Abstract—In today's wireless networks, a variety of Radio Access Technologies (RATs) are present. However, each RAT being controlled individually leads to suboptimal utilization of network resources. Due to the remarkable growth of data traffic, interworking among different RATs is becoming necessary to overcome the problem of suboptimal resource utilization. Software Defined Networking (SDN) facilitates the unified control and management of different RATs. In this paper, we propose an SDN based architecture which can handle multiple RATs together. Users can be offloaded from one RAT to another based on loads of different networks, channel conditions and priority of users. We consider the optimal RAT selection problem in a Long Term Evolution (LTE)-Wireless Fidelity (WiFi) network where we aim to maximize the total system throughput subject to constraints on the blocking probability of high priority users and the offloading probability of low priority users. The problem is formulated as a Constrained Markov Decision Process (CMDP). We reduce the effective dimensionality of the action space by eliminating the provably suboptimal actions. We propose low-complexity online heuristics for RAT selection which can operate without the knowledge regarding the statistics of system dynamics. We implement the proposed algorithms in an SDN based evaluation platform developed by us using Network Simulator-3 (ns-3). It is observed that the proposed algorithms offer near-optimal performances and outperform traditional RAT selection algorithms under realistic network scenarios including user mobility.

Index Terms—User association, LTE-WiFi offloading, CMDP, Software defined networking.

I. INTRODUCTION

In the recent years, the number of mobile subscribers has increased exponentially with a rise in the popularity of data-intensive applications such as video, social networking. To address the problem of increasing data traffic consumption and demand for high data rate, small cells are being deployed by network operators. Moreover, interworking with IEEE 802.11 based Wireless Local Area Network (WLAN) (popularly known as Wireless Fidelity (WiFi) network) Access Points (APs) is also becoming popular. The reason behind this is twofold. First, WiFi AP deployment is low cost as they operate in unlicensed band. Moreover, while the Third Generation Partnership Project (3GPP) 4G Long Term Evolution (LTE) Base Stations (BSs) primarily target to provide ubiquitous coverage to support mobility of users, WiFi APs aim to provide high data rate in hotspot regions. Such networks where different types of Radio Access Technologies (RATs) are present and a user can be associated with any RAT, are known as Heterogeneous Networks (HetNets). With the upcoming 3GPP Fifth Generation (5G) (see [2], [3]) networks, the future wireless network is expected to be a mixture of variety of RATs, necessitating a tighter interworking among RATs. Therefore, for an efficient interworking between various RATs in a 5G based HetNet, the need for a unified framework enabling a global view of different RATs where control and management decisions are taken by a common entity becomes even more important. In the absence of a global view, as observed in today’s network (e.g., LTE and Wifi), the utilization of network resources becomes suboptimal. In the 3GPP 5G, Non 3GPP Interworking Function (N3IWF) is standardized for seamlessly integrating non-3GPP RATs such as WLAN with the 5G core. Even though the 3GPP 5G standardization introduces a unified core, Radio Access Network (RAN) level decisions are still taken in a RAT-specific manner. However, optimal performance can be obtained if the common RAN functionalities of different RATs such as admission control and mobility management are controlled and managed within a unified framework. The concept of Software Defined Networking (SDN) (see [4]–[6]) may be instrumental in achieving unified control of RATs.

The basic idea behind SDN is the split of control and data plane elements in a network. While the control plane comprises of control and management protocols, the data plane comprises of protocols for data transfer. Using SDN, the control plane functionalities can be decoupled from the network elements of various RATs and aggregated in a centralized control plane. The interface which separates control and data planes, enables the configuration of elements of data plane using policy-based rules provided by control plane elements. The control plane elements have information regarding network parameters (e.g., load, channel states of users) of all RATs which are used to design the policy-based rules. These parameters are forwarded by RAT-specific elements (viz., 5G New Radio (NR) or LTE BS and WiFi AP) to the control plane elements. Since the control plane has a global view of the entire network, this approach allows the design of policy-based rules in an optimal manner. Therefore, optimal utilization of network resources can be achieved contrary to distributed control in traditional network.

Motivated by this, we propose an architecture where different RATs are controlled and managed using an SDN controller. As an example in this paper, we consider an LTE-WiFi HetNet where users of different priorities are present. Control and management functionalities of these RATs are unified at the SDN controller. Radio resource control and management messages received by LTE BSs and WiFi APs are forwarded to the SDN controller. Contrary to traditional networks where LTE BSs and WiFi APs take individual control and management
decisions, in the proposed architecture, the SDN controller takes control and management decisions within a unified framework.

Note that although this paper considers LTE and WiFi RATs, the proposed architecture can be easily applied to 3GPP 5G NR also. Unlike the 3GPP 5G which does not possess an integrated RAN, the proposed architecture can handle radio access functionalities of multiple RATs in a unified manner. Among the RAN functionalities, we consider the RAT selection problem. We assume that high priority users (like voice over internet protocol, live streaming) are always served using cellular network since WiFi may not provide the required Quality of Service (QoS) in terms of delay and packet loss. Low priority users are best effort class of users which may be served using cellular network or WiFi.

In our earlier works [7]–[9], we have addressed the trade-off between the total system throughput and the blocking probability of high priority users in an LTE-WiFi HetNet. Since low priority users are best effort in nature, blocking probability of low priority users need not be taken into consideration. However, maximizing the total system throughput subject to a blocking probability constraint, may lead to a ‘ping-pong’ kind of behavior since the optimal policy may result in offloading [10] of a low priority user from LTE to WiFi and back to WiFi again within a short time interval when a high priority user arrival is followed by a departure. Similar instances can occur where a departure is followed by an arrival. To address the issue of additional control signaling traffic in the backhaul due to this ping-pong behavior, we incorporate an additional constraint on the offloading probability of low priority users (i.e., fraction of offloaded low priority users). We thus aim to maximize the total system throughput subject to the high priority user blocking probability and the low priority user offloading probability constraints.

The above problem is modeled as a Constrained Markov Decision Process (CMDP) problem. We establish the sub-optimality of various actions in different states of the system and thereby, reduce the effective dimensionality of the action space. However, even after reducing the size of the action space, conventional Dynamic Programming (DP) methods to solve the CMDP problem are computationally prohibitive under large state spaces. Moreover, DP methods require the knowledge of transition probabilities between different states which depend on the unknown statistics of system dynamics, viz., the arrival rates of low and high priority users. These are hard to gather in reality. To address these issues, we propose two online RAT selection heuristic algorithms. Unlike DP based algorithms, the proposed algorithms do not require the knowledge of the statistics of system dynamics. Moreover, the proposed algorithms have low computational and storage complexities. These features make the algorithms suitable for practical online implementation.

We implement the proposed RAT selection algorithms in an SDN based evaluation platform developed using Network Simulator-3 (ns-3) (a discrete event network simulator) [11] according to the proposed SDN based LTE-WiFi network architecture. Experimental results illustrate that the proposed algorithms provide near-optimal performances. Moreover, performances of the proposed algorithms are compared with state-of-the-art RAT selection scheme under various practical scenarios including user mobility.

A. Related Work

Due to the emergence of the SDN paradigm, unification of control plane functionalities of different RATs is facilitated. In [5], an SDN based architecture is proposed to achieve control and management of various RATs in an end-to-end manner. RAT selection and offloading are among the control plane functionalities which are traditionally implemented either in distributed [12]–[17] or centralized [7]–[9], [18]–[22] manner. An overview of existing RAT selection techniques in HetNets and their performance evaluation is presented in [23].

