Hap-SliceR: A Radio Resource Slicing Framework for 5G Networks With Haptic Communications

Adnan Aijaz, *Member, IEEE*

***Abstract*—It is expected that the emerging 5G networks will not only support diverse use cases, but also enable unprecedented applications such as *haptic communications*. Therefore, *network slicing* will provide the required design flexibility. Radio resource slicing would be an indispensable component of any network slicing solution. This paper proposes Hap-SliceR, which is a novel radio resource slicing framework for 5G networks with haptic communications. First, Hap-SliceR derives a *network-wide* radio resource slicing strategy for 5G networks. The optimal slicing strategy, which is based on a *reinforcement learning* approach, allocates radio resources to different slices while accounting for the dynamics and utility requirements of different slices. Second, Hap-SliceR provides customization of radio resources for haptic communications over 5G networks. The radio resource allocation requirements of haptic communications have been translated into a unique radio resource allocation problem. A low-complexity heuristic algorithm has been developed for resource allocation. Finally, a comprehensive performance evaluation of Hap-SliceR has been conducted based on a recently proposed 5G air-interface design.**

***Index Terms*—5G, haptic communications, LTE-A, radio resource allocation, radio resource slicing, virtualization.**

I. Introduction

THE emerging 5G mobile communications networks will support a range of use cases spanning different vertical industries. The research community has a consensus that 5G networks must be designed in a flexible manner to cater for the diverse and rather orthogonal service requirements of different vertical applications. Such flexibility is possible through *network slicing* [1], which will be indispensable for 5G network design. The fundamental principle of network slicing is to create multiple logical networks, over a common physical infrastructure, with each network tailored to the specific needs of a use case.

Recently, the notion of *5G-enabled Tactile Internet* [2]–[4] is emerging, which is envisioned to enable the delivery of real-time control and physical haptic[1] experiences in perceived real-time. It is expected that 5G networks will underpin the Tactile Internet at the wireless edge through ultrareliable, ultraresponsive, and intelligent network connectivity. By enabling *haptic communications* [5] over 5G networks, the Tactile Internet will provide the medium for physical interaction in remote environments with complete immersion.

Ongoing standardization efforts toward 5G reveal that the legacy LTE-A networks will continue to develop in a backward-compatible way and become an integral component of the 5G ecosystem. The overall 5G wireless access solution will consist of *evolved* LTE-A radio access network (RAN), complemented with novel technological enhancements and architectural designs, coexisting with potentially new radio access technologies (in new spectrum) [6]. Recently, some studies have investigated the viability of a 5G network, based on LTE-A air interface, for ultrareliable and low latency communications [7], [8]. Exploring the potential of LTE-A networks for haptic communications will pave the way toward realizing the Tactile Internet.

A. Problem Definition and Challenges

Although network slicing requires slicing of both radio access and core network resources, our focus in this paper is strictly on RAN slicing with emphasis on radio/wireless resource slicing. Network slicing can be achieved in various ways. However, radio resource slicing would be an indispensable component of any network slicing solution.

Radio resource slicing would be enabled through virtualization of radio/wireless resources [9]. Such virtualization-based radio resource slicing must be able to achieve tight *isolation* and provide application-specific *customization*, while ensuring efficient utilization across different radio slices. Isolation means that any change in one slice due to traffic load, channel conditions, etc., should not affect other slices, e.g., result in reduction of radio resources. Customization provides the flexibility of implementing application-specific radio resource management schemes within each slice.

One way of achieving radio resource slicing is through independent slicing of radio resources at the base station level. In this respect, techniques like NVS [10], VBTS [11], and LTEVirt [12] have been developed. However, such techniques not only require modifications to legacy base station schedulers but also perform suboptimally (from a resource utilization perspective) as user distribution, average user channel conditions, and user-traffic requirements may vary significantly across base stations over fine timescales. Therefore, in order to create effective RAN slicing, *network-wide* radio resource slicing across multiple base stations is much more attractive.

Manuscript received October 11, 2016; revised December 10, 2016; accepted December 21, 2016.

The author is with the Telecommunications Research Laboratory, Toshiba Research Europe, Ltd., Bristol BS1 4ND, U.K. (e-mail: adnan.aijaz@toshiba-trel.com).

Digital Object Identifier 10.1109/JSYST.2017.2647970

[1]The word haptic refers to any form of interaction involving the sense of touch.

Such network-wide resource slicing not only provides dynamic allocation of resources across a set of base stations but also holds minimum footprint for wide-scale adoption by requiring minimal changes to legacy base stations. Network-wide slicing can be logically achieved in an entity located at a higher level (compared to the base station) in mobile network hierarchy. Some recent examples of network-wide radio resource slicing include NetShare [13], CellSlice [14], and AppRAN [15].

State-of-the-art network-wide radio resource slicing solutions exhibit certain limitations that complicate their application in future 5G cellular networks. First, current solutions mostly employ admission control mechanisms to ensure isolation across slices. Such mechanisms may result in degraded quality-of-experience for users. Second, most existing solutions do not *independently* leave customization of allocated radio resources to base stations. This not only results in suboptimal scheduling decisions, but also leaves little room for innovating radio resource scheduling for new applications, e.g., haptic communications. Third, a common limitation of most existing solutions is the inability to deal with highly dynamic wireless environments (varying traffic demands and channel conditions). Each slice follows different dynamics, and therefore, calculation of a single fixed optimal *slicing period*[2] is not feasible. Last, but not the least, some existing solutions are based on complex architectural enhancements, which require protocol-level modifications.

B. Contributions and Outline

Our objective in this paper is to investigate the problem of radio resource slicing for 5G networks along with radio resource customization for haptic communications. We propose Hap-SliceR, which is a novel radio resource slicing framework for 5G networks with haptic communications. Hap-SliceR provides a twofold contribution. First, Hap-SliceR derives a novel *network-wide* radio resource slicing strategy for 5G networks. Second, Hap-SliceR provides novel customization of radio resources for haptic communications over 5G networks. The key contributions of this paper are further summarized as follows.

1) Based on virtualization of radio resources, we develop a novel network-wide radio resource slicing strategy for 5G networks that allocates radio resources to different slices in a flexible way, while accounting for the utility requirements of different vertical applications. For maximum utilization of scarce radio resources, it is particularly important to have a slicing strategy that *implicitly* accounts for the slice dynamics. Hap-SliceR adopts a *reinforcement learning* (RL) [16] approach for dynamic radio resource slicing. In this respect, we first model the slicing problem as a Markov decision process (MDP) and, then, derive an optimal radio resource slicing strategy based on the application of Q-learning [17] technique.
2) The radio resource slicing strategy based on Q-learning algorithm is particularly attractive as it does not require *a priori* knowledge of the wireless environment. However, its performance can be slower in practice. We improve the efficiency of the radio resource slicing strategy through a *post-decision state (PDS) learning* [18] approach that exploits partially known system dynamics.
3) We develop a novel radio resource customization algorithm for haptic communications. We first identify the key requirements of haptic communications from a radio resource allocation perspective and then formulate a unique radio resource allocation problem that captures these requirements along with the specific power and resource block (RB) allocation constraints of uplink and downlink multiple access schemes in LTE-A networks.
4) The radio resource allocation problem becomes particularly challenging to solve due to its combinatorial nature. To reduce complexity, we decompose the original optimization problem and transform it into a binary integer programming (BIP) problem through an optimal power allocation strategy. We propose a low-complexity greedy heuristic algorithm for resource allocation.
5) We conduct a comprehensive performance evaluation of Hap-SliceR through system-level simulation studies, based on a recently proposed 5G air-interface design [8]. Performance is also benchmarked against state-of-the-art solutions.

The rest of the paper is organized as follows. Section II covers the related work. In Section III, we present the radio resource slicing strategy for Hap-SliceR. Section IV provides the preliminaries on haptic communications. This is followed by radio resource customization for Hap-SliceR in Section V. Section VI describes the practical aspects of Hap-SliceR. Performance evaluation has been conducted in Section VII. Finally, the paper is concluded in Section VIII.

