IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 19, 2020, accepted December 28, 2020, date of publication January 8, 2021, date of current version January 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049945

Modeling and Evaluation of Software Defined Networking Based 5G Core Network Architecture

ABDULAZIZ ABDULGHAFFAR, ASHRAF MAHMOUD, (Member, IEEE), MARWAN ABU-AMARA, AND TAREK SHELTAMI

Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Abdulaziz Abdulghaffar (g201703470@kfupm.edu.sa)

This work was supported by the King Fahd University of Petroleum and Minerals (KFUPM) and the Department of Computer Engineering.

ABSTRACT Nowadays, the increase in the number of mobile users and cellular traffic leads to new challenges in the fifth-generation (5G) of cellular networks. The increase in the demand for high data rates brings challenges like scalability and flexibility in the 5G network. Software-defined networking (SDN) is a network paradigm that separates the control plane and data plane in the network and ease the management of the network. In this work, an SDN based 5G core architecture is proposed, in order to introduce flexibility and ease of management in the network. Another benefit of using SDN is to make the network vendor-independent. Furthermore, the explanation of initial attachment and handover procedures in the proposed architecture is provided. A network simulator is built to evaluate the performance of proposed architecture, in terms of end-to-end delay, throughput and resource utilization of controller, under different network factors. A performance comparison, in terms of end-to-end delay, between proposed SDN based 5G architecture and traditional 5G architecture is provided. Results show that the proposed architecture provides 18% to 62% less end-to-end delay, under different factors for different procedures, compared to the traditional 5G architecture. A comparison with previous works is also provided, which indicates similar trends in delay between our work and previous studies.

INDEX TERMS Fifth generation cellular network, 5G, software-defined network (SDN), OpenFlow, simulation, end-to-end delay, throughput, utilization.

I. INTRODUCTION

The fifth-generation (5G) standard of cellular communications is the successor of the fourth-generation (4G) cellular network. 4G provides data rates of 100 Mbps and 1 Gbps for mobile and stationary users, respectively. However, the demand for data rates is increasing rapidly because of multimedia applications [1] and the massive increase in the number of devices. According to Cisco, mobile data traffic in 2022 will increase seven times the amount of traffic in 2017, i.e., an increase from 11.5 Exabytes per month in 2017 to 77.5 Exabytes per month by 2022 [2]. Next Generation Mobile Networks (NGMN) presented several use cases for 5G networks, for example, ultra-reliable communication, massive Internet of things [3] and lifeline communication, etc. Each of these use cases has its own expectations [4]. So the 5G architecture should be programmable in order to fulfill these expectations [5] and allow the service providers to implement new services flexibly.

Software-defined networking (SDN) is a technology that separates the control plane and data plane in a network and centralize the control plane in a single entity. This separation and centralization improve network flexibility and also help to reduce the capital expenditure (CAPEX) of the network [6]. By incorporating SDN concepts in the 5G network, the network will be more adaptable to the changing demands and the network operators will be able to fulfill the requirements of different applications [7]. The SDN controller is able to programmatically control the data plane switches using open standards, like OpenFlow [8], [9].

Several research works have been done in order to apply the advantages of SDN in the 5G mobile core network. However, most of these works implement SDN on LTE architecture and extend it to propose a 5G architecture, instead of using SDN on 3GPP (3rd Generation Partnership Project) 5G architecture [10], which is shown in Fig. 1. Moreover, most of the studies do not provide a detailed performance

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

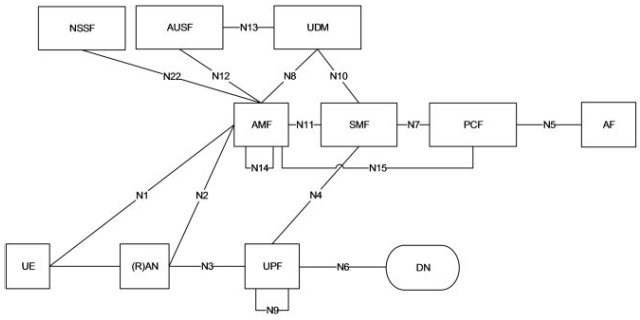


FIGURE 1. 5G Service-based architecture [10].

evaluation of their proposed SDN based 5G core network. So, this work aims to propose an SDN based architecture for 5G core network, which will be based on the standard 3GPP 5G architecture. Furthermore, two control plane procedures, including Registration (initial attachment) and Handover, are considered. The performance of the SDN based 5G architecture is evaluated based on different metrics, including end-to-end delay, throughput at the controller and the resource utilization of the controller. The performance of the proposed architecture is compared against the traditional 5G architecture in terms of end-to-end delay observed during control plane operations. Results show that the proposed architecture outperformed the traditional 5G architecture. A comparison with previous works is also provided, which indicates similar trends in delay between our work and previous studies.

The rest of this paper is organized as follows: Section 2 provides a brief background of 5G architecture and SDN, while Section 3 explains the related work present in the literature. Section 4 describes our proposed architecture and Section 5 explains the initial attachment (registration) and handover procedures in our architecture. Modeling details and implementation of the proposed architecture are presented in Section 6. Section 7 details the results of the simulation, provides a comparison with the traditional 5G architecture and explains the effect of various factors on the results. It also provides a comparison with previous works and contains details on the validation of our simulator. Lastly, Section 8 concludes the research work and provides potential future work directions.

II. BACKGROUND

A brief background on 5G architecture and software-defined networking are presented in this section, in order to assist in understanding the basic concepts of these technologies.

A. 5G ARCHITECTURE

5G service-based architecture (SBA) given in [10], is shown in Fig. 1. Network functions that are part of the control plane are Network Slice Selection Function (NSSF), Access and Mobility Management Function (AMF), Authentication Server Function (AUSF), Session Management Function (SMF), Unified Data Management (UDM), and Policy Control Function (PCF). Whereas other entities are Application Function (AF), Data Network (DN), User Plane Function (UPF), (Radio) Access Network ((R)AN), which is also known as Next Generation Radio Access Network (NG-RAN), and User Equipment (UE).

Each entity provides different functions. AMF provides functions like access control, registration, and mobility management. Session management functions, like establishing, modifying, and releasing sessions, are provided by SMF. Along with that, SMF also provides the support of IP address allocation to the UE and configures routing decisions at UPF to facilitate traffic routing to proper destinations. UPF is similar to the gateways (SGW and PGW) in the LTE network, and it provides the functionality of packet inspection, routing and forwarding. It can also support the quality of service (QoS) requirements for the user plane. Another important functionality of UPF is that it connects to the data network, and the UPF connected to the data network is known as Protocol Data Unit (PDU) Session Anchor [11]. PCF provides a policy framework to direct network behavior, while UDM provides user identification functionality and generates authentication credentials. UDM also supports SMS and subscription management. AUSF supports authentication functions, whereas NSSF selects the network slice that will serve the UE. NSSF is generally a part of AMF. More details on these network functions are available in the 3GPP standard and literature [10], [12].

B. SOFTWARE-DEFINED NETWORK

Software-defined networking (SDN) is a paradigm that enables the separation of control and data plane in a network. SDN does that by extracting the control plane functions from the forwarding devices, like switches and routers, and centralizing these functions on an SDN controller. In traditional network entities, like switches and routers, the control plane is tightly coupled within each entity, along with the data plane. The benefits of centralizing the control plane in one entity includes simplifying network management and introducing programmability in the network [13]. It also helps network administrators to easily upgrade the services provided by the network from a centralized source [14], as compared to the traditional network where each device has to be configured manually [15].

In the SDN network, the controller is responsible to decide which actions should be performed on the packets, like forwarding or dropping, and install these rules in the forwarding elements, e.g., switches. These rules are termed as flow rules and each forwarding element maintains these rules in a table known as flow table. This flow table dictates the operation of a forwarding device. SDN controller communicates with the forwarding devices on the southbound interface and the communication protocol used is known as OpenFlow protocol [8]. The OpenFlow protocol is standardized by the Open Networking Foundation (ONF) [16], and its major benefit is that it allows interoperability between devices in a multi-vendor

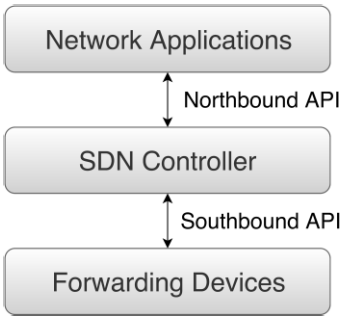


FIGURE 2. Simple SDN architecture.

network environment. On the other hand, different network applications run on the northbound interface of the controller [17]. This is depicted in Fig. 2.

III. RELATED WORK

In this section, the most relevant work is presented that implements SDN concepts in mobile core network in order to move towards the next generation or 5G network. Most of the research work present in literature tries to incorporate SDN concepts in LTE architecture and extend it to propose a new 5G architecture. Whereas some articles propose a clean slate architecture, and some integrate SDN with the standardized 5G architecture. An example of a work that provides a clean slate architecture is SoftNet [18], while examples of works that extend LTE architecture to propose a 5G architecture are presented in [19]–[24]. Finally, samples of the research work that use the standardized 5G architecture are presented in [25]–[27].

SoftNet [18] completely re-designs the core network using SDN. It consists of a core network and a unified radio access network (RAN). The access points in the RAN are connected to servers that are placed at the edge of the core network. Several control plane network functions are supported including, centralized network control and Quality of Service (QoS) control, etc. SoftNet is claimed to be adaptable, efficient, scalable, and simple in design. SoftNet has two different types of network control, decentralized and centralized. The decentralized control is used on a system level to improve scalability and flexibility, while the centralized control is used on a component level to improve efficient resource utilization. The authors provide performance evaluation to compare the signaling cost of SoftNet and LTE networks, and the results of simulations shows that the signaling overhead is reduced in SoftNet compared to the LTE network.

Sama *et al.* [19] proposed an architecture to introduce programmability and flexibility in the LTE core network using SDN. In order to achieve their desired goal, the authors implemented MME and the control plane of SGW (SGW-C) as applications on top of the SDN controller. Whereas the data plane of SGW (SGW-D) is implemented as an OpenFlow switch and forwards the traffic based on flows installed by the controller. The authors kept PGW the same as the traditional LTE core network. The radio functions of eNB are kept unchanged, while they added the support of OpenFlow protocol for data plane management. Using OpenFlow protocol, the controller periodically receives load status of switches and then performs load balancing on the switches. The authors provide analytical modeling of their proposed architecture and quantify the signaling load of the proposed architecture and the traditional LTE architecture and provide their numerical results. From the results, the authors claimed that introducing SDN in the core network reduces the signaling load in the network. Similar architecture is used by Pagé and Dricot in [23], and from qualitative analysis the authors claimed that the end-to-end delay is reduced by using lightweight OpenFlow messages. However, the work did not provide any analytical model or implementation and performance evaluation of the proposed solution.

The study in [20], [21] by Nguyen and Kim, extends the work in [19] by separating the control plane and data plane of PGW and moving the control plane (PGW-C) on top of the SDN controller, named as the mobile controller (MC). Whereas the rest of the architecture is reused from [19]. However, the single controller does not solve the problem of scalability in the network. The analytical modeling of this work is provided by the authors, where they investigate the signaling load of three architectures, including traditional LTE architecture, architecture proposed by Sama [19] and OEPC architecture proposed by the authors. They also carried out experiments to find the numerical results of the provided analytical model and concluded from the results that their proposed architecture reduces the signaling load compared to the traditional LTE architecture and the architecture proposed by Sama [19]. Similarly, the research work done by Jain *et al.* [22] uses the architecture that is presented in [21]. The main goal of the authors is to compare the performance of the core network with SDN and NFV in terms of throughput and latency. From their results, the authors claimed that the SDN based core network is suitable for data plane traffic, while it is not suitable for signaling traffic.

The authors in [24] utilize LTE network entities to propose a new cloud based 5G core architecture. The architecture contains two level of cloud network hierarchy, namely the edge cloud and the central cloud. As the name suggest, the edge cloud is closer to the user (UE) and it provides latency sensitive services to the UE. Whereas the central cloud manages the service and network resources in a centralized manner using SDN and NFV technologies. The authors provide an analytical model that selects the appropriate cloud with the aim to maximize resource and network utilization while minimizing resource cost and network load. The numerical results show that there is a tradeoff between resource cost and network cost when the number of deployed clouds is increased in the network. However, the authors did not provide any simulation or prototype-based performance evaluation of their architecture. Tadros *et al.* [28] study different control plane architectures in 5G networks and compare them in terms of throughput and latency. The three architectures presented in

this work are centralized, distributed, and locally centralized-physically distributed (LC-PD) control plane architectures. These architectures are implemented in Mininet-WiFi emulator [29] and from the results the authors claimed that LC-PD architecture provide lower latency and higher throughput compared to other two architectures. However, the proposed architecture is not based on the standard 3GPP 5G architecture.

