IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 10, 2020, accepted March 2, 2020, date of publication April 6, 2020, date of current version April 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986132

Software-Defined Networking-Enabled Heterogeneous Wireless Networks and Applications Convergence

HAMADA ALSHAER, (Senior Member, IEEE), AND HARALD HAAS, (Fellow, IEEE)
School of Engineering, Institute for Digital Communications, The University of Edinburgh, Edinburgh EH9 3JL, U.K.

Corresponding author: Hamada Alshaer (h.alshaer@ed.ac.uk)

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) (TOUCAN Project) under Grant EP/L020009/1. The work of Harald Haas was supported in part by the EPSRC through the Established Career Fellowship under Grant EP/R007101/1, in part by The Wolfson Foundation, and in part by the Royal Society.

ABSTRACT A software-defined networking (SDN) architecture is capable of integrating all radio frequency and optical wireless small cell networks (e.g. fifth generation (5G), long-term evolution (LTE) femtocell, wireless fidelity (WiFi), light fidelity (LiFi)) in one network domain. This paper considers a SDN-enabled heterogeneous network (HetNet) comprised of LiFi, LTE femtocell and WiFi access points (APs). The HetNet control plane maintains the state of the network topology and wireless resources, which can support the development of intelligent service provisioning and efficient data communications in x generation (x G) wireless networks. The SDN applications use the network state to provide services in the data plane. However, when the state of network and wireless resources constantly changes, the SDN applications cannot provide reliable and guaranteed services to the wireless user equipments. This paper develops a queuing theoretic framework, which provides a performance evaluation for the SDN-enabled HetNet and applications convergence. A traffic engineering (TE) scheme is developed to support dynamic agnostic downlink flows routing to APs and differentiated granular services across the HetNet. Network and user centric policies are developed to make applications aware of network resource availability on the northbound and southbound interfaces of a SDN controller. Numerical models are introduced to study the impact of the computation and communication resources of northbound and southbound interfaces on the SDN-enabled HetNet scalability and the quality-of-service (QoS) guarantee of applications. Also, simulation scenarios are conducted to evaluate the performance of the TE scheme in provisioning effective and reliable services for subscribers.

INDEX TERMS SDN controller, traffic engineering, heterogeneous wireless networks, QoS, LiFi, VLC, WiFi, LTE, software agents, 5G.

I. INTRODUCTION

Visible light communication (VLC) systems and light fidelity (LiFi) attocellular networks have been technologically enhanced to support high data rate point-to-point (p2p) and multiuser wireless communications [1]. Radio and optical wireless access points (APs) can coexist and operate in a small cell network without causing interference to each other, as shown in Fig. 1. Next generation small cell networks are expected to have more APs and utilize 200x more spectrum than the fourth generation (4G), which can support the unprecedented growth in traffic volume, service diversity and bandwidth-intensive applications [2], [3].

With increasing small cell densification, the distances between users and APs are reduced. More wireless links become available for users to meet their quality-of-service (QoS) requirements [4]. A wireless user equipment (UE) can benefit from the wide service coverage of radio frequency (RF) wireless systems, and enjoy secure high-data rate communications in LiFi attocellular networks. Traffic and user offloading in an integrated long-term evolution (LTE) femtocell/wireless fidelity (WiFi)/LiFi network is viewed as a significant progress towards a true integration of RF and optical wireless technologies. The recent

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

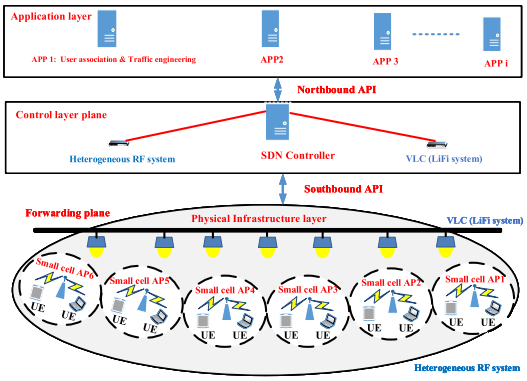

FIGURE 1. A SDN architecture for HetNet and applications convergence.

3rd generation partnership project (3GPP) 5G standard report [5] explains that the non-stand alone (NSA) architecture enables the fifth generation (5G) radio access network (AN) and its new radio (NR) interface to be used in conjunction with the existing LTE infrastructure packet core network. This makes the NR technology available without network replacement, while enabling the provided 4G services to enjoy the capacities offered by the 5G New Radio [5]–[7].

However, the current 3GPP wireless networks integration options are not flexible to enable efficient multi-radio connectivity [8], yet alone the integration of radio and optical wireless technologies. To this end, any non-3GPP access network like LiFi or WiFi should have a termination point at the LTE packet data network gateway (P-GW) to enable UEs route their traffic through an integrated LTE/WiFi/LiFi HetNet, following the mobile internet protocol (IP) principles [9]. The existing 3GPP architecture [8] does not solve the problem of efficient multi-radio and optical wireless networking, because every time the data path is switched from LTE to WiFi or LiFi and vice-versa, some packets are lost, while simultaneous usage is not possible [9].

A software-defined networking (SDN)-enabled heterogeneous wireless network (HetNet) is comprised of LiFi, WiFi and LTE APs, which should be designed to support diverse services with different data traffic profiles. These can be provided by operators and/or trusted third parties. Smart 5G-enabled UEs, which have multiple air interfaces, can be set by preference to communicate through different wireless networks. For example, a UE may be set to download music through the WiFi interface. A UE streams live high definition (HD) videos and establishes secure wireless communications through the LiFi interface. A UE with high mobility makes on-the-move voice over internet protocol (VoIP) calls through the LTE interface. In HetNets, resource availability, users mobility, traffic volume, connection duration and service requirements change constantly. To this end, they should be planned, and resources be dimensioned to support autonomous and adaptive reliable services provisioning to UEs and service providers (SPs) offering virtual networks [10]. They should also support intelligent traffic routing in the user plane to cope with the UE mobility during active communications. A centralized SDN architecture provides a platform that can meet these requirements [11], [12].

The SDN architecture decouples the control plane and the data plane of the HetNet, as shown in Fig. 1 [13]. The network events (e.g. AP failure and services disruption), traffic statistics and payload analysis are presented in information reports that describe the network state. The SDN controller collects it through the southbound and advertises it periodically through the northbound application programming interfaces (APIs) to the application plane [11]. The SDN applications manage, through the controller, OpenFlow (OF) rules in the different APs to support the offered services in the data plane. A fitting traffic engineering (TE) scheme can leverage the centralized view of the network state to support intelligent traffic flows routing and network function virtualization (NFV) modules [14]. For example, it can route traffic flows from failed, congested or underutilized APs to others, which can improve network resource efficiency and support provisioning mission-critical applications [9].

The proposed SDN architecture provides an agnostic network platform which can embody the integration of radio and optical wireless technologies in line with the 5G standardization roadmap [7], [8]. The SDN controller can be interconnected through its east and west interfaces to the 5G core network via a non-3GPP inter-working function (IWF) on the N3 interface (N3IWF). This N3IWF interfaces the 5G core network control plane (CP) and user plane (UP) functions via N2 and N3 interfaces, respectively [6]–[8]. The N2 and N3 reference points are used to connect standalone non-3GPP accesses to 5G core network CP and UP functions, respectively [6]–[8]. This enables the SDN-enabled HetNet to benefit from the full services supported in the first phase of 5G. This paper extends the SDN concepts to manage, in an agnostic manner, LiFi, WiFi and LTE small cells interconnected through the SDN architecture shown in Fig. 1. Note that the SDN control plane implementation for wireless networks is more complex than the wired optical networks [15]. They must account for physical constraints, including channel gain variations, bandwidth availability and granularity, heterogeneous multiuser access and flows routing reconfiguration speed.

This paper evaluates a queueing model that helps to better understand the dynamic of traffic volume on the north and south bound interfaces of the SDN controller and wireless APs in the HetNet data plane. We develop an analytical mathematical framework for modelling the proposed SDN architecture shown in Fig. 1. We are motivated by developing an agnostic TE framework for a SDN-enabled HetNet, which supports autonomous traffic load routing, efficient resource and service resilience provisioning. These help to support a flexible and intelligent integration of the LTE/WiFi/LiFi HetNets. Analytical and simulation analysis have been developed to corroborate the mathematical analysis and the comparison

of the proposed TE routing policies. The convergence of multiple heterogeneous wireless networks enable them to operate seamlessly under the SDN architecture, which is an important enabler for efficient service provisioning.

The remainder of this paper is organized as follows. Section II presents the HetNet system model. Section III states the research problem and challenges in the network, service and application dimensions. Section IV summarizes the main research work in the literature regarding queuing theory for TE in HetNet, SDN-enabled switch and controller modelling. It also summarizes the contributions of this paper. Section V introduces mathematical queuing models for the different planes of SDN-enabled HetNet and discusses the analytical results. Section VI introduces the proposed TE framework modules. Section VII explains the simulation results. Section VIII introduces the key future research challenges. Finally, Section IX concludes this paper and presents the main findings.

II. SDN-ENABLED HetNet SYSTEM MODEL

The SDN-enabled HetNet system model is composed of data (forwarding), control and application planes. A number of SDN-enabled heterogeneous LiFi, WiFi and LTE wireless APs are configured in the data plane. They are managed by a SDN controller that runs in the control plane. An orthogonal frequency-division multiple access (OFDMA) protocol allocates the available resources on the downlink channels of LiFi and LTE APs. A wireless resource is a slot that occupies a space in the time and frequency domains. It is the smallest resource unit which can be allocated to a UE. An average number of slots, $\overline{S}$, are available for data transmission in the downlink part of each time frame, T_F. The wireless channel states are represented by a number of modulation and coding schemes (MCS) i, $1 \leq i \leq I$. A UE uses the MCS_i to transmit a number of bits, b_i, per slot. At each frame time-step, T_F, a UE has the probability, p_i, to use the MCS_i.

The software switch, Open vSwitch [16], runs in the APs, which makes them OpenFlow (OF) enabled switches. They are connected to the controller through a tree topology via an OF-enabled switch, as shown in Fig. 1. A set of northbound representational state transfer (REST) APIs enable communications between the application plane and the control plane, which support various applications. For example, users association and traffic engineering, security, access control, network resilience, etc., as shown in Fig. 2. Similarly, a set of southbound open APIs (e.g., OF protocol) enables communications between the data plane and the control plane, which supports the reporting of a global HetNet view (i.e. the state of resources in the data plane and network topology connectivity) and setting traffic forwarding rules in the routing tables of AP switches. Whenever AP switches receive new flows, it requests the controller to set forwarding rules in the switches that will handle them [17].

The downlink channel gains of wireless APs vary in time and space, which justify the assumption of the general (G) service distribution of the APs. Each small cell has a single AP to serve the UEs. Each AP switch has a buffer of size K packets. The arrivals of UEs and downlink flows follow a Markovian (M) Poisson distribution. Thus, an M/G/1/K queuing model [18] can be adopted to describe the networking operations of AP switches in the data plane, as shown in Fig. 2. As the controller manages N switches in the data plane, an M/G/1/K/N queuing model can be adopted to describe the traffic packets handling operations on the southbound interface of the SDN controller. The traffic flows that arrive at the switch follow a Poisson process with arrival rate, λ_s. The service times at the AP switches are assumed to follow an exponential distribution with an average of $\frac{1}{\mu_s}$. The service time at the SDN controller is assumed to follow an exponential distribution with average rate, $\frac{1}{\mu_c}$. A traffic flow with a pre-set rule in the AP switch is served and forwarded with a probability, $\overline{\phi_s} = 1 - \phi_s$. However, if a traffic flow does not have a rule set in the AP switch, a packet-in is sent to the SDN controller, with a probability, ϕ_s, to define its forwarding rule in the AP switch. The input traffic rate from the AP switch to the controller and vice-versa can be expressed statistically as, $\phi_s\lambda_s$, and $\psi_s\phi_s\lambda_s$, respectively. Note that ψ_s denotes the probability of the traffic without preassigned rules are successfully assigned rules by the controller.

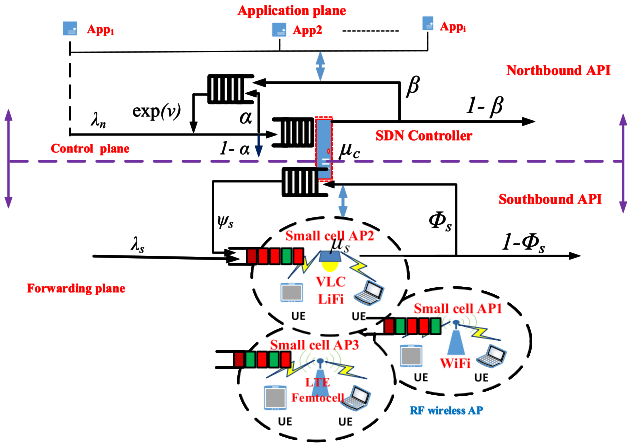

FIGURE 2. SDN-enabled HetNet architecture and applications convergence modelling.

Some of the SDN applications, which need to use the same resource advertised as available in the current network state, may try again to access it in the next round of network state advertisement. The retrial access happens due to a huge volume of requests generated from the SDN applications on the northbound interface; or resource insufficiency and unavailability in the data plane. This happens when the applications require constant networks state update and simultaneous increase of network resources. To this end, the M/M/1 retrial queue with geometric loss and feedback [19] can be adopted to model the access of applications to the data plane through the northbound interface of the SDN controller. The arrival rate of applications requests sent through the northbound interface is denoted by λ_n. The requests of applications which cannot be admitted at the controller join a retiral queue with a

retrial probability, α, or dropped with a probability, $\overline{\alpha} = 1-\alpha$. After a random amount of time, the requests in the retrial queue retry again following an exponential distribution with average rate $\frac{1}{\nu}$ to access the network resources. An application that requires a single service leaves the SDN controller with a probability ($\overline{\beta} = 1 - \beta$). If it requires further services like an increase of bandwidth or other resources, it rejoins the retrial queue with a probability β.

