover, it increases its utilization when there are policies deployed before the packet arrives at a node. Similarly, from Fig. 8, it is clear that the availability of the policies increases utilization of class 1. Also, at $p = 0.5$ utilization of class 3 is more, this is because of class switching from class 1 to class 3.

From Fig. 11 we infer that the response time for the given class is less if the policies are available. Therefore it is a good choice to use learning-based policy deployment in the nodes to reduce the response time of the application. From Fig. 10 we can conclude that in both the cases that are at $p = 0.5$ and $p = 0.9$, the response time increases exponentially when the node reaches its utilization close to 1. The average queue length increases when the load on a node reaches its maximum capacity, and the throughput of the class increases with more increase in probability.

From Fig.12 and 13 we can infer that node and class throughput is more when the probability of having policies in the node are more. Therefore, deploying the policies in advance helps the SP to utilize and plan the network resources. Since mobility is involved in the cellular network deploying the policies in advance is a complex operation, lot of packets like *Packetin, Port-status, Flow-removed* messages are exchanged between controller and switches, which increases the traffic in the network hence leading to more resource consumption. Such an increase in resource consumption can be reduced using learning based techniques. Figure 14 gives the statistics of the queue in every node for different classes. It helps the SP to plan the resource allocation for a different slice.

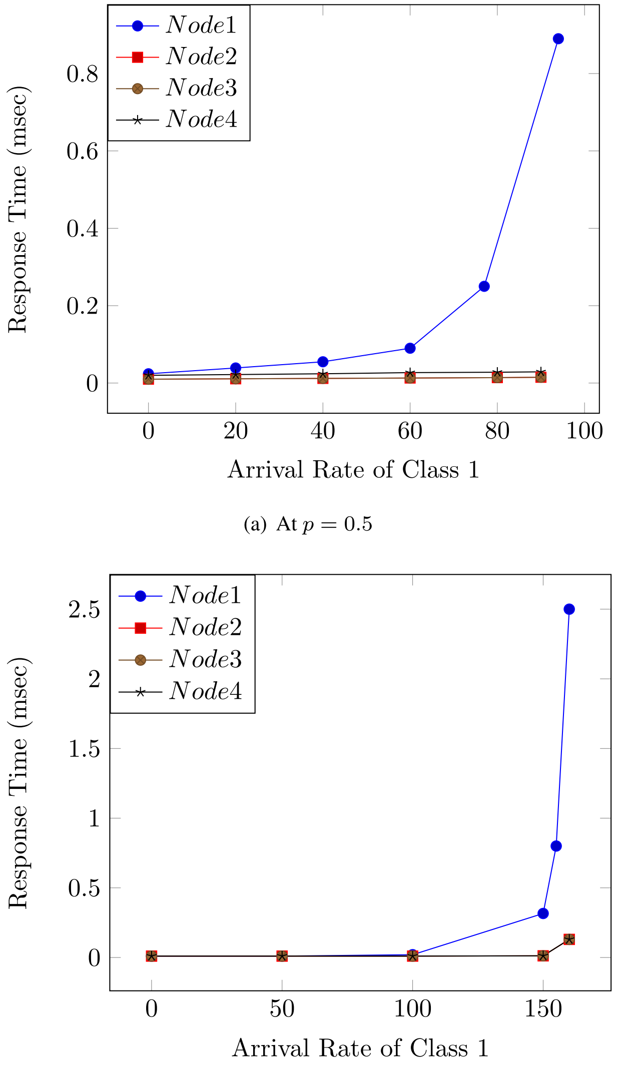
From Fig 8 and 9, it is observed that as the number of users of any class/ application increases, the utilization of the whole network increases. Furthermore, as the utilization of a given node reaches the maximum, the waiting time for the



(a) At $p = 0.5$

(b) At $p = 0.9$

FIGURE 10. Class-wise response time.