The study in [26] by Eichhorn *et al.* proposed an SDN based architecture for a sliced 5G core. Network slice is a customized logical network on top of physical architecture that provides specific network features and capabilities [30]. The proposed architecture decouples the control plane from data plane components. The authors integrate the SDN controller between SMF and UPF, where the UPF is implemented as an SDN switch. The controller is connected to the SMF on its northbound interface and the UPF (the SDN switch) is connected to its southbound interface. Thus, decoupling the control plane from the data plane. The proposed architecture is based on 3GPP's 5G service-based architecture. However, the authors did not provide any analytical model or performance evaluation of their architecture.

Nayak *et al.* [27] proposed a centralized SDN architecture in 3GPP's 5G architecture. The control plane functions of RAN are moved to the core network, which made the gNB a simple data plane entity. All control plane functions are controlled by a single control plane entity in the core network. The control plane functions of RAN are merged with AMF in the core network, the new AMF is known as evolved AMF (eAMF). The authors provide the signaling costs of registration and handover procedures in 5G and the proposed architecture. It has been claimed that the proposed architecture helps in the reduction of signals between the RAN and the core network. The authors also implemented both architectures in the NS-3 simulator [31] and showed that the proposed architecture outperformed 5G architecture.

Most of the work done in the literature uses the LTE architecture in order to propose a 5G architecture based on SDN. Furthermore, most of the previous research works have not provided an extensive performance evaluation of the work done. So, in this work, 3GPP's standard 5G core architecture is used to propose a new 5G core architecture based on SDN and provide the performance evaluation of the architecture in terms of end-to-end delay, throughput at the controller and resource utilization.

IV. PROPOSED ARCHITECTURE

A new 5G core architecture is proposed based on the SDN concepts. The proposed architecture is shown in Fig. 3.

In this architecture, all the control plane network functions, like AMF, SMF etc., are moved on top of the SDN controller. All these network functions are realized as applications and they communicate with the SDN controller on its northbound interface. Whereas the data plane of the 5G network, i.e., the UPF, is implemented as an SDN switch. The UPF communicates with the controller on its southbound interface

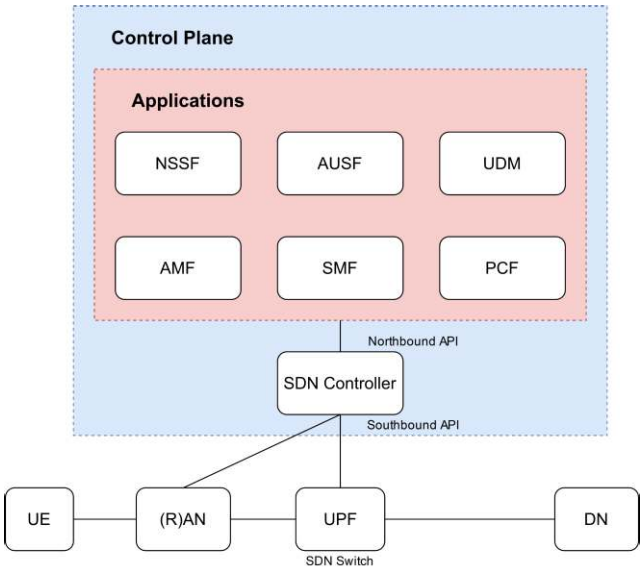


FIGURE 3. The proposed architecture.

using the OpenFlow protocol. The access network is kept unchanged, however, in order to make the gNBs OpenFlow compatible, it is assumed that each gNB is connected with an SDN switch that can communicate with the SDN controller. The SDN controller is responsible to manage the data plane and to install the flow rules in the data plane switches, including UPF and gNB's switches. The benefit of using this architecture is that it makes the user plane easy to manage and flexible as it is managed by a centralized controller [26]. On the other hand, centralizing the control plane operations in one entity leads to an increase in the resources required at the controller, and decreasing the resources needed at the data plane nodes. However, the controller is generally located in the data centers where the resources are not a major concern. Another drawback of using a single controller is the issue of scalability in the network. However, for the purpose of this study, we are dealing with one logical controller and we are not tackling the issue of scalability. More implementation-related details for the proposed architecture are presented in section 6.

Incorporating the changes mentioned above in 5G architecture will lead to a change in the sequence of messages exchanged during different control plane procedures. In this work, we will focus on registration procedure and Xn based intra and inter UPF handover procedures. Xn is an interface between different RAN (NG-RAN) nodes [32]. In the subsequent sections, we will explain these procedures and provide their call flows.

V. PROCEDURES IN PROPOSED ARCHITECTURE

A. INITIAL ATTACHMENT (REGISTRATION) PROCEDURE

The initial attachment procedure in our proposed architecture is shown in Fig. 5. Initially, the UE sends a registration request to the (R)AN, which is forwarded to the appropriate AMF

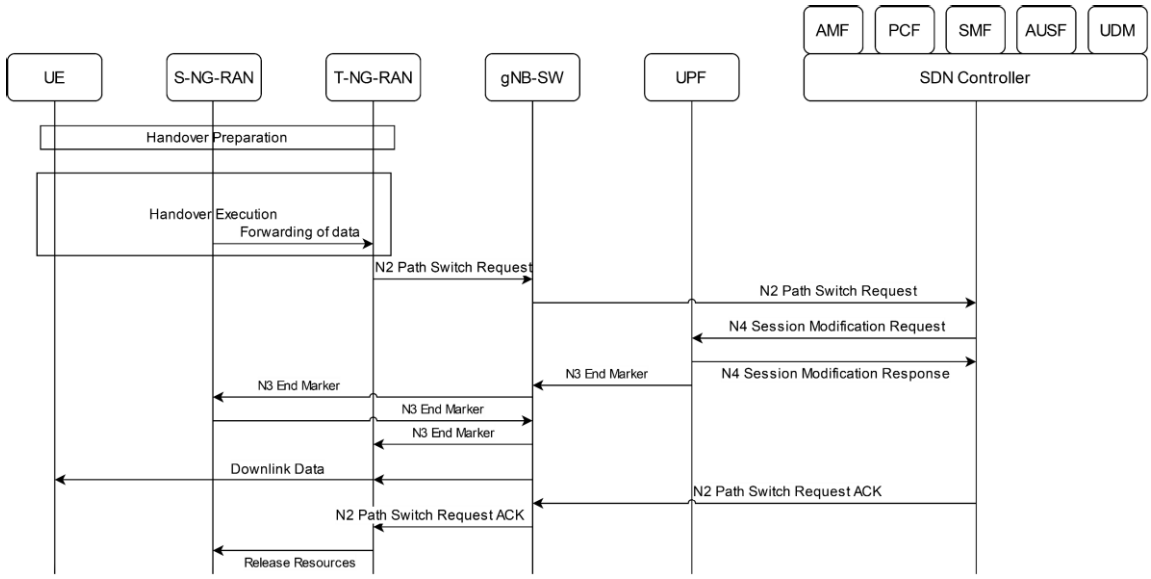


FIGURE 4. Xn based handover without UPF re-allocation in proposed architecture.

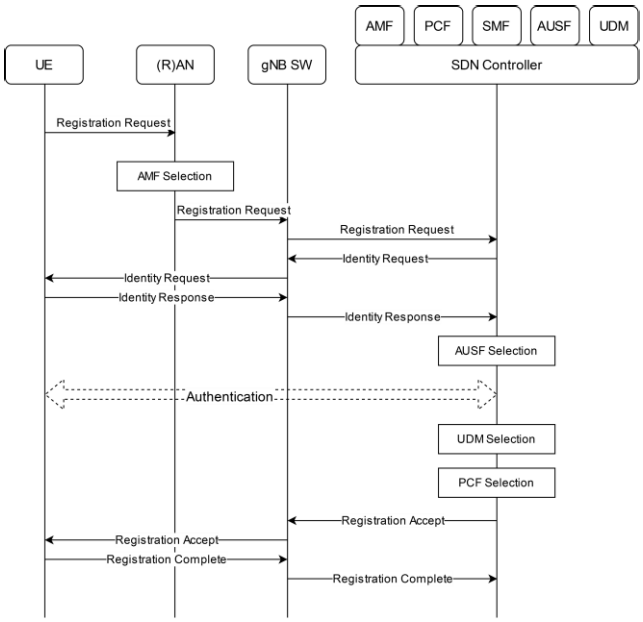


FIGURE 5. Initial Attachment procedure in the proposed architecture.

after its selection from the (R)AN. In this case the AMF is running as an application on the controller, so (R)AN sends the registration request to the controller which is then forwarded to the AMF application. If the SUbscription Permanent Identifier (SUPI) is not provided by the UE, the AMF application then sends an Identity request message to UE to retrieve its SUbscription Concealed Identifier (SUCI). The UE then replies with Identity response which includes the SUCI. The AMF will select AUSF which will initiate UE authentication. After that, the AMF selects the UDM and PCF of the UE. Once the registration procedure is completed, the AMF assigns a Globally Unique Temporary Identity (5G-GUTI) to the UE and sends it to the UE through registration accept message. The UE then acknowledges the reception of 5G-GUTI using Registration complete message. It is important to mention that the messages exchanged between the (R)AN and the SDN controller are embedded in OpenFlow packets. Also, the messages that will be exchanged between different control plane entities (i.e., the applications on SDN controller) will be handled within the SDN controller and are not shown in the Fig. 5.

B. HANDOVER PROCEDURE

1) Xn BASED HANDOVER WITHOUT UPF RE-ALLOCATION (INTRA-UPF)

The handover procedure without UPF re-allocation is shown in Fig. 4. In this procedure the target and source gNBs are attached to a common UPF, i.e., the UPF is shared by the gNBs. Initially, the Target Next Generation Radio Access Network (T-NG-RAN) will send a path switch request to the AMF (application on SDN controller) indicating that the UE has moved to a new target cell and a handover is required. It will also include the information regarding Packet Data Unit (PDU) sessions that are needed to be switched. After that, SMF will send a session modification request to UPF in order to switch the PDU sessions requested by the T-NG-RAN. Once the PDU sessions are switched, the UPF will respond with the session modification response message. After switching the path, the UPF will send an end marker to the Source NG-RAN (S-NG-RAN) notifying it about the change in the path and then UPF will start sending downlink traffic to T-NG-RAN. Once this process is done, the AMF will send path switch request acknowledgment (ACK) to the T-NG-RAN, which in turn will send a message to S-NG-RAN

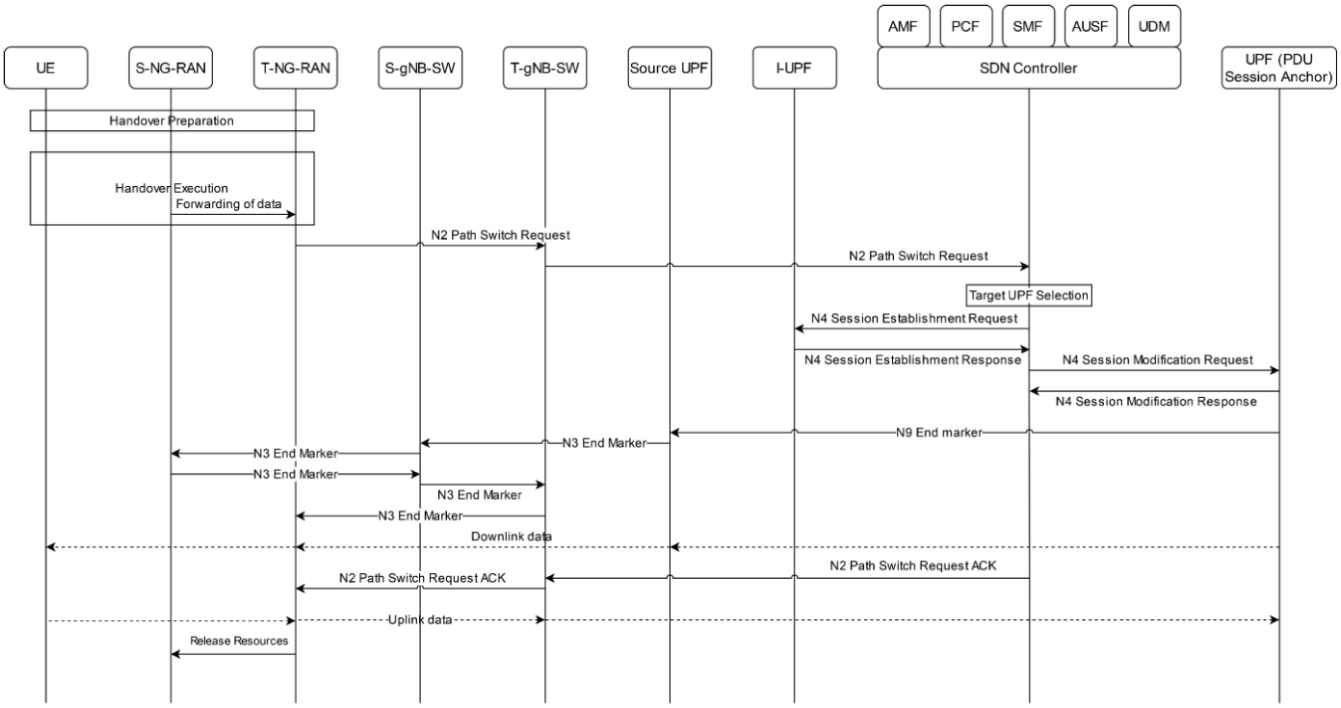


FIGURE 6. Xn based handover with UPF re-allocation in proposed architecture.

to release its resources and confirming the completion of the handover procedure.