III. PROBLEM STATEMENT

Network function virtualization (NFV) orchestration, user access and traffic engineering applications provide services across the HetNet to support the QoS and quality-of-experience (QoE) requirements of users. These run applications on UEs that request services from external endpoints like an edge computing or a cloud. The developed network access and traffic engineering application manages users association to APs and resource allocation to user applications in an efficient-manner. It takes into consideration user priority, HetNet resource granularity and allocation efficiency. When the number of deployed APs and active users increases in the network, it becomes more challenging for the control plane to apply fast forwarding rules and support concurrent applications access to the available resources in the data plane. Note that every service traffic flow requires proper forwarding rules to be set up in the APs that serve the users requested it. For example, suppose that there are user association and traffic engineering, and user localization (tracking) applications running in the SDN application plane. The former requires information regarding the resources availability to engineer users association to APs and resource allocation to users applications, whereas the latter requires information regarding the users in the network. However, both applications compete for the available network resources they need to transport their traffic back and forth among servers, SDN controller and users. During the next state collection time period, the SDN controller advertises on the northbound interface the network state in terms of the available bandwidth on the downlink channel of AP_i. Based on the received network state, both applications make attempts to request the controller to configure forwarding rules in the AP_i to use the same available bandwidth resource. However, this may no longer be enough to support both of them. In this case, one of the applications needs to retry again during the next state collection time period to access the same resource in the forwarding plane. To this end, the constant changes in the global state of the HetNet make it challenging for the SDN applications to provide consistent and ubiquitous services to their UEs in the data plane. A research challenge lies in developing a mathematical model that can evaluate the consolidation of the control plane in a centralized SDN controller and its impact on the network performance and scalability. Moreover, the HetNet has APs with different transmission ranges, service coverages and data rates. The UEs have various connection preference trade-offs among the bandwidth, latency, throughput and seamless connectivity. The centralized controller collects the channel state information, QoS requirements of users and load state of APs. A research challenge lies in developing user association to APs and agnostic traffic flow routing algorithms. These should run on any AP, independent of its underlying wireless technology, while meeting the service requirements of user applications.

IV. RELATED WORK

The operations underlying the OF switch, SDN controller and their interconnections have been investigated by researchers to develop mathematical models that can evaluate the performance and scalability of SDN-enabled networks and services provisioning. The following subsections discuss the main developed models for the SDN switch, controller, architecture and software-defined wireless networks.

A. SWITCH AND CONTROLLER MODELLING

An OF switch should realise fast look up tables that match flows to forwarding instructions. The SDN controller should have enough processing capacity to handle all the messages coming from the OF switches on the southbound interface and the applications running on the northbound interface. The controller handles a large number of packets per second just to monitor the network state. This is not just a burden on the controller, but it also increases the load on the network control plane. A software Open vSwitch [16] can run with different hardware and operating systems, which support scalable OF operations in network switches. An OF switch is implemented on a networking field-programmable gate array (NetFPGA) platform [20]. It is used to conduct performance analysis on forwarded traffic flows. An OF switch [21] is deployed on a Linux platform to evaluate the performance of multiple forwarding switches and compare switching and routing flows at layers 2 and 3, respectively. A modular and parametrised implementation of a hardware-based OF switch is implemented on three different platforms: NetFPGA, FPGA ML605 and DE4 models. The performance evaluation of these platforms shows that the OF switch design can be implemented across them with minor performance variations [22].

In [23], the authors used a network calculus-based approach [24] to derive upper bounds for the buffer length of a single OF switch and controller. The upper bounds for the service busy period of the switch and controller were derived to calculate the worst delay of packet processing. However, the derived bounds are deterministic and apply only to an OF switch configured in fixed networks. They cannot be used to derive an end-to-end delay bound for traffic flows traversing multiple OF switches. In [17], the authors modelled an OF switch and controller as an M/M/1 and an M/M/1/S queuing systems, respectively, where S denotes the controller queue length on the southbound interface. The developed model considers a bidirectional feedback between the switch and controller.

In [25], the authors proposed an approach that identifies the elephant flows based on some SDN information (e.g. flows, packets or bytes counters) on the different ports of network switches. This research study is useful to understand the relationship between the collected traffic statistics and the OF rules that need to be configured in the OF switches to handle long and short traffic flows. However, the proposed approach may not support a large-scale data centre. Note that a single OF rule can be configured in an OF switch to handle an aggregate of traffic flows shaped through a leaky bucket controller (LBC) at the network edges to receive the same forwarding services. In [26], the authors developed a network calculus-based admission control mechanism that ensures enough allocation of bandwidth and buffer space for complied traffic flows. These are serviced by the switches along a path set up by a SDN controller. However, the buffer and bandwidth allocation are based on a loose bound, which cannot accurately quantify the bandwidth for the different flow aggregates. In [27], the authors considered a Jackson network composed of some OF switches. They evaluated the performance of traffic in the data plane under finite and infinite buffer size scenarios at the switches. The packets, which are sent by the controller to install OF rules in the switch, cannot be sent back to it again. In [17], the authors developed an approximate model for the operations underlying the southbound interface. It does not distinguish between the external traffic and the packets sent by the controller to the switch. The packet-in requests for setting rules in the switches are modelled as a batch of flows, with size X, arriving at the controller queue, which is modelled as an $M^{[X]}$/M/1 queuing model [27]. This work applies mostly to fixed networks. It also does not consider link dynamics in the forwarding plane. Most of the above developed SDN models consider a single switch in the data plane interconnected with a controller, which require further analysis to investigate the support of real-time applications. Also, none of the developed models [20]–[22] provides a comprehensive and generic framework for evaluating the performance of SDN-enabled wireless networks. The developed queuing model helps to evaluate the impact of packet arrival rate and flow size of external traffic on the performance of the controller and switch. However, more research work is still needed to understand the performance of a SDN controller that manages multiple AP switches in a wireless network.

B. SOFTWARE-DEFINED WIRELESS NETWORKS ARCHITECTURE

SDN architectures and solutions are increasingly adopted in the context of wireless networks to enable network programmability and support mobile operators to manage resource allocation and service provisioning to end users. In [13] the authors proposed a software defined radio access network (SoftRAN) architecture that encapsulates all the base stations in a radio access network (RAN) represented as a virtualized big base station. This centralizes the control and management of RAN. The paper only explains the benefits of this architecture to support applications and use-cases in wireless networks. In [28] the authors discuss the capabilities of 3GPP network management and evolved-UMTS (Universal Mobile Telecommunications Service) Terrestrial Radio Access Network (E-UTRAN) architectures to support self-organizing wireless networks, automation, mobility and load balancing. This is an overview paper, which only explains self-organizing networks (SON) 3GPP standardization activities. It discusses the 3GPP architecture, without explaining how this can support small cell wireless networks integration.

A SDN architecture can be used to interconnect data centres to support traffic engineering functionalities in ultra-dense networks (UDNs). The virtual cell technology has been widely used to improve the user experience of UDNs. By adopting the virtual cell technology, a new type of UDN, termed user-centric UDNs, could be developed [29]. The main idea of user-centric UDNs is to cluster APs dynamically to provide better services for end-users. Based on the virtual cells in UDNs, a new type of beam forming technology named the balanced beam forming algorithm was developed to optimize the network capacity. In [30] the authors summarize the main challenges in dense user-centric cloud random access networks (C-RAN). They advocate the application of dense user-centric C-RAN approaches to UDNs using some cloud computing techniques.

In [31] the authors developed an analytic model for evaluating the queueing delays and channel access times at nodes in 802.11 based WiFi networks. Each node is modelled as a discrete time G/G/1 queuing system model, which is used to derive closed form solutions to obtain the values of the delay and queue length. In [9] the authors proposed a high-level architecture for tight integration of WiFi-LTE systems. The idea is to integrate WiFi into a viable 3GPP architecture. A Markov chain model framework was developed to analyse the performance of the proposed LTE-WiFi UDNs. It provides some insights regarding the gained benefits in terms of average data rate per user and service coverage in the considered area. In [3] the authors provide a comparison study for different AP densities with various bandwidth in an UDN. The paper [3] proposes an approach for increasing the average user capacity in an UDN to 1 Gbps.

The support of SDN for control and forwarding isolation for traffic flows and the centralized network management provides a platform for realizing traffic engineering based on traffic measurement and management [32]. The integration of SDN in legacy wired networks can improve traffic engineering [33]. In [34] the authors proposed a dynamic cell specific uplink/downlink (UL/DL) frame reconfiguration mechanism, which can customize the TD(time division)-LTE frame ratio to efficiently meet the QoS requirements of UEs. This research study only considers the local state of frame and traffic load in cells, without considering the network state. In [35] the authors demonstrated the capability of a SDN management algorithm to re-direct traffic flows to mobile femtocell, RF AP or LTE eNodes. The obtained results show

some improvement in the user experience, which is quantified in the increase of their service satisfactions, throughput and the reduction of waiting times. The algorithm also improves cellular resource utilization through offloading traffic flows to WiFi and mobile femtocells. It ensures load balancing based on flow admission control to each supported traffic technology.

The above related work handled the research problems from different angles, though more analysis is required to investigate services provisioning performance and applications support in the network. The related studies are focused on some entities in the proposed SDN architecture, whereas a robust SDN solution should consider the different SDN planes. The previous developed approaches propose solutions that cannot run in an agnostic manner, independent of the wireless technologies underlying the different APs in the SDN-enabled HetNet.

C. CONTRIBUTIONS

The contributions of this paper can be described as threefold. The first uses queuing theory principles and models to develop a mathematical framework for modelling the inter-planes of the SDN-enabled HetNet and applications convergence. The second fold introduces a TE scheme that runs at the network and medium access layers. Routing policies are developed to support TE and load balancing on the network layer and multiple services on the medium access layer. The proposed TE scheme handles the dynamics of the SDN-enabled HetNet through enabling the AP switches to dynamically serve flows per frame basis. The SDN controller exposes appropriate feedback information for applications to improve the QoS and QoE of their UEs. The third fold develops network and service provisioning automation mechanisms, which can support service resilience, dynamic APs selection and resource allocation for UEs.

V. INTER-PLANE INTERFACES MODELLING

The SDN controller has horizontal (i.e. east, west) and vertical (i.e. northbound, southbound) interfaces, though, in this paper, we will focus on modelling the vertical interfaces. A buffer on the northbound interface keeps and passes the requests generated from the SDN applications to access the data plane. Likewise, a buffer on the southbound interface passes the packet-in rules and incoming packets to the controller and switches, respectively, as shown in Fig. 2. The following subsections introduce mathematical queuing models for the northbound and southbound interfaces, and data plane switches.

A. NORTHBOUND INTERFACE MODELLING

The SDN applications view the state of resources in the HetNet data plane through the northbound interface of the control plane. They offer services via the centralized SDN controller by requesting to set up their forwarding rules in the data plane switches through a southbound protocol (e.g. OpenFlow). The applications may renege on accessing the data plane, because of resource unavailability or controller processing capacity constraint. This reneging process influences the performance of network services provided throughout the HetNet data plane.

Based on the HetNet system model, the SDN applications generate requests following a Poisson process with arrival rate, λ_n. The M/M/1 queuing model describes the packets buffering and processing at the northbound interface of the controller. We adopt the M/M/1 retrial queueing system model with geometric loss and feedback [19] to model the requests processing for data plane access through the northbound interface, as shown in Fig. 2. The state of the SDN-enabled HetNet is described by a pair $(\zeta(t), N(t))$, where $\zeta(t)$ denotes the number of busy controllers and $N(t)$ denotes the number of requests in the retrial buffer at time t. A stochastic process $(\zeta(t), N(t)) : t \geq 0)$ is formed as a time-homogeneous Markov process with a state space (ζ, N) as a limiting variable of $(\zeta(t), N(t))$. When the controller receives a small number of requests from the network applications to access the data plane, the average queue length of the controller is given as follows [19]:

$$
\begin{aligned}
&E(N : \zeta = 0) \\
&= C\Bigg[\frac{\alpha(\lambda_c + v) + \beta\mu_c}{v} \\
&\quad \times F\left(\frac{\alpha(\lambda_c + 2v) + \beta\mu_c}{\alpha v}; \frac{\overline{\alpha}(\lambda_c + v) + \overline{\beta}\mu_c}{\overline{\alpha} v}; \frac{\alpha\lambda_c}{\overline{\alpha} v}\right) \\
&\quad - \alpha F\left(\frac{\alpha(\lambda_c + v) + \beta\mu_c}{\alpha v}; \frac{\overline{\alpha}(\lambda_c + v) + \overline{\beta}\mu_c}{\overline{\alpha} v}; \frac{\alpha\lambda_c}{\overline{\alpha} v}\right) \\
&\quad - \frac{\alpha\lambda_c(\lambda_c + v) + \beta\lambda_c\mu_c}{v(\overline{\alpha}(\lambda_c + v) + \mu_c\overline{\beta})} \\
&\quad \times F\left(\frac{\alpha(\lambda_c + 2v) + \beta\mu_c}{\alpha v}; \frac{\overline{\alpha}(\lambda_c + 2v) + \overline{\beta}\mu_c}{\overline{\alpha} v}; \frac{\alpha\lambda_c}{\overline{\alpha} v}\right)\Bigg],
\end{aligned}
$$

where $F(a; b; w)$ is the Kummer's function; and C is a normalizing constant, given by [19]:

$$
\begin{aligned}
C = &\Bigg[\frac{\mu_c\overline{\beta} + \lambda_c\overline{\alpha}}{\lambda_c} \\
&\times F\left(\frac{\alpha(\lambda_c + V) + \beta\mu_c}{\alpha V}; \frac{\lambda_c\overline{\alpha} + \mu_c\overline{\beta}}{\overline{\alpha} V}; \frac{\alpha\lambda_c}{\overline{\alpha} V}\right)\Bigg]^{-1}. \qquad (1)
\end{aligned}
$$

However, when the number of requests generated from the SDN applications increases, the controller may become busy on its northbound and southbound interfaces. In this case, the average queue length of the northbound interface is expressed as follows [19]:

$$
\begin{aligned}
&E(N : \zeta = 1) \\
&= C \cdot \frac{\alpha\lambda_c(\lambda_c + v) + \beta\lambda_c\mu_c}{(\overline{\alpha} v(\lambda_c + v) + \mu_c\overline{\beta} v)} \\
&\quad \times F\left(\frac{\alpha(\lambda_c + 2v) + \beta\mu_c}{\alpha v}; \frac{\overline{\alpha}(\lambda_c + 2v) + \overline{\beta}\mu_c}{\overline{\alpha} v}; \frac{\alpha\lambda_c}{\overline{\alpha} v}\right). \qquad (2)
\end{aligned}
$$

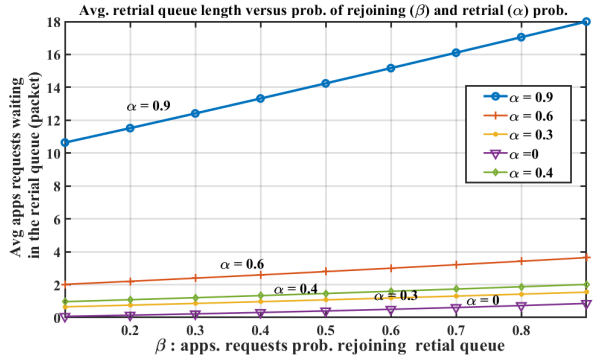

FIGURE 3. Average retrial queue length versus β.