Centralized RAT selection strategies can be implemented in an SDN based framework. An integrated user association 1 and interference management problem in a two-tier HetNet is considered in [19]. Although the authors propose a computationally efficient algorithm, this approach is not adaptable to fast changes in network parameters. In [22], an admission control algorithm which maximizes the users’ quality of experience is proposed in a macro cell-small cell HetNet. The authors show that the optimal policy performs better than the random policy. In our earlier work [7], we propose a computationally efficient network-initiated RAT selection algorithm which maximizes the total system throughput subject to a blocking probability constraint. However, it requires the knowledge of the state transition probabilities of the underlying Markov chain which depend on the statistics of unknown system dynamics. Subsequently, we propose learning algorithms in [8] which can work without the knowledge of the statistics of unknown system dynamics and unlike [7], can be implemented online. The convergence speed of the traditional Q-learning based algorithm in [8] is further improved in [9] by exploiting the structural properties of the optimal policy.

RAT selection and offloading problems in the context of SDN-based HetNets are investigated recently in the literature [24]–[28]. The authors in [26] consider an SDN-enabled dynamic path selection problem in a multi-RAT system and propose an algorithm which chooses the path based on the rate obtained. The rate obtained takes into account radio and load conditions and performance requirements of different flows. In [27], a QoS-aware RAT selection algorithm is proposed in an SDN based HetNet. RAT selection is performed based on a metric which takes into account, bit rate requirements of users and capabilities of different RATs. In [28], the performance of a multi-class user association heuristic which scales well with the LTE-WiFi HetNet system, is evaluated in an SDN based testbed.

Among the distributed solutions, the authors in [12] propose an association scheme which maximizes the network utility subject to constraints on user requirements. The proposed scheme is based on the utility obtained from past associations of users. In [17], a low complexity RAT selection algorithm is proposed for an LTE network comprising macro, pico and femto cells. The proposed algorithm achieves a near-optimal performance with a theoretical guarantee on the performance.

Contrary to distributed approaches which focus on optimizing individual user utilities and hence, often may not provide the globally optimal solution, centralized approaches provide a framework for overall system optimization. Moreover, in [29], the authors demonstrate that network-centric resource al-

1The terminologies “association” and “RAT selection” are used interchangeably throughout this paper.
location approaches perform better than distributed approaches in LTE-WiFi systems. Hence, we focus on network-initiated centralized approaches for RAT selection and offloading in an SDN based multi-RAT network. The trade-off involving the total system throughput, the blocking probability and the offloading probability in a dynamic system within an optimization framework has not been considered in the literature before. Furthermore, unlike others, we investigate the role of offloading in improving the system performance at the time instances of arrivals and departures of users.

B. Our Contributions

In this paper, we propose an SDN based architecture for a multi-RAT system where users of different priorities and channel conditions are present. We consider the problem of optimal RAT selection in an LTE-WiFi HetNet where we aim to maximize the total system throughput subject to constraints on the high priority user blocking probability and the low priority user offloading probability. Our contributions are summarized as follows:

- We propose an SDN-based architecture for the control and management of various RATs. While the 3GPP 5G specifications [2] introduce a unified core but handle the RAN functionalities in a RAT-specific manner, our architecture handles the RAN functionalities of various RATs in an integrated manner.
- Optimal association problem of maximizing the total system throughput subject to constraints on the high priority user blocking probability and the low priority user offloading probability is formulated as a CMDP problem.
- We prove the sub-optimality of certain actions in different states. This reduces the size of the effective action space.
- We propose two low complexity heuristics for RAT selection. They do not require the knowledge of the user arrival rates and hence, can be implemented online.
- We develop an SDN based LTE-WiFi evaluation platform in ns-3 to implement the RAT selection algorithms. Building of the platform requires significant modifications of existing ns-3 elements. Simulation results illustrate that the algorithms perform close to optimality.
- We also compare the performances of the proposed algorithms with that of traditional RAT selection algorithm under realistic scenarios including user mobility.

The rest of the paper is organized as follows. In Section II, we present the system architecture. Section III describes the system model. In Section IV, the problem formulation within the framework of CMDP is described. In Section V, we derive the suboptimal actions and eliminate them from the action space. We describe the proposed algorithms in Section VI with a comparison of storage and computational complexities. Performance analysis of the proposed algorithms in the SDN-based evaluation platform is provided in Section VII. Section VIII concludes the paper.

II. PROPOSED SYSTEM ARCHITECTURE

In this section, we propose an architecture which unifies the control and management of multiple RATs using an SDN controller. We demonstrate the details of the architecture considering LTE and WiFi as example RATs in this paper. Control and management functionalities of these two RAT-specific access networks are aggregated in the SDN controller, as illustrated in Fig.1. For this purpose, Radio Resource Management (RRM) module of LTE BS is moved to the SDN controller. LTE RRM comprises radio bearer control, radio admission control, connection mobility control, inter-cell interference coordination and dynamic resource allocation. In this work, we focus on aspects related to RAT selection, viz., radio bearer, admission and mobility control. RRM control messages sent by the users are forwarded from LTE BS to the controller. Similarly, LTE BS forwards the control messages sent by the SDN controller to the users. For example, RRC connection request (response) sent by the user (controller) reaches the controller (user) via LTE BS. Similarly, WiFi AP forwards the association request message to the SDN controller. Other functionalities of LTE BS except RRM, remains in LTE BS. Therefore, all other legacy control message exchanges (not related to RRM) in LTE remain as it is there in existing LTE network. For the communication between LTE BS/ WiFi AP and the SDN controller, we define a control plane protocol which encapsulates LTE and WiFi control messages (such as RRC connection request and association request) by identifying them using headers. The protocol works over Transmission Control Protocol (TCP). Channel condition information of users which are required for association decisions, is forwarded to the SDN controller at the time of association of a user.

Scalability is an important system parameter from the architecture design perspective. In our case, since only RRM related control procedures in LTE and control and management procedures in WiFi are handled by the controller, the proposed architecture is more scalable than a fully centralized SDN architecture where all the control and management functionalities of LTE and WiFi may be moved to the controller. Moreover, the proposed architecture requires lesser changes to the existing network than a typical fully centralized architecture. We consider LTE and WiFi as candidate RATs to demonstrate the capabilities of the proposed architecture. However, it can be used to incorporate other RATs such as 5G NR.

III. SYSTEM MODEL
The system model considered in this paper is analogous to those of [7]–[9]. However, in these works, channel states of users are not taken into consideration. The system model consists of an LTE BS and a WiFi AP. As described in Fig. 2, both LTE AP and WiFi AP which is present inside the coverage area of LTE BS, are connected to the SDN controller via high capacity lossless links. High and low priority users are assumed to be present anywhere within the coverage area of LTE BS. Since low priority users present outside the coverage area of WiFi AP always associates with LTE BS, in the state space, we take into consideration only those low priority users which are present in the common coverage area of LTE BS and WiFi AP. In this case, the controller needs to choose a decision whether the user is to be associated with WiFi AP or LTE BS, based on the system state. Both high and low priority users are allotted resources in LTE from a common set of resources. High priority users are provided a Guaranteed Bit Rate (GBR) following which available resources in LTE are uniformly allocated among low priority users. We further assume that in LTE, high and low priority users can be in either “good” or “bad” channel state. Based on the location of users, the coverage area of an LTE BS is assumed to be divided into two regions, viz., cell center and cell edge regions. Since cell edge users are present in the vicinity of the cell boundary, usually they receive weaker signal strength than that of cell center users. Therefore, it is assumed that users present in the cell center region have good channels, whereas cell edge users have bad channels. Cell center/ cell edge region can be chosen based on the average Channel Quality Indicator (CQI) experienced by the users in LTE. If the average CQI of a user exceeds a certain threshold, then the user is called a cell center user. Otherwise, it is called a cell edge user. Since RAT selection decisions are made for a sufficiently long period of time, we assume that users are distributed in cell edge/cell center region depending on their average radio conditions. We assume that instantaneous fading effects are averaged out over the timescale over which decisions are taken. We consider that the users are stationary, and the channel state of a user does not change with time once the user is admitted. The channel states of incoming users are assumed to be known at the SDN controller, however, the channel states in LTE are random (good/bad) with finite probabilities. Since WiFi AP has a small coverage area, it is assumed that the channel states of users in WiFi are always good.