II. Related Work

In the literature, a number of recent studies have investigated the concept of wireless resource virtualization. In [19], Zaki *et al.* proposed an LTE air-interface virtualization scheme based on hypervisor, which virtualizes the evolved Node B (eNB) into a number of virtual eNBs. Kokku *et al.* [10] presented NVS, which is a flow-level wireless resource virtualization framework. The wireless resources of base stations are sliced between different flow groups enabling customized flow scheduling per slice. The same authors proposed CellSlice [14], which is a gateway-level solution for wireless resource virtualization. CellSlice indirectly influences the base station scheduling decisions to impose slice-specific resource allocation. In [13], Mahindra *et al.* presented the design and implementation of NetShare, which is a network-wide radio resource management framework for RAN sharing. NetShare introduces a two-level scheduler split to effectively manage and allocate RAN wireless resources among multiple entities that share the network. He and Song introduced AppRAN [15], which is an application-oriented framework for RAN sharing. AppRAN defines a serial of abstract applications with distinct QoS requirements and periodically computes application-level resource allocation for each radio element at a central controller. In [20], Kalil *et al.* proposed a wireless resource virtualization framework for RAN sharing in LTE networks. The resource scheduling problem for each slice

[2]In the context of radio resource slicing, the slicing period is defined as the time after which the size of each allocated slice would be recalculated.

has been formulated as an integer programme that is solved using an iterative algorithm. Fu and Kozat proposed a game-theoretic framework for wireless network virtualization [21].

Hap-SliceR is fundamentally different compared to state-of-the-art wireless resource virtualization solutions. First, it provides a dynamic radio resource slicing strategy, which is based on an RL approach. Second, it only abstracts radio resources and not the complete base station. This ensures minimal changes to LTE standards and protocols for implementation. Third, Hap-SliceR leaves customization of slices at the base station that results in more robust scheduling decisions for different applications. Fourth, it can be easily implemented in any standard cloud computing infrastructure at the edge of RAN. Last, but not the least, Hap-SliceR provides novel customization of radio resources for haptic communications. To the best of our knowledge, the problem of radio resource allocation for haptic communications has not been investigated before.

III. Hap-SliceR—Radio Resource Slicing

In this section, we explain the radio resource slicing strategy for Hap-SliceR. Besides fulfilling the utility requirements of the respective vertical application, the slicing strategy must ensure maximum utilization of scarce radio resources. Rather than following a static slice allocation, the slicing strategy must provide dynamic slice management. The slicing period plays an important role in ensuring maximum utilization of wireless resources. However, it is not feasible to calculate a single fixed optimal slicing period owing to diversity of slice dynamics. Hap-SliceR provides a dynamic slicing strategy that implicitly accounts for slice dynamics. It adopts an RL approach to radio resource slicing wherein the entity responsible for slicing not only learns the size of each slice, but also updates it dynamically. RL is particularly attractive to tackle dynamic changes in wireless environments. It provides the network with necessary cognitive capabilities to modify the slicing strategy in accordance with the requirements of different slices. However, RL is generally applicable to problems that are either MDP or its variant. Therefore, we first model radio resource slicing problem as a semi-MDP (SMDP), which is particularly suited to wireless environments. The functional architecture of radio resource slicing and the overall RL-based radio resource slicing methodology is illustrated in Fig. 1.

A. SMDP Model for Radio Resource Slicing

We consider the existence of a slicing controller within a generic 5G network, as shown in Fig. 1, which is responsible for radio resource slicing. We assume that the 5G network comprises B base stations and needs to support $\mathcal{M}$ different vertical applications; hence, $\mathcal{M}$ different radio resource slices are required. Let, $\mathcal{X}$ denote the overall radio resource pool. The overall radio resource pool can be abstracted in different ways; for example, through a three-dimensional (3-D) resource grid [22] wherein radio resources are distributed in time, frequency, and spatial (base station) domains.

We start by defining network states, actions, state dynamics, and rewards in context of the generic SMDP framework [23]. The interaction of the slicing controller with the wireless environment is represented by the tuple $[\mathcal{S}, \mathcal{A}, \mathcal{P}\left(s, s^{*}\right), \mathcal{R}\left(s, a\right)]$, where $\mathcal{S}$ denotes the set of possible states, $\mathcal{A}$ denotes the set of possible actions, $\mathcal{P}\left(s, s^{*}\right)$ captures the state dynamics and denotes the state transition probability $s \rightarrow s^{*}$, and $\mathcal{R}\left(s, a\right)$ is the reward/cost associated with the trigger of action a in state s, which is fed back to the slicing controller.

States: Let, s_t^m denote the state of mth slice at time t, which is represented by the tuple $\{\Lambda_t^m, \mathcal{U}_t^m, \Psi_t^m\}$ such that $\Lambda_t^m, \mathcal{U}_t^m$, and Ψ_t^m denote the resource allocation, sum utility, and resource utilization for the mth slice at time t, respectively. The resource allocation for the mth slice, Λ_t^m, is given by a matrix $[\mathcal{J}_{b,i}]_{B\times|\mathcal{X}|}$, where each entry $\mathcal{J}_{b,i} \in \{0, 1\}; i \in \mathcal{X}$ denotes whether the ith resource unit (RU), which represents the minimum possible resource allocation, is allocated to the bth base station or not. The utility function $\mathcal{U}_t^m$ is specific to each slice depending on the respective vertical application. The resource utilization, Ψ_t^m is defined as the ratio of used resources to the allocated resources.

Actions: In each state, an action is performed by the slicing controller. However, not all the actions are feasible in a given state. We denote it by $\mathcal{A}(s) \subset \mathcal{A}$, the set of feasible actions in state s. The action taken by the slicing controller for the mth slice at time t is represented by $a_t^m = \{\alpha_{m,b,i}\}$, where $\alpha_{m,b,i} \in \{0, 1\}; i \in \mathcal{X}$ denotes whether the ith RU, is allocated to the mth slice at the bth base station or not.

State Dynamics: The wireless environment perceives the action and transitions to a new state $s^{'}$. The state dynamics for the mth slice are characterized by the state transition probability $\mathcal{P}(s, s^{'}) = \delta(s, s^{'}) \cdot \mathcal{T}(s, a)$, where $\delta(s, s^{'})$ represents the transition rate between states s and $s^{'}$, and $\mathcal{T}(s, a)$ denotes the expected sojourn time for the respective state-action pair.

Reward: The state transition yields a reward for the mth slice given by $r_t^m \in \mathcal{R}\left(s, a\right)$. This reward is fed back to the slicing controller. In our case, we express the reward function as the sum of a slice utility term and a utilization term as follows:

$$\mathcal{R}\left(s, a\right) = \kappa \cdot \mathcal{U}\left(s, a\right) + \varpi \cdot \Psi(s, a) \tag{1}$$

where $\mathcal{U}\left(s, a\right)$ and $\Psi(s, a)$, respectively, denote the sum utility and resource utilization associated with the trigger of action a in state s, and κ and ϖ are constants.

The ultimate objective of the slicing controller is to find the optimal slicing strategy (policy) π^{*}, which is a mapping from $\mathcal{S}$ to $\mathcal{A}$, that maximizes the expected long-term discounted reward for each state:

$$\pi^{*}(s) = \arg\max_{\pi} V_{\pi}(s). \tag{2}$$

Let, $\Pi_{\pi}(s) = s, s_1, s_2, \ldots$ be the trajectory of the Markov chain when strategy π is adopted. The long-term discounted reward of state s is the discounted sum of rewards earned on the trajectory and is given by

$$\mathcal{R}\left(s, \pi(s)\right) + \varphi\mathcal{R}\left(s_1, \pi(s_1)\right) + \varphi^2\mathcal{R}\left(s_2, \pi(s_2)\right) + \cdots \tag{3}$$

where φ is the discount factor $(0 < \varphi < 1)$, which determines the present value of future rewards. The optimization objective in (2), which denotes the state-value function of an arbitrary