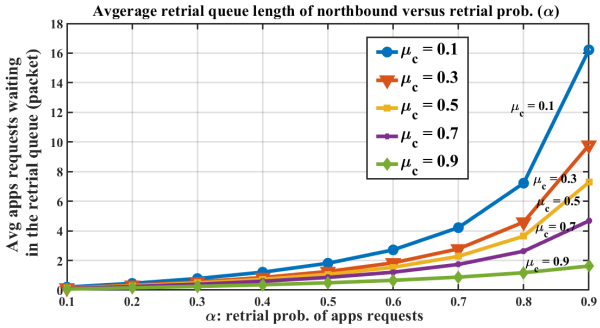

FIGURE 4. Average retrial queue length versus α.

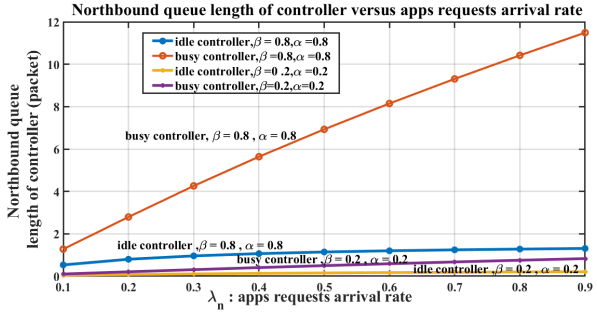

FIGURE 5. Average queue length of northbound controller versus λ_n.

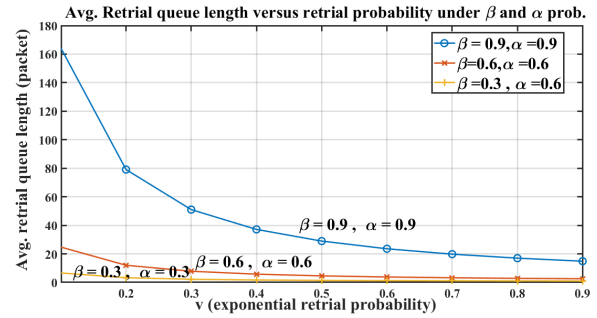

FIGURE 6. Average retrial queue length versus v.

When the SDN controller runs beyond its capacity, the network is down or resources in the data plane are overbooked, the excess requests generated from the SDN applications are kept in the retrial queue or are dropped. In this case, the average length of the retrial queue is expressed as follows [19]:

$$\begin{aligned} &E(N) \\ &= E(N:\zeta=0)+E(N:\zeta=1) \\ &= C\Bigg[\frac{\alpha(\lambda_c+v)+\beta\mu_c}{v} \\ &\quad\times F\left(\frac{\alpha(\lambda_c+2v)+\beta\mu_c}{\alpha v};\frac{\overline{\alpha}(\lambda_c+v)+\overline{\beta}\mu_c}{\overline{\alpha}v};\frac{\alpha\lambda_c}{\overline{\alpha}v}\right) \\ &\quad-\alpha F\left(\frac{\alpha(\lambda_c+v)+\beta\mu_c}{\alpha v};\frac{\overline{\alpha}(\lambda_c+v)+\overline{\beta}\mu_c}{\overline{\alpha}v};\frac{\alpha\lambda_c}{\overline{\alpha}v}\right)\Bigg]. \end{aligned} \tag{3}$$

Analytical models have been developed in MATLAB, which evaluate (1), (2) and (3). They give some indications regarding the relationship among the applications requests, queue length of the northbound interface, retrial queue length and controller processing rate. Based on (3), the request retrial probability, α, influences the retrial queue length more than the rejoining probability, β, as shown in Fig. 3. This is attributed to the fact that not all the rejoining requests wait in the retrial queue length. The applications, which are allocated resources in the previous round, are quickly served by the controller. Based on (3), the controller service rate, μ_c, directly influences the retrial queue length. this even grows more rapidly with the increase of the requests retrial probability, α, as shown in Fig. 4. This also happens when the controller is busy or requested resources in the network data plane become unavailable during the access time of applications. This emphasises the importance of the controller processing rate and requested resource availability for applications to provide reliable services in the HetNet data plane. Based on (1), the retrial and rejoining probabilities significantly impact the queue length of the northbound interface, as shown in Fig. 5. When the exponential probability, v, of sending requests to the controller increases, it is obvious to see that the retrial queue length decreases. However, based on (2) and (3), the queue length of the northbound interface still depends on the controller processing rate, retrial and rejoining probabilities, as shown in Fig. 6. When the rejoining and retrial requests decrease, the average retrial queue length decreases as well and vice-versa. This case indicates that the controller can manage the allocation of resources that are requested by the different SDN applications. New flow rules can also be set in the APs based on the retrial and rejoining probabilities range.

B. SOUTHBOUND INTERFACE MODELLING

As mentioned before, the SDN applications can manage the data plane traffic through requesting the controller to configure their OF rules in the APs to deliver network services. Thus, the APs and controller may become a bottleneck for provisioning services in the data plane, where traffic flows

wait for a longer time in the queue of APs to set up their OF rules. The queuing model, M/G/1/K/N, which captures the operations of the SDN controller on the southbound interface, is proposed to investigate the impact of the buffer size, number of switches and traffic flow rate on the network performance. It is used to evaluate the queue length and response time of the controller and AP switches under traffic load.

The external traffic load rate, λ_s, and the traffic load rate of SDN applications, λ_n, represent the total traffic load rate at the southbound interface of controller, which also follows a Poisson distribution with a traffic arrival rate, expressed as: $\lambda_c = \lambda_n + \phi_s \lambda_s$. The traffic load intensity at the controller is expressed as: $\rho_c = \frac{\lambda_c}{\mu_c}$. When a new flow arrives at an AP that does not have a correspondent forwarding rule, an in-packet request is sent to the controller to define the forwarding rule. However, if the in-packet request arrives, where the controller has already K packets in its buffer, the $K+1$ packet is dropped. A stochastic relationship between the distribution of the controller queue at an arbitrary time and an arrival time of an in-packet rule request is used to describe a Markov process [18]. This is used to derive the relationship formulas for the blocking probability, throughput and response time. Let P_{B_c} denotes the probability that an arriving in-packet rule request is blocked by the controller. It (P_{B_c}) can be expressed as follows [36]:

$$P_{B_c} = \frac{\rho_c (N-K) P_K}{(1-P_o) + \rho_c (N-K) P_K}, \tag{4}$$

where P_K denotes the probability that there are K in-packet requests in the controller queue at an arbitrary time. And the probability P_o denotes the queue of the controller is empty at an arbitrary time, given by [37]:

$$P_o = \frac{\rho_c - 1}{\rho_c^{2((1+\sqrt{\rho_c}s^2-\sqrt{\rho_c}+k)/(2+\sqrt{\rho_c}s^2-\sqrt{\rho_c}))} - 1}. \tag{5}$$

And P_K is expressed as follows [37]:

$$P_K = \frac{\rho_c^{((1+\sqrt{\rho_c}s^2-\sqrt{\rho_c}+K)/(2+\sqrt{\rho_c}s^2-\sqrt{\rho_c}))}(\rho_c - 1)}{\rho_c^{2((1+\sqrt{\rho_c}s^2-\sqrt{\rho_c}+K)/(2+\sqrt{\rho_c}s^2-\sqrt{\rho_c}))} - 1}, \tag{6}$$

where $s^2 \approx 1/2$ denotes the squared coefficient variation of the service process. L_c denotes the average number of in-packet requests in the controller queue at an arbitrary time, which is expressed as follows [37]:

$$L_c = N - \frac{1-P_o}{\rho_c} - (N-K)P_K. \tag{7}$$

And T_c denotes the average response time from the arrival to service time completion, which is expressed as follows [18]:

$$T_c = \frac{1}{\lambda_c} - \frac{(N-K)\mu_c P_K}{1-P_o}. \tag{8}$$

And γ_c denotes the controller throughput in terms of the number of processed in-packet requests per unit of time, and given by [18]:

$$\gamma_c = \frac{1-P_o}{\mu_c}, \tag{9}$$

where μ_c denotes the average service time of controller.

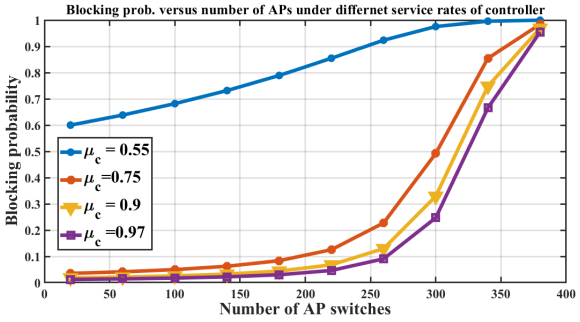

FIGURE 7. Southbound controller blocking probability versus number of switches.

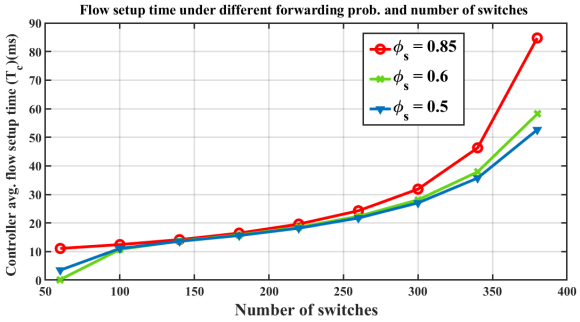

FIGURE 8. Controller average flow set up time versus number switches.

As a result, based on (4), the probability that a traffic flow be blocked by the controller increases when the number of AP switches increases. But, it decreases, when the controller service rate increases, as shown in Fig. 7. The controller service rate influences the blocking probability of new traffic flows, that arrive at the AP switches, without pre-assigned forwarding rules. To this end, the capacity of SDN controller in terms of processing rate and buffer space should be carefully dimensioned to support both the SDN applications and AP switches. Based on (8), the flow set up time for traffic flows, that arrive at the APs without pre-assigned forwarding rules, increases in terms of the in-packet rule requests sent to the controller, as shown in Fig. 8. This becomes more obvious when the number of AP switches exceeds a critical value. For example, when the number of switches exceeds 200, the flow set up time and blocking probability start to grow rapidly, as shown in Fig. 8 and Fig. 7. This means that the controller needs an effective mechanism that can proactively assign flow rules in the AP switches for traffic flows. This significantly reduces the controller set up time, which shortens the sojourn time of traffic packets arrive at the AP switches. The average sojourn time for packets that arrive at an AP, in the HetNet, is defined as the expected amount of time to wait before they

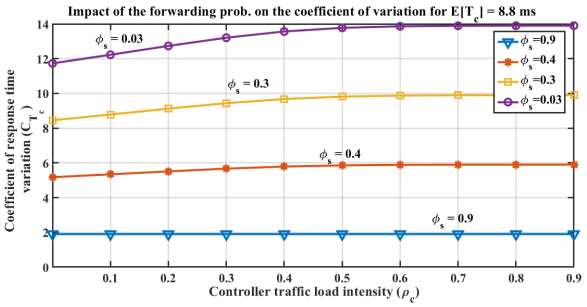

FIGURE 9. Controller response time variation versus traffic load.

are handled by the AP. This mainly includes the controller flow set up time. Based on (8), the coefficient of response time variation, C_{T_c}, for the sojourn times is dependent on the traffic flows, that arrive without pre-assigned flow-rules, at the southbound interface of the SDN controller, as shown in Fig. 9. The coefficient of response time variation (C_{T_c}) is defined as the ratio of the standard deviation, σ, to the average of controller service (response) time (E[T_c]), namely: $C_{T_c} = \frac{\sigma}{E[T_c]}$. It is calculated for an average controller service time value of E[T_c] = 8.8 ms. When the traffic load, arriving without pre-assigned flow-rules, increases at the AP switches, the coefficient of response time variation decreases. With a smaller ϕ_s, less traffic packets are subject to the delay imposed by the controller and therefore the deviation from the average value for these packets is much higher, as shown in Fig. 9. The more traffic flows require flow-rules assignment in the APs by the controller, the more traffic flows have a longer sojourn time caused by the controller service time and queueing waiting time. The controller response time influences the total packet sojourn time more than the AP switches [17].

There is not a clear approach to identify the amount of resources (buffer space and channel bandwidth) required by a flow to guarantee its performance requirements, except for a peak bandwidth assignment. Here, the idea is to dynamically allocate a sufficient capacity to traffic flows served by the AP switches. These provide bandwidth based on the wireless channel conditions. Thus, the M/G/1/K queuing system model provides a good approximation for buffering packets at the southbound buffer and the users access to the wireless channels of AP switches in the data plane.

An embedded Markov chain of the M/G/1/K queue is developed, in which the embedded points are selected immediately after each packet transmission completion. A switch state is defined at a particular time instant as the number of packets in its queue at that time. Then, the considered switch has $K + 2$ states from state 0 to state $K + 1$, including K waiting packets in the queue and one packet being transmitted. However, when observing the switch immediately after each packet transmission completion, the utilized embedded Markov chain has only $K + 1$ states from state 0 to state $K + 1$ [36].

The packet blocking probability, P_{B_s}, is the probability that there are K packets waiting in the buffer of the AP switch, which is given by [37]:

$$P_{B_s} = P[N = K] = \frac{(1 - \rho_s)\rho_s^K}{(1 - \rho_s^{K+1})}, \tag{10}$$

where ρ_s denotes the traffic load intensity, expressed as, $\rho_s = \frac{\lambda_s}{\mu_s}$. The switch throughput, γ_s, represents the number of packets that are completely served when the AP switches reach an equilibrium point, given as follows [37]:

$$\gamma_s = \lambda_s(1 - P_{B_s}). \tag{11}$$

The response time of packets served in the AP switches is given as follows [37]:

$$T_s = \frac{\rho_s + \rho_s^{K+1}(1 + K\rho_s - K)}{\lambda_s(1 - \rho_s^{K+1})(1 - \rho_s)}. \tag{12}$$

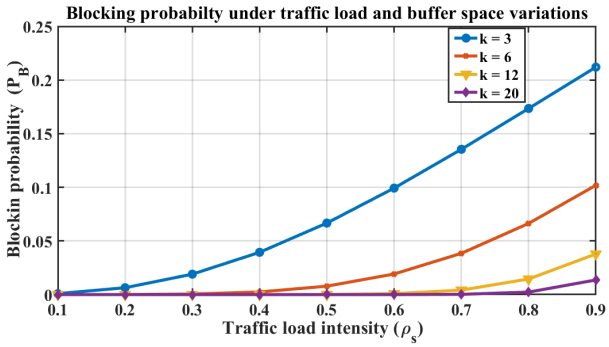

FIGURE 10. AP switch blocking probability versus traffic load intensity.