2) Xn BASED HANDOVER WITH UPF RE-ALLOCATION (INTER-UPF)

The handover procedure when the UPF will be re-allocated is shown in Fig. 6. Unlike the previous procedure, in this scenario both the target and source gNBs are connected to different UPFs. Similar to the previous handover procedure, the T-NG-RAN will send path switch request to the AMF. However, in this case, the UE has moved out of the serving area of the UPF connected to the S-NG-RAN, so the SMF has to select a target UPF (Intermediate-UPF) connected to the T-NG-RAN and then sending a session establishment request to the selected UPF (I-UPF). I-UPF then responds with a session establishment response message. After that, SMF will exchange session modification messages with the PDU session anchor in order to switch the PDU sessions. The End marker will be sent by the PDU session anchor on the old path indicating that a new downlink is established, and the downlink traffic will be forwarded on the new path. The AMF will then send a path switch request ACK to the T-NG-RAN, similar to the previous handover procedure. T-NG-RAN will inform S-NG-RAN to release its resources once the handover procedure is complete.

VI. MODELING AND IMPLEMENTATION OF THE PROPOSED ARCHITECTURE

This section provides a detailed explanation of implementing the proposed architecture in the network simulator.

The visual implementation of the proposed architecture is shown in Fig. 7. The radio access network consists of 5G base stations (gNBs) that are connected to the UEs using radio links. As mentioned in section 4 that the base stations are connected with OpenFlow enabled switches that are termed as gNB-switch and they enable the base stations to communicate with the SDN controller using OpenFlow protocol. These gNB-switches are deployed in the edge network to benefit from traffic offloading and enforce QoS [33], [34]. The core network consists of the SDN controller and several UPFs which facilitate UE traffic routing between the radio access network and data network. The UPF connected to the data network (DN) is known as the PDU session anchor. The SDN controller is responsible to control all data plane switches, including UPFs and gNB-switches.

In order to evaluate the performance of SDN based 5G core network, we developed a Python [35] based discrete event simulator. First, we develop an event graph of our network and then translate it into a simulation model. An event graph represents a discrete event system visually and provides an easy way to translate it into a simulation model [36]. The event graph shows the events that occur in the network and how different events can take place based on several conditions. A simplified event graph of the proposed architecture is shown in Fig. 8. The event graph consists of three network elements, namely the base station, OpenFlow (OF) switch and the SDN controller. At the start, all state variables of queues (Q) and servers (S) are initialized to zero (0). Each entity is modeled as an M/M/1 node with a single server, single queue, where the requests arrive at the base station

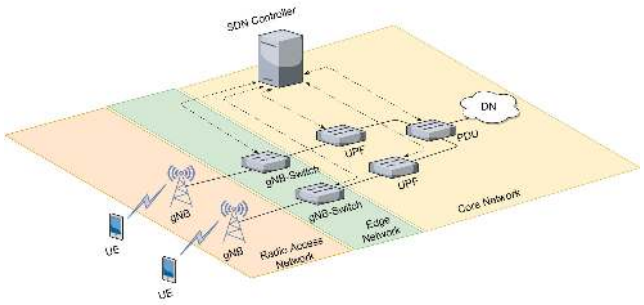


FIGURE 7. Implementation of the proposed network architecture.

based on Poisson distribution and the service times of the nodes follow an exponential distribution. The arrival event on each node will increment the queue state variable of the respective node. If the server of that node is free (indicated by 0), the service will start, and the state variable of the queue will be decremented, and the server will be set to busy (represented by 1). After some service time, the departure event will be scheduled, and the request will leave the node. After departure, if the queue of that node contains other requests, the service will start for the request at the head of the queue (based on FIFO). Similar events will occur in each node. However, the arrival at the base station is triggered based on different arrival rates from the UE or upon receiving a response from the switch. In the case of OpenFlow switch, the requests arrive from both, the base station and the SDN controller. Whereas, in the SDN controller the requests only arrive from the datapath switches.

A. SIMULATOR VALIDATION

We validate our simulation model using the Mininet emulator [37] and the analytical model, and the details of each are presented in the following subsections.

1) MININET EMULATION

We first validate our simulator with mininet emulation, we set up a network environment shown in Fig. 7. In this network, Open vSwitch [38] provides the implementation of all data plane switches whereas the Ryu controller [39] is used as the SDN controller. Ryu is a python based SDN controller and all network applications running on top of the controller are written in python programming language. The requests arrive at one of the gNB's based on several arrival rates, and then each request follows a specific route in the network based on their type as shown in the call flows in section 5. We calculated the end-to-end delay for each type of request against various request arrival rates and then compared the results from our simulator with a mininet emulated network. The results obtained are discussed in the next section.

2) ANALYTICAL MODEL BASED ON JACKSON NETWORK

In order to verify our simulator, we refer to the model presented in [40]. The system is modeled using Jackson network [41] so the following assumptions are made. The first

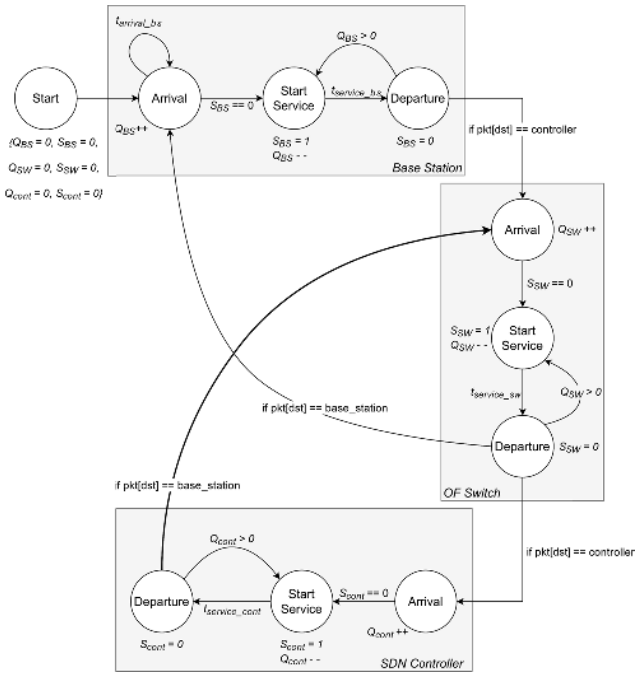


FIGURE 8. Event graph for the proposed architecture.

assumption is that the network has a steady-state, and the evaluation will be independent of time. The second assumption is that the arrivals and the service times at different network nodes are independent from each other. While, the arrival and service processes are considered as Poisson processes, which is the third assumption. This let us model each network node, including the base station, OpenFlow switch and SDN controller, as M/M/1 queuing system and to find the total arrival rate at each node separately.

In the system shown in Fig. 9, there is only one base station, one controller and one switch. The initial arrival rate at the base station is λ. The probability that a packet arrived at the base station will be forwarded to the switch is q_{bs}. The probability that the switch will forward the packet to the controller is given by q_{sw}, while the probability that the controller will send the traffic back to the switch is q_c. Where the service rates of the base station, the switch and the controller are represented by μ_{bs}, μ_{sw}, and μ_c, respectively.

First, we will find the analytical model for registration requests. The total arrival rate at the base station is given as:

$$\begin{aligned} \tau_{bs} &= \textit{initial arrival} + \textit{arrival from switch} \\ \tau_{bs} &= \lambda + ((1 - q_{sw}) \times \lambda_{sw})) \end{aligned} \tag{1}$$

The total arrival intensity at the switch is given as

$$\begin{aligned} \tau_{sw} &= \textit{arrival from BS} + \textit{arrival from conroller} \\ \tau_{sw} &= q_{bs} \times (\lambda + ((1 - q_{sw}) \times \lambda_{sw})) + q_c \times (q_{sw} \times (\lambda_{sw})) \end{aligned} \tag{2}$$

While the total arrival intensity at the controller is given as

$$\tau_c = \textit{arrival from switch}$$

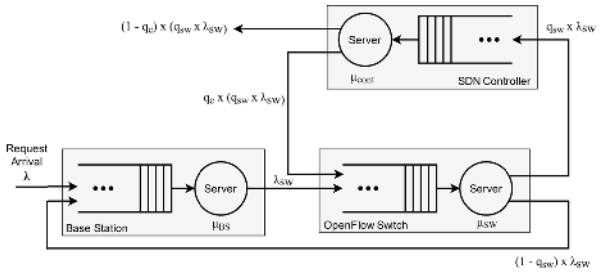


FIGURE 9. Queueing model of proposed architecture.

$$\tau_c = q_{sw} \times \lambda_{sw} \tag{3}$$

In order to get the probabilities, we refer to the call flow for a registration request, as shown in Fig. 5. The probabilities are,

$$q_{sw} = \frac{3}{5} = 0.6; \quad q_c = \frac{2}{3} = 0.6667; \; q_{bs} = 1$$

where, the arrival at the switch can be found from the call flow of registration procedure in Fig. 5, so λ_{sw} is,

$$\lambda_{sw} = 5\lambda$$

Putting these values in (1), (2), and (3) we get,

$$\tau_{bs} = \lambda + ((1 - 0.6) \times 5\lambda)) = 3\lambda \tag{4}$$

$$\begin{aligned}\tau_{sw} &= 1 \times (\lambda + ((1 - 0.6) \times 5\lambda)) + 0.6667 \times (0.6 \times (5\lambda)) \\ &= 3\lambda + 2\lambda = 5\lambda \end{aligned} \tag{5}$$

$$\tau_c = 0.6 \times 5\lambda = 3\lambda \tag{6}$$

By referring to (4), (5), and (6) and Fig. 5, it can be noticed that the total arrival intensity at each node is equal to the number of messages it receives during a single registration request, multiplied by the initial arrival rate at the base station.

Since each node is an M/M/1 queue locally, the load for each node is given by,

$$\rho = \frac{\tau}{\mu}$$

In order to find the total time for a packet (delay) in the network for a registration request, we use Little's formula [42].

$$E\,[D_{TOT}] = \frac{1}{\lambda}\left(\frac{\rho_{bs}}{1 - \rho_{bs}} + \frac{\rho_{sw}}{1 - \rho_{sw}} + \frac{\rho_c}{1 - \rho_c}\right) \tag{7}$$

For intra-UPF handover requests, we can simply get the total arrival intensities on all the nodes from the Fig. 4, and the values are:

$$\begin{aligned} \tau_{t\_bs} &= 3\lambda \quad \tau_{upf} = \lambda \\ \tau_{t\_sw} &= 4\lambda \quad \tau_c = 3\lambda \\ \tau_{s\_bs} &= \lambda \end{aligned}$$

where τ_{t_bs} is the arrival intensity at the target base station (or g-NB) and τ_{s_bs} is the arrival intensity at the source base station. While τ_{t_sw} and τ_{upf} represent the arrival intensities at the target gNB switch and the UPF, respectively.

The total delay for a packet in the network is,

$$\begin{aligned} E\,[D_{TOT}] = \frac{1}{\lambda}\Big(&\frac{\rho_{t\_bs}}{1 - \rho_{t\_bs}} + \frac{\rho_{t\_sw}}{1 - \rho_{t\_sw}} + \frac{\rho_{s\_bs}}{1 - \rho_{s\_bs}} \\ &+ \frac{\rho_{upf}}{1 - \rho_{upf}} + \frac{\rho_c}{1 - \rho_c}\Big) \end{aligned} \tag{8}$$

Similarly, we can obtain the arrival rates for inter-UPF handover from the Fig. 6.

$$\begin{aligned} \tau_{t\_bs} &= 3\lambda \quad \tau_{s\_bs} = \lambda \;\; \tau_{pdu} = \lambda \\ \tau_{t\_sw} &= 3\lambda \quad \tau_{s\_sw} = 2\lambda \;\; \tau_c = 4\lambda \\ \tau_{t\_upf} &= \lambda \quad \tau_{s\_upf} = \lambda \end{aligned}$$

where, τ_{t_upf} and τ_{s_upf} represent the arrival rates at target and source gNB UPFs respectively, and τ_{pdu} shows the total arrivals at the PDU session anchor.

The total delay for a packet is given by,

$$\begin{aligned} E\,[D_{TOT}] = \frac{1}{\lambda}\Big(&\frac{\rho_{t\_bs}}{1 - \rho_{t\_bs}} + \frac{\rho_{t\_sw}}{1 - \rho_{t\_sw}} + \frac{\rho_{t\_upf}}{1 - \rho_{t\_upf}} \\ &+ \frac{\rho_{s\_bs}}{1 - \rho_{s\_bs}} + \frac{\rho_{s\_sw}}{1 - \rho_{s\_sw}} + \frac{\rho_{s\_upf}}{1 - \rho_{s\_upf}} \\ &+ \frac{\rho_{pdu}}{1 - \rho_{pdu}} + \frac{\rho_c}{1 - \rho_c}\Big) \end{aligned} \tag{9}$$

VII. RESULTS AND DISCUSSION

A. SIMULATION PARAMETERS

In order to carry out simulations, we explore different factors that are important in cellular networks and find their levels. The important factors are processing times of nodes and the propagation time of packets. The subsequent paragraphs explain different factors and their levels.