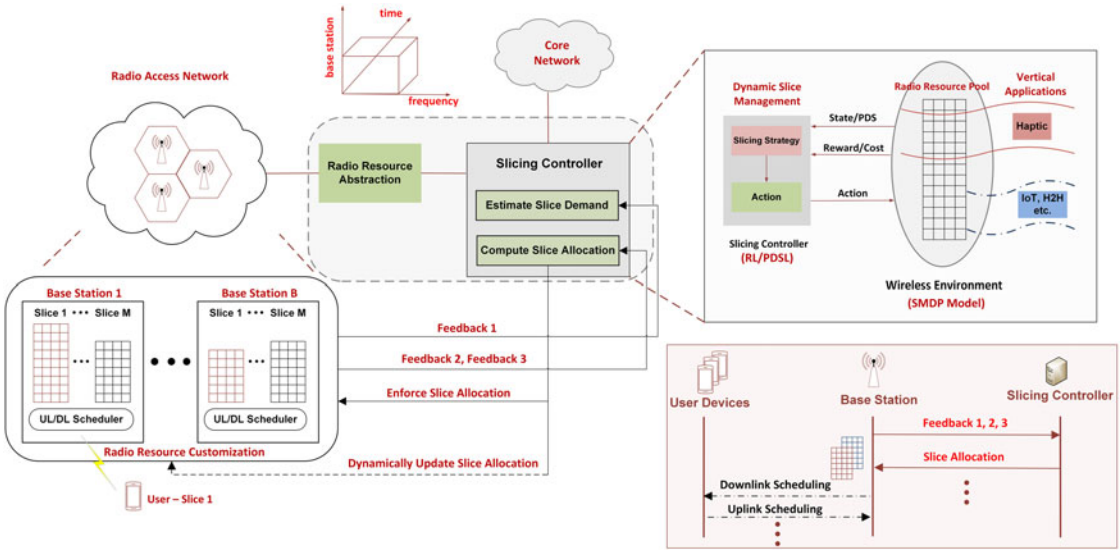

Fig. 1. Functional architecture of radio resource slicing in a cellular network. Radio resource slicing methodology in Hap-SliceR is also illustrated.

policy, can be expressed as follows:

$$\begin{aligned} V_{\pi}(s) = & \ \mathbb{E}\left\{\sum_{t=0}^{\infty}\varphi^{t}\mathcal{R}\left(s_{t},a_{t}\right)\big|(s_{0}=s,a_{0}=a)\right\} \\ = & \ \mathbb{E}\left\{\mathcal{R}\left(s_{0},a_{0}\right)\big|(s_{0}=s,a_{0}=a)\right\} \\ & + \mathbb{E}\left\{\sum_{t=1}^{\infty}\varphi^{t-1}\mathcal{R}\left(s_{t},a_{t}\right)\big|(s_{0}=s,a_{0}=a)\right\}. \end{aligned} \tag{4}$$

Further, due to the Markov property, (4) can be expressed as

$$\begin{aligned} V_{\pi}(s) = & \ \mathbb{E}\left\{\mathcal{R}\left(s,a\right)\right\} + \varphi\sum_{s^{'}\in\mathcal{S}}\mathcal{P}(s,s^{'},\pi(s)) \\ & \times \mathbb{E}\left\{\sum_{t=1}^{\infty}\varphi^{t}\mathcal{R}\left(s_{t},a_{t}\right)\big|(s_{1}=s,a_{1}=a)\right\} \\ = & \ \bar{\mathcal{R}}\left(s,a\right) + \varphi\sum_{s^{'}\in\mathcal{S}}\mathcal{P}(s,s^{'},\pi(s))V_{\pi}(s^{'}). \end{aligned} \tag{5}$$

According to the Bellman's optimality criterion [16], there is at least one optimal strategy in a single environment setting. Therefore, the state-value function for the optimal strategy is given by

$$V_{\pi^{*}}(s) = \arg\max_{a\in\mathcal{A}(s)}\left\{\bar{\mathcal{R}}\left(s,a\right) + \varphi\sum_{s^{'}\in\mathcal{S}}\mathcal{P}(s,s^{'},a)V_{\pi}(s^{'})\right\}. \tag{6}$$

B. RL-Based Slicing Strategy

In the previous section, an optimal policy is derived based on $\bar{\mathcal{R}}\left(s,a\right)$ and $\mathcal{P}(s,s^{'},a)$. The state-transition probability depends on a number of factors, such as traffic load, user arrival and departure rates, decision-making algorithms, etc., and therefore, it might not be easily obtained in wireless environments. In this context, model-free RL turns out to be a good fit for deriving the optimal strategy as it does not require the expectation of reward/cost and the state transition probability to be known *a priori*. Among the various existing RL algorithms, we select $\mathcal{Q}$-learning [17] due to its simplicity.

The slicing controller interacts with the wireless environment (radio resource pool) over discrete time-steps of fixed duration τ. The action-value function (also referred to as the $\mathcal{Q}$-value) of the state-action pair $(s,\pi(s))$, denoted by $\mathcal{Q}\left(s,\pi(s)\right)$, is defined as the expected long-term discounted reward of state s when strategy π is used. The objective is to find the optimal strategy π^{*} that maximizes the action-value function of each state s:

$$\pi^{*}(s) = \arg\max_{a\in\mathcal{A}(s)}\mathcal{Q}\left(s,a\right) \qquad \forall\, s,\pi. \tag{7}$$

With $\mathcal{Q}$-learning, the slicing controller learns the optimal $\mathcal{Q}$-values iteratively, based on the available information. At time-step t, the slicing controller in state s selects action a. It earns a reward $\hat{\mathcal{R}}$ and ends in state $s^{'}$ at time-step $t+\upsilon$, as the actions can span several time-steps. The $\mathcal{Q}$-learning process can be represented by the following updated equation:

$$\mathcal{Q}\left(s,a\right) \leftarrow \mathcal{Q}\left(s,a\right) + \Theta\left[\hat{\mathcal{R}} + \varphi^{\upsilon}\max_{a^{'}\in\mathcal{A}(s)}\mathcal{Q}(s^{'},a^{'}) - \mathcal{Q}\left(s,a\right)\right] \tag{8}$$

where Θ is the learning rate and $\hat{\mathcal{R}}$ is the discounted accumulation of all single time-step rewards $\mathcal{R}_{\tau}$, received while executing action a for a duration τ and is given by

$$\hat{\mathcal{R}} = \sum_{j=0}^{\upsilon-1}\varphi^{j}\mathcal{R}_{\tau}. \tag{9}$$

Further, by updating the $\mathcal{Q}$-values over sufficiently large duration, and by adjusting Θ and φ, $\mathcal{Q}\left(s,a\right)$ is guaranteed to converge to $\mathcal{Q}_{\pi^{*}}\left(s,a\right)$ [17].

The overall slicing strategy is given as Algorithm 1. Initially, the $\mathcal{Q}$-values are set to zero. Before the application of $\mathcal{Q}$-learning algorithm, the slicing controller performs an initial slice allocation for different slices based on traffic demand estimation for each slice. This is done for state initialization of different slices. Existing radio resource slicing solutions either

Algorithm 1: RL-BASED SLICING STRATEGY.

Initialization
$t \leftarrow 0$; $\mathcal{Q}(s,a) \leftarrow 0,\ \forall\ s \in \mathcal{S},\ a \in \mathcal{A}$
Input $\mathcal{X}, \mathcal{M}, \tau$
Execute Algorithm 2 for initial resource slicing
Iterate
while (*learing period is active*) **do**
  for *each slice* **do**
    observe state s
    generate a random number $\Omega \in \{0,1\}$
    if $\Omega \leq \epsilon$ **then**
      select a randomly
      else
        select $a = \arg\max_{a \in \mathcal{A}(s)} \mathcal{Q}(s,a)$
      end
    end
    $\hat{\mathcal{R}} \leftarrow 0, \quad \upsilon \leftarrow 0$
    while *(slice m in state s)* **do**
      observe $\mathcal{R}_\tau$
      $\hat{\mathcal{R}} \leftarrow \hat{\mathcal{R}} + \varphi^{\upsilon}\mathcal{R}_\tau$
      $\upsilon \leftarrow \upsilon + 1$
    end
    observe state $s^{'}$
    update $\mathcal{Q}(s,a)$ according to (8)
    $t \leftarrow t + \upsilon$
  end
end

use *bandwidth-based* or *resource-based* provisioning for allocating radio resources to different slices. In case of the former approach, resource allocation to each slice is defined in terms of aggregate throughput that will be obtained by the flows of that slice. On the other hand, the latter approach allocates a fraction of overall resources to each slice. Since every slice has different requirements, we adopt a more generic approach and consider a combination of bandwidth-based and resource-based provisioning of resources for each slice. Let, $\mathcal{W}_B^m$ denote the bandwidth demand (e.g., in kb/s) of the mth slice, which is given by $\sum_{j \in \mathcal{F}^m} \mathcal{G}_j$, where $\mathcal{F}^m$ is the set of flows for the mth slice and $\mathcal{G}_j$ denotes the minimum required bandwidth for the jth flow. Similarly, let $\mathcal{W}_R^m$ denote the resource demand of the mth slice in terms of minimum required fraction of RUs. The bandwidth requirement are translated into radio resource requirements by using the transmission rate of the mth slice, given by R_m, that is actually determined by the base rate, which refers to the minimum supported data rate by a base station based on the lowest order supported modulation and coding scheme (MCS). The initial resource slicing for the mth slice is given in Algorithm 2. In practice, traffic demand estimation by the slicing controller is facilitated by the RAN, as shown by Feedback 1 in Fig. 1. The lowest order MCS information is known *a priori* and can be embedded into Feedback 1.