(a) At $p = 0.5$

(b) At $p = 0.9$

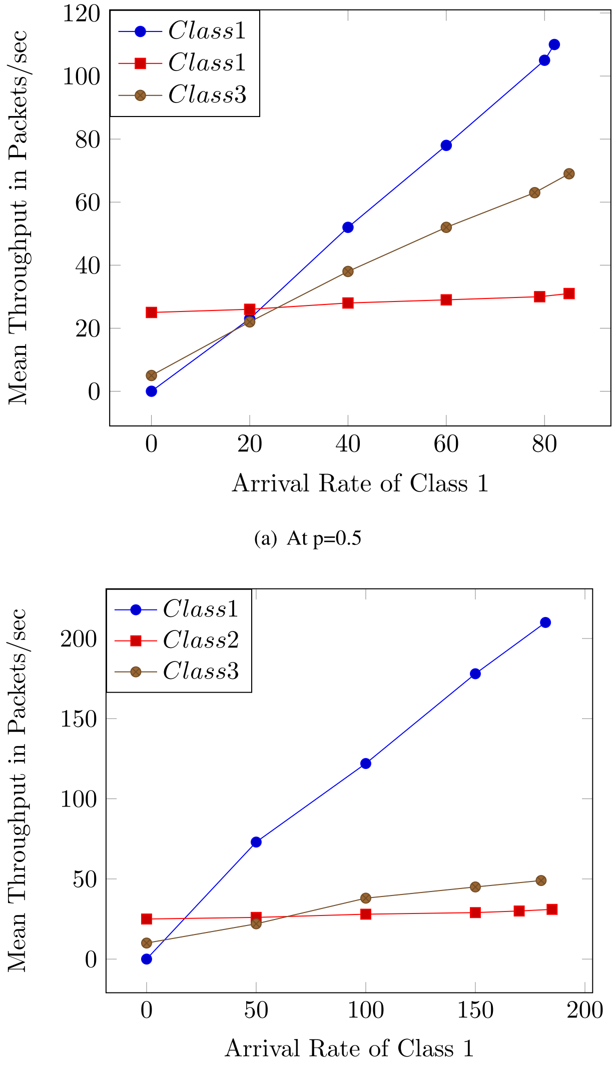
FIGURE 11. Node-wise response time.

remaining class increases, as shown in Fig 10 and 11. For example, if the arrival of a class or application say video stream at node 1 increases and its utilization reaches a maximum, the impact of this can be seen on other applications like autonomous driving, etc. provided if those application needs to be processed in the node 1. If there is no sharing of processor for different application in a given node then the impact of one on another cannot be seen. Dependency between the application can be decided by the transition probability matrix.

C. APPLICATION OF THE MODEL TO THE NETWORK DIMENSIONING AND MANAGEMENT

In this section, we discuss some applications of our model to network dimensioning and management. Suppose that a Service provider (SP) collects statistics on flow delay and flow arrivals using tools such as Cisco NetFlow. With this sole information, the SP can compute the moments of the total rate. This way, the SP would have more detailed information than that provided by Simple Network Management (SNMP) (one of the problems of SNMP is that it does not capture traffic variation at short time scales). The information on flows can be collected on the node or the class at a node we want to monitor. It can also be collected at the edges of the backbone. Combined with the routing information in the edge routers, this will give information on flows on each class in the backbone. Our model can then be used to compute the traffic on the node of the backbone, by only monitoring the edges.