As a result, based on (10), the blocking probability at the AP switches increases in terms of the traffic load intensity and decreases when the buffer size gets larger, as shown in Fig. 10. This reflects the importance of ensuring a proper buffer size at the different AP switches, because it can control the number of admitted downlink traffic flows. Based on (12), the AP switches continue to serve flows with minimal response time, irrespective of the buffer size until the traffic load intensity exceeds a specific value, as shown in Fig. 11. For example, after the load intensity exceeds 0.6, the impact of buffer size starts to become clearer on the switch response time, as shown in Fig. 11. The throughput of AP switches has an impact on the QoE of UEs. When the QoS offered to the UEs is improved, this enhances the offered service. For example, based on (11), despite the increase in traffic load intensity, the buffer size remains a key in enhancing the satisfaction of users about the received throughput, as shown in Fig. 12. This is reflected in the data rates of services offered to the UEs.

VI. DYNAMIC TRAFFIC ENGINEERING SCHEME

The available resources on the downlink channels of the APs can be viewed as virtual wireless paths that provide data connectivity to the UEs, as shown in Fig. 13 [38]. Four service classes are proposed to serve traffic flows generated from

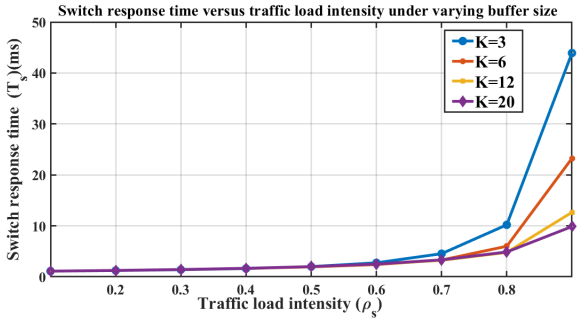

FIGURE 11. AP switch response time versus traffic load intensity.

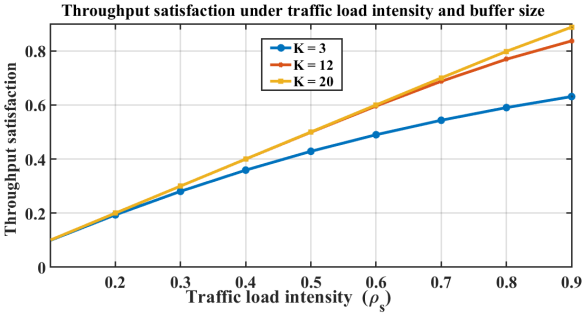

FIGURE 12. AP switch throughput versus traffic load intensity.

user applications, namely: Deterministic guaranteed service (DGS), delay-sensitive service (DSS), throughput sensitive service (TSS) and best effort service (BES). The DGS class is offered to traffic flows that have a constant bit-rate requirement. The DSS class is offered to traffic flows that have a stringent delay requirement. The TSS class is offered to traffic flows that have a minimum throughput requirement but tolerate some delay bound violation. The BES class is offered to traffic flows that have no particular QoS guarantees. The software agents, that run in the AP switches, compute the statistical information of downlink traffic flows and monitor the available bandwidth of wireless interfaces and queue length. They also determine the channel conditions of UEs and their MCS values. Local and generic network channel gain and information matrices are formed and updated periodically. The local matrix records the information of the UEs that can be only associated with a single AP. Whereas, the generic matrix records the information of the UEs that have a choice to be associated with one AP from a set of APs. The local matrix, $\mathbf{\Gamma}_{\boldsymbol{l}}$, records the users' tag, flow class, SNR, data rate of UEs and AP mark. The generic matrix, $\mathbf{\Gamma}_{\boldsymbol{g}}$, records the users' tag, flow class, SNR, data rate of UEs and potential APs which can serve them. A downlink traffic flow is tagged with an identifier of the UE requesting it. A row in $\mathbf{\Gamma}_{\boldsymbol{l}}$ and $\mathbf{\Gamma}_{\boldsymbol{g}}$ has six elements, which are defined in this order: user's tag, AP mark, SNR value, data rate value, flow class and number. These information matrices are used by traffic routing and scheduling modules running under the proposed TE scheme. The traffic routing module routes flows to the different

Algorithm 1 Traffic Engineering Scheme

Input : Local and generic matrices, $\mathbf{\Gamma}_{\boldsymbol{l}}$ and $\mathbf{\Gamma}_{\mathrm{g}}$
Output: 1. Network layer: Downlink flows routing to APs, $\mathbf{\Gamma}_{\mathrm{n}}$
Output: 2. Mac layer: traffic packets scheduling, $\mathbf{\Gamma}_{\mathrm{c}}$
Output: 3. Network-Mac layer: Flow routing and traffic adaptation, $\mathbf{\Gamma}_{\mathrm{nc}}$
Initialize: Maximum number of users and flows per class i. Number of slots and frame duration per AP switch, $(\overline{S}, T_{\mathrm{f}})$;
$t \leftarrow 0$;
$\mathbf{\Gamma}_{\mathrm{n}} \leftarrow 0$; $\mathbf{\Gamma}_{\mathrm{c}} \leftarrow 0$; $\mathbf{\Gamma}_{\mathrm{nc}} \leftarrow 0$;
while *true* **do**
  The SDN controller receives updated $\mathbf{\Gamma}_{\boldsymbol{l}}$ and $\mathbf{\Gamma}_{\mathrm{g}}$;
  $y \leftarrow$ size ($\mathbf{\Gamma}_{\mathrm{g}}$); y(1,1) gives the number of rows ;
  $z \leftarrow$ size($\mathbf{\Gamma}_{\boldsymbol{l}}$); z(1,1) gives the number of rows ;
  $\mathbf{\Gamma}_{\mathrm{n}}(:,:) \leftarrow \mathbf{\Gamma}_{\boldsymbol{l}}(:,:)$;
  Routing on the networking layer;
  for $k \leftarrow 1$ *to* $y(1,1)$ **do**
    class $\leftarrow \mathbf{\Gamma}_{\mathrm{g}}(\mathrm{k},5)$;
    if class $\in$ DGS, apply smallest delay routing polciy ;
    if class $\in$ DSS, apply minimum network delay routing polciy;
    if class $\in$ TSS, apply maximum throughput policy;
    if class $\in$ BES, apply probabilistic routing policy;
    $\mathbf{\Gamma}_{\mathrm{n}}(\mathrm{z(1,1)+k},:) = \mathbf{\Gamma}_{\mathrm{g}}(\mathrm{k},:)$;
  end
  $\mathbf{\Gamma}_{\mathrm{n}}$;
  Scheduling traffic on the MAC layer per AP ;
  for $v \leftarrow 1$ *to* T_f **do**
    $\overline{S}_{\mathrm{DGS}} \leftarrow \sum_{i=\mathrm{DGS}\in\Gamma_{\mathrm{n}}} \overline{S}_i$;
    $\overline{S}_{\mathrm{rem}} \leftarrow \overline{S} - \overline{S}_{\mathrm{DGS}}$;
    if $\overline{S}_{\mathrm{rem}} > 0$;
    slots are allocated to flows $j \in$ { DSS,TSS} proportional to their queue size;
    $\overline{S}_j \leftarrow \overline{S}_j + \left\lfloor \overline{S}_{\mathrm{rem}} \frac{Q_i}{\sum_{j,j\neq i} Q_j} \right\rfloor$;
    if $\overline{S}_{\mathrm{rem}} \geq (\overline{S}_{\mathrm{DSS}} + \overline{S}_{\mathrm{TSS}})$
    $\overline{S}_{\mathrm{DSS}} \leftarrow \min(\overline{S}_{\mathrm{DSS}}, \overline{S}_{\mathrm{rem}})$;
    $\overline{S}_{\mathrm{TSS}} \leftarrow \min(\overline{S}_{\mathrm{TSS}}, (\overline{S}_{\mathrm{rem}} - \overline{S}_{\mathrm{DSS}}))$;
    $\overline{S}_{\mathrm{BES}} \leftarrow \overline{S}_{\mathrm{rem}} - \overline{S}_{j,j\in\mathrm{DSS,TSS}}$;
    $\mathbf{\Gamma}_{\mathrm{c}}(\mathrm{j},:) \leftarrow \mathbf{\Gamma}_{\mathrm{n}}(\mathrm{j},:)$;
    if $\overline{S}_{f_i} = 0$;
    Change MSTR of this flow class or do reassocaition to another AP ;
    $\mathbf{\Gamma}_{\mathrm{nc}}(\mathrm{i},:) = \mathbf{\Gamma}_{\mathrm{c}}(\mathrm{i},:)$;
  end
  $t \leftarrow t + \tau$;
end

AP switches. The scheduling module forwards traffic packets to the medium access control of the selected APs, following the procedure explained in Algorithm 1.

A. TRAFFIC FLOWS ROUTING MODULE

The TE scheme uses this module to select an AP for each active UE that requests downlink traffic flows, as explained in Algorithm 1. This routes flows on the network layer and schedules their packets (frames) on the MAC layer based on their class and QoS requirements. The presence of UEs in the network are detected by the software agents, running in the APs, based on their active wireless interfaces and service coverage. When the UEs are detected by a single AP, they are associated with it and their traffic is classified as local. Whereas, when the UEs are detected by multiple APs, the module uses routing policies to select an AP that can better serve them. It also classifies their traffic as generic and shares it among the selected APs. The UEs are cleared from their APs in the local and generic matrices, when it shows that their idle time exceeds or the SNR drops below programmable preassigned thresholds for these parameters. While we consider N AP switches, let n be a subset of APs that detect and can serve a UE.

Traffic routing policies are broadly classified into two types: deterministic and non-deterministic policies [39]. The former routes downlink traffic flows to APs based on criterion functions, whereas the latter routes traffic flows based on probabilities, ω_i, assigned to APs, as shown in Fig. 13. The module runs a probabilistic policy, which will be discussed later along with three deterministic routing policies: smallest delay, minimum network delay and maximum throughput. Obviously, these policies mainly run on the network layer and therefore use information related to this routing level.

A deterministic smallest delay routing policy has a criterion function, $f_{\text{sd}}(\vec{\boldsymbol{q}}_{\boldsymbol{s}}, \vec{\boldsymbol{\mu}}_{\boldsymbol{s}})$, of the average queue length vector, $\vec{\boldsymbol{q}}_{\boldsymbol{s}}$, and service rate vector, $\vec{\boldsymbol{\mu}}_{\boldsymbol{s}}$, of APs in the network. The SDN controller advertises the state of network in terms of the queue length and service rate vectors, as follows: $\vec{\boldsymbol{q}}_{\boldsymbol{s}} = (q_{s_1}, q_{s_2}, \ldots, q_{s_n})$ and $\vec{\boldsymbol{\mu}}_{\boldsymbol{s}} = (\mu_{s_1}, \mu_{s_2}, \ldots, \mu_{s_n})$, $\forall n \leq N$, where q_{s_i} denotes the number of packets waiting in the queue and those under service in the AP switch i; and μ_{s_i} denotes the service rate of the AP switch i. The policy, $f_{sd}(\vec{\boldsymbol{q}}_{\boldsymbol{s}}, \vec{\boldsymbol{\mu}}_{\boldsymbol{s}})$, routes a downlink traffic flow to an AP i that has the smallest ratio of the queue length to the service rate, expressed as follows [40]:

$$f_{\text{sd}}(\vec{q}_s, \vec{\mu}_s) = \min_{i=1,2,\ldots,n} \frac{q_{s_i}+1}{\mu_{s_i}}. \tag{13}$$

If more than one AP has the same minimum queue length to service rate ratio, the policy selects the AP i which has the maximum service rate, μ_{s_i}. Since $\frac{q_{s_i}+1}{\mu_{s_i}}$ provides the average delay of the next admitted flow in the network, the smallest delay traffic routing policy provides an indication of how to improve the response time of provisioned services.

The minimum average network delay policy has a criterion function, $f_{\text{nd}}(\vec{\boldsymbol{q}}_{\boldsymbol{s}},\vec{\boldsymbol{\mu}}_{\boldsymbol{s}})$, of the average queue length, $\vec{\boldsymbol{q}}_{\boldsymbol{s}}$, and service rates, $\vec{\boldsymbol{\mu}}_{\boldsymbol{s}}$, of APs in the network. It routes a downlink traffic flow to an AP i that minimizes the delay of all waiting and in service packets at the APs which can serve a UE, given as follows [41]:

$$\begin{aligned} f_{\text{nd}}(\vec{q}_s, \vec{\mu}_s) &= \frac{1}{\sum_{j=1,q_{s_j}\neq 0}^{n} \mu_{s_j}} \\ &\quad + \sum_{k=1,q_{s_k}\neq 0}^{n} \left[\frac{\mu_{s_k}}{\sum_{j=1,q_{s_j}\neq 0}^{n} \mu_{s_j}} f_{\text{nd}}(q_s, \mu_s) \right]; \\ f_{\text{nd}}(0, \ldots, q_{s_k}, \mu_{s_k}, \ldots, 0, 0) &= \frac{q_{s_k}}{\mu_{s_k}}; \\ f_{\text{nd}} &= \min_{k=1\ldots n} \left\{ f_{nd}(\vec{q}_s, \vec{\mu}_s) \right\}. \end{aligned} \tag{14}$$

Alternatively, this policy routes a flow to the AP i that has the shortest queue, which can be heuristically expressed as follows [41]:

$$f_{\text{ld}}(\vec{q}_s, \vec{\mu}_s) = \min_{i=1,2,\ldots,n} \frac{q_{s_i}+1}{\mu_{s_i}^2}. \tag{15}$$

The maximum average throughput policy has a criterion function, $f_{\text{mt}}(\vec{\boldsymbol{q}}_{\boldsymbol{s}},\vec{\boldsymbol{\mu}}_{\boldsymbol{s}},\boldsymbol{\lambda}_{\boldsymbol{f}})$, of the flow average traffic arrival rate, λ_f, in addition to the average queue length vector, $\vec{\boldsymbol{q}}_{\boldsymbol{s}}$, and service rate vector, $\vec{\boldsymbol{\mu}}_{\boldsymbol{s}}$, of APs. A flow is routed to an AP i that maximizes the average network throughput during the next inter-arrival time period. This policy also minimizes the average network response time. The criterion function, $f_{\text{mt}}(\vec{\boldsymbol{q}}_{\boldsymbol{s}},\vec{\boldsymbol{\mu}}_{\boldsymbol{s}},\boldsymbol{\lambda}_{\boldsymbol{f}})$, is expressed as follows [40]:

$$\begin{aligned} f_{\text{mt}}(\vec{q}_s, \vec{\mu}_s, \lambda_f) &= \sum_{i=1}^{n} \lambda_f \left[\sum_{k=1}^{q_{s_i}-1} (1 - \frac{q_{s_i}}{k})(\frac{\mu_i}{\lambda_f + \mu_i})^k \right. \\ &\quad \left. - q_{s_i} \ln(\frac{\lambda_f}{\lambda_f + \mu_i}) \right]; \\ f_{\text{mt}} &= \max_{i=1\ldots n} \left\{ f_{\text{mt}}(\vec{q}_s, \vec{\mu}_s, \lambda_f) \right\}. \end{aligned} \tag{16}$$

Alternatively, this policy $f_{\text{mt}}(\vec{\boldsymbol{q}}_{\boldsymbol{s}},\vec{\boldsymbol{\mu}}_{\boldsymbol{s}},\boldsymbol{\lambda}_{\boldsymbol{f}})$ routes a flow to an AP i that maximizes the expected number of flow service completions before the next expected flow arrival. With this policy rule, the AP that satisfies [41]:

$$f_{\text{mt}}(\vec{q}_s, \vec{\mu}_s, \lambda_f) = \max_{i=1\ldots n} \left\{ \left\{ \frac{\mu_{s_i}}{\lambda_f + q_{s_i} + \mu_{s_i}} \right\}^{q_{s_i}} \right\} \tag{17}$$

is selected.