Since $\mathcal{Q}$-learning is an online iterative learning algorithm, it performs two distinct types of actions. In the *exploration* mode, the slicing controller randomly selects one of the possible actions in order to enhance its future decisions. On the contrary, in the *exploitation* mode, the slicing controller prefers actions it has tried in the past and found effective. We assume that

Algorithm 2: Initial Resource Slicing.

Input: $\{\mathcal{X}, \mathcal{W}_B^m, \mathcal{W}_R^m\}$
Overall resource demand:
$\mathcal{W}_O^m = \mathcal{W}_R^m + 1/R_m \cdot \left(\sum_{j \in \mathcal{F}^m} \mathcal{G}_j\right)$
Slice allocation:
allocate RUs to slice m based on $\mathcal{W}_O^m$

the slicing controller in state s explores with probability ϵ and exploits previously stored $\mathcal{Q}$-values with probability $1-\epsilon$. In any state s, not all the actions are feasible. To maintain isolation between slices, the slicing controller must ensure that the same RU is not allocated to two different slices, i.e.,

$$\sum_{m=1}^{M} \alpha_{m,b,i} = 1 \qquad \forall\, i, b. \tag{10}$$

The state of each slice is observed at each time-step. If the network is in state s, action a is maintained and the discounted accumulation of single time-step rewards is updated according to (9). However, if the state of slice has changed to $s^{'}$, action a is terminated, and $\mathcal{Q}$-values are updated according to (8) until the end of learning period.

Note that Feedbacks 2 and 3, as illustrated in Fig. 1, play an important role in deriving the optimal slicing strategy. Feedback 2 provides current utilization of radio resources allocated to mth slice at a base station. The resource utilization is measured as the ratio of used RUs to the allocated RUs. Feedback 3 provides the necessary parameters for calculating the sum utility of a slice. The slicing controller maps these parameters to slice-specific utility function.

C. PDS Learning-Based Slicing Strategy

The Algorithm 1, which is based on $\mathcal{Q}$-learning algorithm derives an optimal strategy under the assumption that environment dynamics are completely unknown *a priori*. Consequently, its performance, in terms of convergence, can be slower. In practice, partial knowledge about environment dynamics can be available. By exploiting such partially known dynamics, the runtime performance of radio resource slicing strategy can be significantly improved through PDS learning (PDSL) algorithm [18].

The PDS describes the state of the system *after* the known dynamics have taken place, but *before* the unknown dynamics have occurred. We define the PDS of the mth slice as $\hat{s}_t^m \in \mathcal{S}$ (since the set of PDSs is the same as the set of states). Based on the PDS, the transition from state s_t^m to s_{t+1}^m can be represented as:

1) State at time t: $s_t^m = \{\Lambda_t^m,\ \mathcal{U}_t^m,\ \Psi_t^m\}$.
2) PDS at time t: $\hat{s}_t^m = \{\hat{\Lambda}_t^m,\ \hat{\mathcal{U}}_t^m,\ \hat{\Psi}_t^m\}$, where $\hat{\Lambda}_t^m = \hat{\Lambda}_{t+1}^m$, $\hat{\mathcal{U}}_t^m = \mathcal{U}_t^m \pm \delta\mathcal{U}$, $\hat{\Psi}_t^m = \Psi_t^m \pm \delta\Psi$.
3) State at time $t+1$: $s_{t+1}^m = \{\Lambda_{t+1}^m, \mathcal{U}_{t+1}^m, \Psi_{t+1}^m\}$.

By introducing the PDS, the state transition probability for the mth slice can be split into known and unknown parts such that the former accounts for transition from current state to the PDS, i.e., $s \rightarrow \hat{s}$, and the latter accounts for the transition from PDS to the next state, i.e., $\hat{s} \rightarrow s^{'}$. Hence, the state transition

probability for the mth slice is given by

$$\mathcal{P}(s^{'}, s, a) = \sum_{\hat{s}\in\mathcal{S}} \mathcal{P}_c(\hat{s}, s, a) \cdot \mathcal{P}_u(s^{'}, \hat{s}, a) \tag{11}$$

where the subscripts c and u denote the known and unknown components, respectively. Similarly, the reward function is factorized as follows:

$$\mathcal{R}(s, a) = \mathcal{R}_c(s, a) + \sum_{\hat{s}\in\mathcal{S}} \mathcal{P}_c(\hat{s}, s, a) \cdot \mathcal{R}_u(\hat{s}, a). \tag{12}$$

Next, we define the PDS value function, which plays a similar role as the action-value function in the $\mathcal{Q}$-learning algorithm. The optimal PDS value function, denoted by $\hat{V}^*$, is given by

$$\hat{V}^*(\hat{s}) = \mathcal{R}_u(\hat{s}, a) + \varphi \sum_{s^{'}\in\mathcal{S}} \mathcal{P}_u(s^{'}, \hat{s}, a) V^*(s^{'}) \tag{13}$$

where $V^*(s^{'})$ denotes the optimal state-value function and is given by

$$V^*(s) = \arg\max_{a\in\mathcal{A}(s)} \left\{ \mathcal{R}_c(s, a) + \sum_{\hat{s}\in\mathcal{S}} \mathcal{P}_c(\hat{s}, s, a) \hat{V}^*(\hat{s}) \right\}. \tag{14}$$

Given the optimal PDS value function, the optimal strategy is given by

$$\pi^*_{\mathrm{PDS}}(s) = \arg\max_{\pi} \hat{V}^*(\hat{s}). \tag{15}$$

In order to implement the PDSL algorithm, we assume that the state transition probability for the mth slice can be determined from conditionally independent resource allocation, sum utility, and resource utilization transition probabilities as follows:

$$\mathcal{P}(s^{'}, s, a) = \mathcal{P}^{\Lambda}(\Lambda^{'}, \Lambda, a) \cdot \mathcal{P}^{\mathcal{U}}(\mathcal{U}^{'}, \mathcal{U}, a) \cdot \mathcal{P}^{\Psi}(\Psi^{'}, \Psi, a).$$

To keep the state-space finite, we define thresholds Y_1, Y_2, Y_3, and Y_4, based on which the state of sum utility as well as resource utilization of a slice can be characterized as very high (VH), high (HI), medium (M), low (LO), and very low (VL). The state transition probabilities are represented as a matrix $[\mathcal{P}_{j,n}]_{5\times5}$, where each entry $\mathcal{P}_{j,n}$ denotes the transition probability from state j to state n. The PDSL algorithm exploits known dynamics that include the stochastic knowledge of mean arrival and departure rate of users for a slice, mean resource demand per user for a slice, and the information of previous sum utility and resource utilization states through Feedbacks 2 and 3.

The PDSL-based slicing strategy is outlined as Algorithm 3. Similar to Algorithm 2, the initial resource slicing for each slice is performed by using Algorithm 2. In the main body of the algorithm, the PDSL-based slicing strategy selects an action in a greedy manner in each time-step, as opposed to randomized selection in $\mathcal{Q}$-learning based slicing strategy. Further, updating of the state and PDS value functions provides information about state-value function at many states. Hence, the PDSL-based slicing strategy significantly enhances the efficiency as compared to the $\mathcal{Q}$-learning-based slicing strategy. The computational and memory complexity of PDSL, as compared to $\mathcal{Q}$-learning, has been evaluated in [18].

Algorithm 3: PDSL-BASED SLICING STRATEGY.