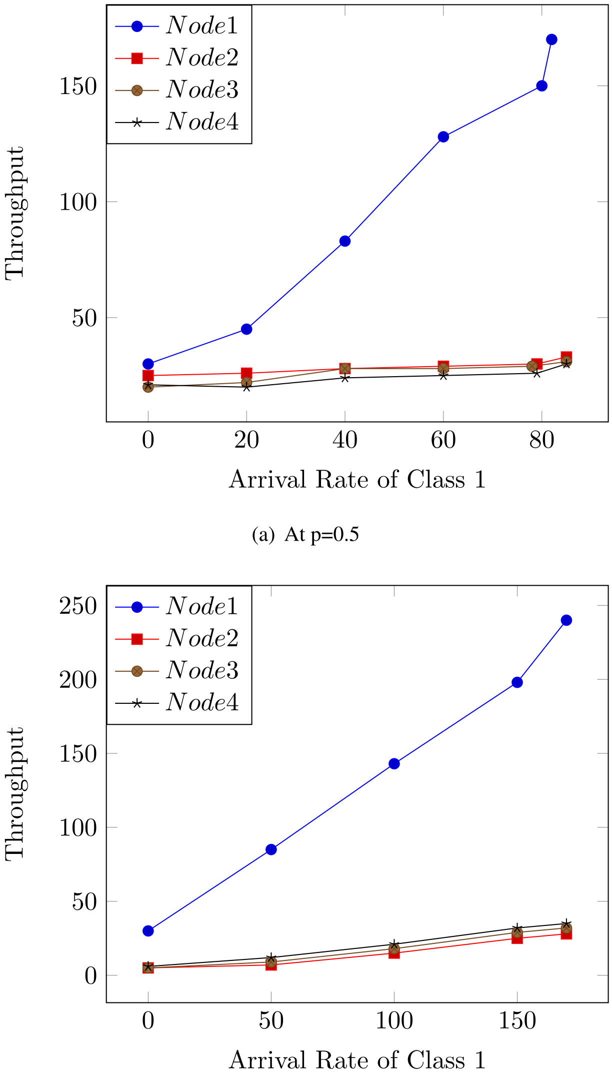
The detailed information provided by our model on the traffic helps to dimension backbone nodes. Given the characteristics of flows composing the traffic, the nodes of the backbone network can be dimensioned to avoid congestion. Note that for highly variable traffic, dimensioning the nodes



(a) At p=0.5

(b) At $p = 0.9$

FIGURE 12. Class-wise throughput.



(a) At p=0.5

(b) At $p = 0.9$

FIGURE 13. Node-wise throughput.

of the backbone based only on the average utilization is not enough to avoid congestion. Rate variation also needs to be considered.

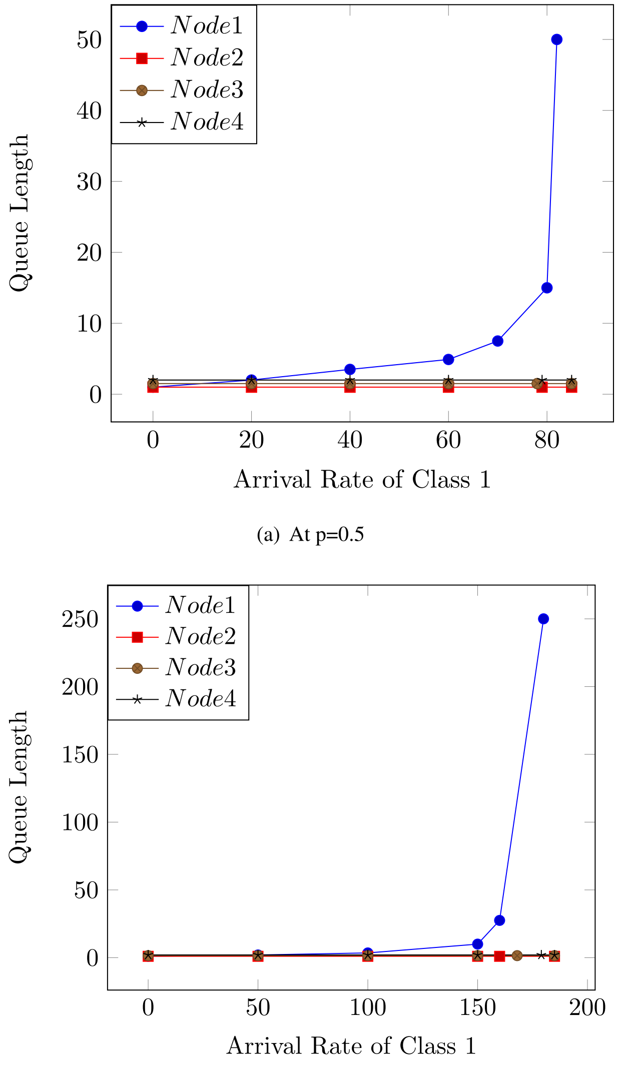
Computing the traffic in the backbone using the information on flows is not the only application of our model to network dimensioning and management. A fundamental problem the operator faces is the planning and up-gradation of the backbone nodes or links in order to maintain the absence of congestion. One may ask the following questions:

i) What is the impact on the node/class utilization caused by a change in the distribution of policies, for example, to the arrival of a new application where policies are not deployed?

ii) What is the impact on the node/class utilization caused by a change in policies/slices, for example, to an increase in the number of users in the congested access networks?

We illustrate this application by the following two examples. The first example shows the impact of a change in the flow arrival rate λ on the traffic, and hence on the dimensioning of the core network. The second example shown is the impact of the availability of the policies in the nodes.

When the arrival rate of flows increases, the bandwidth of the backbone links requirement and nodes with high processing has to be increased as well. Similarly, load balancing should be used to reduce the burden on a given node, which can be done quickly using SDN. The availability of the policies reduces the number of messages exchanged between the node and a controller, which means high priority class packets are reduced, and application packets are propagated quickly out of the network, which gives high utilization. If policies are not there, then packets need to wait till the policies are deployed which in turn gives more delay. It is because we are considering In-band OpenFlow architecture, in which packet needs to travel through other nodes to reach the controller, and while deploying the policies also it should follow the same path in the reverse direction.



(a) At p=0.5

(b) At $p = 0.9$

FIGURE 14. Mean queue length.

V. CONCLUSION

In this paper, we have proposed an analytical model to determine the type of queue in an In-band Open-Flow architecture for uncongested backbone link, and we have extended this work to a multiclass network system using multiclass queuing theory. The proposed queuing system assumes that the flows are heterogeneous: requests come from a variety of sources, from customers to OpenFlow switches and from OpenFlow switches to the controller, with different intensity. The objective is to analyze non-ergodic traffic patterns and load-dependent network architecture like mobile networks under network slicing. The applicability of the proposed model is determined by the parameters such as the average waiting time for multiclass equipment both OpenFlow switch and controller, average queue length on controller and Open-Flow switch, and switch processor utilization for each slice. For each traffic class, routing probabilities were randomly defined so that a particular choice of distribution does not bias the simulation. Given the routing matrix, the statistics of the network provides the knowledge of valuable resource consumptions for every class at every node. The performance of every slice can be evaluated using this method, which helps the management of the resources efficiently among all the slices. The experimental results shows that it is advisable to keep the policies in advance in the switches before the flow arrives at the system. We will be using a scheduling algorithm that reduces the response time of the high priority class packets in our future work.

REFERENCES

[1] A. Asava. (Apr. 2015). *OTT Is Not Just For Consumers! How Businesses Can Benefit*. [Online]. Available: https://www.kandy.io/blog/ott-not-just-consumers-how-businesses-can-benefit

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008, doi: 10.1145/1355734.1355746.

[3] F. Malik. (Jan. 2018). *How to Configure 5G Wireless Technology and Future Mobile Communication?* [Online]. Available: https://www.loginworks.com/blogs/configure-5g-wireless-technology-future-mobile-communication/

[4] O. Sallent, J. Pérez-Romero, R. Ferrús, and R. Agustí, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 166–174, Oct. 2017.

[5] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.

[6] A. Ludwig, S. Dudycz, M. Rost, and S. Schmid, "Transiently secure network updates," *SIGMETRICS Perform. Eval. Rev.*, vol. 44, no. 1, pp. 273–284, Jun. 2016, doi: 10.1145/2964791.2901476.