First, we look at the processing times of different network nodes, including the SDN controller, OpenFlow switch and base station, reported in the literature. For the SDN controller, the processing time is less than 0.5 ms as reported in [43]. While the study done by Metter *et al.* [44] reports the processing time of 0.1 ms for the OpenDaylight controller [45] and 0.2 ms for the Ryu controller [39]. The processing time for OpenFlow switches reported in the literature is 10 μs for a hardware-based switch [46], and for a software-based switch it can go up to 140 μs as reported in [47]. Other research work like Sattar *et al.* [48] also reports the processing time of 10 μs, 25 μs and 40 μs. While the processing time of 75 μs is also reported in the literature [49]. The packet processing time on the 5G base station, also known as gNB (next-generation NodeB) [50], is reported in [51], [52] to be around 1 ms, whereas, the processing time of 0.6 ms is mentioned in [53], [54].

The next important factor in the simulation is the propagation delay in the network. The distance between the radio access network and the core network can go up to hundreds of kilometers [55]. The transmission delay in a 200 km optical fiber is nearly 1 ms [56]. While for a link length of 100 km and 20 km, the transmission delays are 0.5 ms and 0.1 ms respectively [57]. In our simulation, the links between the radio access network and the edge are configured with a delay

TABLE 1. Simulation parameters and their values.

Parameters	Values	Unit
Switch Processing time [46-49]	10 µs, 25 µs, 40 µs, 75 µs, 140 µs	µsec/requests
Base station Processing time [51-54]	1 ms, 0.6 ms	msec/requests
SDN Controller processing [43, 44]	0.1 ms, 0.2 ms	msec/requests
Propagation delays [56, 57]	1 ms (200 km), 0.5 ms (100 km), 0.1 ms (20 km)	ms

of 0.1 ms. While the link between the core and the edge are configured with 0.5 ms and 1 ms, depending on the different simulation scenarios. All links are considered as 10 Gbps links. Table 1 provides a summary of different simulation parameters and their values present in the literature.

B. SIMULATOR VALIDATION

In this section, we will provide the results for simulator validation. First, we will validate the simulator against the Mininet emulation tool. And then we will validate our simulator with the analytical model explained in the previous section.

1) MININET EMULATION

The values of service rate of the base station, OpenFlow switch and SDN controller are obtained by running a simulation in Mininet emulator [37] and monitoring the service time per packet in all three network entities. Table 2 provides the values of the parameters used in this simulation.

The results from mininet emulator and our simulator are shown in Fig. 10 a, b, and c. Fig. 10a shows the result for the registration procedure, and the delay for smaller values of arrival rate is around 6 ms. When the request arrival intensity is increased, the delay also starts to increase. This is expected because when the number of packets in the network increases, the packets must wait longer in the queues which results in higher overall end-to-end delay. However, when the arrival rate crosses 500 requests per second, the number of packets in the network increases to the extent that the network becomes unstable and the delay increases exponentially.

The intra and inter UPF handover results are given in Fig. 10 b and c, respectively. At lower arrival rates, the delay to complete a single handover is around 10 ms in mininet emulator. However, it can be noticed that the mininet results are not very close to the simulator results. The reason for this lies in the Python module that was used in mininet for sending packets from the base station. The module used was scapy [58], and at lower arrival rates the module takes more time to send a packet. While at higher arrival rates,

TABLE 2. Mininet simulation parameters.

Parameter	Value	Unit
Arrival rate	Varied from 0.5 to 700	requests/sec
Base station Processing time	0.6, 0.47	ms
OpenFlow Switch Processing time	0.006	ms
SDN Controller processing	0.23	ms
Transmission time between nodes	0.56	µs
Simulation time	10 (600)	Minutes (seconds)
Number of replications	30	

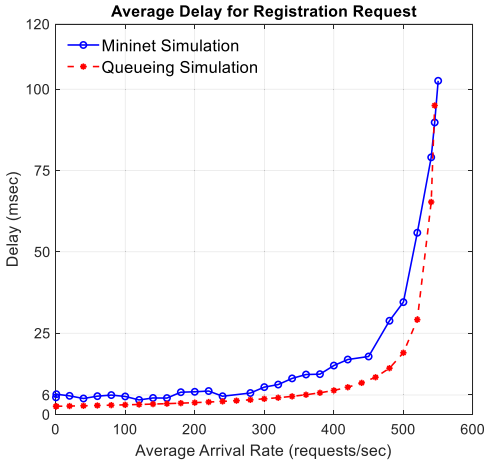
the values were much stable. Another issue with the mininet emulator is that it is very processor dependent. It can be seen in all the graphs that the mininet results fluctuate and the curve is not stable. However, we can see from the graphs that our simulator shows similar behavior to the mininet network emulation and the working of our simulator can be validated. Next, we will try to verify our simulator using the analytical model.

2) ANALYTICAL MODEL BASED ON JACKSON NETWORK

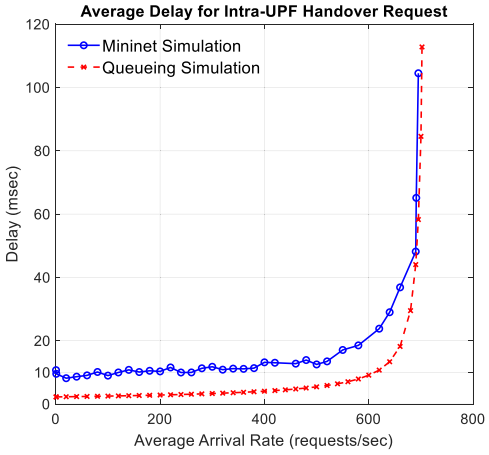
In order to verify our simulator, we refer to the equations derived in section 6 for the average delay of packets in the network. To obtain the results, we plug in the values of the initial arrival rate (λ) and the service rate (μ) of each node in (7), (8) and (9). The values of the service rate for each node are obtained from table 2.

The result of the registration procedure is shown in Fig. 11a. For handover procedures, Fig. 11b shows the delay for intra-UPF handover and Fig. 11c gives the delay for inter-UPF handover. For the registration procedure, the values of the initial arrival rate at the base station are varied from 0.5 to 550 requests per second, and the delay from the simulation and analytical model is plotted. From Fig. 11a, we can see that the results obtained from the analytical model and the simulator are almost identical and both curves follow similar trends.

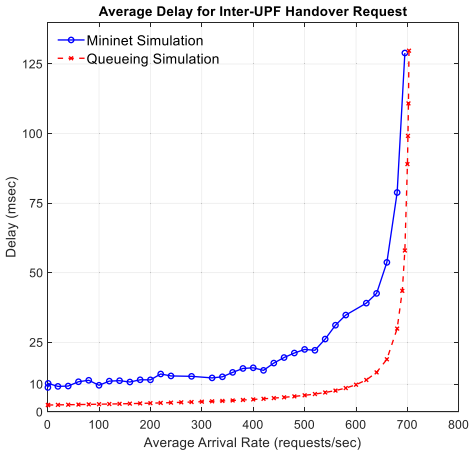
In case of handover, the arrival rate is varied from 0.5 to 700 requests per second. At lower arrival rates, the result of the analytical model and the simulator is almost the same in Fig. 11 b and c. At higher arrival rates, around 500 requests per second, the values of the analytical model become slightly higher as compared to the delay from the simulator. However,



(a)

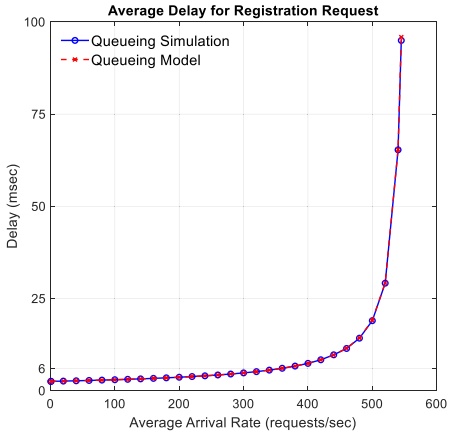


(b)

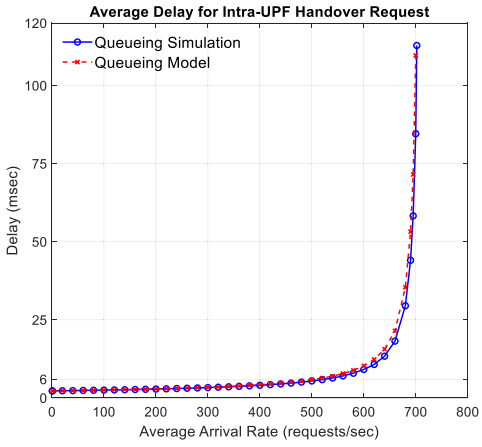


(c)

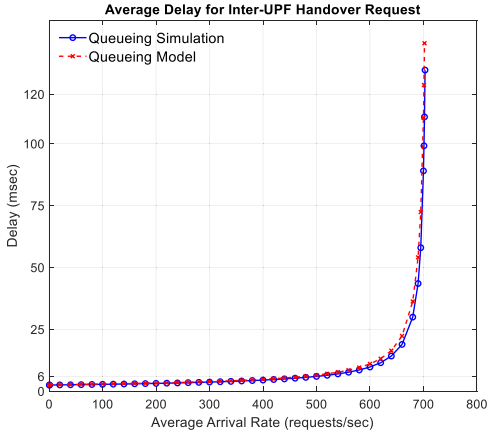
FIGURE 10. Simulation vs Mininet emulation. delay in requests (a) Registration (b) Intra-UPF handover (c) Inter-UPF Handover.



(a)



(b)



(c)

FIGURE 11. Simulation vs Model. Delay in Requests (a) Registration (b) Intra-UPF handover (c) Inter-UPF Handover.

the overall trend of the result from the analytical model is still similar to the simulator. So, it can be observed from all the graphs that the analytical model follows a similar behavior as the results from our simulator. Hence, we can verify our simulator.

C. RESULTS OF NETWORK SIMULATOR

In order to do extensive simulations and find the bottlenecks in the network, we create four simulation scenarios based on different levels for each factor. For each scenario, we find the end-to-end delay for the requests, average throughput, and

TABLE 3. Simulation scenarios.

Parameters	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Switch Processing time	10 µs	10 µs	10 µs	10 µs
Base station Processing time	1 ms	1 ms	0.6 ms	0.6 ms
SDN Controller Processing	0.2 ms	0.2 ms	0.2 ms	0.2 ms
Propagation delays – Access to Edge	0.1 ms	0.1 ms	0.1 ms	0.1 ms
Propagation delays – Edge to Core	0.5 ms	1 ms	0.5 ms	1 ms
Propagation delays (rest of the links)	10 µs	10 µs	10 µs	10 µs

utilization of resources at the SDN controller, for all procedures in the 5G network. The simulation runs for 10 minutes, and the results are obtained over the average of 30 replications. The four scenarios are given in table 3 below:

1) END-TO-END DELAY FOR ALL SCENARIOS

End-to-end delay for a procedure is the time that the request takes to complete. So, end-to-end delay is obtained by subtracting the time when the request is finished with the time when it started.

$$end\ to\ end\ delay = end\ time - start\ time$$

The final value of the end-to-end delay is the mean of delays for all the requests that arrived in the network.

a: REGISTRATION REQUEST

The result of the average delay in the registration request is shown in Fig. 12. The arrival rate for scenarios 1 and 2 is varied from 0.5 to 325 requests per second. While for scenarios 3 and 4 it is varied from 0.5 to 545 requests per second. For scenario 1, the delay starts at around 6.7 ms for low arrival rates, while for scenario 2 the delay at low arrival rate is around 9.2 ms as we see in Fig. 12a. However, in both scenarios the delay increases drastically after 300 requests/sec. On the other hand, for scenarios 3 and 4 the delay starts from 5.4 ms and 7.9 ms respectively, which is shown in Fig. 12b. Whereas, in this case the delay increases after 500 requests/sec.