Initialization
$t \leftarrow 0; \quad \hat{V}_t(\hat{s}) \leftarrow 0, \ \forall \ s \in \mathcal{S}$
Input $\mathcal{X}, \mathcal{M}$
Execute Algorithm 2 for initial resource slicing
Iterate
while (*learing period is active*) **do**
  for *each slice* **do**
    take a greedy action
    $a_t = \arg\max_{a\in\mathcal{A}(s)} \{\mathcal{R}_c(s_t, a) + \sum_{\hat{s}\in\mathcal{S}} \mathcal{P}_c(\hat{s}, s_t, a)\hat{V}_t(\hat{s})\}$
    observe the tuple: $\{s_t, a_t, \hat{s}_t, \mathcal{R}^t_u, s_{t+1}\}$
    $V_t(s_{t+1}) = \arg\max_{a\in\mathcal{A}(s)} \{\mathcal{R}_c(s_{t+1}, a) + \sum_{\hat{s}\in\mathcal{S}} \mathcal{P}_c(\hat{s}, s_{t+1}, a)\hat{V}_t(\hat{s})\}$
    $\hat{V}_{t+1}(\hat{s}_t) \leftarrow (1-\Theta^t)\hat{V}_t(\hat{s}_t) + \Theta^t [\mathcal{R}^t_u + \varphi V_t(s_{t+1})]$
    $t \leftarrow t + \upsilon$
  end
end

IV. HAPTIC COMMUNICATIONS OVER 5G NETWORKS

The haptic sense (sense of touch) establishes a link between humans and unknown environments in a similar way as the auditory and visual senses. Differing from these senses, the haptic sense occurs bilaterally, i.e., a touch is sensed by imposing a motion on an environment and feeling the environment by a distortion or reaction force.

The functional architecture of haptic communications over the Tactile Internet is shown in Fig. 2. The *master* domain consists of a human and a human system interface, i.e., a haptic device that converts human input to a haptic input. The *slave* domain consists of a slave robot or a teleoperator that interacts with the remote environment. The *network* domain provides the medium for bilateral communication between master and slave domains. On the forward path, the haptic device sends command signals to control the operation of the teleoperator, which sends a haptic feedback on the reverse path. Through command and feedback signals, energy is exchanged between master and slave domains thereby closing a global control loop involving the human, the communication network, and the remote environment [24].

A. Radio Resource Allocation Requirements

Haptic communications creates unique requirements for radio resource allocation due to its bidirectional nature and presence of a global control loop. We identify and motivate these requirements, which would be used for radio resource customization (Section V), as follows.

1) Unlike traditional multimedia applications, uplink and downlink sessions are coupled in haptic communications due to the bidirectional exchange of haptic information [5]. Therefore, to ensure tracking performance between master and slave domains, *joint* resource allocation in uplink and downlink is necessary.
2) In order to ensure the stability of the global control loop, which arises due to bidirectional exchange of haptic information [24], a *bounded delay* requirement must be considered during resource allocation.

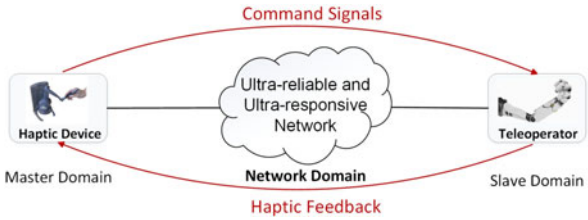

Fig. 2. Illustration of haptic communications over 5G networks.

3) Typical haptic systems use local control loops to overcome device dynamics and display high frequency haptic effects to the user [5]. It is particularly important to periodically refresh the local control loops at the master and slave domains. Therefore, from a radio resource allocation perspective, a *minimum rate* must be guaranteed throughout the haptic session.
4) The bidirectional flow of command and feedback signals between master and slave domains results in symmetrical traffic in uplink and downlink. Therefore, haptic communications requires *symmetric* resource allocation in uplink and downlink.

Note that the simultaneous consideration of these requirements is unique to haptic communications only.

V. Hap-SliceR—Radio Resource Customization

Radio resource customization provides the flexibility of implementing specific radio resource management schemes within the allocated slice. The main focus of radio resource customization in Hap-SliceR is radio resource allocation for haptic communications over LTE-A networks.

A. Network Model for Radio Resource Allocation

We consider an LTE-A RAN within the 5G ecosystem, termed as 5G LTE-A. Unlike the legacy LTE-A RAN, the 5G LTE-A RAN is assumed to be empowered with the required *networking* capabilities, which are critical for realizing the Tactile Internet [3]. The 5G LTE-A RAN employs conventional multiple access schemes, i.e., single carrier frequency division multiple access (SC-FDMA) in the uplink and orthogonal frequency division multiple access (OFDMA) in the downlink. We assume that the slicing controller allocates a slice $\mathcal{X}^T$, from the overall resource pool $\mathcal{X}$, for haptic communications. Further, $\mathcal{X}^T$ is split into uplink and downlink RB sets, indexed by $\mathcal{L}^{\mathrm{UL}} \triangleq \{1,\ldots,l,\ldots,L^U\}$ and $\mathcal{L}^{\mathrm{DL}} \triangleq \{1,\ldots,l,\ldots,L^D\}$, respectively. The haptic devices, which are randomly distributed, are indexed by the set $\mathcal{K} \triangleq \{1,\ldots,k,\ldots,K\}$. The channel is assumed to exhibit block fading characteristics and the coherence time is greater than the transmission time interval (TTI). In the uplink, certain restrictions apply for power and RB allocation owing to the use of SC-FDMA [25]. First, a single RB can only be allocated to at most one user (*exclusivity* restriction). Second, multiple RBs allocated to a user must be adjacent (*adjacency* restriction). Third, the transmit power on all RBs allocated to a user should be equal. On the other hand, the use of OFDMA in the downlink only requires the exclusivity restriction.

For the uplink model, assume that a set of consecutive RBs, $\mathcal{L}_k^{\mathrm{UL}}$, is allocated to a user k in the current TTI. The uplink signal-to-noise ratio (SNR) for the kth user over the lth RB is given by $\gamma_{k,l}^{\mathrm{UL}} = \frac{P_{k,l}^{\mathrm{UL}}|h_{k,l}^{\mathrm{UL}}|^2}{\sigma^2}$, where $P_{k,l}^{\mathrm{UL}}$ is the uplink transmission power of kth user over the lth RB, $h_{k,l}^{\mathrm{UL}}$ is the channel coefficient, and σ^2 denotes the noise power. Using Shannon's capacity formula, the upper bound on the achievable data rate for the kth user in the uplink is given by

$$R_k^{\mathrm{UL}} = B \cdot L_k^{\mathrm{UL}} \log_2(1 + \mu \cdot \gamma_{\mathrm{eff},k}^{\mathrm{UL}}) \tag{16}$$

where B is the bandwidth of each RB, $L_k^{\mathrm{UL}} = \mid \mathcal{L}_k^{\mathrm{UL}} \mid$ denotes the cardinality of set $\mathcal{L}_k^{\mathrm{UL}}$, $\gamma_{\mathrm{eff},k}^{\mathrm{UL}}$ is the uplink effective SNR [26] for the kth user, and μ denotes the power control policy [27] to be discussed later. As per [28], the effective SNR for SC-FDMA transmission is given by

$$\gamma_{\mathrm{eff},k}^{\mathrm{UL}} = \frac{1}{L_k^{\mathrm{UL}}} \sum_{l \in \mathcal{L}_k^{\mathrm{UL}}} \gamma_{k,l}^{\mathrm{UL}} \tag{17}$$

where $\gamma_{k,l}^{\mathrm{UL}}$ is the instantaneous uplink SNR for the kth user over the lth RB. Similarly, for the downlink model, we assume that a set of RBs, $\mathcal{L}_k^{\mathrm{DL}}$, is allocated to a user k in the current TTI. The upper bound on the achievable data rate for the kth user in the downlink is given by

$$R_k^{\mathrm{DL}} = B \cdot L_k^{\mathrm{DL}} \log_2(1 + \mu \cdot \gamma_{\mathrm{eff},k}^{\mathrm{DL}}) \tag{18}$$

where $L_k^{\mathrm{DL}} = \mid \mathcal{L}_k^{\mathrm{DL}} \mid$ and $\gamma_{\mathrm{eff},k}^{\mathrm{DL}}$ is the downlink effective SNR [26] for the kth user such that

$$\gamma_{\mathrm{eff},k}^{\mathrm{DL}} = -\beta \log\left(\frac{1}{L_k^{\mathrm{DL}}} \sum_{l \in \mathcal{L}_k^{\mathrm{DL}}} e^{-\gamma_{k,l}^{\mathrm{DL}}/\beta}\right) \tag{19}$$

where β is a parameter depending on MCS and $\gamma_{k,l}^{\mathrm{DL}}$ is the instantaneous downlink SNR for the kth user over the lth RB.