[7] Z. Bozakov and A. Rizk, "Taming sdn controllers in heterogeneous hardware environments," in *Proc. 2nd Eur. Workshop Softw. Defined Netw.*, Oct. 2013, pp. 50–55.

[8] J. Naous, D. Erickson, G. A. Covington, G. Appenzeller, and N. McKeown, "Implementing an openflow switch on the netfpga platform," in *Proc. 4th ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS)*, New York, NY, USA, 2008, pp. 1–9, doi: 10.1145/1477942.1477944.

[9] A. Lazaris, D. Tahara, X. Huang, E. Li, A. Voellmy, Y. R. Yang, and M. Yu, "Tango: Simplifying SDN control with automatic switch property inference, abstraction, and optimization," in *Proc. 10th ACM Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, New York, NY, USA, 2014, pp. 199–212, doi: 10.1145/2674005.2675011.

[10] N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "CacheFlow: Dependency-aware rule-caching for software-defined networks," in *Proc. ACM Symp. SDN Res. (SOSR)*, New York, NY, USA, 2016, pp. 6-1–6-12, doi: 10.1145/2890955.2890969.

[11] A. Khan and N. Dave, "Enabling hardware exploration in software-defined networking: A flexible, portable openflow switch," in *Proc. IEEE 21st Annu. Int. Symp. Field-Program. Custom Comput. Mach. (FCCM)*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 145–148, doi: 10.1109/FCCM.2013.15.

[12] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Flowvisor: A network virtualization layer," *OpenFlow Switch Consortium*, vol. 1, p. 132, Oct. 2009.

[13] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and performance evaluation of an openflow architecture," in *Proc. 23rd Int. Teletraffic Congr. (ITC)*, 2011, pp. 1–7. [Online]. Available: http://dl.acm.org/citation.cfm?id=2043468.2043470

[14] K. Mahmood, A. Chilwan, O. Østerbø, and M. Jarschel, "Modelling of openflow-based software-defined networks: The multiple node case," *IET Netw.*, vol. 4, no. 5, pp. 278–284, 2015.

[15] K. Sood, S. Yu, and Y. Xiang, "Performance analysis of software-defined network switch using *M/Geo/1* model," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2522–2525, Dec. 2016.

[16] W. Stallings, "Software-defined networks and OpenFlow," *Internet Protocol J.*, vol. 16, no. 1, pp. 2–14, 2013.

[17] *OpenFlow Switch Specification Version 1.3.1*, Open Netw. Found., Menlo Park, CA, USA, Jan. 2012.

[18] S. B. Fred, T. Bonald, A. Proutiere, G. Régnié, and J. W. Roberts, "Statistical bandwidth sharing: A study of congestion at flow level," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 111–122, Aug. 2001, doi: 10.1145/964723.383068.

[19] R. Sommer and A. Feldmann, "NetFlow: Information loss or win?" in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas. (IMW)*, New York, NY, USA, 2002, pp. 173–174, doi: 10.1145/637201.637226.

[20] U. N. Bhat, *An Introduction to Queueing Theory: Modeling and Analysis in Applications*, 2nd ed. Basel, Switzerland: Birkhäuser, 2015.

[21] S. Balsamo and A. Marin, "Queueing networks," in *Formal Methods for Performance Evaluation*. Berlin, Germany: Springer, 2007, pp. 34–82, doi: 10.1007/978-3-540-72522-0_2.

[22] S. Ata, M. Murata, and H. Miyahara, "Analysis of network traffic and its application to design of high-speed routers," *IEICE Trans. Inf. Syst.*, vol. E83-D, no. 5, pp. 988–995, May 2000.

[23] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, "On the nonstationarity of Internet traffic," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 29, no. 1, pp. 102–112, Jun. 2001, doi: 10.1145/384268.378440.

[24] L. Kleinrock, *Communication Nets; Stochastic Message Flow and Delay*. New York, NY, USA: Dover, 1972.