The reason behind this is that for the first two scenarios the value of base station processing is 0.1 ms, which roughly translates to the processing capacity of 1K packets per second. And since the base station handles 3 packets for each request from the UE, so after 300 requests per second the base station starts handling packets close to 1000, and

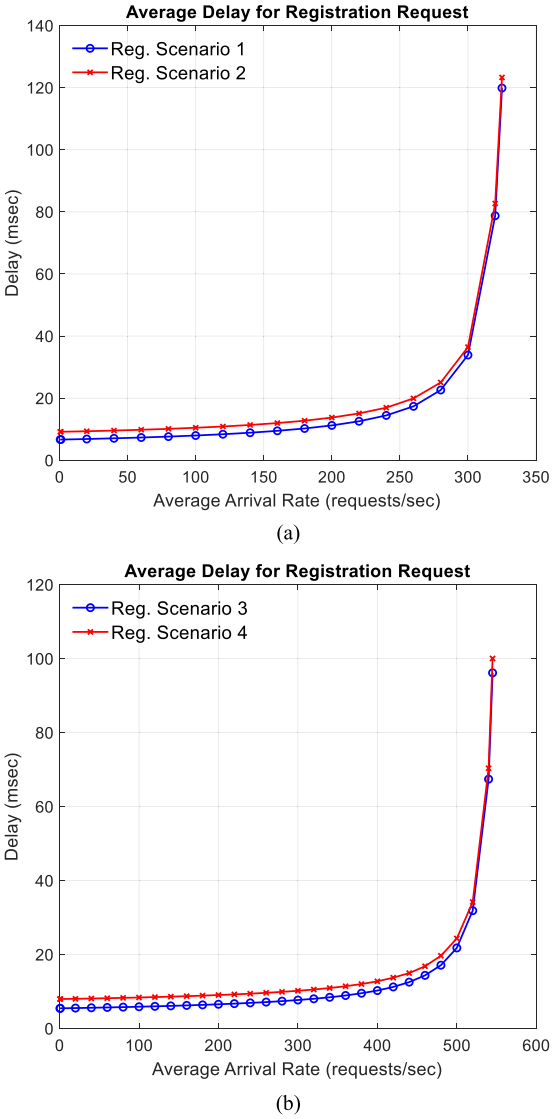


FIGURE 12. Delay for Registration Requests (a) Scenario 1 & 2 (b) Scenario 3 & 4.

hence, the delay increases significantly. Similarly, for scenarios 3 and 4, the processing time is 0.6 ms which translates into 1.6K packets per second. So, after 500 arrivals per second, the number of packets at the base stations reaches around 1500 and causes the delay to increase.

b: Xn BASED HANDOVER WITHOUT UPF RE-ALLOCATION

The results for Intra-UPF handover requests end-to-end delay are shown in Fig. 13. The arrival rate for scenarios 1 and 2 is similar to the registration procedure, but for scenarios 3 and 4 the arrival rate is varied from 0.5 to 550 requests per second. For scenarios 1 and 2, shown in Fig. 13a, the delay starts from 6.2 ms and 7.9 ms respectively, at lower arrival rates. While for scenarios 3 and 4, the average delay at a small number of arrivals is 4.7 ms and 6.7 ms respectively, as shown in Fig. 13b. We can also observe similar behavior to registration results, in terms of the threshold after which

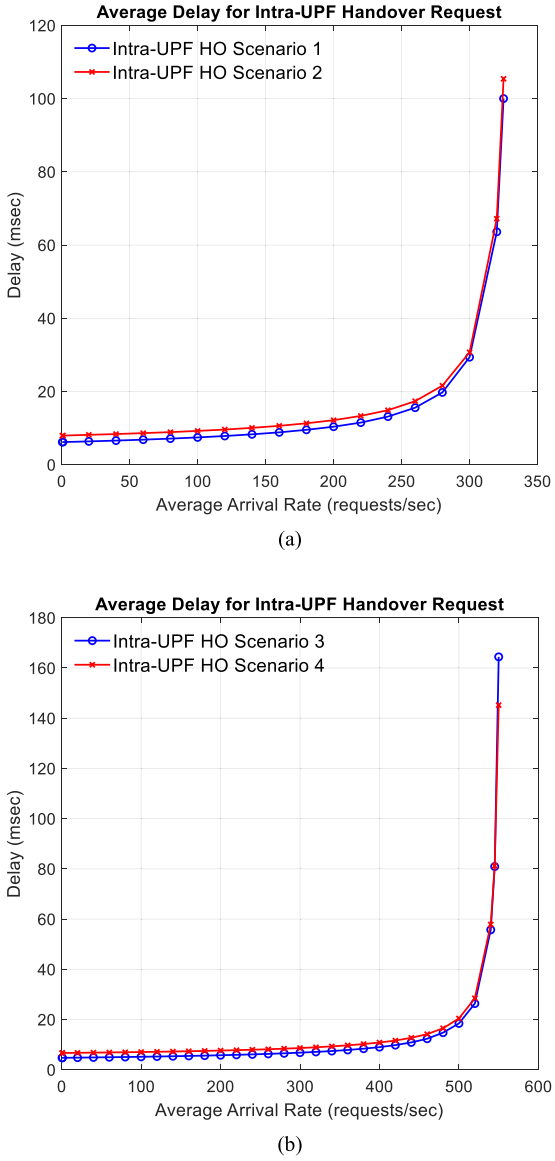


FIGURE 13. Delay for Intra-UPF Handover Requests (a) Scenario 1 & 2 (b) Scenario 3 & 4.

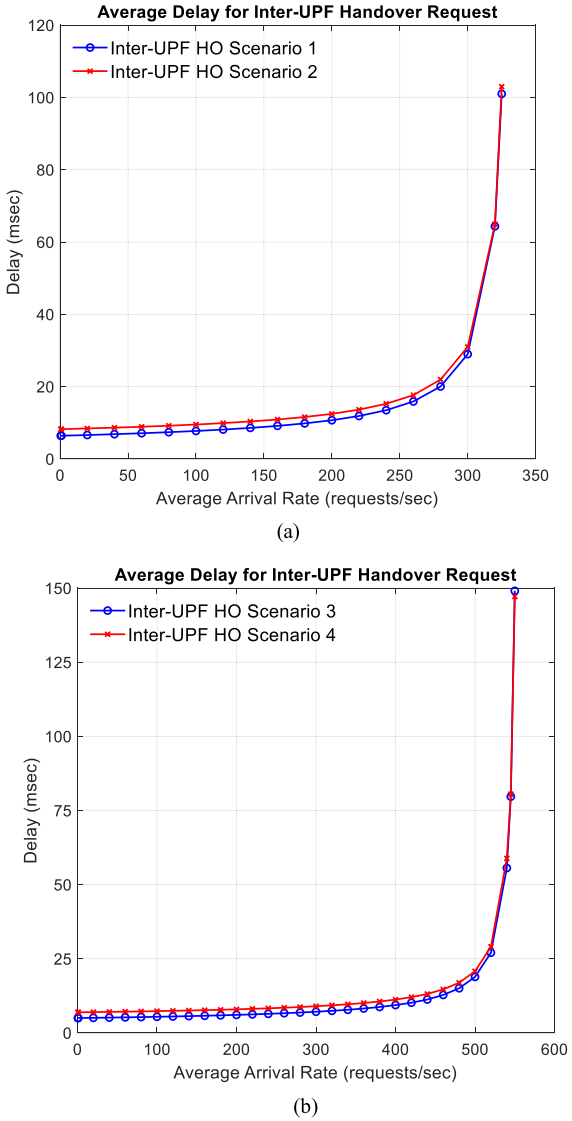


FIGURE 14. Delay for Inter-UPF Handover Requests (a) Scenario 1 & 2 (b) Scenario 3 & 4.

the results start to become unstable. Because in intra-UPF handover, the base station also processes 3 packets for each request, similar to the registration procedure, which results in having a similar value of the threshold.

It is also important to mention that the delays for scenarios 2 and 4 are higher as compared to scenarios 1 and 3, respectively. The reason for this is that the packet propagation delay between the edge and the core network is higher in scenarios 2 and 4 as compared to their counterparts, which results in higher overall delay to complete a single request.

c: Xn BASED HANDOVER WITH UPF RE-ALLOCATION

Average delay results for an inter-UPF handover request is given in Fig. 14. The arrival rates for inter-UPF handover is similar to the one used in intra-UPF handover. At low arrival rates, the delay for scenarios 1 and 2 is 6.4 ms and 8.2 ms respectively, as shown in Fig. 14a. On the other hand, the delay of 5 ms and 7 ms is observed for scenarios 3 and 4, respectively. Looking closely at the results, we observe that the delay for scenario 3 is lower than scenario 1, because in scenario 3 the processing time of the base station is lower as compared to scenario 1, which results in reducing overall delay. A similar analogy can also be applied to scenarios 2 and 4. Like previous results, we see similar trends in these results too.

2) AVERAGE THROUGHPUT AT CONTROLLER FOR ALL SCENARIOS

Throughput measures the number of packets handled by the controller in one-unit time. To find the throughput at the controller, we find the total number of packets processed by the SDN controller and divide it by the total simulation time.

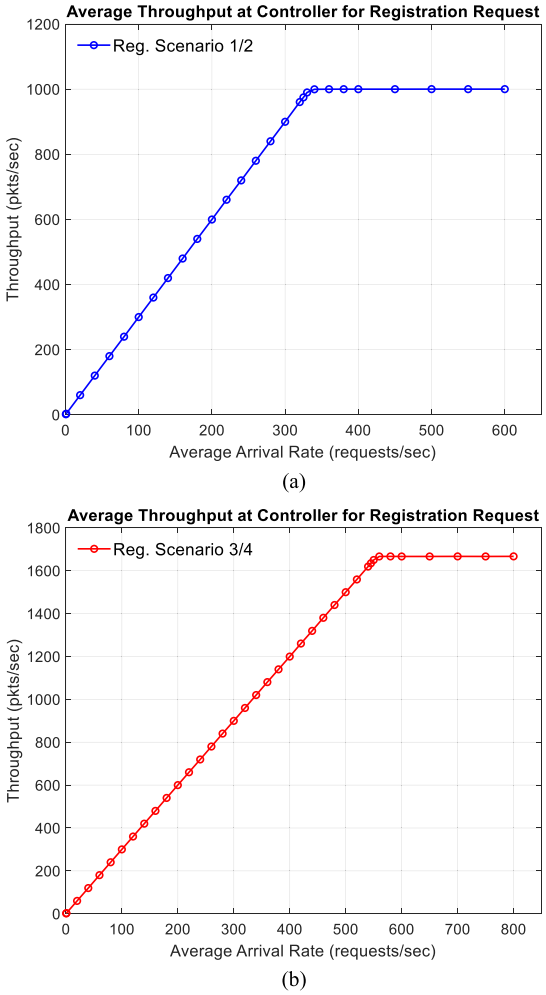


FIGURE 15. Throughput at SDN controller for registration request (a) Scenario 1 & 2 (b) Scenario 3 & 4.

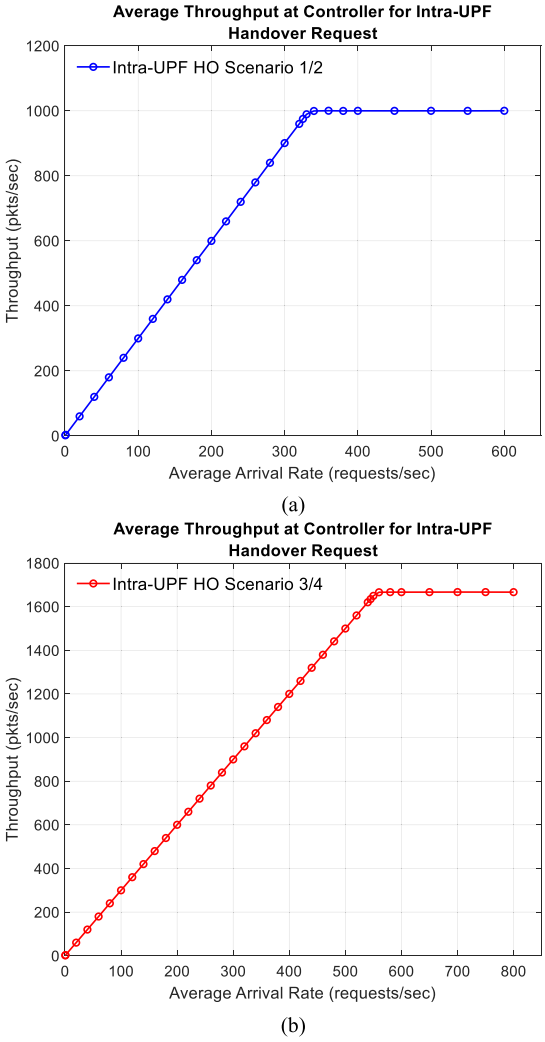


FIGURE 16. Throughput at SDN controller for Intra-UPF Handover request (a) Scenario 1 & 2 (b) Scenario 3 & 4.

The average throughput is obtained by taking the mean of throughput across all simulation replications.

$$Throughput = \frac{Packet\ sprocessed}{clock}$$

a: REGISTRATION REQUEST

Fig. 15 shows the throughput at the controller during the registration procedure. We can see from Fig. 15a that when the arrival rate increases, the throughput at the controller also increases linearly, which is as expected. The throughput keeps increasing until it reaches the threshold, which is around 330 requests/sec for scenarios 1 and 2. In these scenarios, the processing time of the base station was set to 1 ms, which translates into 1K requests per second, as mentioned earlier. So, after the arrival rate crosses the threshold, the throughput starts to stabilize at 1000 packets/sec.