For haptic communications, bounded delay is a key service requirement. Due to the time varying nature of wireless channel, deterministic delay guarantees cannot be provided. In this respect, the statistical QoS theory provides an efficient approach to characterize the dynamics of the delay distribution. In the literature, statistical QoS guarantees have been extensively investigated in the context of *effective bandwidth* $(\mathcal{B}_E)$ [29] and *effective capacity* $(\mathcal{C}_E)$ [30] functions. Specifically, for a dynamic queuing system, under sufficient conditions, the queue length process, $\mathcal{Q}(t)$ converges in distribution to a random variable $\mathcal{Q}(\infty)$ such that $-\lim_{z_t \to 0} \frac{\log(Pr\{\mathcal{Q}(\infty) > z_t\})}{z_t} = \theta_k$, which states that the probability of the queue length exceeding a certain threshold z_t decays exponentially fast as z_t increases, and the parameter θ_k determines the decaying rate.

In context of delay QoS guarantees, we define θ_k as the statistical QoS exponent of the kth user that characterizes its steady-state delay violation probability such that

$$\mathcal{V}_k = Pr\{d_k > d_k^T\} = e^{-\theta_k \varphi d_k^T} \tag{20}$$

where d_k is the delay, d_k^T is the delay bound, and φ is jointly determined by the arrival and service processes [31]. A smaller θ_k indicates a looser QoS constraint whereas a larger θ_k implies

a more stringent QoS requirement. Owing to the presence of a global control loop, the delay bound is the same in both uplink and downlink, and therefore, (20) is applicable in both scenarios.

B. Radio Resource Allocation Problem

Considering the radio resource allocation requirements of haptic communications, which have been discussed in Section IV, we formulate the resource allocation problem as follows:

$$\min_{\mathcal{L}_k^{\mathrm{UL}},P_{k,l}^{\mathrm{UL}},\mathcal{L}_k^{\mathrm{DL}},P_{k,l}^{\mathrm{DL}}} \Delta = \left| \sum_{k\in\mathcal{K}} R_k^{\mathrm{UL}} - \sum_{k\in\mathcal{K}} R_k^{\mathrm{DL}} \right| \tag{21}$$

s.t.

$$R_k^f \geq R_{min}^f \quad \forall\, k \in \mathcal{K},\ f \in \{\mathrm{UL},\mathrm{DL}\} \tag{22a}$$

$$P_{k,l}^{\mathrm{UL}} = P_{k,n}^{\mathrm{UL}} \quad \forall\, k \in \mathcal{K}, \quad \forall\, l,n \in \mathcal{L}^{\mathrm{UL}} \tag{22b}$$

$$\mathcal{L}_k^f \cap \mathcal{L}_j^f = \varnothing \quad \forall\, k \neq j,\ k,j \in \mathcal{K},\ f \in \{\mathrm{UL},\mathrm{DL}\} \tag{22c}$$

$$\left\{ \min(\mathcal{L}_k^{\mathrm{UL}}), \min(\mathcal{L}_k^{\mathrm{UL}})+1, \ldots, \max(\mathcal{L}_k^{\mathrm{UL}}) \right\} \cap$$
$$\left\{ \mathcal{L}_1^{\mathrm{UL}} \cup \cdots \mathcal{L}_{k-1}^{\mathrm{UL}} \cup \mathcal{L}_{k+1}^{\mathrm{UL}} \cup \cdots \mathcal{L}_k^{\mathrm{UL}} \right\} = \varnothing\ \forall\, k \in \mathcal{K} \tag{22d}$$

$$\mathcal{V}_k^f \leq \Gamma \qquad \forall\, k \in \mathcal{K},\ f \in \{\mathrm{UL},\mathrm{DL}\} \tag{22e}$$

$$\left| R_k^{\mathrm{UL}} - R_k^{\mathrm{DL}} \right| \leq \eta \quad \forall\, k \in \mathcal{K}. \tag{22f}$$

The objective of the optimization problem in (21) is power and RB allocation for different users in both uplink and downlink while satisfying the resource allocation requirements of haptic communications as well as the specific constraints of SC-FDMA and OFDMA.

Owing to the joint resource allocation requirement, the objective function aims to minimize the difference of sum rate in the uplink and the downlink. The constraint (22a) captures the minimum rate requirement in both uplink and downlink. The constraints (22b)–(22d) are specific to SC-FDMA, i.e., (22b) is the equal power allocation constraint, (22c) is the exclusivity restriction, and (22d) models the adjacency restriction. Note that (22c) also applies to the downlink. Further, constraint (22e) appears due to the bounded delay requirement of haptic communications in both uplink and downlink. Note that the delay bound is similar in both uplink and downlink. Finally, owing to the symmetric resource allocation requirement, we couple the uplink and the downlink through constraint (22f).

The optimization problem (21) is particularly challenging to solve due to its combinatorial nature. Besides, standard Lagrangian duality techniques cannot be employed due to the analytical intractability associated with the adjacency restriction of RB allocation in the uplink. In order to solve (21), we first introduce a power control policy that dictates optimal power allocation for a given delay bound. After that, we formulate an equivalent BIP and propose a low-complexity heuristic algorithm for resource allocation.

C. Power Control Policy

The power control policy, denoted by μ, gives the relationship between R_k, θ_k, and allocated power. Conventionally, the power control policy is expressed as a function of SNR only. However in our case, it is a function of both SNR and QoS exponent.

Lemma 1: The optimal power control policy (which is a function of θ_k and $\gamma_{\mathrm{eff},k}$) for the kth user is given by

$$\mu_{\mathrm{opt},k} = \frac{1}{\gamma_{\mathrm{eff},k}} \left[\left(\frac{\gamma_0}{\gamma_{\mathrm{eff},k}} \right)^{\frac{1}{q-1}} - 1 \right] \tag{23}$$

where $q = -\theta_k B L_k / \log 2$ and γ_0 is the cutoff effective SNR.

Proof: The proof directly extends from [27].

Note that the power control policy dictates both uplink and downlink power allocation. Since the power control policy depends on the statistical QoS exponent, it is important to determine θ_k in order to guarantee a specific delay QoS requirement given by $\left(d_k^T, \Gamma\right)$. As per [31], the parameter φ in (20) is determined by the intersection point of effective bandwidth and effective capacity curves (as a function of θ_k) such that $\mathcal{B}_E(\theta_k^*) = \mathcal{C}_E(\theta_k^*) = \varphi$. We assume that the traffic generated by haptic devices follows a Poisson arrival process with rate λ. Using the moment generating function of the Poisson process, given by $e^{\lambda(e_k^\theta - 1)}$ [29], the parameter φ can be obtained as $\varphi = \lambda \cdot \frac{e^{\theta_k^*}-1}{\theta_k^*}$. Substituting for φ in (20) and using the constraint (22e), we get

$$\theta_k^* = \log\left(1 - \frac{\log\Gamma}{\lambda d_k^T}\right). \tag{24}$$

D. BIP Formulation

To make the optimization problem (21) more tractable, we first define an uplink RB allocation matrix [32] that ensures the adjacency of allocated RBs for each user in the uplink. The RB allocation matrix is of the order $\mid \mathcal{L}^{\mathrm{UL}} \mid \times N$, where each row corresponds to the RB index and each column corresponds to a feasible (meeting adjacency restriction) RB allocation pattern, and N denotes the total number of feasible allocation patterns given by $N = 0.5 \times (\mid \mathcal{L}^{\mathrm{UL}} \mid^2 + \mid \mathcal{L}^{\mathrm{UL}} \mid)$. The basic idea of this RB allocation matrix, for the case of three RBs, is illustrated by

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \tag{25}$$

In any allocation pattern (i.e., any column), a "1" is placed when the RB is allocated to a user, otherwise a "0" is placed.

Next, we define a RB indicator vector $\boldsymbol{x} \triangleq [\boldsymbol{x}_k^{\mathrm{UL}}\ \boldsymbol{x}_k^{\mathrm{DL}}]_{K\times 2}$, where $\boldsymbol{x}_k^{\mathrm{UL}} = [x_{k,n}]_{N\times 1}$ and $\boldsymbol{x}_k^{\mathrm{DL}} = [x_{k,l}]_{L^D\times 1}$, such that each entry $x_{k,n} \in \{0,1\}$ and $x_{k,l} \in \{0,1\}$ indicates resource allocation for the kth user in the uplink and the downlink, respectively. In the uplink, the variable $x_{k,n}$ denotes whether the nth allocation pattern is allocated to the kth user or not. On the other hand, in the downlink, the variable $x_{k,l}$ indicates whether the lth RB is allocated to the kth user or not.