Let high and low priority user arrivals be independent Poisson processes with means $\lambda_H$ and $\lambda_L$, respectively. Following [30], the service times for high and low priority users are assumed to be exponentially distributed with means $\frac{1}{\mu_H}$ and $\frac{1}{\mu_L}$, respectively. We also assume that low priority users use applications such as video where the duration of a session does not depend on the number of users.

**Remark 1.** One LTE BS-one WiFi AP scenario (considered for brevity for notation) in this paper can be extended easily for multiple BSs-multiple APs case. If coverage areas of BSs (APs) do not overlap, users present inside the coverage area of each BS (AP) can be considered as a tuple in the state space. In the case of overlapping coverage areas, the problem can be cast into the non-overlapping coverage area case and analysis follows in a similar way. This can be performed using some simple criterion, say, mapping every geographical point to the BS (AP) which provides the highest signal-to-noise ratio.

### A. State Space

The system can be viewed as a controlled continuous time stochastic process $\{X(t)\}_{t \geq 0}$, similar to [7]–[9]. Any state $s$ in the state space $\mathcal{S}$ is expressed as $s = (i_G, i_B, j_G, j_B, k_G, k_B)$, where $i_G, i_B$ denote the number of high priority users associated with LTE BS with good and bad channels in LTE, $j_G, j_B$ denote the number of low priority users associated with LTE BS with good and bad channels in LTE, and $k_G, k_B$ denote the number of low priority users associated with WiFi AP, however, with respect to LTE they have good and bad channels, respectively. Channel states of users in WiFi are not explicitly mentioned since channel states of users in WiFi are always assumed to be good. The arrival and departure of high and low priority users with good and bad channel states in LTE are taken as decision epochs. It is evident that the system changes state only at the decision epochs. Moreover, since the system is Markovian, it is sufficient to observe the system state at these decision epochs and not at other time points.

An arrival or a departure of a user with good/bad channel state in LTE is referred to as an event. Whenever an event happens, the system changes state. Let the set of all events be denoted by $E$. Let us denote the arrival of a high and low priority user with good (bad) channel by $E_1(E_3)$ and $E_2(E_4)$, respectively. Let the departure of a high and low priority user with good (bad) channel be denoted by $E_5(E_6)$ and $E_7(E_8)$, respectively. We assume that the departure of a low priority user from WiFi with good and bad channel in LTE are denoted by $E_9$ and $E_{10}$, respectively. Note that, the channel states of users in WiFi do not appear in the event space because the channel states of users in WiFi are assumed to be good. At every decision epoch, a decision is chosen by the controller based on the event and the current system state. Based on the decision chosen, the system moves to a different state with finite probability.

Let the LTE system be composed of $C_L$ resource blocks. We assume that $s = (i_G, i_B, j_G, j_B, k_G, k_B) \in \mathcal{S}$ if $i_G + j_G + i_B + j_B \leq C_L$, $(j_G + j_B) \leq N$ and $(k_G + k_B) < W$, where $N$ is a sufficiently large positive integer (incorporated for analytical tractability). The first condition signifies that a high priority user is admitted only when sufficient resources are available. The first condition is under the assumption that a high priority user with bad channel requires $p_r(>1)$ times as many resource blocks as required by a high priority user with good channel. The quantity $W$ signifies the maximum number of users that can be supported in WiFi to guarantee a specified minimum per-user throughput. Note that the per-user throughput of WiFi decreases monotonically with load [31]. Let the GBR required by a high priority user be denoted by $R_{L,H}$. A fixed number of resource blocks are allocated to a high priority user based on the channel condition of the user. Low priority users are best-effort in nature. Therefore, the available resources in LTE are allocated uniformly among low priority users. The data rate obtained by a low priority user depends on the channel states of the users and the number of high priority users. We assume that the bit rate of a low priority user with bad channel is $\frac{1}{d}(d > 1)$ times that of a low priority user with good channel, where $d$ is a constant.
B. Action Space

Let us denote the action space by $A$. Action $A_1$ blocks an arriving user or does nothing during a departure. Actions $A_2$ and $A_3$ correspond to association with LTE and WiFi, respectively. Note that actions $A_1$, $A_2$, $A_3$ are identical to those of $[7]$. Under action $A_4$, a high priority user is associated with LTE and a low priority user with bad channel is offloaded to WiFi. Action $A_5$ performs offloading of a low priority user with bad (good) channel from LTE (WiFi) to WiFi (LTE) when a user departs from WiFi (LTE). Action $A_6$ associates a high priority user with LTE and offloads a low priority user with good channel to WiFi. Action $A_7$ offloads a low priority user with good (bad) channel from LTE (WiFi) to WiFi (LTE) when a user departs from WiFi (LTE). In case of high and low priority user arrivals, the feasible action sets are $\{A_1, A_2, A_4, A_6\}$ and $\{A_2, A_3\}$, respectively. The feasible action set for departures comprises $A_3$, $A_5$ and $A_7$. Note that blocking is a feasible action for high priority users only when the system is non-empty. Low priority users are blocked only when $(j_G+j_B)$ becomes $N$.

<table>
<thead>
<tr>
<th>$a$</th>
<th>$E_l$</th>
<th>${(\lambda_{i_B},i_G,j_G,j_B,k_G,k_B)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$E_1 \cap (E_2 \cup E_3)^\delta$</td>
<td>$({i_B, i_G, j_G, j_B, k_G, k_B})$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>$E_2$</td>
<td>$({i_B + 1, i_G, j_G, j_B, k_G, k_B})$</td>
</tr>
<tr>
<td>$A_3$</td>
<td>$E_3$</td>
<td>$({i_G + 1, j_B, i_B, j_G, k_G, k_B})$</td>
</tr>
<tr>
<td>$A_4$</td>
<td>$E_4$</td>
<td>$({i_B, i_G, j_G, j_B, k_G, k_B + 1})$</td>
</tr>
<tr>
<td>$A_5$</td>
<td>$E_5$</td>
<td>$({i_B, i_G, j_G, j_B, k_G + 1, k_B})$</td>
</tr>
<tr>
<td>$A_6$</td>
<td>$E_6$</td>
<td>$({i_B + 1, i_G, j_G, j_B, k_G + 1, k_B})$</td>
</tr>
<tr>
<td>$A_7$</td>
<td>$E_7$</td>
<td>$({i_B + 1, i_G, j_G, j_B, k_G + 1, k_B + 1})$</td>
</tr>
</tbody>
</table>

C. Transition Probabilities

Let $\delta = (i_G',i_B',j_G',j_B',k_G',k_B')$, and $e_{\{1\leq i \leq 6\}}$ be a set of 6 dimensional vectors where all elements except $i_B'$ element (which is ‘$1$’) is zero. Let $p_g$ denote the probability that the channel state of the arriving user in LTE is good. Then,

$$p_{s's}(a) = \begin{cases} 
\frac{\lambda_{i_B}p_g}{\lambda_{i_B}p_g + \lambda_{i_B}1 - p_g}, & s' = \hat{s} \\
\frac{\lambda_{i_B}1 - p_g}{\lambda_{i_B}p_g + \lambda_{i_B}1 - p_g}, & s' = \hat{s} + e_1 \\
\frac{\lambda_{i_B}1 - p_g}{\lambda_{i_B}p_g + \lambda_{i_B}1 - p_g}, & s' = \hat{s} + e_2 \\
\frac{\lambda_{i_B}1 - p_g}{\lambda_{i_B}p_g + \lambda_{i_B}1 - p_g}, & s' = \hat{s} + e_3 \\
\frac{\lambda_{i_B}1 - p_g}{\lambda_{i_B}p_g + \lambda_{i_B}1 - p_g}, & s' = \hat{s} + e_4 \\
\frac{\lambda_{i_B}1 - p_g}{\lambda_{i_B}p_g + \lambda_{i_B}1 - p_g}, & s' = \hat{s} + e_5 \\
\frac{\lambda_{i_B}1 - p_g}{\lambda_{i_B}p_g + \lambda_{i_B}1 - p_g}, & s' = \hat{s} + e_6.
\end{cases}$$

Values of $i_B',i_G',j_G',j_B',k_G',k_B'$ as a function of action $a$ and event $E_l$ are described in Table I.