The proposed TE scheme also runs a non-deterministic probabilistic policy [40] which aims to minimize the average response time of all service flows in the network data plane. The UEs that can only be served under a single AP generate local or dedicated traffic, $\vec{\lambda}_l = (\lambda_1, \ldots, \lambda_n)$, $\forall n \leq N$. Whereas, the UEs that can be served under multiple APs generate a generic traffic load, which is shared among the reachable APs based on splitting probabilities, $\vec{\omega}_s = (\omega_{s_1}, \ldots, \omega_{s_n})$, $\forall\, n \leq N$, as shown in Fig 13. When the probability of flow routing to AP i, $\omega_{s_i} \neq 1$, the routing decision is considered non-deterministic. When $\lambda_{s_i} + \lambda\omega_{s_i} \leq \mu_{s_i}$, the AP i is considered unsaturated and the response time is calculated based on (12).

Let λ_{s_i} and ρ_{s_i} be the local traffic arrival rate and load intensity at AP switch i; and λ_n and ρ_n denotes the total generic traffic arrival rate and load intensity at n APs, respectively. An AP i is assigned a generic traffic load based on, $\rho_n r_{s_i} \omega_{s_i}$, where $r_{s_i} = \frac{\mu_{s_i}}{\min_{i=1}^{n} \mu_{s_i}}$; and $\rho_n = \lambda_n(\min_{i=1}^{n} \mu_{s_i})$. The minimum average network response time optimization can be formulated as a non-linear programming problem with linear constraints, as follows [40]:

$$\begin{aligned} \min_{\omega_{s_1},\omega_{s_2},\ldots,\omega_{s_n}} & \frac{1}{\lambda_n + \sum_{j=1}^{n} \lambda_{s_j}} \sum_{i=1}^{n} \frac{\rho_n r_{s_i} \omega_{s_i} + \rho_{s_i}}{1 - (\rho_n r_{s_i} \omega_{s_i} + \rho_{s_i})} \\ s.t.\ & \omega_{s_i} \geq 0, \quad 1 \leq i \leq n \\ & \sum_{i=1}^{n} \omega_{s_i} = 1 \\ & \rho_n r_{s_i} \omega_{s_i} + \rho_{s_i} < 1, \quad 1 \leq i \leq n. \end{aligned} \tag{18}$$

The solution of this optimization problem stated in (18) results in obtaining these optimal traffic load sharing probabilities: $\omega_s^* = [\omega_{s_1}^*, \omega_{s_2}^*, \ldots, \omega_{s_n}^*]$. These determine the proportions of generic traffic flows assigned to each AP that can serve the UEs.

An adaptive control algorithm uses the collected SDN information to maintain a similar level of controlled generic loads at n APs. A heuristic probabilistic load sharing algorithm is developed in [40] to solve this nonlinear programming problem. The proposed TE scheme receives the information regarding the local and generic matrices in addition to the queue length and service rates of the APs every τ seconds [42], [43], as explained in Algorithm 1. These parameters are used in a function developed in MATLAB to solve the optimization problem (18). The splitting probabilities, ω_{s_i}, are kept constant during the measurement intervals; and they are updated upon the receipt of a new measurement every τ. The local traffic flows are first associated with their APs. Then, the generic traffic flows are routed to the APs that can serve them, using the above routing policies, as explained in Algorithm 1.

B. GRANULAR RESOURCE ALLOCATION MODULE

The scheduling module supports granular differentiated services on the MAC layer of LiFi and LTE APs. It allocates the wireless interface bandwidth to the UEs based on the number of bits that can be transmitted at their assigned MCS per frame. The smallest data unit allocation per resource block, b, differs in each wireless technology. Therefore, b_k represents different values in LiFi and LTE. For LTE, the number of transmitted bits per resource block (RB) pair using MCS_i is, $b_i = 2\ R_{\text{re}} b_{\text{re}_i}$, where b_{re_i} represents the number of bits transmitted per resource element using MCS_i [44]. For LiFi, b_i represents the number of transmitted bits per slot using MCS_i. Whereas, for WiFi, it represents the number of transmitted bits per frame (i.e. contention period).

A two-level hierarchical parametrized scheduler is deployed in the AP switches to adaptively allocate wireless resources. It can support various policies for sharing the AP wireless interface and scheduling polices to provide multiple services. A traffic flow classifier module is integrated with the scheduler. It classifies packets arriving from the application and transport layers based on an identifier of their traffic class, (f_{id}), and forwards them into a queue designated for their service class [38]. The upper interface medium access class scheduler distributes the available bandwidth among the supported services according to a criterion function of AP state and performance parameters. The lower interface medium access service scheduler supports light scheduling functionalities that serve separate queues designated for the supported service classes [38].

One of the QoS parameters considered for service classes is the maximum sustained traffic rate (MSTR) in bits per second (bps) [44], which is an upper bound for the user throughput. In our context, each service class can be associated with a different MSTR. The average download duration of a UE is, $\bar{t}_{\text{on}} = \frac{\overline{Q}}{\overline{L}}$, where $\overline{Q}$ denotes the average number of active UEs; and $\overline{L}$ denotes the average number of UEs that are served per unit of time. The average throughput of a UE can be calculated as, $\gamma = \frac{\bar{x}_{\text{on}}}{\bar{t}_{\text{on}}}$, where $\bar{x}_{\text{on}}$ denotes the average size of downlink data in bits during the ON period. A UE may receive a different amount of resources at each frame to achieve its guaranteed bit rate (GBR (κ)), which varies with the MCS_i. It is assigned a slightly greater bit rate than its GBR, called delivery bit rate (DBR (ξ)) to compensate for the data rate losses in the outage periods, given by [45]:

$$\xi = \frac{\kappa}{1 - p_0}, \tag{19}$$

where p_0 denotes the wireless channel outage portability. A UE that achieves its ξ (DBR) using MCS_i needs g_i slots, given as follows:

$$g_i = \frac{\xi\ T_F}{b_i}. \tag{20}$$

A UE with an outage probability, $p_0 = 1$, is not allocated any slot (i.e. $g_0 = 0$). The average number of slots per frame needed by m UEs belonging to a service class to obtain their DBR is given by:

$$\overline{g}(m) = m \sum_{i=1}^{I} p_i \overline{g}_i. \tag{21}$$

The hierarchical scheduler supports four service schedulers, which serve the proposed DGS, DSS,TSS and BES classes. There is a limit, $M_{\max}$, to the number of simultaneous DGS, DSS and TSS flows accepted in a small cell. The BES traffic flows can simultaneously transfer data, when there are enough resources remained after serving the higher priority services. The upper class scheduler dynamically shares the downlink wireless bandwidth of each AP among the supported services based on their backlogged traffic and QoS requirements, as explained in Algorithm 1. This improves the buffer utilization and wireless interface bandwidth of APs, while guaranteeing the requirements of services. The DGS scheduler assigns fixed time slots for flows, which are

negotiated in the initial service access phase and deducted from each downlink frame, as explained in Algorithm 1. The aggregate departure rates $\mu(m)$ per service class is expressed as follows [44], [45]:

$$\mu(m) = \frac{\overline{S}}{\max(m\overline{g}, \overline{S})} m \frac{R}{\overline{x}_{\text{on}}}, \tag{22}$$

where R denotes the MSTR of the service class offered to m UEs. The first part $\frac{\overline{S}}{\max(m\overline{g},\overline{S})}$ represents the ratio of the global departure rate achieved by the m concurrent transfers, when there are m active UEs needing $m\ \overline{g}$ slots on average to obtain their R. The second part $m\frac{R}{\overline{x}_{on}}$ corresponds to the rate at which any of the m active UEs completes its transfer, assuming there are always enough available slots in the frame to satisfy the R. The service class parameter intensity, ρ, is given by [44], [45]:

$$\rho = \frac{\overline{x}_{\text{on}}}{\overline{t}_{\text{off}} R}, \tag{23}$$

where $\overline{t}_{\text{off}}$ denotes the average OFF duration.

The steady-state probabilities, $\pi(m)$, can be expressed as follows [44], [45]:

$$\pi(m) = \frac{M!}{(M-m)!} \frac{\rho^m}{m! \prod_{i=1}^{m} \frac{S}{\max(i\overline{g}, S)}} \pi(0), \tag{24}$$

where $\pi(0) = 1 - \rho$. The average number of active UEs is given as follows:

$$\overline{Q} = \sum_{m=1}^{M} m\pi(m). \tag{25}$$

The average number of UEs, $\overline{L}$, that complete their transfer per unit of time is given by:

$$\overline{L} = \sum_{m=1}^{M} \mu(m)\pi(m). \tag{26}$$

The average throughput $\overline{X}$ obtained by each active UE is given as follows:

$$\overline{X} = \frac{\overline{x}_{\text{on}}}{\overline{t}_{\text{on}}} = \frac{\overline{x}_{\text{on}} \sum_{m=1}^{M} \mu(m)\pi(m)}{\sum_{m=1}^{M} m\pi(m)}. \tag{27}$$

The average utilization $\overline{U}$ of the frame is defined as the weighted sum of the ratios between the average number of slots needed by the m UEs to reach their R and the average number of slots they obtain:

$$\overline{U} = \sum_{m=1}^{M} \frac{m\overline{g}}{\max(m\overline{g}, S)} \pi(m). \tag{28}$$

A feasible Earliest Due Date (FEDD) scheduler policy [46] selects, at each slot per frame, a UE with the earliest deadline among the UEs which have good channel gains. We recognize that this policy is not always throughput optimal. The TSS scheduler uses the Modified Largest Weighted Delay First (MLWDF) policy proposed in [47], [48]. It schedules UE i^* such that [47], [48]

$$i^* = arg \max_i \left\{ \delta_i [D_i(t)]^{\iota} \mu_i(t) \right\}, \tag{29}$$

where $D_i(t)$ is the head-of-the-line packet delay for UE i at time t, $\mu_i(t)$ is its instantaneous channel data rate at time t and δ_i and ι are arbitrary positive constants. It is recommend a value of [48]

$$\delta_i = \frac{a_i}{\overline{\mu}_i} \tag{30}$$

for the UE i, where a_i is a weight that may be based on delay requirements, and $\overline{\mu}_i$ is the user's long-term average data rate. This policy tries to balance weighted delays, and for the given choice of δ_i, reduces to a channel-aware scheduling strategy when the users are otherwise equal. The DSS scheduler uses the exponential rule introduced in [47] and analysed in [49]. It schedules the user i such that [49]

$$i^* = \arg\max_i \left\{ \delta_i \mu_i(t) \exp\left(\frac{a_i D_i(t) - a\overline{D}}{1 - \sqrt{(a\overline{D})}}\right) \right\}, \tag{31}$$

where $a\overline{D} = \frac{1}{M} \sum_i D_i(t)$, and M is the total number of users. This policy tends to equalize the weighted delays when the differences are large, and falls to the Proportional Fair (PF) policy when the differences are small. A PF algorithm serves each UE i at peak channel conditions. At each scheduling time slot, the BES scheduler policy selects the UE i with the highest priority value which is calculated as follows:

$$i = \arg\max_i \left\{ \frac{\mu_i(t)}{\overline{\mu}_i} \right\}. \tag{32}$$

The class scheduler aims to allocate a sufficient number of slots from the frames to each service so that its active UEs together can achieve R, as in Algorithm 1. If a UE that receives the BES is in an outage state, it will experience a temporarily degraded throughput. If at any given time, the total number of available slots is not enough to satisfy R of all users, excluding those in outage, given a BES class, they will all experience equally degraded throughput. The service scheduler policies can be dynamically selected or updated for each service class to best meet its QoS requirements. The class scheduler policies can also be dynamically updated to efficiently support the QoS requirements of supported service classes. After the class scheduler policy allocates the slots to the DGS class, the remaining slots are allocated to DSS and TSS flows proportional to their reported backlogged traffic. If DSS or TSS flows cannot be assigned slots at an AP, the algorithm either changes the sustainable data rate of the traffic class or routes them to another AP that can provide the requested service. This process is updated per SDN period, τ, and a number of frames, T_{f}, to adapt the QoS requirements of supported services by the scheduler module.

C. NETWORK RESILIENCE

When the performance level of the provided DGS, DSS or TSS is degraded below a predetermined threshold or an AP becomes unresponsive (i.e. fails), the TE scheme runs the restoration Algorithm 2 to maintain service provisioning to the affected UEs. In a centralized tree-type SDN-enabled network, the dynamic traffic routing Algorithm 2 running in the SDN controller dynamically employs the wide coverage WiFi or LTE capacity to support a single AP failure recovery. It does not place any spare network capacity for mitigating the impact of an AP failure to provide service continuity with an acceptable QoS level to UEs during the AP failure. The agents running in the APs periodically send the load of each traffic class in addition to the information matrices, $\mathbf{\Gamma}_{\boldsymbol{l}}$, $\mathbf{\Gamma}_{\mathrm{g}}$, to the SDN controller. This uses Algorithm 1 to periodically compute and update the information matrix, Γ_{nc}. When an AP fault occurs, the SDN controller produces a new matrix $\mathbf{\Gamma}_{\text{nc-r}}$, which is used to reroute traffic from the failed AP. Algorithm 2 uses a traffic re-routing policy that considers the tree-topology constraints, volume and QoS requirements of the failed flows. The SDN controller reroutes the affected traffic flows based on the reported information of the failed AP before a failure occurrence. The local traffic flows are either routed to the LTE or WiFi AP by using the probabilistic routing policy. Whereas the generic traffic flows are managed according to the other routing policies following Algorithm 1.