We formulate the BIP equivalent of the optimization problem in (21) as follows:

$$\min_{\boldsymbol{x}} \ddot{\Delta} = \left| \sum_{k=1}^{K}\sum_{n=1}^{N} x_{k,n} R_{k,n}^{\mathrm{UL}} - \sum_{k=1}^{K}\sum_{l=1}^{L} x_{k,l} R_{k,l}^{\mathrm{DL}} \right| \tag{26}$$

s.t.

$$x_{k,f}(x_{k,f}-1)=0 \qquad \forall\, k\in\mathcal{K}, f\in\{n,l\} \tag{27a}$$

$$\sum_{k=1}^{K}\sum_{n=1}^{N} x_{k,n}M_{l,n}=1 \quad \forall\, l\in\mathcal{L}^{\mathrm{UL}} \tag{27b}$$

$$\sum_{n=1}^{N} x_{k,n}=1 \qquad \forall\, k\in\mathcal{K} \tag{27c}$$

$$\sum_{k=1}^{K} x_{k,l}=1 \qquad \forall\, l\in\mathcal{L}^{\mathrm{DL}} \tag{27d}$$

$$\left|x_{k,n}R^{\mathrm{UL}}_{k,n}-x_{k,l}R^{\mathrm{DL}}_{k,l}\right|\le\eta \quad \forall\, k\in\mathcal{K}. \tag{27e}$$

The constraint (27a) is a pure binary integer constraint that ensures that $x_{k,n}, x_{k,l}\in\{0,1\}$. The constraint (27b), where $M_{l,n}$ denotes the lth row and nth column of the matrix $\boldsymbol{M}$, ensures the exclusivity of allocated RBs in the uplink. The constraint (27c) ensures that at most one allocation pattern is chosen for each user in the uplink. Similarly, the constraint (27d) ensures that each RB can be exclusively allocated to one user in the downlink. Finally, the constraint (27e) is the binary version of the constraint (22f). In transforming (21) to (26), the power allocation is performed using the optimal power control policy, given by (23). The transmit power in the uplink $P_{k,n}$ is a function of $R^{\mathrm{UL}}_{\min}$, θ_k, and $\gamma^{\mathrm{UL}}_{\mathrm{eff},k}$, and is obtained by setting $R^{\mathrm{UL}}_{k}=R^{\mathrm{UL}}_{\min}$ [using (23)] and $P^{\mathrm{UL}}_{k,l}=\frac{P_{k,n}}{L_{k,n}}$, where $L_{k,n}$ denotes the cardinality of set of RBs $(\mathcal{L}_{k,n})$ allocated to user k, when the nth allocation pattern is used. The power allocation in the downlink is performed similarly, i.e., using (23) and setting $R^{\mathrm{DL}}_{k}=R^{\mathrm{DL}}_{\min}$. Hence, the constraints (22a) and (22b) are implicitly accommodated during power allocation. Although the optimization problem (26) is more tractable and apparently simpler than (21), the solution is still exponentially complex.

Algorithm 4: HEURISTIC RB ALLOCATION ALGORITHM.

Step 1: Initialization
$\mathcal{L}^{UL}=\{1,\cdots,L^U\}$, $\mathcal{L}^{DL}=\{1,\cdots,L^D\}$,
$\mathcal{N}=\{1,\cdots,N\}$, $\mathcal{I}^{UL}=\varnothing$, $\mathcal{I}^{DL}=\varnothing$
step 2: Initial DL Scheduling
while $\mathcal{L}^{DL}\neq\varnothing$ **do**
  allocate RB l^* to the user k^* with the best channel
end
step 3: Optimized uplink Allocation; $\mathcal{N}_k=\varnothing$;
while $\mathcal{L}^{UL}\neq\varnothing$ **do**
  $\forall\, n\in\mathcal{N}, \forall\, k$ calculate δ_k
  $(k^*,n^*)=\arg\min(\mid\delta_k\mid)$; $\mathcal{N}_{k^*}=\mathcal{N}_k\cup a^*$
  if $\mid\delta_k\mid\le\eta$ **then**
    $s\leftarrow s+1$, where s denotes successful allocation
    else
      $\mathcal{I}^{UL}=\mathcal{I}^{UL}\cup k^*$
    end
  end
end
step 4: Optimized downlink Allocation; $\mathcal{L}^{DL}_k=\varnothing$
while $\mathcal{L}^{DL}\neq\varnothing$ **do**
  $\forall\, l\in\mathcal{L}^{DL}, \forall\, k$ calculate χ_k
  $(k^*,l^*)=\arg\min(\mid\chi_k\mid)$; $\mathcal{L}^{DL}_{k^*}=\mathcal{L}^{DL}_k\cup l^*$
  if $\mid\chi_k\mid\le\eta$ **then**
    $s\leftarrow s+1$, where s denotes successful allocation
    else
      $\mathcal{I}^{DL}=\mathcal{I}^{DL}\cup k^*$
    end
  end
end
step 5: Iterate
while $(\mathcal{I}^{UL}\neq\varnothing)\|(\mathcal{I}^{DL}\neq\varnothing)$ **do**
  repeat Steps 3 and 4 with $\mathcal{L}^{DL}_{k^*}$ and $\mathcal{L}^{UL}_{k^*}$ respectively
end

E. Low-Complexity Heuristic Algorithm

In state-of-the-art LTE-A networks, packet scheduling is a key resource allocation technique and is done *independently* in the uplink and the downlink. The downlink scheduler dynamically determines the terminals to transmit to along with the set of RB, for each of these terminals, on which the downlink transmission should take place. The uplink scheduler performs a similar task in the uplink, i.e., to dynamically determine, for each TTI, which users are to transmit and on which uplink resources. The uplink scheduler also controls the transport format of users. For a detailed description of packet scheduling, interested readers are referred to [33].

However, joint uplink and downlink scheduling necessitates an information exchange mechanism between uplink and downlink schedulers. As both uplink and downlink schedulers are located inside the eNodeB, implementation of such an information exchange mechanism is not an insurmountable task in practice. With this assumption, we propose a low-complexity heuristic algorithm to solve (26) for efficient implementation in practical scenarios. The key steps of the proposed algorithm (given in Algorithm 4) are described as follows. The heuristic algorithm follows a steepest descent approach in which the feasible allocation is determined as the one that maximizes the decrease in objective function. Initially the downlink scheduler performs channel-aware scheduling wherein an RB is allocated to a user with the best channel. In the main body of the algorithm, based on the initial downlink allocation, the uplink scheduler performs optimized allocation in the uplink. For each user and for all the allocation patterns, it computes the difference in the uplink and the downlink rates, given by δ_k, such that $\delta_k=\sum_{n'\in\mathcal{N}_k\cup a}B\cdot\mid\mathcal{N}_k\mid\log_2(1+\mu\cdot\gamma^{\mathrm{UL}}_{\mathrm{eff},k})-\sum_{l'\in\mathcal{L}^{\mathrm{DL}}_k}B\cdot\mid\mathcal{L}^{\mathrm{DL}}_k\mid\log_2(1+\mu\cdot\gamma^{\mathrm{DL}}_{\mathrm{eff},k})$. The objective is to find the user and allocation pattern pair (k^*,n^*) that minimizes δ_k. The user allocation is feasible if the symmetric resource allocation constraint is met. Otherwise, it is appended to an infeasible allocation set, which is initially empty. Based on the optimized uplink scheduling, the downlink scheduler performs optimized downlink scheduling in a similar way as the uplink scheduler. The objective is to find the user and RB pair (k^*,l^*) that minimizes χ_k, which is given by $\chi_k=\sum_{l'\in\mathcal{L}^{\mathrm{UL}}_k}B\cdot\mid\mathcal{L}^{\mathrm{UL}}_k\mid\log_2(1+\mu\cdot\gamma^{\mathrm{UL}}_{\mathrm{eff},k})-\sum_{l'\in\mathcal{L}^{\mathrm{DL}}_k\cup\, l}B\cdot\mid\mathcal{L}^{\mathrm{DL}}_k\mid\log_2(1+\mu\cdot\gamma^{\mathrm{DL}}_{\mathrm{eff},k})$. After achieving optimized uplink and downlink allocation, the algorithm continues until the uplink and the downlink infeasible allocation sets are empty.