D. Rewards and Costs

Based on the system state and the action, a finite reward rate is obtained. In WiFi, the total throughput is a function of the total load of WiFi comprising low priority users with good and bad channels in LTE. Let the reward rate for state $s$ and action $a$ be denoted by $r(s,a)$. Under full buffer traffic WiFi model [31], let $R_{W,D}(k)$ be the per-user throughput when $k$ users are present in WiFi. $R_{W,D}(k)$ is a function of success and collision probabilities (which signify the contention-driven medium access of WiFi users) and slot times for idle, busy (due to collision) and successful transmissions. The reward rate under a state-action pair is the sum of throughput of all users in LTE and WiFi under an action. Let us define

$$R(\hat{i}_G,i_B,j_G,j_B,k_G,k_B) = (i_G + i_B)R_{L,H} + (j_B + j_G)R_{L,L}\mathbb{1}_{(j_G+j_B)>0} + (k_G + k_B)R_{W,D}(k_G + k_B),$$

where $R_{L,L}$ is the data rate obtained when single LTE resource block is allocated to a low priority data user with good channel condition. The complete description of reward rates in state $s$ under different event-action pairs is given in Table II.

```
<table>
<thead>
<tr>
<th>$a$</th>
<th>$E_l$</th>
<th>$r(s,a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$E_1$</td>
<td>$R_{L,H}$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>$E_2$</td>
<td>$R_{W,D}(i_G + i_B)$</td>
</tr>
<tr>
<td>$A_3$</td>
<td>$E_3$</td>
<td>$R_{L,H}$</td>
</tr>
<tr>
<td>$A_4$</td>
<td>$E_4$</td>
<td>$R_{W,D}(i_G + i_B)$</td>
</tr>
<tr>
<td>$A_5$</td>
<td>$E_5$</td>
<td>$R_{W,D}(i_G + i_B)$</td>
</tr>
<tr>
<td>$A_6$</td>
<td>$E_6$</td>
<td>$R_{W,D}(i_G + i_B)$</td>
</tr>
<tr>
<td>$A_7$</td>
<td>$E_7$</td>
<td>$R_{W,D}(i_G + i_B)$</td>
</tr>
</tbody>
</table>
```

We consider cost functions due to blocking and offloading, respectively. Let the cost rates for blocking and offloading in state $s$ under action $a$ be denoted by $c_b(s,a)$ and $c_o(s,a)$, respectively. Whenever a high priority user is blocked, $c_b(s,a)$ is unity, else it is zero. Therefore,

$$c_b(s,a) = \begin{cases} 
1, & \text{if high priority users are blocked,} \\
0, & \text{otherwise}. 
\end{cases}$$

Whenever one low priority user is offloaded from one RAT to another, $c_o(s,a)$ is unity, else it is zero. Therefore,

$$c_o(s,a) = \begin{cases} 
1, & \text{if } a = (A_4||\ldots||A_7), \\
0, & \text{otherwise}. 
\end{cases}$$

IV. Problem Formulation & Solution Techniques

We target to determine an association policy which maximizes the total system throughput subject to constraints on the blocking probability of high priority users and the offloading probability of low priority users. A policy is a mapping from a state to an action specifying the action to be chosen in a state. Arrivals and departures of high and low priority users can occur at any point in time. Therefore, the considered problem is formulated as a continuous time CMDP problem. In this case, a stationary randomized optimal policy, i.e., a mixture of pure policies with finite probabilities, is known to be optimal [32].
A. Problem Formulation

Let $\mathcal{M}$ be the set of memoryless policies. We assume that
the underlying Markov chains corresponding to the memoryless
policies are unichain to guarantee that the Markov chains have unique
stationary distributions. Let $V^M$, $C^B,M$ and $C^O,M$ denote
the average reward, the cost due to blocking of high priority
users and the cost due to offloading of low priority users over
infinite horizon under policy $M \in \mathcal{M}$, respectively. Let the
total reward, costs due to blocking and offloading till time $t$ be
denoted by $R(t)$, $C_B(t)$ and $C_O(t)$, respectively. The objective
of the problem is as follows:

Maximize: $V^M = \lim_{t \to \infty} \frac{1}{t} E_M[R(t)],$

subject to: $C^B,M = \lim_{t \to \infty} \frac{1}{t} E_M[C_B(t)] \leq B_{\max}$
and (2)

$C^O,M = \lim_{t \to \infty} \frac{1}{t} E_M[C_O(t)] \leq O_{\max},$

where $E_M$ is the expectation operator corresponding to policy
$M$, and $B_{\max}, O_{\max}$ are constraints on the blocking probability
of high priority users and the offloading probability of low
priority users, respectively. As the optimal policy is stationary,
the limits in Equation (2) exist.

B. Conversion to Discrete-Time MDP and Lagrangian Approach

Optimal policy can be obtained using a combination of Relative
Value Iteration Algorithm (RVIA) [33] and Lagrangian approach
[32]. The approach adopted is analogous to that of
[9]. However, due to the presence of an additional constraint,
in this paper, we describe the approach to capture the notational
specificities. For fixed Lagrange Multipliers (LMs) $\beta_b$ and
$\beta_o$, the equivalent unconstrained reward function is

$r(s, a; \beta_b; \beta_o) = r(s, a) - \beta_b c_b(s, a) - \beta_o c_o(s, a).$

DP based optimality equation for the considered Semi Markov
Decision Process (SMDP) $\forall s, s' \in \mathcal{S}$ is

$V(s) = \max_a [r(s, a; \beta_b; \beta_o) + \sum_{s'} p_{s,s'}(a)V(s') - \rho \bar{V}(s, a)],$

where $V(s), \bar{V}(s, a), \rho$ denote the value function of state $s \in \mathcal{S}$,
the mean transition time from state $s$ when action $a$ is chosen
and the optimal average reward of the system, respectively.
Since the sojourn times are known to be exponentially distributed,
this is a special case of controlled continuous time
Markov chain. Therefore, the following equation holds,

$0 = \max_a [r(s, a; \beta_b; \beta_o) - \rho + \sum_{s'} q(s'|s, a)V(s')],$

where $q(s'|s, a)$ denote controlled transition rates which satisfy
$q(s'|s, a) \geq 0$, for $s' \neq s$ and $\sum_{s'} q(s'|s, a) = 0$. We scale
the transition rates by a positive scalar quantity which makes it equivalent to time scaling. This scales the average
reward for every policy, however, without changing the optimal
policy. Therefore, let $-q(s, a) \in (0, 1), \forall a$ (without loss of
generality). Hence, $q(s'|s, a) \in [0, 1]$ for $s' \neq s$. Add $V(s)$
to both sides of Equation (3). The optimality equation for
an equivalent discrete-time MDP ($\{X_n\}$ say) with controlled
transition probabilities $p_{ss'}(a)$ is as follows:

$V(s) = \max_a [r(s, a; \beta_b; \beta_o) - \rho + \sum_{s'} p_{ss'}(a)V(s')],$

where $p_{ss'}(a) = q(s'|s, a)$ for $s' \neq s$ and $p_{ss'}(a) = 1 +
q(s'|s, a)$ for $s' = s$. For the rest of the paper, we focus on
the equivalent discrete-time MDP in Equation (4), instead of
the original continuous-time MDP.

For fixed values of $\beta_b$ and $\beta_o$, we use RVIA to solve the
unconstrained maximization problem (see Equation (4)) using
the following scheme.