[25] S. Kamath, S. Singh, and M. S. Kumar, "Performance analysis of out-band openflow switch architecture: The single node case," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS) (IEEE ANTS)*, Indore, India, Dec. 2018, pp. 1–6.

[26] D. Bertsimas and D. Nakazato, "The distributional little's law and its applications," *Oper. Res.*, vol. 43, no. 2, pp. 298–310, Apr. 1995, doi: 10.1287/opre.43.2.298.

[27] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik, *Quantitative System Performance—Computer System Analysis Using Queueing Network Models*. Upper Saddle River, NJ, USA: Prentice-Hall, 1984.

[28] S. Balsamo and A. Marin, "Queueing networks," in *Proc. Int. School Formal Methods Design Comput., Commun. Softw. Syst.* Berlin, Germany: Springer, 2007, pp. 34–82.

[29] X. Chao and M. Pinedo, "Networks of queues with batch services, signals and product form solutions," *Oper. Res. Lett.*, vol. 17, no. 5, pp. 237–242, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016763779500026G

[30] R. Sherwood. (Jan. 2013). *Cbench (Controller Benchmark)*. Available: https://github.com/mininet/oflops/tree/master/cbench

[31] USAT. *What are the Main MTU and MSS Design Considerations?* [Online]. Available: http://usatcorp.com/faqs/main-mtumss-design-considerations/

[32] F. Wamser, R. Pries, D. Staehle, K. Heck, and P. Tran-Gia, "Traffic characterization of a residential wireless Internet access," *Telecommun. Syst.*, vol. 48, nos. 1–2, pp. 5–17, Oct. 2011.

SANTHOSHA KAMATH received the B.E. degree from Vishweshwarayya University Belgaum, Karnataka, India, in 2004, and the M.Tech. degree in network engineering from Manipal University, in 2009. He is currently pursuing the Ph.D. degree in the area of software defined networking with the Manipal Institute of Technology, MAHE, Manipal, India. His research interests include performance analysis and enhancement of the mobile core networks for OTT application using SDN.

SANJAY SINGH (M'05–SM'13) received the bachelor's degree from the Institution of Electronics and Telecommunications Engineers, New Delhi, India, in 2001, and the M.Tech. and Ph.D. degrees from the Manipal Institute of Technology, Manipal, India, in 2003 and 2010, respectively.

In 2004, he joined the Department of Information and Communication Technology, Manipal Institute of Technology, MAHE, where he is currently a Professor. He is also heading the Centre for Artificial and Machine Intelligence (CAMI), Manipal Institute of Technology. His work spans various areas, including wireless communications, artificial intelligence, machine learning, neural networks, fuzzy logic, and information retrieval. He is a member of ACM.

M. SATHISH KUMAR received the bachelor's degree in electronics and communication engineering from the Institution of Engineers, India, the master's degree in engineering specializing in optical communication from the College of Engineering, Anna University, Guindy, and the Ph.D. degree from the National Institute of Technology Karnataka (NITK) Surathkal for his thesis on Performance Enhancement of Optical CDMA Networks and Hybrid Multiplexed Optical Communication Systems.

He started his academic career as a Lecturer in electronics and communication engineering at Dr. T. M. A. Pai Polytechnic, Manipal, in 1992. In 1998, he joined the Electronics and Communication Engineering Department, MIT, Manipal, as a Lecturer. He continued till 2008 as a Reader in the Electronics and Communication Engineering Department, MIT. From 2008 to 2010, he was a Research Assistant Professor with the School of Electrical Engineering and Computer Science, Seoul National University, South Korea. He joined back MIT, Manipal, in 2010, as a Professor at the Electronics and Communication Engineering Department. He is currently heading the Electronics and Communication Engineering Department. He works in the area of wireless communications, optical communications, and nanophotonics. He was a recipient of the International Scholar Exchange Fellowship of the Korea Foundation for Advanced Studies, from 2006 to 2007.