For scenarios 3 and 4, we can see in Fig. 15b that the threshold, in this case, is around 550 requests/sec. Since in these scenarios the processing time of the base station was reduced from 1 ms to 0.6 ms, and hence the capacity of the base station increases to 1666 packets/sec. Which is also clear from the result that the throughput becomes stable at 1666 packets/sec after the threshold.

b: Xn BASED HANDOVER WITHOUT UPF RE-ALLOCATION

The throughput at the controller for intra-UPF handover procedure, shown in Fig. 16, is very similar to the registration procedure because in both procedures the controller handles three packets for a single request. So, the threshold and the value at which the throughput stabilizes are similar in both cases.

c: Xn BASED HANDOVER WITH UPF RE-ALLOCATION

Fig. 17 shows the average throughput at the SDN controller during the inter-UPF handover procedures. The results for inter-UPF handover are slightly different than the previous two procedures. The reason behind this is that during inter-UPF handover, the controller processes a total of four packets

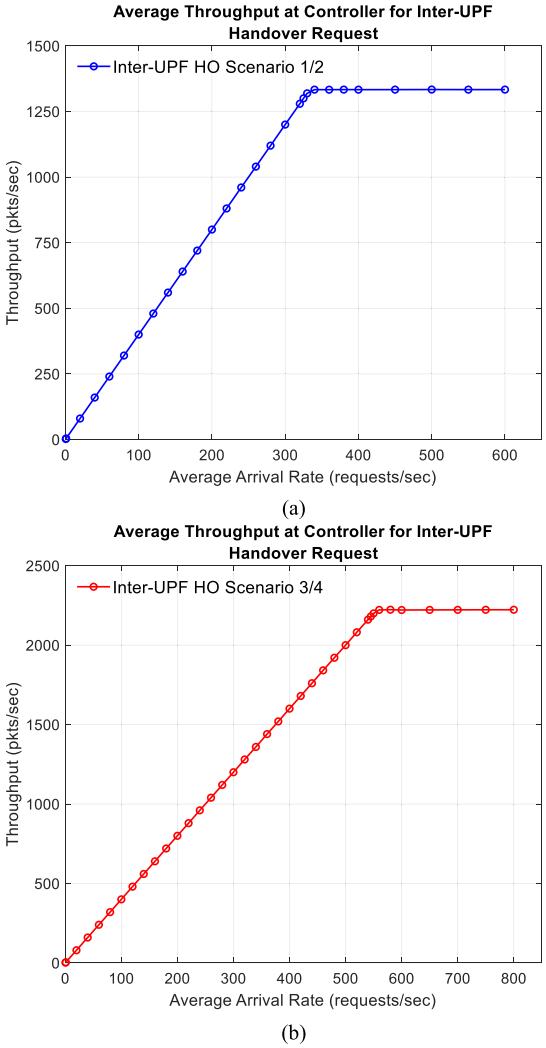


FIGURE 17. Throughput at SDN Controller for Inter-UPF handover request (a) Scenario 1 & 2 (b) Scenario 3 & 4.

for a single handover request, which results in higher throughput.

We can see from Fig. 17a and 17b that the threshold after which the throughput becomes stable is still the same. This is because the processing time for the base station does not change as compared to the previous two procedures. But the only difference is that the throughput stabilizes at 1333 packets/sec for scenarios 1 and 2. Whereas, in scenarios 3 and 4, the maximum throughput achieved is 2222 packets/sec, which is four times the threshold at which the values start to stabilize.

3) RESOURCE UTILIZATION AT CONTROLLER FOR ALL SCENARIOS

Resource utilization shows the proportion of simulation time during which the controller was busy. To find the utilization of the controller, the throughput at the controller is divided

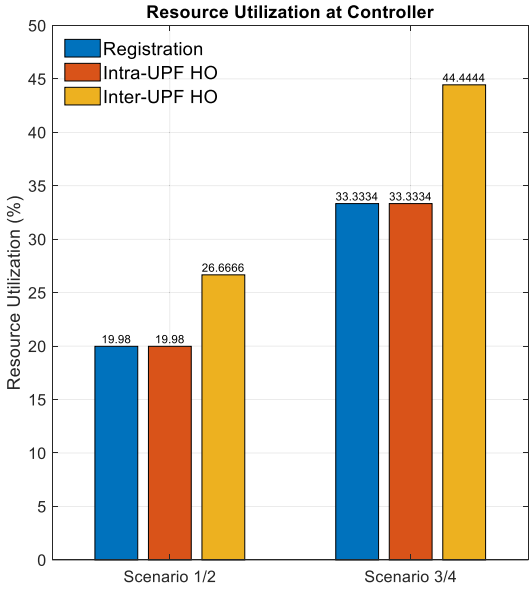


FIGURE 18. Resource utilization at SDN controller.

with its capacity.

$$Utilization\ of\ Controller = \frac{Throughput\ at\ Controller}{Capacity\ of\ Controller}$$

In order to find the percentage utilization, multiply the above equation with 100. Fig. 18 shows the percentage of resource utilization of the SDN controller for different procedures in all scenarios. The processing time of the SDN controller was fixed at 0.2 ms in all scenarios as given in table 3. This processing time translates into the capacity of 5K requests per second. For scenarios 1 and 2, the resource utilization during registration and intra-UPF handover procedures is 19.98%. Whereas 26.67% of resources are utilized during inter-UPF handover. On the other hand, the utilization is increased for scenarios 3 and 4 because the processing time of the base station is reduced to 0.6 ms in these scenarios. In the case of registration and intra-UPF handover, 33% of utilization is recorded. While for inter-UPF handover, only 44.44% of controller resources are utilized.

From this analysis, we can deduce that the bottleneck in the architecture is the base station. Since the processing time of the base station is higher than the rest of the nodes. And because of this, the maximum controller resource utilization achieved is only 44%.

D. COMPARISON WITH TRADITIONAL 5G ARCHITECTURE

This section provides the results for the proposed architecture and the traditional 5G architecture for all possible combinations of different factors, that are selected, and their levels. Also, a full factorial design of these results is performed to analyze the factors that have a higher impact on the results. A full factorial design considers all possible combinations of all levels across all factors. The result of such design, called Analysis of Variance (ANOVA), identifies the significance of

TABLE 4. Simulation factors and their levels.

Factors (SDN 5G/ TRAD. 5G)	Levels (SDN 5G/ TRAD. 5G)	Values
Switch / UPF Processing time	SW-1 / UPF-1	10 µs
	SW-2 / UPF-2	140 µs
Base station Processing time	BS-1	1 ms
	BS-2	0.6 ms
SDN Controller / Network Functions (NF) processing	CONT-1 / NF-1	0.1 ms
	CONT-2 / NF-2	0.2 ms
Propagation delays	PROP-1	1 ms
	PROP-2	0.5 ms
Arrival rate		1 request/sec
Propagation delay between NFs (TRAD. 5G)		1 ms
Simulation time		3600 seconds
Number of replications		30

each factor on the output (or response) variable, along with the effect of interactions of different factors [59].

The response variable selected for full factorial design is end-to-end delay, and the comparison between the traditional 5G architecture and the proposed SDN based 5G architecture is provided. Different factors are selected, and their levels are obtained from table 1. The first three factors are controller (CONT), switch (SW) and the base station (BS) processing times, and the fourth factor is the propagation delay (PROP) between the edge and the core network. For the traditional 5G architecture, we use processing times of network functions (NF) and UPF, instead of the controller and switch processing time, respectively. The propagational delay between 5G network functions (NF) is 1 ms [60], [61]. With four factors and each factor having two levels, a total of 16 simulation runs are required for each procedure. The simulation ran for 3600 seconds and the arrival rate was fixed at 1 request per second, in order to get the results having the minimal effect of the external factors, like queueing delays, etc. The selected factors and their levels are given in table 4, along with some simulation parameters:

1) REGISTRATION REQUEST

Table 5 provides the end-to-end delay results for the registration procedure in the SDN based 5G architecture. From the table, we can see that the delay ranges from 5.15 ms to 9.81 ms. The effect of controller processing (CONT) is very slight on the end-to-end delay, with the difference of 0.3 ms between its two levels. Switch processing (SW), on the other hand, has a higher impact on the results, with a

TABLE 5. End-to-end delay (ms) for Registration request in the proposed architecture.

		PROP-1		PROP-2	
		BS-1	BS-2	BS-1	BS-2
CONT-1	SW-1	8.87	7.65	6.36	5.15
	SW-2	9.52	8.302	7.00	5.80
CONT-2	SW-1	9.16	7.95	6.66	5.46
	SW-2	9.81	8.6	7.31	6.1

difference of 0.65 ms between its levels. Base station (BS) processing shows a much higher influence and the difference in its two levels is around 1.2 ms. Whereas the highest impact on the delay is recorded from the propagation delay (PROP), and the effect of changing its level from PROP-2 to PROP-1 introduces a delay of 2.5 ms in results. However, all the differences are constant across all the simulations, which means that the factors are not interacting in this case.

We perform similar simulations on the traditional 5G network, and the delay is significantly higher compared to the proposed SDN based 5G network. The average delay in the traditional 5G network is 15.82 ms, with the lowest and the highest being 13.41 ms and 18.23 ms, respectively. Table 6 provides the percentage of improvement in the end-to-end delay for our proposed SDN based 5G architecture. On average, the SDN based 5G architecture shows 53% less delay in completing a registration procedure. The highest improvement of 62.37% is obtained when all the factors are at their second level except for SW/UPF factor which is at the first level. On the other hand, the lowest improvement recorded is 44.39%. The reason behind this significant improvement is the number of messages required in the registration request. The proposed architecture requires fewer messages as compared to the traditional 5G architecture, as shown in the call flow of registration.

Applying ANOVA on the results in table 5, we quantify the main factors that impact the end-to-end delay the most. Only the main factors contribute to results, and all order of interactions between the factors is 0%. Out of the main effects, the PROP factor has a major effect on the results with a 76% variation. After that, BS shows the variation of 17.75% on the delay results. SW and CONT comes next, with a contribution of 5% and 1% to overall results. For the traditional 5G architecture, the PROP factor has the highest impact on the delay results with 70.30%. BS and NF come next with 16.20% and 13.50% variation on results, respectively. While UPF does not affect the results as expected.

2) Xn BASED HANDOVER WITHOUT UPF RE-ALLOCATION

The results for intra-UPF handover in the proposed architecture is given in table 7. The delay ranges from 4.56 ms to 8.62 ms. We can see similar results, in terms of impact on the delay. CONT has the least impact, with a constant difference

TABLE 6. Percentage (%) improvement in registration E2ED results in the proposed architecture.

		PROP-1		PROP-2	
		BS-1	BS-2	BS-1	BS-2
CONT-1 / NF-1	SW-1 / UPF-1	48.19%	51.95%	56.50%	61.60%
	SW-2 / UPF-2	44.39%	47.85%	52.12%	56.75%
CONT-2 / NF-2	SW-1 / UPF-1	49.75%	53.29%	57.61%	62.37%
	SW-2 / UPF-2	46.19%	49.47%	53.47%	57.96%

TABLE 7. End-to-end delay (ms) for Intra-UPF handover request in the proposed architecture.

		PROP-1		PROP-2	
		BS-1	BS-2	BS-1	BS-2
CONT-1	SW-1	7.79	6.48	6.00	4.56
	SW-2	8.43	7.13	6.62	5.19
CONT-2	SW-1	7.98	6.68	6.17	4.75
	SW-2	8.62	7.32	6.8	5.38

of 0.2 ms between its levels. SW has a slightly higher effect, with the difference of 0.65 ms between SW-1 and SW-2. However, the results for BS and PROP factors are slightly different than the registration procedure. When propagation delay is set to PROP-1, the increase in delay from BS-2 to BS-1 is 1.31 ms, but when the propagation delay is fixed at PROP-2, this increase is 1.44 ms. Similar behavior is shown by propagation delay. When the BS is set on the first level, the PROP causes an increase of 1.81 ms in delay. But, when the BS is fixed on the second level, the PROP results in the increases of 1.94 ms. This behavior shows that there is some interaction between the PROP and the BS in the case of intra-UPF handover, this will be evident from ANOVA analysis.

The highest delay to perform an intra-UPF handover in the traditional 5G network is around 11 ms. The lowest delay observed is 6.84 ms and the average delay required to complete an intra-UPF handover is 8.9 ms. PROP factor has the highest impact on the results with an increase of 2 ms from PROP-2 to PROP-1. BS factor comes after that with the difference of 1.6 ms between its two levels. NF and UPF have the least impact with the difference of 0.4 ms and 0.13 ms, respectively. Table 8 gives the percentage of improvement in the proposed architecture. The proposed SDN based 5G architecture provides 20.25% to 34.39% less end-to-end delay during intra-UPF handover compared to traditional 5G architecture.

TABLE 8. Percentage (%) Improvement in Intra-UPF handover E2ED results in the proposed architecture.

		PROP-1		PROP-2	
		BS-1	BS-2	BS-1	BS-2
CONT-1 / NF-1	SW-1 / UPF-1	25.45%	26.61%	28.99%	33.33%
	SW-2 / UPF-2	20.25%	20.42%	22.75%	25.43%
CONT-2 / NF-2	SW-1 / UPF-1	26.38%	27.71%	30.20%	34.39%
	SW-2 / UPF-2	21.42%	21.88%	24.11%	27.00%

On average, the proposed architecture provides 26% better performance than the traditional 5G architecture.