$V_{n+1}(s) = \max_a [r(s, a; \beta_b; \beta_o) + \sum_{s'} p_{ss'}(a)V_n(s') - V_n(s)],$

(5)

where $V_n(s)$ is the estimate of value function of state $s$ at $n^{th}$
iteration, and $s^*$ is a fixed state. We aim to obtain the optimal
values of $\beta_b$ and $\beta_o$, viz., $\beta^*_b$ and $\beta^*_o$, which maximize
the average reward subject to cost constraints. The gradient
descent routines to update the values of $\beta_b$ and $\beta_o$ at $k^{th}$
iteration are as follows,

$\beta_{b,k+1} = \beta_{b,k} + \frac{1}{k}(B^{\pi_{b,k}} - B_{\max}),$

$\beta_{o,k+1} = \beta_{o,k} + \frac{1}{k}(O^{\pi_{o,k}} - O_{\max}),$

where $\beta_{b,k}$ and $\beta_{o,k}$ are the values of $\beta_b$ and $\beta_o$ at $k^{th}$
iteration, and $B^{\pi_{b,k}}, O^{\pi_{o,k}}$ are the high priority user blocking
probability and low priority user offloading probability at $k^{th}$
iteration, respectively. Note that the optimal policy for the
CMDP is a randomized policy which is randomized in at most
two states [34].

V. Action Elimination

The DP equations (Equations (4) and (5)) are exploited to prove
the sub-optimality of certain actions in different states. Using this,
the number of actions to be considered in different
states can be reduced. This fact is utilized in analyzing the
combinational complexities of the RAT selection algorithms,
as described later. The sub-optimality of different actions is
established with the help of some lemmas.

A. Suboptimal Actions for Departures

The subsequent lemmas describe the sub-optimality of certain
actions in a subset of states. Specifically, whenever a
high/low priority user departs from LTE, $A_5$ is better than $A_7$.
Therefore, in this case, $A_7$ is a suboptimal action. Similarly,
in case of a low priority user departure from WiFi, $A_7$ is a
suboptimal action.

Lemma 1. $A_5$ is better than $A_7$ in case of high/low priority user
departure from LTE (events $E_5, E_6, E_7$ and $E_9$).

Proof. See Appendix A.

Lemma 2. $A_5$ is better than $A_7$ in case of low priority user
departure from WiFi (events $E_9$ and $E_{10}$).

Proof. See Appendix B.

The physical significance of Lemmas 1 and 2 is that whenever there is a departure of a user, if we choose to offload a
low priority user, it is always better to choose the user with
good (bad) channel condition for offloading to LTE (WiFi).
Intuitively, since a bad user degrades the throughput of all
other low priority users in LTE, it is better to offload a bad
user to WiFi. Since we have assumed that in WiFi, every user
experiences good channel condition, offloading a user with bad
channel condition in LTE to WiFi improves the total system throughput. Similar argument holds for the offloading of a user with good channel condition to LTE.

B. Suboptimal Actions for Arrivals

We characterize the suboptimal action in the case of high priority user arrivals. As described in the subsequent lemma, whenever there is a high priority user arrival, then irrespective of the channel condition of the user, action \( A_6 \) is better than \( A_4 \). In other words, whenever a high priority user is associated with LTE, and we decide to offload an existing low priority user to WiFi, it is always better to choose a user with bad channel condition rather than choosing one with good channel.

**Lemma 3.** \( A_4 \) is better than \( A_6 \) in case of high priority user arrivals (events \( E_1 \) and \( E_3 \)).

**Proof.** Proof is similar to Lemma 2.

VI. PROPOSED RAT SELECTION ALGORITHMS

The CMDP problem described in Section IV can be solved using DP techniques which are computationally prohibitive. For example, in policy iteration [33], the computational complexity (which is \( O(|A|S^2) \)) is exponential in the cardinality of the state space. This is known as the curse of dimensionality. Although elimination of suboptimal actions in Section V reduces the size of action space, still the complexity remains exponential in \( |S| \). Furthermore, to compute the optimal policy, we need to know the state transition probabilities which are governed by the statistics of arrival processes. In practice, statistics of arrival processes may be unknown. This is known as the curse of modeling. Although learning based approaches [8], [9] do not require the knowledge of statistics of arrival processes, usually their convergence rates are very slow. To address these issues, we propose low-complexity algorithms for RAT selection. Unlike DP based methods, they do not require the knowledge of the statistics of arrival processes and hence, can be implemented online.

A. Myopic with Constraint Satisfaction Algorithm

In this subsection, we propose an algorithm which is myopic, i.e., it optimizes based on the current reward without considering the future utility. However, the proposed Myopic with Constraint Satisfaction Algorithm (MCSA) (described in Algorithm 1) is designed in such a way that it satisfies the associated constraints on the blocking probability of high priority and the offloading probability of low priority users.

We first determine the event corresponding to the current decision epoch. Then we determine the best action (denoted by \( a^* \)) which provides the highest immediate reward (Line 4). Now, based on the event, we choose different actions. If the current event is low priority user arrival (events \( E_2 \) and \( E_4 \)), then we always choose the action \( a^* \), irrespective of the channel condition of the user (Line 6). Since the set of actions for low priority user arrivals \( \{A_2, A_3\} \) has no direct impact on the high priority user blocking probability and the low priority user offloading probability, whenever a low priority user arrives, we always choose the action which is best in the myopic sense. However, when a high priority user arrives (events \( E_1 \) and \( E_3 \)), we increment the counter which keeps track of the number of high priority user arrivals (denoted by \( A_H \)). We block the incoming high priority user if the current value of blocking probability (which is \( B_H \)) is less than \( B_{max} - \epsilon_B \) (Line 16). The factor \( \epsilon_B \) is incorporated to ensure that the blocking probability of high priority users remains below \( B_{max} \) in the long run. However, if \( B_H \) exceeds \( B_{max} - \epsilon_B \), then actions are chosen based on the current value of offloading probability of low priority users (denoted by \( O_L \)). If \( O_L \) is less than \( O_{max} - \epsilon_O \), then the action \( a^* \) (Line 11) is selected. Note that, similar to the margin \( \epsilon_B \) on \( B_{max} \), we consider a margin \( \epsilon_O \) on \( O_{max} \) to guarantee that the offloading probability of low priority users is less than \( O_{max} \) in the long run. However, if the offloading probability constraint is not satisfied (\( O_L > O_{max} - \epsilon_O \)), then \( A_4 \) is chosen because selection of \( A_4 \) or \( A_5 \) may increase the value of \( O_L \) (Line 12). Depending on whether action involving blocking \( \{A_1\} \) or offloading \( \{A_4, A_5\} \) is chosen, we update the values of \( B_H \) and \( O_L \), respectively (Line 20-21). Procedures followed in case of departures are similar. Initially, we increment the counter (denoted by \( D \)). If \( O_L \) exceeds \( O_{max} - \epsilon_O \), action \( A_1 \) is chosen since it reduces the value of \( O_L \) (Line 29). Otherwise, we act in a myopic manner (Line 27). Based on the action selected, we then update the value of \( O_L \) (Line 31-32). Since the algorithm does not need the knowledge of unknown system dynamics \( \lambda_H \) and \( \lambda_L \), unlike DP methods, it does not suffer from the curse of modeling. Note that when the considered problem does not have a feasible solution, then except few initial iterations, actions involving blocking and offloading are never chosen.