Algorithm 2 Service Provisioning Restoration From a Single AP Failure

Input : Local and generic matrices, $\mathbf{\Gamma}_{\boldsymbol{l}}$, $\mathbf{\Gamma}_{\mathrm{g}}$, $\mathbf{\Gamma}_{\mathrm{nc}}$
Network fault event: An AP becomes unresponsive or service flows' throughput or delay drops below a specific threshold;
Output: Updated Network-MAC layer matrix: re-routing failed downlink flows to operational APs, $\mathbf{\Gamma}_{\text{nc-r}}$
for $i \leftarrow 1$ *to* $N_{\mathrm{f}} \in \mathbf{\Gamma}_{\boldsymbol{l}}$ **do**
  if $f \in \mathbf{\Gamma}_{\boldsymbol{l}}$, route local traffic flows to LTE or WiFi based on their traffic load. **if** $f \in \mathbf{\Gamma}_{\mathrm{g}}$, route local traffic flows baed on Algorithm 1. Update $\mathbf{\Gamma}_{\mathrm{nc}}$ based on the rerouted traffic flow and their correspoding inforamation.
end
Remove the row corresponding to failed APs from $\mathbf{\Gamma}_{\mathrm{nc}}$;
$\mathbf{\Gamma}_{\text{nc-r}}(:,:) \leftarrow \mathbf{\Gamma}_{\mathrm{nc}}(:,:)$

VII. SIMULATION AND RESULTS

A discrete time simulation environment has been developed in MATLAB to evaluate the proposed TE scheme in the HetNet. This is comprised of 10 LiFi APs, one LTE femtocell and another WiFi APs, which provide services to UEs uniformly distributed in a room of size $16 \times 6 \times 4$ ($W \times L \times H$), as shown in Fig. 14. The user mobility pattern follows a random waypoint model [50], [51] with a uniform distribution low

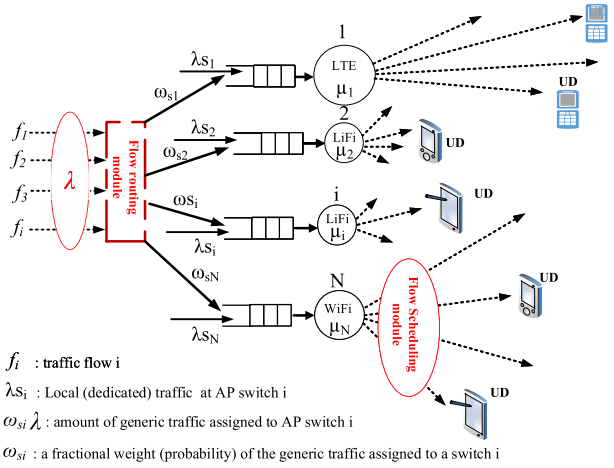

FIGURE 13. A traffic engineering scheme for SDN-enabled HetNet.

movement speed between 0.1 and 0.4 m/s, which correspond to the users movement in the room. The downlink physical data rates of LiFi APs, the LTE femtocell AP and the WiFi AP are taken to be 45, 14 and 54 Mbps, respectively. The channel gains of LiFi APs are calculated based on the Lambertian model [52], which considers signal reflections from all the room walls. The main parameters of the LiFi attocellular network are summarized in Table 1 [53] and [54], [55] with a service coverage radius of 3 m and total number of OFDMA subcarriers on downlink channel, $\mathrm{N_{sc}} = 1024$. The main parameters of the LTE femtocell system are summarized in Table 1 [56] and [57] with service coverage radius 20 m. The channel access parameters of the WiFi AP are summarized in Table 1 [58] and [31]. The interface queue length of all APs uses a drop tail policy; and it is set at 5000 packets. The maximum number of users that can be simultaneously connected to a LiFi AP, LTE femtocell or WiFi AP, is set to 8, 10 and 20, respectively. Similarly, the maximum number of each traffic connection class is set to communicate through the LiFi, LTE or WiFi AP switches.

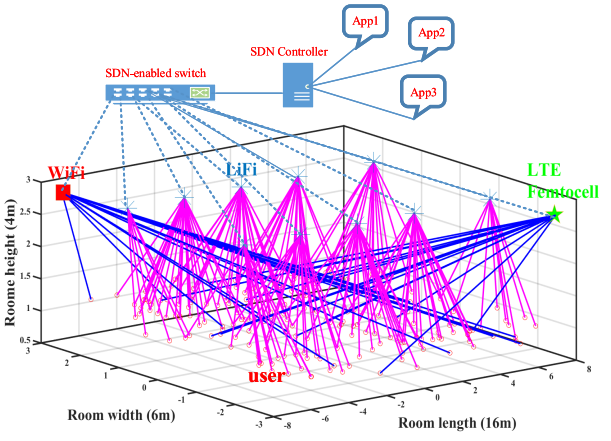

FIGURE 14. SDN-enabled HetNet simulation environment.

A number of applications share the spectrum and buffer resources in the SDN-enabled HetNet, as shown in Fig. 14.

ON/OFF traffic sources generate DGS, DSS, TSS and BES traffic flows. An ON period is considered as a sequence of arrivals with less than t seconds between two OFF events. The duration of ON and OFF periods are exponentially distributed. The traffic flows of DGS, DSS, TSS and BES are generated with data rate at 64 kbps, $1-5$ Mbps, $12-18$ Mbps, $0.5-6$ Mbps, respectively. The guaranteed bit rate (GBR) of DGS, DSS and TSS is set to 64 kbps, 2 Mbps and 12 Mbps, respectively. A throttling policy schedules BES flows according to their MSTR. This is set to 1 Mbps, but it can be adjusted by the controller according to the service requirements of the other traffic classes and network state. The average downlink size of BES traffic, $\bar{x}_{\text{on}}$, is set to 2 Mbps. The packet size of DGS, DSS,TSS and BES flows is set to 200, 256, 512, 1512 bytes, respectively. The packet arrival rate, λ, of DGS, DSS, TSS and BES 90, 700, 350, 620 packet per second (pps), respectively. Each time a random number of UDs are active in the network, there are 30 % DGS UDs, 25 % DSS UDs, 25 % TSS UDs and 20 % BES UDs are distributed across the HetNet.

We have conducted a number of simulation scenarios to evaluate the impact of the proposed TE scheme on the service provided at the network, MAC and across network-MAC layers. The conducted scenarios are evaluated in two network settings: Random HetNet setting (RNS) and SDN-enabled HetNet setting (SNS). In the RNS, UEs receive services in a default HetNet that runs without the SDN support capabilities. For example, the UEs are associated to APs providing them with the best received signal strength (RSS). Whereas, in the SNS, the TE scheme supports service provisioning to UEs, which considers the network information feedback regarding the resource availability of APs, service class and target network performance.

The throughput distribution within LiFi, WiFi and LTE cells varies based on their path loss models, which changes in terms of the distance and transmission power. These define the virtual boundaries of circles (rings) forming the cells coverage. Within these cells, the throughput is considered uniform, but varies based on the number of UEs and traffic load, as shown in Fig. 15. A cell capacity is defined as the maximum traffic load that can meet its QoS requirements. The maximum average data arrival rate per UE is evaluated, given that all the UEs are uniformly distributed within the coverage area of each AP in the HetNet. The flow throughput and cell capacity in the different AP cells decrease, as the traffic load and radius increase. Deploying multiple optical wireless technologies, such as LiFi APs, in an indoor environment is important for meeting the bandwidth and throughput requirements of UEs with heterogeneous wireless air interfaces.

The MAC scheduler shares the limited communication bandwidth of the AP wireless channel among the supported traffic services. The interface bandwidth of LiFi APs is shared in proportion to the number of UEs and generated traffic load per class, which is also influenced by the network setting, as shown in Fig. 16. Each evaluation point in this figure

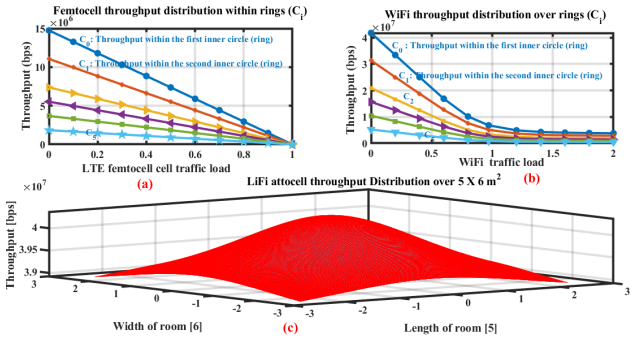

FIGURE 15. Throughput distribution under small cells: (a) LTE femtocell (b) WiFi cell (c) LiFi attocell.

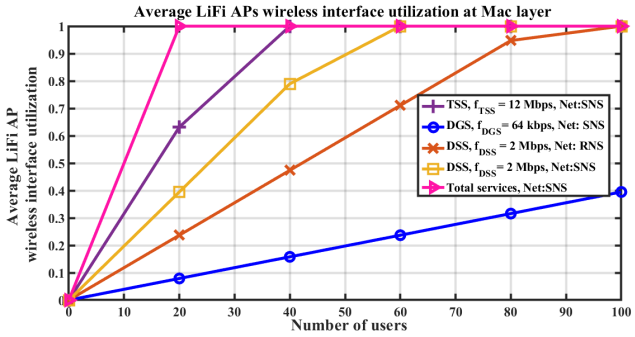

FIGURE 16. Average interface utilization of LiFi APs at MAC layer.

corresponds to the average utilization of service at each AP. The different scheduling functionalities can better manage the wireless interface of LiFi APs in SNS than RNS. This explains the admission of more DSS flows, which increases the utilization of LiFi AP interfaces in SNS. More DSS flows are dropped in RNS than SNS, which decreases the LiFi interface utilization, as shown in Fig. 16. Furthermore, in RNS, some downlink traffic flows are blocked due to the unfairness of load distribution among the different APs in the HetNet.

The effective capacity is defined as the maximum sustainable constant data arrival rate by the channel process under some throughput or delay (QoS) constraints. The proposed TE scheme is evaluated under the minimum throughput and delay requirements of TSS and DSS traffic flows, respectively. The transmission of TSS real-time video streaming requires a large bandwidth and a minimum stable end-to-end delay. As a result, Fig. 17 shows the average flow throughput of TSS flows achieved under their minimum throughput requirements. We observe that the TSS flows are guaranteed a minimum bandwidth of 12 Mbps in SNS, which is difficult to achieve in RNS, as shown in Fig. 17. This is attributed to the fact that in SNS, traffic flows are balanced among the network APs; and wireless interfaces of APs are shared in proportion to traffic flows class and volume. In SNS, the TE scheme enhances the performances of HetNet by increasing the number of admitted traffic flows with minimum

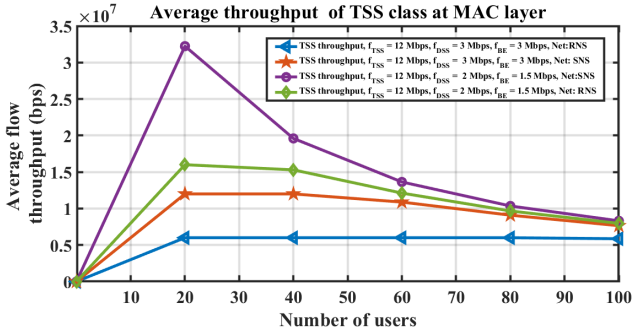

FIGURE 17. Average flow throughput in TSS class at MAC layer.

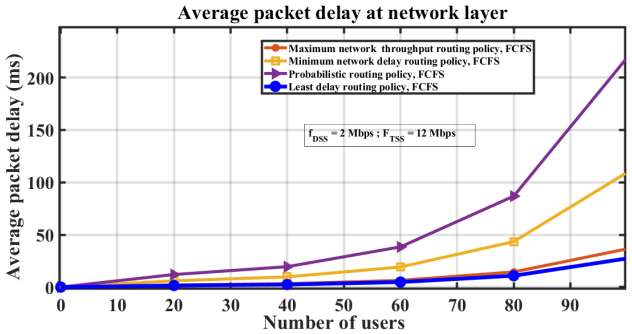

FIGURE 19. Network response time for different traffic routing policies.

throughput requirement, which increase, in turn, the network throughput, as shown in Fig. 17.

The DSS traffic packets experience a delay at the MAC layer of APs due to the resource contention and network congestion, which can be better maintained within strict performance bounds in SNS. The aggregate scheduler on the MAC layer ensures the resource allocation to the DSS traffic packets. Thus, they experience a shorter delay in SNS than RNS. The UEs are associated to APs based on service guarantee policies at the MAC layer, which can better exploit the LiFi and LTE/WiFi channels for data transmission in SNS than RNS, as shown in Fig. 18. The average delay of DSS traffic grows exponentially, when the traffic volume of the other classes increases significantly, because the aggregate scheduler has to allocate some resources to the other traffic classes. Also, when the number of UEs that request the DSS increases significantly, their delay exceeds the upper delay bound requirement.

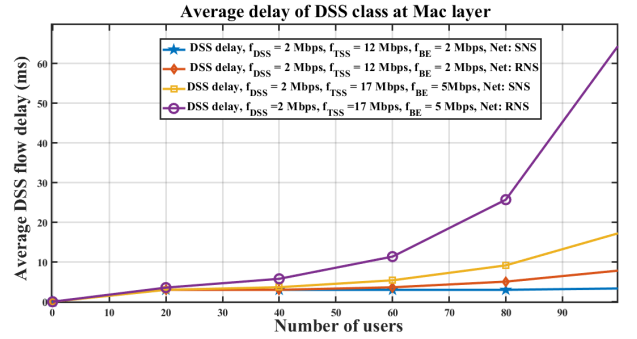

FIGURE 18. Average flow delay in DSS class at MAC layer.

The proposed TE scheme is evaluated on the network layer in two scenarios. The first considers class-based traffic flows routing according to their minimum service requirement and HetNet resource allocation efficiency. The second scenario focuses on the HetNet resilience from a single AP failure, which considers re-routing the failed traffic flows. The network response time is defined as the amount of time required for a packet to be routed on the network layer and scheduled at the MAC layer to reach the UEs in the downlink direction. Traffic packets are routed according to different routing policies, which have unequal network response times, as shown in Fig. 19. Obviously, packets delay increases as the number of users grows in the network. The smallest network delay and minimum throughput routing policies could maintain the delay within acceptable upper bounds, compared to the other policies, as shown in Fig. 19. The minimum network delay routing policy requires more time to find an AP that can minimize the whole network delay. Similarly, the probabilistic routing policy requires to classify the local and generic traffic, which require more time as flows grow in number and volume, making it difficult to achieve a minimum network response time.

The proposed TE scheme is evaluated across the network-MAC layer in two scenarios. The first discusses HetNet dimensioning according to the minimum data rate requirement and maximum target network throughput per active TSS UE. The second scenario discusses the capability of TE scheme to support reliable HetNet. Users transfer traffic during the ON state at a minimum data rate, r, where the distribution of UEs, m, follows a binomial probability density function. The binomial cumulative distribution function is used to calculate the probability of the maximum target total HetNet throughput, R (MSTR), per TSS UE, considering different number of UEs in SNS and RNS configurations. As a result, we observe that the TE scheme enables the HetNet to efficiently allocate the bandwidth of each AP to maximize the target HetNet throughput, R, which supports the UEs receiving the TSS class. We notice that with a probability of 0.9, a total maximum sustainable throughput of 12 Mbps and 13 Mbps can support the TSS UEs in RNS and SNS configurations, respectively, as shown in Fig. 20. As the number of UEs increases, the target total throughput increases per TSS UE in SNS than RNS. For example, with a probability of 0.9, a maximum throughput of 32 Mbps and 38 Mbps are guaranteed per TSS UE in RNS and SNS, respectively. An extra 6 Mbps are added to the R (MSTR) of each active TSS UE in SNS than RNS, as shown in Fig. 20.