Applying ANOVA analysis, it is evident that the main effects contribute 99.93% of the variation in the end-to-end delay results. CONT has the least contribution with 0.6%. While PROP has the highest impact with 60.28% of variation. SW and BS contribute 6.94% and 32.1% of end-to-end delay variation. The first order interaction, and specifically the interaction between the BS and the PROP, explains 0.07% of the variation in the end-to-end delay results. This is not very significant; however, it explains why the increase in table 7 was not constant in the case of BS and PROP. Traditional 5G architecture shows similar trends to the registration request ANOVA results, with the PROP factor having most of the impact on the results with 59.24%. BS explains 38.15% of the variation in the results. The effect of NF is reduced to 2.37%, compared to the registration procedure, because in intra-UPF handover the number of messages exchanged with NF is less compared to the registration procedure. While the contribution of UPF on the results is 0.23%. We can also see that in the traditional 5G architecture there are no interactions between any factors.

3) Xn BASED HANDOVER WITH UPF RE-ALLOCATION

End-to-end delay results for inter-UPF handover procedure in the proposed architecture is given in table 9. CONT shows similar behavior, with the least effect on the results, with an increase of 0.3 ms from CONT-1 to CONT-2. SW factor shows a higher impact compared to previous procedures, from SW-1 to SW-2, the induced delay is around 1.02 ms. However, CONT and SW factors are still constant across all the simulation runs. The other two factors, BS and PROP, show exactly similar behavior to intra-UPF handover results. When the BS level is changed from BS-2 to BS-1, the increase of 1.3 ms and 1.4 ms is observed with PROP-1 and PROP-2, respectively. While PROP shows an increase of 1.83 and 1.94 ms, with BS-1 and BS-2, respectively.

In the traditional 5G network, the average delay to complete a request is 9.2 ms. While the highest and the lowest

TABLE 9. End-to-end delay (ms) for Inter-UPF handover request in the proposed architecture.

		PROP-1		PROP-2	
		BS-1	BS-2	BS-1	BS-2
CONT-1	SW-1	7.95	6.65	6.14	4.72
	SW-2	8.98	7.68	7.15	5.74
CONT-2	SW-1	8.24	6.94	6.41	5.00
	SW-2	9.28	7.98	7.43	6.03

TABLE 10. Percentage (%) improvement in Inter-UPF handover E2ED results in the proposed architecture.

		PROP-1		PROP-2	
		BS-1	BS-2	BS-1	BS-2
CONT-1 / NF-1	SW-1 / UPF-1	24.93%	25.95%	28.44%	32.38%
	SW-2 / UPF-2	18.29%	18.12%	20.38%	22.22%
CONT-2 / NF-2	SW-1 / UPF-1	25.70%	26.79%	29.48%	33.16%
	SW-2 / UPF-2	19.16%	19.15%	21.71%	23.48%

delay recorded are 11.48 ms and 6.98 ms, respectively. The effect of PROP and BS factors on the results is similar to intra-UPF handover. However, the impact of NF and UPF on the results is increased, compared to intra-UPF handover, with the difference of 0.5 ms and 0.4 ms, respectively. The percentage of improvement for different combinations in the proposed architecture is given in table 10. On average, the proposed architecture shows 24% better results than the traditional 5G architecture. For different combinations, the performance of the proposed architecture is 18.29% to 33.16% better when compared to the traditional 5G architecture.

From ANOVA analysis, the main effects explain most of the variation in the end-to-end delay, with 99.95%. CONT contributes 1.3% of variations in the results. PROP has the most significant impact on the variation of results with 54.4%. SW contributes 16.15%, and BS is responsible for 28.1% of variations in the results. In this procedure also, the interaction between PROP and BS explains 0.05% of variations in results, which is also evident from table 9. The ANOVA in the case of traditional 5G architecture shows a similar trend to intra-UPF handover results, with PROP and BS factors explaining most of the variation in the results. But in this case, the effect of NF and UPF is increased to 3.58% and 2.26%. Also, there are no interactions between any factors, similar to previous procedures.

TABLE 11. Delay comparison for registration procedure.

Research Work	Delay (ms)	Evaluation Method
Proposed architecture	7.48	Simulation
Jia [62]	23	Simulation
An et al. [63]	50	Simulation
Nguyen et al. [64]	100	Simulation

E. COMPARISON WITH PREVIOUS WORKS

In this section we provide a performance comparison between our proposed architecture and previous research works present in the literature. We compare the end-to-end delay observed while performing registration and handover procedures in these architectures. We only consider delay for low traffic load scenario; in this way we reduce the delays introduced by external factors, like queueing delays etc.

Table 11 compares the end-to-end delay for registration procedure. From this table, the average delay to complete a registration request in our proposed architecture is 7.48 ms. Based on simulation results, Jia [62] reported a delay of 23 ms during network attachment process in their architecture. However, the author did not provide sufficient information regarding the simulator used and the simulation environment. Furthermore, the performance of the network against various traffic loads is not investigated. An *et al.* [63] proposed a network architecture based on SDN and NFV principles. The authors carried out simulations to measure the delay required for attachment procedure. In their simulation, the processing time of network nodes was varied from 0 to 3 ms depending on the traffic load. While the message propagation time was set to 1 ms. From the results, the average delay reported is 50 ms during low load scenario. Figure 7 in [63] illustrates the trend in the delay against the number of devices. From the graph, the delay increases with the increase in the number of devices, which is the similar trend we observe from our results. Authors in [64] proposed a load balancing mechanism in order to tackle the issue of scalability in 5G networks. The authors used Open5GCore [65] platform to conduct their simulations. Figure 6 in [64] demonstrates the average registration delay versus the request arrival rate. When the request rate increases, the delay also increases until it reaches around 1 second at 450 requests per second. While at low arrival rates, the average delay to complete a registration request is around 100 ms.

The comparison of delay for handover procedure is shown in table 12. We only consider intra-UPF handover procedure since it is the most widely reported in the literature. The average delay to complete a handover in our proposed architecture is 6.61 ms. Prados-Garzon *et al.* [66] suggested an OpenFlow based handover procedure in SDN based 5G architecture. The authors simulated the proposed architecture in ns-3 simulator [31] and reported the time of 8.31 ms to execute a handover procedure. We can observe that their reported delay is very close to our results, and the reason behind this is that the

TABLE 12. Delay comparison for handover procedure.

Research Work	Delay (ms)	Evaluation Method
Proposed architecture	6.62	Simulation
Prados-Garzon et al. [66]	8.31	Simulation
Contreras et al. [67]	45	Prototype
Bilen et al. [68]	150	Simulation

values of simulation parameters, like processing times and propagation delay, used in their work are very similar to what we have used in our simulation. Figure 4 of [66] compares the delay for handover with the data rate per UE, and as expected, the delay increases exponentially when data rate exceeds 1 Gbps.

Contreras *et al.* [67] proposed an SDN based mobility management solution for 5G networks. The authors evaluated the performance by implementing a prototype of their architecture. The testbed comprises of a single Ryu SDN controller, along with several data plane switches that run Open vSwitch and one hundred emulated users. Table 1 of [67] reports the delay to complete a handover request against several arrival rates. As the arrival rate increases, the congestion rises in the network which causes the delay to increase. At low arrival rate, the average delay to complete a handover request is 45 ms as reported by the authors. An SDN based mobility management approach in ultra-dense 5G networks is proposed in [68]. The performance of the proposed scheme is evaluated using simulation in MATLAB-Simulink [69]. From their results, the average time to execute a handover request is around 150 ms. This delay starts to increase as the simulation time increases, as shown in figure 6 of [68].

From this comparison, it is evident that our proposed architecture provides less end-to-end delay during registration and handover procedures compared to other architectures present in the literature. In general, we observe similar trends is delay between our work and previous studies.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we explain the basic operation of the standard 5G core architecture and propose a software-defined networking (SDN) based 5G core architecture. We also explain two very basic procedures in cellular networks, namely initial attachment and handover. A comparison of these procedures in the 5G architecture and the proposed SDN based 5G architecture is provided.

A network simulator is developed in Python to evaluate the performance of the proposed SDN based 5G architecture and the traditional 5G architecture. The simulator was verified with the mininet emulator and queueing model. Different simulations were conducted considering different factors to evaluate the performance in terms of end-to-end delay, throughput at controller and resource utilization of controller, and to identify the potential bottlenecks.

The factors included in the simulations are, the controller (CONT) and network function (NF) processing times, switch (SW) and UPF processing times, base station (BS) processing time and propagation delay (PROP). The results show that the major bottleneck in the architecture is the base station. The performance comparison between the proposed architecture and the traditional 5G architecture shows that the proposed architecture achieved less end-to-end delay while performing registration and handover operations. On average, the proposed architecture provides 53% less delay in case of registration request. While for handover scenarios, our architecture outperforms traditional 5G architecture with 26% reduced delay for intra-UPF handover. In case of inter-UPF handover, the average reduction in the delay is 24% compared to the traditional 5G architecture. Also, a full factorial design is made to analyze the major factors impacting the results, and the results show that the PROP factor was responsible for most of the variations in the end-to-end delay results. A performance comparison between our proposed architecture and previous research works is also presented. From this comparison, our proposed architecture shows similar trend in delay results while providing less end-to-end delay during registration and handover procedures.

Several research directions are possible for future work. First, analysis of the proposed architecture under different data plane traffic loads. Second, investigating the load balancing and scalability of SDN controllers is also an important topic. Another direction is implementing the techniques that can enforce some quality of service (QoS), for example, efficient traffic routing of some delay-sensitive applications and introducing load balancing on the forwarding devices. A real test-bed implementation of the proposed architecture and its performance evaluation is another important research direction.

REFERENCES

[1] *TELCOMA GLOBAL | 5g Technology Introduction*. Accessed: Dec. 6, 2018. [Online]. Available: https://telcomaglobal.com/blog/17780/5g-technology-introduction

[2] CISCO. *VNI Mobile Forecast Highlights, 2016–2021. CISCO*. Accessed: Feb. 4, 2019. [Online]. Available: https://www.cisco.com/assets/sol/sp/vni/forecast_highlights_mobile/index.html#~Region

[3] A. AbdulGhaffar, S. M. Mostafa, A. Alsaleh, T. Sheltami, and E. M. Shakshuki, "Internet of Things based multiple disease monitoring and health improvement system," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 3, pp. 1021–1029, Mar. 2020, doi: 10.1007/s12652-019-01204-6.

[4] NGMN. *NGMN 5G White Paper*. Accessed: Feb. 22, 2019. [Online]. Available: https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2015/NGMN_5G_White_Paper_V1_0.pdf

[5] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, P. Bertin, and A. Kerbellec, "On evaluating different trends for virtualized and SDN-ready mobile network," in *Proc. IEEE 6th Int. Conf. Cloud Netw. (CloudNet)*, Sep. 2017, pp. 1–6, doi: 10.1109/CloudNet.2017.8071534.

[6] G. C. Valastro, D. Panno, and S. Riolo, "A SDN/NFV based C-RAN architecture for 5G mobile networks," in *Proc. Int. Conf. Sel. Topics Mobile Wireless Netw. (MoWNeT)*, Jun. 2018, pp. 1–8, doi: 10.1109/MoWNet.2018.8428882.

[7] S. B. Hadj Said, B. Cousin, and S. Lahoud, "Software defined networking (SDN) for reliable user connectivity in 5G networks," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–5, doi: 10.1109/NETSOFT.2017.8004219.

[8] O. N. Foundation. *OpenFlow Switch Specification*. Accessed: Mar. 2, 2019. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[9] S. Staff. *What is OpenFlow? Definition and How it Relates to SDN*. Accessed: Mar. 15, 2020. [Online]. Available: https://www.sdxcentral.com/networking/sdn/definitions/what-is-openflow/

[10] *System Architecture for the 5G System*, Standard ETSI TS 123 501 3GPP TS 23.501, 3GPP, 2018.

[11] S. Dredge. *What is the 5G User Plane Function (UPF)? Metaswitch*. Accessed: Nov. 16, 2019. [Online]. Available: https://www.metaswitch.com/knowledge-center/reference/what-is-the-5g-user-plane-function-upf

[12] G. Brown, "Service-based architecture for 5G core networks," Huawei Technol. Co. Ltd, Shenzhen, China, White Paper, 2017.

[13] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2815638.

[14] F. Hu, *Network Innovation Through OpenFlow and SDN: Principles and Design*. Boca Raton, FL, USA: CRC Press, 2014.

[15] P. Goransson, C. Black, and T. Culver, *Software Defined Networks: A Comprehensive Approach*. San Mateo, CA, USA: Morgan Kaufmann, 2016.

[16] *Open Networking Foundation*. Accessed: Mar. 25, 2020. [Online]. Available: https://www.opennetworking.org/

[17] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/jproc.2014.2371999.