**Algorithm 1** Myopic with Constraint Satisfaction Association Algorithm.

1. Initialize \( D = 0 \), \( A_H = 0 \), \( B_H = 0 \), \( O_L = 0 \), \( F_B = 0 \), \( F_O = 0 \).
2. while TRUE do
3. Determine event \( E \) in the current decision epoch.
4. if \((E = E_2)[E_4]\) then
5. Select action \( a = a^* \).
6. else if \((E = E_1)[E_3]\) then
7. if \((E = E_2)[E_4]\) then
8. \( A_H \leftarrow A_H + 1 \).
9. if \( B_H > (B_{max} - \epsilon_B) \) then
10. procedure \( HP\)-CONSTRAINT-VIOLATION
11. if \( O_L < (O_{max} - \epsilon_O) \) select \( a = a^* \in A \setminus A_1 \).
12. else select \( a = A_2 \).
13. \( F_B \leftarrow I_{||a=A_2||A_6} \).
14. end procedure
15. else
16. Select action \( a = A_1 \).
17. end if
18. procedure \( UPDATE\)-BP-OP
19. \( F_B \leftarrow I_{||a=A_1||A_4} \).
20. \( B_H \leftarrow B_H + (A_H + D) + F_B \).
21. \( O_L \leftarrow O_L \frac{(A_H + D) + F_O}{(A_H + D + 1)} \).
22. end procedure
23. else
24. procedure \( DEPARTURE\)-POLICY
25. \( D = D + 1 \).
26. if \((E = E_2)[E_4]\) then
27. Select action \( a = a^* \).
28. else
29. Select action \( a = A_1 \).
30. \( F_B \leftarrow I_{||a=A_2||A_6} \).
31. \( O_L \leftarrow I_{||a=A_4||A_6} \).
32. \( O_L \leftarrow I_{||a=A_4||A_6} \).
33. end procedure
34. end if
35. end while
B. State-aware Myopic with Constraint Satisfaction Algorithm

In this subsection, we describe the shortcomings of MCSA and propose a State-aware Myopic with Constraint Satisfaction Algorithm (SMCSA) which addresses these shortcomings.

Whenever the current values of blocking and offloading probabilities are lower than the respective constraints, the proposed MCSA blocks an incoming high priority user. Hence, when the arrival rate of high priority users is small, it may lead to unnecessary blocking of high priority users. On the other hand, the optimal policy may result in a lower value of blocking probability of high priority users than that of MCSA, depending on $B_{\text{max}}$. In this case, the optimal policy corresponding to the unconstrained problem may result in a high priority user blocking probability which is significantly lower than $B_{\text{max}}$. Intuitively, MCSA blindly aims to satisfy the constraints of the considered problem without a consideration of the system state. Thus, MCSA always results in high priority user blocking probability values which are close to the given constraints, irrespective of $\lambda_H$ and $\lambda_L$. Due to similar reasons, MCSA results in a high value of offloading probability of low priority users which is always close to $O_{\text{max}}$, irrespective of $\lambda_H$ and $\lambda_L$.

To address this, we propose SMCSA which is described in Algorithm 2. The procedures for low priority user arrivals are same as that of Algorithm 1. In the case of high priority user arrival, when the constraints on the blocking probability and the offloading probability are not satisfied, the procedure is exactly same as that of Algorithm 1. However, when the constraints on the blocking probability and the offloading probability are met, we modify the RAT selection strategy in the following way. We divide the entire state space into multiple regions based on the number of high and low priority users in the system. Let us divide the entire state space into $P$ regions denoted by $R_1, R_2, \ldots, R_P$. For a given region $R_n (1 \leq n \leq P)$, let the probability of blocking and offloading be denoted by $q(n)$ and $p(n)$ ($0 \leq q(n) \leq 1$, $0 \leq p(n) \leq 1$), respectively, where $q(n)$ and $p(n)$ are increasing functions of $n$, and $q(P) = p(P) = 1$. Whenever an event happens, we determine the current state of the system and evaluate the region in which it falls. If it falls in $R_n$, we block (choose $A_1$) the user with probability $q(n)$ and accept (choose $A_2$) with probability $(1 - q(n))$ (Line 17). Similarly, if it falls in $R_n$ and the optimal action involves offloading, we offload with probability $p(n)$ and choose the other action with probability $(1 - p(n))$ (Line 11-12). Similar procedures are followed for the departures. If the constraint on $O_L$ is met and the optimal action involves offloading, we offload with probability $p(n)$ and choose the other action with probability $(1 - p(n))$ (Line 25-26). The procedures for the update of $B_H$ and $O_L$ are same as those of Algorithm 1.

The key advantage of SMCSA is that when the value of $\lambda_H$ is low, we block the incoming high priority users with low probability. As $\lambda_H$ increases and the system gradually fills up with high priority users, the probability of blocking increases. Hence, effectively, the system observes less blocking probability than that of MCSA, when $\lambda_H$ is low. As $\lambda_H$ increases, blocking probability of high priority users increases since $q(n)$ is an increasing function of $n$. The performance of the resulting policy in the case of SMCSA is closer to the optimal policy than that in the case of MCSA. This is because unlike MCSA, the blocking is state-dependent. The blocking probability of high priority users gradually increases with $\lambda_H$, similar to the optimal policy. Therefore, the problem of high blocking probability (which is close to $B_{\text{max}}$) of high priority users for all values of $\lambda_H$, as seen in MCSA, does not arise in SMCSA. Similar observation holds in the case of offloading probability of low priority users also. As $\lambda_L$ grows, the offloading probability of low priority users gradually rises. Similar to MCSA, SMCSA does not require the knowledge of $\lambda_H$ and $\lambda_L$ and hence, is practically implementable.

C. Comparison of Complexities

In this subsection, we analyze the computational and storage complexities associated with the proposed algorithms and the optimal policy. We summarize the complexities of MCSA and SMCSA in Table III. Storing the optimal action for every state results in a storage complexity of $O(|S|)$). Furthermore, the computation of the optimal policy using policy iteration has the worst case computational complexity of $O(|A||S|)$, making it computationally restrictive.

In MCSA, based on every event, we need to calculate the best action $a^*$, resulting in a per-iteration computational complexity of $O(|A|)$. As discussed in Section V, action elimination reduces the effective cardinality of the action space. Although this does not reduce the theoretical computational complexity of MCSA, in practice, the computation time may reduce. MCSA requires to store the values of $A_H, D, B_H$ and $O_L$. However, it need not store any information regarding the state space. Hence, the storage complexity of MCSA is $O(1)$.
The per-iteration worst case computational complexity of SMCSA is also $O(|A|)$ because when the current values of the constraints are below the specified values, the associated procedures are same as that of MCSA. However, the complexity involved with the probabilistic state-aware blocking and offloading is $O(1)$ because no comparison among the actions is required. Apart from $A_H, D, B_H$ and $O_L$, SMCSA needs to store the information regarding the regions $R_n (1 \leq n \leq P)$ and corresponding probabilities $q(n)$ and $p(n)$. Therefore, the storage complexity of SMCSA is $O(P)$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Storage complexity</th>
<th>Computational complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCSA</td>
<td>$O(1)$</td>
<td>$O(</td>
</tr>
<tr>
<td>SMCSA</td>
<td>$O(P)$</td>
<td>$O(</td>
</tr>
</tbody>
</table>

**Remark 2.** A well-studied approach for MDP problems is the investigation of structural properties, see e.g., [7], [35], which often leads to threshold-based optimal policy. In [7], which does not consider offloading probability of low priority users, we prove optimality of threshold policies. Although the computational complexity of the resulting algorithm in [7] is lower than that of policy iteration, it is still exponential in one of the parameters of the state space. The problem addressed in this paper does not result in a threshold-based optimal policy. However, the proposed algorithms provide significantly lower computational complexities compared to what would have been achieved corresponding to a threshold structure.

**VII. Simulation Results**

In this section, we evaluate the performances of the proposed algorithms in an SDN based evaluation platform built using ns-3. The evaluation platform is constructed by the modification of existing components of ns-3 in such a manner that it conforms to the proposed architecture. We observe performances of the proposed algorithms and the optimal policy in terms of the blocking probability of high priority users, the offloading probability of low priority users and the total system throughput. We also compare the performances of the proposed algorithms with the association scheme adopted in existing network where SDN is not present. In this scenario, the association scheme results in on-the-spot offloading [36], where low priority users are always associated with WiFi and high priority users are associated with LTE. However, when capacity is reached in LTE, high priority users are blocked. We also compare the performances of our proposed algorithms with on-the-spot offloading in the face of user mobility.