The proposed TE scheme supports the HetNet to survive from a single AP failure, which enables it to autonomously handle the affected flows that were receiving services from a failed AP, as explained by Algorithm 2. As mentioned before, an AP is identified as failed, when it can no longer provide

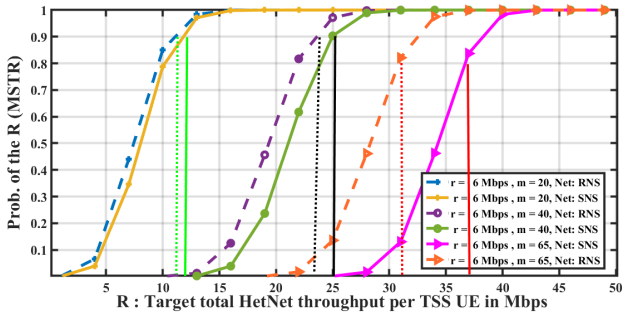

FIGURE 20. Target total HetNet throughput per TSS UE.

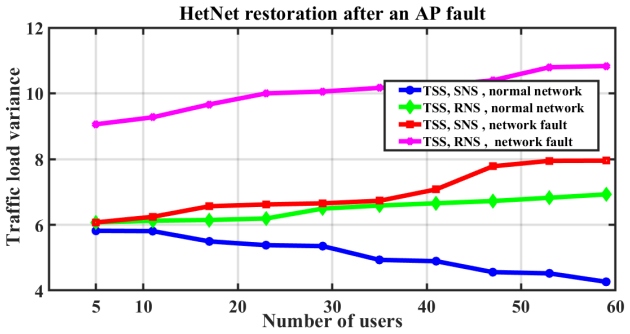

FIGURE 22. Traffic load variance of TSS flows after network fault.

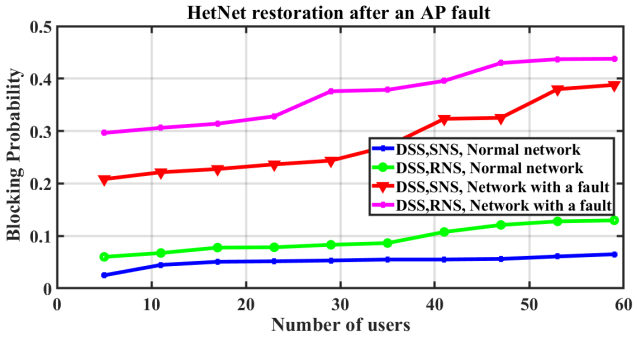

FIGURE 21. Blocking probability of DSS flows after network fault.

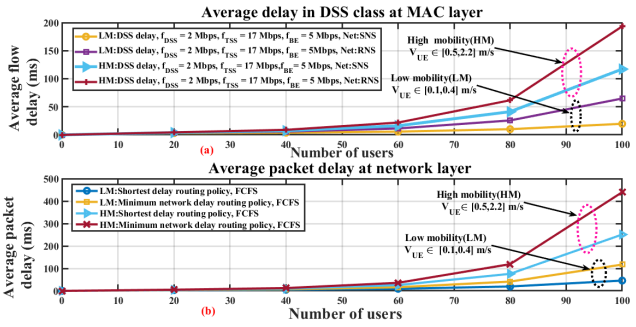

FIGURE 23. Mobility impact on average (a) DSS delay at MAC layer and (b) packet delay at network layer.

services to the users. While the affected flows from an AP failure can be rerouted to neighbouring LiFi APs, or wider coverage APs like WiFi or LTE AP, the network resources may not be sufficient to accommodate them. A computer simulation is carried out, which involves the four neighbouring LiFi APs in the first two columns, LTE and WiFi APs in the SDN-enabled HetNet topology shown in Fig. 14. The focus is put on studying the performance of the affected DSS and TSS service flows from a single AP failure, subject to delay and throughput constraints, respectively. As discussed before, the DSS and TSS flows should receive their minimum e2e delay and throughput guarantees, respectively, to provide the requested services to the UEs. Therefore, some of these flows are blocked during an AP failure, because of being routed to an overloaded AP or their routing process takes a longer time. The blocking handover request probability is defined as the number of dropped requests divided by the total affected flows that require association again with a new AP due to the failure. When the HetNet runs normally, without any AP failure, the DSS flows have a lower blocking probability in SNS than RNS, as shown in Fig. 21. The probabilistic routing policy considers the affected flows as generic, which can be routed to any of the four APs or LTE/WiFi APs that can provide the requested services. Algorithm 2 introduces a SDN scheme to manage the affected flows, which reduces the number of blocked DSS flows in SNS, as shown in Fig. 21.

The traffic load distribution varies at APs, because of network dynamics, which is expected to increase in case of AP failure. A traffic load variance metric is evaluated to show the spread of the traffic load at the APs involved in serving the flows before and after AP failures. The variance can be used to evaluate the standard deviation, which is a better measure for the traffic load variations at the APs. The standard deviation is the square root of the traffic load variance, which is expressed as follows: $\sqrt{\sigma_i^2} = \frac{1}{N}\sum_{i=1}(\lambda_i - \mu_i)^2$, where N is the number of APs. In SNS, the load variance is smaller than in RNS, because the affected UDs are rerouted based on load balancing and scheduling policies. These keep the load of each AP around the average value measured during the AP failure.

While the previous results are obtained under low UE mobility, in the remaining part we investigate the impact of higher UE mobility on the performance of TE scheme. In essence, we compare the impact of low UE mobility to high mobility on traffic delay at MAC and network layers under the two network settings: RNS and SNS, separately, as shown in Fig. 23. In this paper, a high UE mobility means that the UE's minimum velocity v_{min} is set to 0.5 m/s, whereas the maximum velocity v_{max} varies over the range [0.5, 2.2] m/s. Initially, a UE is connected to a LiFi AP that can meet its service requirements in terms of delay and throughput. Otherwise, the TE scheme connects the UE to LTE or WiFi AP based on their traffic load. The UE is assumed to associate with only one AP at a time, and only vertical handover (VH) is performed from LiFi to LTE/WiFi and vice-versa.

When the number of users is low, the TE scheme has shown an interesting performance by keeping the average

DSS delay and packet delay at the MAC and network layers almost the same under low and high mobility, as shown in Fig. 23 (a) and (b). However, when the number of active users started to exceed 40 in the network, the delay started to pick up under both low and high UE mobility. The gap between low and high mobility impact on the delay becomes obvious, when the number of active users exceeds 68. This can be considered as the maximum number of UEs that the SDN-enabled HetNet can offer services under low and high mobility scenarios, where we can clearly see that SNS alleviates the average DSS delay at MAC and network layers.

As a UE moves across the network under low and high mobility, it requires horizontal handovers (HO) between the LiFi APs and VH among the LiFi/WiFi/LTE APs. While the TE scheme executes both the HO and VH processes, traffic flows and packets experience further delay and possible packet loss. They may also need to be connected again to a new AP to start receiving the requested traffic. This does not only increase the flows or packet delay at the MAC and network layers, but also decreases the network throughput at the MAC layer, as shown in 24 (a). Consequently, it decreases the target total HetNet throughput, R, per TSS UE, as shown in 24 (b). While the UEs move with high mobility under the LiFi APs, they may experience a series of return association to LiFi and LTE/WiFi RF APs, resulting in a ping-pong process. In real-time VH operations, the packet loss is caused due to the time waiting for the transmission and reception of signalling messages associated with network discovery and handover management procedures [59]. This mainly explains the higher delay and lower throughput of traffic flows/packets under high mobility in Fig. 23 (a) and (b), and Fig. 24 (a) and (b). Also, these figures show that the flows/packets delay, throughput and network throughput are improved under low and high mobility in SNS than RNS. This shows that the TE scheme could alleviate the impact of high UE mobility on traffic, service and network performance.

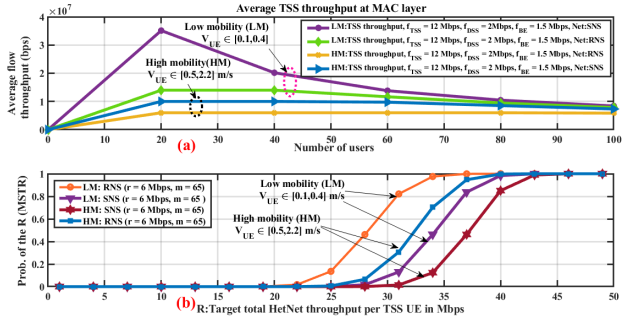

FIGURE 24. Mobility impact on (a) average TSS throughput and (b) target HetNet throughput per TSS UE.

VIII. RESEARCH CHALLENGES

The control and data planes of SDN architectures have limited available solutions to support reliable 5G cells and HetNet (i.e. WiFi/LTE/LiFi cells) integration in one network which we call as 5G+ network. The OpenFlow protocol [12] does not meet the requirements of 5G+ wireless communication and networking protocols [2]. The key research challenges aim to support: (i) dynamic programmability in the wireless operations of APs through the open southbound interfaces, (ii) adaptable efficient traffic, user and signalling management schemes in the control plane and (iii) applications that can leverage the 5G+ network brain (i.e. network control and information state) (iv) SDN for ultra dense 5G+ network management and network-as-a-service. In the following sections, we discuss some open research challenges.

A. SDN-ENABLED ULTRA-DENSE 5G+ NETWORKS MODELLING

The computing and processing capabilities of the SDN controller and OF switches constraint the potential performance of the control and data planes in terms of handling traffic flows volume and network size. Multiple SDN controllers are required to manage and control ultra-dense 5G+ small cell network which provides services in often complex indoor environments (e.g. residential skyscrapers). While each controller manages its local network, it can be interconnected with others from its east and west interfaces or through complex hierarchical SDN architectures. In this context, the following research challenges emerge to develop new solutions for enabling SDN in ultra dense 5G+ network.

- Analytical models are still required to express the capacity boundary conditions of large SDN-enabled ultra-dense 5G+ network. Stochastic network calculus can be utilized for analytical modelling of multiple SDN enabled 5G+ networks, though identifying the envelop of arrival and service curves at the different controllers and OF switches remains a very tedious work. Consequently, it is very challenging to develop closed forms for the end-to-end (e2e) packet delay upper bounds at the queues of controllers and switches.
- The SDN applications can apply, through the SDN controller, flow control, routing, prioritization and QoS enforcement at the AP switches to offer services in the data plane. However, it is still not clear how an application running in one network can influence the OF rules in other networks that are managed by different controllers. More research work is needed to develop centralized mechanisms, which can support reliable scalable applications and services provisioning in inter-5G+ networks.

B. TRAFFIC ANALYSIS IN SDN-ENABLED 5G+ NETWORKS

The different ports in the SDN architectures provide critical information regarding the traffic flows and packets. The following research challenges require to analyse and process the SDN information and traffic characteristics to improve network performance and support SDN applications.

- Correlating the analysis of SDN information with the traffic statistics on the ports of SDN architectures is

one way to distinguish the elephant flows in each traffic data stream and group them based on a certain measure of similarity [25]. The volume and complexity of the traffic heterogeneity call for efficient artificial intelligence techniques to classify flows and detect on the fly anomalies regarding network or information security. Besides, the computation of wireless resource allocation can be very challenging to meet the SDN time scale operations.

- Real-time network capacity dimensioning through steering traffic flows among APs is very challenging in 5G+ networks. A research challenge lies in dynamically allocating the network capacity based on the current and future traffic while guaranteeing their e2e QoS requirements. New approaches should be developed to support deterministic network models, which can use network calculus to compute the optimal paths for each flow through the priority queues of the different APs in the 5G+ networks. A research challenge lies in developing a scheme that can guarantee e2e real-time QoS communication services across the SDN-enabled ultra dense 5G+ networks.
- 5G+ networks business models require clear workloads triggers for reconfiguring the whole network infrastructure to meet their business (i.e service provisioning) requirements. However, the lack of programmable network control plane has so far hindered the realization of software-defined 5G+ networks. A research challenge lies in developing scheduling and automated resource allocation algorithms, which can consider the use of network state information and traffic characteristics to dynamically manage users to APs association.

C. MOBILITY MODELLING IN SDN-ENABLED 5G+ NETWORKS

As a UE velocity increases, it spends shorter time to traverse the AP coverage. This increases the number of vertical handovers during a fixed time interval. The frequent UE handover results in constantly changing OF rules in the network. The following research challenges require solutions to alleviate the potential impact of mobility on the performance of SDN-enabled 5G+ networks.

- Mobility robustness is a challenge, because moving UEs may switch rapidly among the 5G+ cells. A UE transfer or traffic offloading policy from one LiFi AP to a WiFi AP depends on the AP selection and offloading policies, which, in turn, depends on the number of overlapped APs, the overlapping area size, and the user/network preference. A research challenge lies in developing a mobility manager module that can reacts fast and cope with the number of UEs and small cells in the 5G+ networks. It should leverage the RSSI (Received Signal Strength Indicator) and other parameters tracking primitives to trigger a handover process. The mobility manager module periodically checks if a better handover opportunity exists, even if the channel quality experience by the current APs offering services to the active UEs is still acceptable.
- Mobility management functions can base their decisions on network status beyond local radio or optical wireless channel quality at the cell site (e.g., energy, traffic, and interference awareness), while still providing minimal service interruptions during handovers. A research challenge lies in developing optimization techniques for optimal virtual network functions (VNFs) deployment. For example, operators can implement efficient VNFs that offload user traffic at the network edge, while distributing traffic load among the different core nodes following load balancing policies. Small cells clustering can be done more efficiently with network-supported decisions rather than terminal-supported decisions. One particular research challenge is where to place the VNF pool initially; that is, near the edge or near the core of the network. VNFs with real-time constraints are deployed near the edge and those with coordination requirements near the core. Although this split is intuitive, the deployment scenario, where both requirements are present, is still unexplored.

IX. CONCLUSION

Analytical mathematical models have been developed and evaluated by using MATLAB, which clarify the relationship among (a) service requests rate of applications, (b) queue lengths of northbound and southbound interfaces (c) retrial queue length and (d) controller processing rate. The capacity of SDN controller in terms of processing rate and buffer space should be carefully dimensioned to support more SDN applications and AP switches in a reliable-manner. The SDN controller service rate can shorten the retrial queue length by quickly serving the applications that are already handled in a previous SDN time period. New flow-rules can be set in the APs based on the retrial and rejoining probabilities range, which can be incorporated in the controller's brain to assign flow-rules in the AP switches in a proactive-manner. This significantly reduces the controller set up time and traffic flows blocking probability, which support more SDN applications and services in the HetNet data plane.