[18] H. Wang, S. Chen, H. Xu, M. Ai, and Y. Shi, "SoftNet: A software defined decentralized mobile network architecture toward 5G," *IEEE Netw.*, vol. 29, no. 2, pp. 16–22, Mar. 2015, doi: 10.1109/MNET.2015.7064898.

[19] M. R. Sama, S. B. H. Said, K. Guillouard, and L. Suciu, "Enabling network programmability in LTE/EPC architecture using OpenFlow," in *Proc. 12th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2014, pp. 389–396, doi: 10.1109/WIOPT.2014.6850324.

[20] V.-G. Nguyen and Y. Kim, "Signaling load analysis in openflow-enabled LTE/EPC architecture," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2014, pp. 734–735, doi: 10.1109/ICTC.2014.6983272.

[21] V. Nguyen and Y. Kim, "Proposal and evaluation of SDN-based mobile packet core networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, p. 172, Dec. 2015, doi: 10.1186/s13638-015-0395-1.

[22] A. Jain, S. N S, S. K. Lohani, and M. Vutukuru, "A comparison of SDN and NFV for re-designing the LTE packet core," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2016, pp. 74–80, doi: 10.1109/NFV-SDN.2016.7919479.

[23] J. Page and J.-M. Dricot, "Software-defined networking for low-latency 5G core network," in *Proc. Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, May 2016, pp. 1–7, doi: 10.1109/ICMCIS.2016.7496561.

[24] Y. Kim, J. Gil, and D. Kim, "A location-aware network virtualization and reconfiguration for 5G core network based on SDN and NFV," *Int. J. Commun. Syst.*, vol. 34, no. 2, Jan. 2021, Art. no. e4160, doi: 10.1002/dac.4160.

[25] L. Ma, X. Wen, L. Wang, Z. Lu, and R. Knopp, "An SDN/NFV based framework for management and deployment of service based 5G core network," *China Commun.*, vol. 15, no. 10, pp. 86–98, Oct. 2018, doi: 10.1109/CC.2018.8485472.

[26] F. Eichhorn, M.-I. Corici, T. Magedanz, P. Du, Y. Kiriha, and A. Nakao, "SDN enhancements for the sliced, deep programmable 5G core," in *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2017, pp. 1–4, doi: 10.23919/CNSM.2017.8256006.

[27] N. M. Akshatha, P. Jha, and A. Karandikar, "A centralized SDN architecture for the 5G cellular network," in *Proc. IEEE 5G World Forum (5GWF)*, Jul. 2018, pp. 147–152, doi: 10.1109/5GWF.2018.8516960.

[28] C. N. Tadros, M. R. M. Rizk, and B. M. Mokhtar, "Software defined network-based management for enhanced 5G network services," *IEEE Access*, vol. 8, pp. 53997–54008, 2020, doi: 10.1109/ACCESS.2020.2980392.

[29] *Mininet-WiFi | Emulation Platform for Software-Defined Wireless Networks*. Accessed: Aug. 17, 2020. [Online]. Available: https://mininet-wifi.github.io/

[30] S. Kavanagh. *What is Network Slicing? UK's Premier 5G Resource*. Accessed: Dec. 26, 2018. [Online]. Available: https://5g.co.uk/guides/what-is-network-slicing/

[31] *Ns-3. Ns-3 | a Discrete-Event Network Simulator for Internet Systems*. Accessed: Feb. 28, 2019. [Online]. Available: https://www.nsnam.org/

[32] *Xn General Aspects and Principles*, Standard ETSI TS 138 420, 3GPP TS 38.420, 3GPP, 2018.

[33] M. Salih, N. Jawad, and J. Cosmas, "Software defined selective traffic offloading (SDSTO)," in *Proc. IEEE 23rd Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Sep. 2018, pp. 1–7, doi: 10.1109/CAMAD.2018.8514940.

[34] S. D. A. Shah, M. A. Gregory, S. Li, and R. D. R. Fontes, "SDN enhanced multi-access edge computing (MEC) for E2E mobility and QoS management," *IEEE Access*, vol. 8, pp. 77459–77469, 2020, doi: 10.1109/ACCESS.2020.2990292.

[35] *Welcome to Python.org*. Accessed: Mar. 14, 2020. [Online]. Available: https://www.python.org/

[36] Y. E. Osais, *Computer Simulation: A Foundational Approach Using Python*. Boca Raton, FL, USA: CRC Press, 2017.

[37] *Mininet/Mininet: Emulator for Rapid Prototyping of Software Defined Networks*. Accessed: Jan. 2, 2019. [Online]. Available: https://github.com/mininet/mininet

[38] Open vSwitch. Accessed: Oct. 05, 2020. [Online]. Available: https://www.openvswitch.org/

[39] *Ryu SDN Framework*. Accessed: Mar. 15, 2020. [Online]. Available: https://osrg.github.io/ryu/

[40] 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, Sep. 2015, doi: 10.1049/iet-net.2014.0091.

[41] J. R. Jackson, "Networks of waiting lines," *Oper. Res.*, vol. 5, no. 4, pp. 518–521, Aug. 1957, doi: 10.1287/opre.5.4.518.

[42] L. Kleinrock, *Queueing systems: Theory*, vol. 1. Princeton, NJ, USA: Johir Wileii d Sotis, 1975.

[43] *White Paper: SDN Controller Testing, Part 1. IXIA and NEC*. Accessed: Mar. 2, 2019. [Online]. Available: https://www.necam.com/docs/?id=2709888a-ecfd-4157-8849-1d18144a6dda

[44] C. Metter, S. Gebert, S. Lange, T. Zinner, P. Tran-Gia, and M. Jarschel, "Investigating the impact of network topology on the processing times of SDN controllers," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 1214–1219, doi: 10.1109/INM.2015.7140469.

[45] *OpenDaylight: Open Source SDN Platform*. Accessed: Dec. 26, 2018. [Online]. Available: https://www.opendaylight.org/

[46] 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)*, Sep. 2011, pp. 1–7.

[47] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE mobile core gateways, the functions placement problem," in *Proc. 4th Workshop Things Cellular, Oper., Appl., Challenges AllThingsCellular*, 2014, pp. 33–38, doi: 10.1145/2627585.2627592.

[48] D. Sattar and A. Matrawy, "An empirical model of packet processing delay of the open vSwitch," in *Proc. IEEE 25th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2017, pp. 1–6, doi: 10.1109/ICNP.2017.8117597.

[49] Z. Shang and K. Wolter, "Delay evaluation of OpenFlow network based on queueing model," 2016, *arXiv:1608.06491*. [Online]. Available: http://arxiv.org/abs/1608.06491

[50] C.-L. I, H. Li, J. Korhonen, J. Huang, and L. Han, "RAN revolution with NGFI (xhaul) for 5G," *J. Lightw. Technol.*, vol. 36, no. 2, pp. 541–550, Jan. 15, 2018, doi: 10.1109/JLT.2017.2764924.

[51] G. Hains, W. Suijlen, W. Liang, and Z. Wu, "5Gperf: Signal processing performance for 5G," 2018, *arXiv:1810.11451*. [Online]. Available: http://arxiv.org/abs/1810.11451

[52] D. Gutierrez-Estevez, "Architecture and mechanisms for resource elasticity provisioning," in *5G-MoNArch Project Deliverable, D4.1*, 2018.

[53] X.-F. Tao, Y.-Z. Hou, K.-D. Wang, H.-Y. He, and Y. J. Guo, "GPP-based soft base station designing and optimization," *J. Comput. Sci. Technol.*, vol. 28, no. 3, pp. 420–428, May 2013, doi: 10.1007/s11390-013-1343-3.

[54] J. Wang, J. Xu, Y. Yang, and H. Xu, "GPP based open cellular network towards 5G," *China Commun.*, vol. 14, no. 6, pp. 189–198, 2017, doi: 10.1109/CC.2017.7961374.

[55] G. Pujolle, *Software Networks*. Hoboken, NJ, USA: Wiley, 2015.

[56] T. Norp, "5G requirements and key performance indicators," *J. ICT Standardization*, vol. 6, no. 1, pp. 15–30, 2018, doi: 10.13052/jicts2245-800X.612.

[57] A. Mahmoud, A. Abo Naser, M. Abu-Amara, T. Sheltami, and N. Nasser, "Software-defined networking approach for enhanced evolved packet core network," *Int. J. Commun. Syst.*, vol. 31, no. 1, p. e3379, Jan. 2018, doi: 10.1002/dac.3379.

[58] *Scapy - PyPI*. Accessed: Mar. 15, 2020. [Online]. Available: https://pypi.org/project/scapy/
[59] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Hoboken, NJ, USA: Wiley, 1991.
[60] A. Jain, E. Lopez-Aguilera, and I. Demirkol, "Improved handover signaling for 5G networks," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 164–170, doi: 10.1109/PIMRC.2018.8580757.
[61] S. Abe, G. Hasegawa, and M. Murata, "Design and performance evaluation of bearer aggregation method in mobile core network with C/U plane separation," in *Proc. IFIP Netw. Conf. (IFIP Netw.) Workshops*, Jun. 2017, pp. 1–8, doi: 10.23919/IFIPNetworking.2017.8264829.
[62] W.-K. Jia, "Architectural design of an optimal routed network-based mobility management function for SDN-based EPC networks," in *Proc. 11th ACM Symp. QoS Secur. Wireless Mobile Netw.*, Nov. 2015, pp. 67–74, doi: 10.1145/2815317.2815326.
[63] X. An, C. Zhou, R. Trivisonno, R. Guerzoni, A. Kaloxylos, D. Soldani, and A. Hecker, "On end to end network slicing for 5G communication systems," *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 4, Apr. 2017, Art. no. e3058, doi: 10.1002/ett.3058.
[64] V.-G. Nguyen, K.-J. Grinnemo, J. Taheri, and A. Brunstrom, "On load balancing for a virtual and distributed MME in the 5G core," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 1–7, doi: 10.1109/PIMRC.2018.8580693.
[65] *Open5GCore*. Accessed: Nov. 1, 2020. [Online]. Available: https://www.open5gcore.org/
[66] J. Prados-Garzon, O. Adamuz-Hinojosa, P. Ameigeiras, J. J. Ramos-Munoz, P. Andres-Maldonado, and J. M. Lopez-Soler, "Handover implementation in a 5G SDN-based mobile network architecture," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 1–6, doi: 10.1109/PIMRC.2016.7794936.
[67] L. M. Contreras, L. Cominardi, H. Qian, and C. J. Bernardos, "Software-defined mobility management: Architecture proposal and future directions," *Mobile Netw. Appl.*, vol. 21, no. 2, pp. 226–236, Apr. 2016, doi: 10.1007/s11036-015-0663-7.
[68] T. Bilen, B. Canberk, and K. R. Chowdhury, "Handover management in software-defined ultra-dense 5G networks," *IEEE Netw.*, vol. 31, no. 4, pp. 49–55, Jul. 2017, doi: 10.1109/MNET.2017.1600301.
[69] *Simulink—Simulation and Model-Based Design–MATLAB & Simulink*. Accessed: Nov. 1, 2020. [Online]. Available: https://www.mathworks.com/products/simulink.html

ABDULAZIZ ABDULGHAFFAR received the B.Sc. degree in telecommunication engineering from the Department of Telecommunication Engineering, University of Engineering and Technology (UET), Taxila, Pakistan, in 2016. He is currently pursuing the M.S. degree in computer networks with the Department of Computer Engineering, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. His research interests include computer networks, software defined networks, the IoT, and performance evaluation.

ASHRAF MAHMOUD (Member, IEEE) received the B.Sc. degree in electrical and computer engineering from Kuwait University, in 1990, the M.Eng. degree in engineering physics (computer systems) from McMaster University, Hamilton, ON, Canada, in 1992, and the Ph.D. degree in systems and computer engineering from Carleton University, Ottawa, ON, Canada, in 1997. From 1997 to 2002, he was with the Nortel Networks Research and Development, where he focused on the development and evaluation of radio resource management algorithms for broadband and 3G networks. Since 2002, he has been with the Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. His research interests include performance evaluation and simulation techniques, mobile and wireless sensor networks, and the IoT.

MARWAN ABU-AMARA received the M.S. and Ph.D. degrees in electrical and computer engineering from Texas A&M University, USA, in 1991 and 1995, respectively. From 1995 to 2003, he worked with Nortel Networks Corporation, USA, as a Senior Technical Advisor. Since 2003, he has been with the Department of Computer Engineering, KFUPM, Saudi Arabia. His research interests include cloud and Internet security and resiliency, the IoT, and wireless communications.

TAREK SHELTAMI received the Ph.D. degree in electrical and computer engineering from the Department of Electrical and Computer Engineering, Queens University, Kingston, ON, Canada, in April 2003. He worked with GamaEng Inc. as a Consultant on Wireless Networks from 2002 to 2004. Also, he worked in several joined projects with Nortel Network Corporation. He is currently a Professor with the Department of Computer Engineering, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. He joined the Department on August 2004. Before joining KFUPM, he was a Research Associate Professor with the School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, ON, Canada. His research interests include ad hoc networks, WSAN, the IoT, digitization, computer network security, and performance evaluation. He has been a member of the technical program and organizing committees of several international IEEE conferences.