**A. Simulation Setup and Methodology**

We setup an evaluation platform (implemented in ns-3) to simulate the SDN based LTE-WiFi architecture as proposed in Section II. An SDN controller node having two interfaces, one towards LTE BS and the other towards WiFi AP over Internet Protocol (IP) connections, is created. The SDN controller consists of an LTE controller and a WiFi controller as logical entities. RRM functionalities are moved from LTE BS to the SDN controller. WiFi and LTE RRM control signals are forwarded to the SDN controller by WiFi AP and LTE BS, respectively. However, the data plane traffic is routed directly to the gateway. Whenever an event (arrival/departure of users) occurs, a control packet is sent from the user to the corresponding BS/AP which forwards the packet to the SDN controller. The SDN controller then chooses an action following the algorithm implemented in the SDN controller.

The network model is composed of a 3GPP LTE BS and an operator-deployed IEEE 802.11g WiFi AP inside the coverage area of LTE BS. Users are assumed to be stationary. We set the radius of the coverage area of WiFi AP to be around 30 m. WiFi AP is located at nearly 50 m from LTE BS. LTE and WiFi parameters are described in Tables IV and V. LTE and WiFi parameters are selected based on 3GPP [37]- [38] models and saturation throughput [31] IEEE 802.11g WiFi model, respectively. In simulations, we assume that a low priority user can obtain a maximum data rate of 10 Mbps due to access network bottleneck. We set $B_{max} = O_{max} = 0.05, \epsilon_B = \epsilon = 0.01$. In case of SMCSA, we divide the entire state space into two regions, viz., $R_1$ and $R_2$. We keep $q(1) = p(1) = 0$ and $q(2) = p(2) = 1$. We choose $d = p_c = 2$.

### Table IV: LTE Network Model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>High priority user capacity</td>
<td>4 users</td>
</tr>
<tr>
<td>Bit rate of a high priority user</td>
<td>20 kbps</td>
</tr>
<tr>
<td>High priority user packet payload</td>
<td>50 bytes</td>
</tr>
<tr>
<td>Low priority user packet payload</td>
<td>600 bytes</td>
</tr>
<tr>
<td>1x power for BS and MS</td>
<td>46 dBm and 23 dBm</td>
</tr>
<tr>
<td>Noise figure for BS and MS</td>
<td>5 dB and 9 dB</td>
</tr>
<tr>
<td>Antenna height for BS and MS</td>
<td>3.2 m and 1.5 m</td>
</tr>
<tr>
<td>Antenna type for BS and MS</td>
<td>Isotropic Antenna</td>
</tr>
<tr>
<td>Path loss (in km)</td>
<td>$125.4 + 37.6 \log(R)$</td>
</tr>
<tr>
<td>Multi-path fading</td>
<td>Extended Pedestrian A model [40]</td>
</tr>
</tbody>
</table>

### Table V: WiFi Network Model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel bit rate</td>
<td>54 Mbps</td>
</tr>
<tr>
<td>UDP header</td>
<td>224 bits</td>
</tr>
<tr>
<td>Packet payload</td>
<td>1500 bytes</td>
</tr>
<tr>
<td>Slot duration</td>
<td>20µs</td>
</tr>
<tr>
<td>Short inter-frame space (SFIS)</td>
<td>10µs</td>
</tr>
<tr>
<td>Distributed coordination function (DFCS)</td>
<td>50µs</td>
</tr>
<tr>
<td>Minimum acceptable per-user throughput</td>
<td>4.5 Mbps</td>
</tr>
<tr>
<td>1x power for AP</td>
<td>2.5dBm</td>
</tr>
<tr>
<td>Noise figure for AP</td>
<td>4 dB</td>
</tr>
<tr>
<td>Antenna height for AP</td>
<td>2.5 m</td>
</tr>
<tr>
<td>Antenna parameter</td>
<td>Isotropic antenna</td>
</tr>
<tr>
<td>Path loss (in km)</td>
<td>140.3 + 40.7 \log(R)</td>
</tr>
<tr>
<td>Fading</td>
<td>Rayleigh fading</td>
</tr>
</tbody>
</table>

**B. High Priority User Arrival Rate Variation**

Fig. 3a describes the high priority user blocking probability performances of the proposed algorithms, optimal policy and on-the-spot offloading (existing non-SDN scenario) as a function of $\lambda_H$. The blocking probability of the optimal policy increases with $\lambda_H$. Since MCSA blocks high priority users based on $B_{max}$, irrespective of $\lambda_H$, the blocking probability is nearly the same for all $\lambda_H$s. SMCSA is designed in such a way that it blocks high priority users only when the system reaches region $R_2$. We consider two cases, viz., $R_1 : (i_G + 2i_H) \leq C_L - 2$, and $R_1 : (i_G + 2i_B) \leq C_L - 1$, respectively. The high priority user blocking probability of SMCSA gradually increases with $\lambda_H$, similar to the optimal policy. This happens because when the value of $\lambda_H$ is low, we block the incoming high priority users with low probability. As $\lambda_H$ increases and the system gradually fills up with high priority users, the probability of blocking increases as $q(n)$ increases with $n$. Since the size of the region $R_1$ is smaller in the first case, the blocking probability is lower in the second case. In case of on-the-spot offloading, the high priority user blocking probability gradually rises with $\lambda_H$.
C. Low Priority User Arrival Rate Variation

Fig. 3d illustrates the high priority user blocking probability performances of different algorithms as a function of \( \lambda_L \). Similar to Fig. 3a, the blocking probability of MCSA is close to \( B_{\text{max}} \) for all \( \lambda_L \). SMCSA blocks high priority users and offloads low priority users only when the system reaches region \( R_2 \). We consider two cases, viz., \( R_1 : (\lambda_G + 2iB) \leq C_L - 2, (k_G + k_B) \leq 4, (j_G + j_B) \leq 2 \) and \( R_1 : (\lambda_G + 2iB) \leq C_L - 2, (k_G + k_B) \leq 4, (j_G + j_B) < 2 \), respectively. The performance of SMCSA for the first case is close to that of the optimal policy. In the second case, the blocking probabilities are slightly higher than those of the first case because \( R_1 \) is smaller in the second case.

In Fig. 3e, the offloading probability of the optimal policy grows with \( \lambda_L \). The offloading probability of MCSA is close to \( O_{\text{max}} \) for every \( \lambda_L \). In SMCSA, the offloading probability grows with \( \lambda_L \) (similar to the optimal policy) because \( p(n) \) is an increasing function of \( n \). The offloading probability in the second case is slightly larger than in the first case since region \( R_1 \) is smaller in the second case. Since offloading is not possible in the case of on-the-spot offloading, the offloading probability of low priority users is always zero.

In Fig. 3f, we observe that the performances of both MCSA and SMCSA are close to optimal, outperforming on-the-spot offloading algorithm. Though the proposed algorithms take into account only the instantaneous rewards while optimizing, these algorithms facilitate load balancing between LTE and WiFi. The total system throughput of on-the-spot offloading does not increase much with \( \lambda_L \), due to contention among users in WiFi. Fig. 3f demonstrates that indeed our proposed algorithms provide near-optimal performances.

D. Consideration of Mobility

In this section, we evaluate the performances of the algorithms in the presence of user mobility. We consider random waypoint model [41] for user mobility. We set the user speed in the range \([0, 40]\) km/h.