A TE scheme has been developed, which runs on the network, MAC and across network-MAC layers. It has routing policies to route traffic flows to the different APs; and a two-level scheduler to transmit traffic packets on the MAC layer of APs according to their service class requirements and target HetNet performance. These functionalities, which run on the network and MAC layers, provide the SDN applications with routing and scheduling algorithms and policies to offer differentiated services and granular resource allocation in the HetNet data plane. Users can leverage the HetNet diversity to receive the data rate from the APs that

can provide them wireless coverage and the requested services. The simulation scenarios have been evaluated in two network settings, which demonstrate the impact of SDN on services provisioning performance and network autonomy in dynamically offering services to the UEs. A number of performance evaluation scenarios have been conducted, which demonstrate the capabilities of TE scheme to support multiple services, HetNet and applications convergence. The proposed TE scheme guarantees the minimum delay and throughput per service class in active and failed HetNet, while improving the target total HetNet network throughput per TSS UE. It also supports service provisioning restoration applications. The TE scheme algorithms ensure efficient resource allocation and seamless LTE/WiFi/LiFi cells integration. It can be deployed in any AP, independent of its underlying wireless technology.

REFERENCES

[1] H. Haas, *High-Speed Wireless Networking Using Visible Light*. Bellingham, WA, USA: SPIE Newsroom, Apr. 2013.

[2] 5GMF. *5G Mobile Communications Systems for 2020 and Beyond*. 5GMF White Paper V1.1, 2017. [Online]. Available: https://5gmf.jp/en/whitepaper/5gmf-white-paper-1-1/

[3] D. Lopez-Perez, M. Ding, H. Claussen, and A. H. Jafari, "Towards 1 Gbps/UE in cellular systems: Understanding ultra-dense small cell deployments," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2078–2101, Jun. 2015.

[4] X. Ge, J. Ye, Y. Yang, and Q. Li, "User mobility evaluation for 5G small cell networks based on individual mobility model," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 528–541, Mar. 2016.

[5] *Technical Specification Group Services and System Aspects*, document TR 21.915 V15.0.0 (2019-09), 3GPP, Mar. 2019, pp. 1–112, vol. 1, no. 15.

[6] *5G: A Technology Vision*, Huawei Technol., Huawei, Shenzhen, Apr. 2013.

[7] E. Westerberg, "4G/5G RAN architecture: How a split can make the difference," Ericsson, Stockholm, Sweden, Tech. Rep. 6, Jul. 2016. [Online]. Available: https://www.ericsson.com/49ec87/assets/local/reports-papers/ericsson-technology-review/docs/2016/etr-ran-architecture.pdf

[8] *Technical Specification Group Services and System Aspects*, document TR 21.915 V1.1.0 (2019-03), 3GPP, Mar. 2019, pp. 1–112, vol. 1, no. 15.

[9] O. Galinina, A. Pyattaev, S. Andreev, M. Dohler, and Y. Koucheryavy, "5G multi-RAT LTE-WiFi ultra-dense small cells: Performance dynamics, architecture, and trends," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 6, pp. 1224–1240, Jun. 2015.

[10] R. Abhishek, D. Tipper, and D. Medhi, "Network virtualization and survivability of 5G networks: Framework, optimization model, and performance," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6.

[11] 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.

[12] 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, Mar. 2008.

[13] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *Proc. 2nd ACM SIGCOMM workshop Hot Topics Softw. Defined Netw. HotSDN*, New York, NY, USA, Dec. 2013, pp. 25–30.

[14] Y.-D. Lin, P.-C. Lin, C.-H. Yeh, Y.-C. Wang, and Y.-C. Lai, "An extended SDN architecture for network function virtualization with a case study on intrusion prevention," *IEEE Netw.*, vol. 29, no. 3, pp. 48–53, May 2015.

[15] S. Gringeri, N. Bitar, and T. J. Xia, "Extending software defined network principles to include optical transport," *IEEE Commun. Mag.*, vol. 51, no. 3, pp. 32–40, Mar. 2013.

[16] *O. vSwitch*. Accessed: Apr. 9, 2020. [Online]. Available: https://www.openvswitch.org/

[17] 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)*, San Francisco, CA, USA, Sep. 2011, pp. 1–7.

[18] J. W. Cohen, *The Single Server Queue*, vol. 8. New York, NY, USA: North-Holland, 1982.

[19] B. D. Choi, Y. C. Kim, and Y. W. Lee, "The M/M/c retrial queue with geometric loss and feedback," *Comput. Math. with Appl.*, vol. 36, no. 6, pp. 41–52, Sep. 1998.

[20] 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. for Netw. Commun. Syst. ANCS*, San Jose, CA, USA, Nov. 2008, pp. 1–5.

[21] A. Bianco, R. Birke, L. Giraudo, and M. Palacin, "OpenFlow switching: Data plane performance," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.

[22] A. Khan and N. Dave, "Enabling hardware exploration in software-defined networking: A flexible, portable OpenFlow switch," in *Proc. IEEE 21st Annu. Int. IEEE Symp. Field-Program. Custom Comput. Mach.*, Seattle, WA, USA, Jun. 2013, pp. 145–148.

[23] S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour, and D. Simeonidou, "An analytical model for software defined networking: A network calculus-based approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 1397–1402.

[24] R. L. Cruz, "A calculus for network delay. I. network elements in isolation," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.

[25] L. Yang, B. Ng, and W. K. G. Seah, "Heavy hitter detection and identification in software defined networking," in *Proc. 25th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2016, pp. 1–10.

[26] J. Huang, Y. He, Q. Duan, Q. Yang, and W. Wang, "Admission control with flow aggregation for QoS provisioning in software-defined network," in *Proc. IEEE Global Commun. Conf.*, Atlanta, GA, USA, Dec. 2014, pp. 1182–1186.

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

[28] L. Jorguseski, A. Pais, F. Gunnarsson, A. Centonza, and C. Willcock, "Self-organizing networks in 3GPP: Standardization and future trends," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 28–34, Dec. 2014.

[29] J. Kim, H.-W. Lee, and S. Chong, "Virtual cell beamforming in cooperative networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1126–1138, Jun. 2014.

[30] S. Chen, F. Qin, B. Hu, X. Li, and Z. Chen, "User-centric ultra-dense networks for 5G: Challenges, methodologies, and directions," *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 78–85, Apr. 2016.

[31] O. Tickoo and B. Sikdar, "Modeling queueing and channel access delay in unsaturated IEEE 802.11 random access MAC based wireless networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 878–891, Aug. 2008.

[32] Z. Shu, J. Wan, J. Lin, S. Wang, D. Li, S. Rho, and C. Yang, "Traffic engineering in software-defined networking: Measurement and management," *IEEE Access*, vol. 4, pp. 3246–3256, 2016.

[33] M. T. Z. Win, Y. Ishibashi, and K. T. Mya, "QoS-aware traffic engineering in software defined networks," in *Proc. 25th Asia–Pacific Conf. Commun. (APCC)*, Nov. 2019, pp. 171–176.

[34] Y. Lin, Y. Gao, Y. Li, X. Zhang, and D. Yang, "QoS aware dynamic uplink-downlink reconfiguration algorithm in TD-LTE HetNet," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 708–713.

[35] S. Namal, I. Ahmad, A. Gurtov, and M. Ylianttila, "SDN based inter-technology load balancing leveraged by flow admission control," in *Proc. IEEE SDN Future Netw. Services (SDN4FNS)*, Trento, Italy, Nov. 2013, pp. 1–5.

[36] L. Takacs, *Introduction to the Theory of Queues*. New York, NY, USA: Oxford Univ. Press, 1962.

[37] J. M. Smith, "Optimal design and performance modelling of M/G/1/K queueing systems," *Math. Comput. Model.*, vol. 39, nos. 9–10, pp. 1049–1081, May 2004.

[38] H. Alshaer and H. Haas, "Bidirectional LiFi attocell access point slicing scheme," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 909–922, Sep. 2018.

[39] S. Shenker and A. Weinrib, "The optimal control of heterogeneous queueing systems: A paradigm for load-sharing and routing," *IEEE Trans. Comput.*, vol. 38, no. 12, pp. 1724–1735, Dec. 1989.

[40] F. Bonomi and A. Kumar, "Adaptive optimal load balancing in a non-homogenous Multiserver system with a central job scheduler," *IEEE Trans. Comput.*, vol. 39, no. 10, pp. 1232–1250, Oct. 1990.
[41] S. T. Atan, "Solution methods for controlled queueing networks," M.S. thesis, Iowa State Univ., Ames, IA, USA, Jun. 1997.
[42] H. F. Chen, *Recursive Estimation Control Stochastic System*. New York, NY, USA: Wiley, 1985.
[43] A. Kumar, "Adaptive load control of the central processor in a distributed system with a star topology," *IEEE Trans. Comput.*, vol. 38, no. 11, pp. 1502–1512, Nov. 1989.
[44] B. Baynat, "Analytical models for dimensioning of OFDMA-based cellular networks carrying VoIP and best effort traffic," *Int. J. Comput. Netw.*, vol. 4, no. 4, pp. 104–134, 2012.
[45] S. Doirieux, B. Baynat, M. Maqbool, and M. Coupechoux, "An analytical model for WiMAX networks with multiple traffic profiles and throttling policy," in *Proc. 7th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw.*, Jun. 2009, pp. 1–8.
[46] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," *Wireless Netw.*, vol. 8, no. 1, pp. 13–26, 2002.
[47] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queuing system with asynchronously varying service rates," *Probab. Eng. Informational Sci.*, vol. 18, no. 2, pp. 191–217, Apr. 2004.
[48] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150–154, Nov. 2001.
[49] S. Shakkottai and A. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real time data in HDR," *Teletraffic Sci. Eng.*, vol. 1, Jan. 2001, pp. 793–804.
[50] E. Hyytiä and J. Virtamo, "Random waypoint mobility model in cellular networks," *Wireless Netw.*, vol. 13, no. 2, pp. 177–188, Apr. 2007.
[51] E. Hyytia, P. Lassila, and J. Virtamo, "Spatial node distribution of the random waypoint mobility model with applications," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 680–694, Jun. 2006.
[52] J. R. Barry, J. M. Kahn, W. J. Krause, E. A. Lee, and D. G. Messerschmitt, "Simulation of multipath impulse response for indoor wireless optical channels," *IEEE J. Sel. Areas Commun.*, vol. 11, no. 3, pp. 367–379, Apr. 1993.
[53] Z. Chen, N. Serafimovski, and H. Haas, "Angle diversity for an indoor cellular visible light communication system," in *Proc. IEEE 79th Veh. Technol. Conf. (VTC Spring)*, Seoul South Korea, May 2014, pp. 1–5.
[54] H. Alshaer and H. Haas, "SDN-enabled Li-Fi/Wi-Fi wireless medium access technologies integration framework," in *Proc. IEEE Conf. Standards for Commun. Netw. (CSCN)*, Berlin, Germany, Oct. 2016, pp. 1–6.
[55] D. Tsonev, S. Sinanovic, and H. Haas, "Practical MIMO capacity for indoor optical wireless communication with white LEDs," in *Proc. IEEE 77th Veh. Technol. Conf. (VTC Spring)*, Jun. 2013, pp. 1–5.
[56] Y. L. Lee, J. Loo, T. C. Chuah, and A. A. El-Saleh, "Fair resource allocation with interference mitigation and resource reuse for LTE/LTE–A femtocell networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8203–8217, Oct. 2016.
[57] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell networks: A survey," *IEEE Commun. Mag.*, vol. 46, no. 9, pp. 59–67, Sep. 2008.
[58] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
[59] M. A. A. Masri, A. B. Sesay, and A. O. Fapojuwo, "Session state aware handover procedure for VoIP sessions in heterogeneous wireless networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Istanbul, Turkey, Apr. 2014, pp. 3011–3016.

HAMADA ALSHAER (Senior Member, IEEE) received the Ph.D. degree in computer engineering and telecommunications from Pierre et Marie Curie University, France, in 2005. He is currently involved in software defined wireless networking, security, and mobile networks virtualization research development with the LiFi Research and Development Centre, School of Engineering, The University of Edinburgh, U.K. He has researched for over 18 years with British Telecom, Etisalat, INRIA, France, and Leads University, in academic and industrial research development. He has authored over 50 articles in various research topics in electrical and computer engineering, including information technology and systems, and authored a book entitled *Demanding Traffic Control and Management in Next Generation Networks: QoS Analysis, Network Simulation, Performance Modeling* (LAP LAMBERT Academic Publishing, 2009). His research interests include cross-layer design for vehicular networks, software-defined wireless networking, wireless networks virtualization, mobile communications security, optical communications and networking, wireless sensor networks, and intelligent transportation. He was a recipient of the 2009 Royal Academy of Engineering Travel Award and scholarships from UNRWA, French Government, and Pierre et Marie Curie University for academic distinctions. He is a Regular Reviewer of research grants submitted for prestigious funding bodies in Europe, Asia, and Middle East. He has served on the Technical Program Committee for various IEEE conferences, including the IEEE Intelligent Vehicles Symposium, the Vehicular Technology Conference, the IEEE GLOBECOM, ICC, and WCNC, and chaired some of their sessions.

HARALD HAAS (Fellow, IEEE) received the Ph.D. degree from The University of Edinburgh, in 2001. He is currently the Chair of mobile communications with The University of Edinburgh, the Initiator, the Co-Founder, and the Chief Scientific Officer of pureLiFi Ltd., and the Director of the LiFi Research and Development Centre, The University of Edinburgh. He has authored 500 conference papers and journal articles. His main research interests are in optical wireless communications, hybrid optical wireless and RF communications, spatial modulation, and interference coordination in wireless networks. He is a Fellow of the Royal Academy of Engineering. He was a recipient of the Prestigious Established Career Fellowship from the Engineering and Physical Sciences Research Council (EPSRC), U.K., in 2012 and 2017. In 2014, he was selected by EPSRC as one of ten Recognizing Inspirational Scientists and Engineers Leaders in U.K. He was a co-recipient of the EURASIP Best Paper Award for the *Journal on Wireless Communications and Networking*, in 2015, the Jack Neubauer Memorial Award of the IEEE Vehicular Technology Society, and recent best paper awards at VTC-Fall, 2013, VTC-Spring 2015, ICC 2016, ICC 2017, and ICC 2018. He received the Outstanding Achievement Award from the International Solid State Lighting Alliance, in 2016, and the James Evans Avant Garde Award of the IEEE Vehicular Technology Society, in 2019. He gave two TED Global talks, Wireless Data From Every Light Bulb and Forget Wi-Fi: Meet the New Li-Fi Internet, which together have been downloaded more than 5.5 million times. He is an Associate Editor of the IEEE JOURNAL OF LIGHTWAVE TECHNOLOGY.