Mobile users may be offloaded frequently from one RAT to another. This may increase the offloading probability of the overall system. Since the proposed algorithms are designed in such a way that the offloading probability satisfies the constraint, a mobile user may significantly increase the offloading probability of the system. As a result, it may happen that stationary users get very less number of offloading opportunities. Furthermore, a user with mobility is expected to drain a lot of battery due to excessive offloading from one RAT to another. To take into account these factors, we modify the algorithms in the following way. Apart from the constraint on the total offloading probability of low priority users, we consider offloading profile of individual low priority users while offloading. To be precise, whenever an action involving offloading of low priority users \((A_{4}, A_{5}, A_{6}, A_{7})\) is chosen, we choose a user which has not been offloaded till now. If no such user is present, then we choose the user which has been offloaded the earliest before.

In Fig. 4a, we observe that both MCSA and SMCSA outperform on-the-spot offloading in terms of the total system throughput. Similar observation is made in Fig. 4b for different values of \( \lambda_L \). We have not shown the blocking probability and the offloading probability performances since they are exactly...
In this paper, we propose an SDN-based architecture for handling the control and management of multiple RATs. We consider the optimal RAT selection problem in an LTE-WiFi HetNet consisting of users of multiple priorities and channel states. Maximizing the total system throughput subject to constraints on the high priority user blocking probability and the low priority user offloading probability is formulated as a CMDP problem. We prove the sub-optimality of different ac-

VII. CONCLUSION

We prove the lemma using sample path arguments. We consider case of event $E_5$. Proofs for the other events follow in a similar manner. We assume that the system starts at time $t = 0$. Suppose that the system is in state $s_1 = (l_G, l_B, j_G, j_B, k_G, k_B)$, when event $E_5$ occurs at time $t_1$. Consider a policy which chooses $A_2$ in state $(l_G, l_B, j_G, j_B, k_G, k_B)$ and denote it by $\pi_1$. Consider another policy (may be a non-stationary policy) $\pi_2$ which selects $A_5$

in state $(l_G, l_B, j_G, j_B, k_G, k_B)$. As illustrated in Fig. 5, let us assume that under policies $\pi_1$ and $\pi_2$, the system move from state $s_1$ to state $s_2 = (l_G, l_B, j_G, j_B + 1, k_G, k_B - 1)$ and $s_3 = (l_G, l_B, j_G + 1, j_B, k_G - 1, k_B)$, respectively. Since we consider a Markovian system, the inter-arrival and service times are identical for the considered sample paths. We assume that based on the next event $E_t$ and following the policy $\pi_1$, the system moves from state $s_2$ to state $s_4$. Suppose the policy $\pi_2$ is such that in response to event $E_t$, it chooses the same action as that of $\pi_1$. Additionally, it offloads one good user from LTE to WiFi and one bad user from WiFi to LTE. Therefore, under the policy $\pi_2$, the system moves from state $s_3$ to $s_4$. We construct the policy $\pi_2$ in such a way that here onwards, it chooses the same action as that of policy $\pi_1$. Therefore, from state $s_4$ onwards, both the sample paths follow the same trajectory. The difference of value functions of state $s_1$ under policies $\pi_1$ and $\pi_2$ is given by

$$V_{\pi_1}(s_1) - V_{\pi_2}(s_1) = \frac{(C_L - l_G - p_G l_B) R_{L,L}(1 - \frac{1}{d})}{(j_G + j_B + 1)} < 0.$$ 

Therefore, policy $\pi_2$ is strictly better than $\pi_1$. Since the Markov chains under various policies are recurrent in nature, state $s_1$ is visited infinitely often. Upon each visit, action $A_7$ induced by policy $\pi_1$ provides less reward than action $A_5$ corresponding to $\pi_2$. This completes the proof of the lemma.

A. Proof of Lemma 1

We prove the lemma using sample path arguments. We consider case of event $E_5$. Proofs for the other events follow in a similar manner. We assume that the system starts at time $t = 0$. Suppose that the system is in state $s_1 = (l_G, l_B, j_G, j_B, k_G, k_B)$, when event $E_5$ occurs at time $t_1$. Consider a policy which chooses $A_2$ in state $(l_G, l_B, j_G, j_B, k_G, k_B)$ and denote it by $\pi_1$. Consider another policy (may be a non-stationary policy) $\pi_2$ which selects $A_5$

in state $(l_G, l_B, j_G, j_B, k_G, k_B)$. As illustrated in Fig. 5, let us assume that under policies $\pi_1$ and $\pi_2$, the system move from state $s_1$ to state $s_2 = (l_G, l_B, j_G, j_B + 1, k_G, k_B - 1)$ and $s_3 = (l_G, l_B, j_G + 1, j_B, k_G - 1, k_B)$, respectively. Since we consider a Markovian system, the inter-arrival and service times are identical for the considered sample paths. We assume that based on the next event $E_t$ and following the policy $\pi_1$, the system moves from state $s_2$ to state $s_4$. Suppose the policy $\pi_2$ is such that in response to event $E_t$, it chooses the same action as that of $\pi_1$. Additionally, it offloads one good user from LTE to WiFi and one bad user from WiFi to LTE. Therefore, under the policy $\pi_2$, the system moves from state $s_3$ to $s_4$. We construct the policy $\pi_2$ in such a way that here onwards, it chooses the same action as that of policy $\pi_1$. Therefore, from state $s_4$ onwards, both the sample paths follow the same trajectory. The difference of value functions of state $s_1$ under policies $\pi_1$ and $\pi_2$ is given by

$$V_{\pi_1}(s_1) - V_{\pi_2}(s_1) = \frac{(C_L - l_G - p_G l_B) R_{L,L}(1 - \frac{1}{d})}{(j_G + j_B + 1)} < 0.$$ 

Therefore, policy $\pi_2$ is strictly better than $\pi_1$. Since the Markov chains under various policies are recurrent in nature, state $s_1$ is visited infinitely often. Upon each visit, action $A_7$ induced by policy $\pi_1$ provides less reward than action $A_5$ corresponding to $\pi_2$. This completes the proof of the lemma.

B. Proof of Lemma 2

Similar to Lemma 1, we prove this lemma for event $E_9$. Proof for event $E_{10}$ follows in a similar way. Suppose the system is in state $s_1 = (l_G, l_B, j_G, j_B, k_G, k_B)$ when event $E_9$ occurs at time $t_1$. Consider policies $\pi_1$ and $\pi_2$ which choose $A_3$ and $A_7$ in state $s_1$, respectively. We assume that under policies $\pi_1$ and $\pi_2$, the system move from state $s_1$ to state $s_2 = (l_G, l_B, j_G, j_B - 1, k_G, k_B + 1)$ and $s_3 = (l_G, l_B, j_G - 1, j_B, k_G + 1, k_B)$, respectively. We assume that based on the next event $E_t$ and following the policy $\pi_1$, the system moves from state $s_2$ to state $s_4$. Suppose the policy $\pi_2$ is such that for event $E_t$, it chooses the same action as that of $\pi_1$, offloads one bad user from LTE to WiFi and one good user from WiFi to LTE. Therefore, under the policy $\pi_2$, the system moves from state $s_3$ to $s_4$. We construct the policy $\pi_2$ in such a way that here onwards, it chooses the same action as that of policy $\pi_1$. Therefore, from state $s_4$ onwards, both the sample paths follow the same trajectory. The difference of value functions of state $s_1$ under policies $\pi_1$ and $\pi_2$ is given by

$$V_{\pi_1}(s_1) - V_{\pi_2}(s_1) = \frac{(C_L - l_G - p_G l_B) R_{L,L}(1 - \frac{1}{d})}{(j_G + j_B + 1)} > 0.$$ 

Therefore, policy $\pi_1$ is strictly better than $\pi_2$. Due to the recurrent nature of the Markov chain, similar to Lemma 1, $A_5$ is better than $A_7$.

ACKNOWLEDGMENT

The authors thank Ashish Sharma, Rohan Kharade and Abhishek Dandekar for their contributions towards development in the SDN based evaluation platform in ns-3.

REFERENCES