VI. Hap-SliceR—Practical Aspects

Fig. 1 illustrates the functional architecture of radio resource slicing in a cellular network along with the slicing methodology. The abstraction module and the slicing controller are collocated within a single entity, which is logically located between the core network and the RAN. Physically, both can be realized in any standard cloud computing infrastructure closer to the edge of the RAN. The RAN edge not only offers a service environment with low latency but also provides direct access to real-time information that can be easily exploited by the slicing controller.

Unlike other solutions, Hap-SliceR allows base stations to make independent radio resource customization decisions. Hence, the base stations implement legacy radio resource scheduling protocols on fine time scale (on the order of TTIs). On the other hand, the radio resource slicing decisions are made by the slicing controller and exchanged with the RAN on rather coarse time scale (e.g., on the order of several TTIs). The radio resource slicing decisions provide updated allocation of radio resources to each vertical application at each base station based on the slice utility requirements and global view of RAN utilization. The design of the interface between the slicing controller and the RAN is an implementation specific issue, and therefore, beyond the scope of this paper. It can possibly be based on the OpenFlow protocol.

VII. Performance Evaluation

We evaluate the performance of Hap-SliceR through system-level simulation studies. Our simulation model is based on a seven-cell hexagonal grid layout. For simplicity, we consider the scenario of only two vertical applications, and hence two radio slices. The first slice (Slice A) provides connectivity service for haptic communications, whereas the second slice (Slice B) is dedicated to conventional H2H communications. For Slice A, haptic devices are assumed to be randomly distributed throughout the total coverage area according to density λ_A. Similarly, H2H users are Poisson distributed according to density λ_B. The channel model accounts for small-scale Rayleigh fading, large-scale path loss, and shadowing. We assume a fixed overall radio resource pool $\mathcal{X}$, and implement Algorithm 1 to find the optimal slicing policy. We consider two different utility functions in Hap-SliceR. For H2H communications, we assume a rate-based utility function such that $\mathcal{U}(\text{rate}) = \left(1 + e^{-c_1(\text{rate}-c_2)}\right)^{-1}$, where c_1 and c_2 are constants. The utility of haptic communications is jointly dictated by the rate and the delay based utility functions such that the latter (which is inspired and adapted from [34]) is given by $\mathcal{U}(\text{delay}) = -c_3 \tanh(10 \times \Gamma - 1.35) + c_4$, where Γ is the delay bound violation probability, and c_3 and c_4 are constants. Typical end-to-end latency for real-time haptic interaction is $\sim$ 1 to 2 ms. Thus, the air-interface transmission budget is limited to 100–200 μ s [2]. To realize this transmission budget, the physical layer OFDM numerology requires modification. Recently, an OFDM-based 5G air-interface design [8] has been proposed for a latency requirement of 1-2 ms. We adopt the OFDM numerology from [8] to achieve a TTI of $\sim$100 to 200 μs. Considering similar number of OFDM symbols and subcarriers per RB as in legacy LTE-A networks, such modifications of OFDM numerology will demand higher bandwidth per RB; a comprehensive analysis of which has been conducted in [8]. Other simulation parameters, including those for $\mathcal{Q}$-learning, are given in Table I. We consider three distinct types of haptic devices depending on the haptic degrees of freedom (DOF), i.e., 1-DOF, 3-DOF, and 6-DOF device. Typically, haptic data has a resolution of 16 bits/DOF. Assuming a 30% overhead, the air interface rate requirements would be randomly distributed in [0.21, 0.63, 1.25] Mb/s depending on the type of haptic device. The simulation results are averaged over 10^4 TTIs. In terms of traffic dynamics, we adopt a time-varying birth-death process, which determines the active number of traffic flows at any time instant, and consider two different scenarios. A base load scenario wherein the birth intensity (as a function of time) is given by $\lambda_m L_{\text{base}}/(1+t)$, and a high-load scenario in which the birth intensity is given by $\lambda_m L_{\text{high}}/(1+t)$. The death intensity (as a function of time) is given by t in both cases. Note that $\lambda_m; m \in \{A, B\}$ denotes the density of user distribution for the respective slice. The high-load scenario not only generates higher traffic load but also higher traffic dynamics.

TABLE I
Simulation Parameters

Parameter	Value
Cell radius	250 m
Overall resource pool ($\mathcal{X}$)	100 RBs
Path loss	$128.1 + 37.6\log_{10}(r_{\text{km}})$
Standard deviation of Shadowing	8 dB
OFDM symbol duration and cyclic prefix	14.3 μ s and 0.95 μ s
Subcarrier spacing	75 kHz
Learning rate (Θ)	0.8
Discount factor (φ)	0.5
Probability of exploration (ϵ)	0.4
Delay bound violation probability (Γ)	0.05
Spatial density of users $\{\lambda_A, \lambda_B\}$	$\{0.1, 0.1\}$ per m^2
Minimum rate for H2H users	1.5 Mb/s
Birth–death parameters $\{L_{\text{base}}, L_{\text{high}}\}$	$\{5, 20\}$

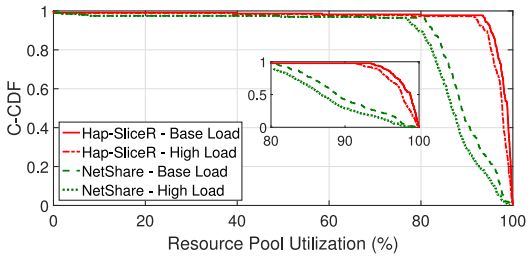

Fig. 3. C-CDF of overall radio resource pool utilization generated over 100 iterations.

First, we evaluate the performance of radio resource slicing strategy in Hap-SliceR. As a baseline for comparison, we adopt NetShare [13], which is a network-wide radio resource slicing solution. Fig. 3 plots the complementary cumulative distribution function (C-CDF) of overall resource pool utilization. As shown by the results, Hap-SliceR maintains a high resource utilization in both base-load and high-load scenarios. For the base load, the utilization is 93.5% in more than 95% of the times and 95% in more than 90% of the times. For the high load, the utilization is 91.5% in more than 95% of the times and 93.5% in more than 90% of the times. Moreover, Hap-SliceR outper-

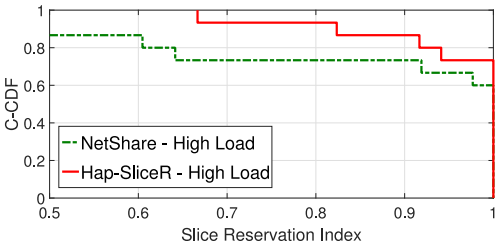

Fig. 4. C-CDF of SRI generated over 30 iterations.

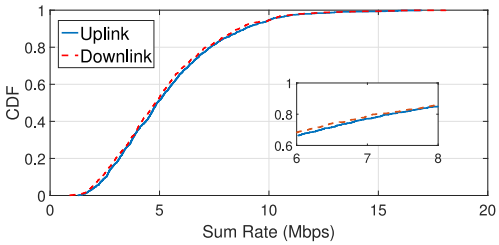

Fig. 5. CDF of uplink and downlink sum rate for the heuristic algorithm generated over 100 iterations ($\eta = 50$ kb/s, $\lambda = 2.5$ (per TTI)).

forms NetShare in both scenarios by performing up to 16% and 18% better in more than 80% of the times for base-load and high-load scenarios, respectively. This is due to the RL-based slicing strategy that inherently tackles slice dynamics. The results demonstrate that a fixed slicing period (as employed by NetShare) is not suitable in dynamic traffic conditions.

We define slice reservation index (SRI) metric to measure how well the slice allocation requirements of different slices are fulfilled. The SRI is given by $\mathcal{I}_R = \frac{\left(\sum_m \mathcal{W}_a^m / \mathcal{W}_O^m\right)^2}{\mathcal{M} \sum_m \left(\mathcal{W}_a^m / \mathcal{W}_O^m\right)^2}$, where $\mathcal{W}_a^m$ and $\mathcal{W}_O^m$, respectively, represent the slice allocation and slice resource demand, and $\mathcal{M}$ denotes the number of slices (vertical applications). We plot the C-CDF of SRI in Fig. 4 for the high-load scenario. As shown by the results, Hap-SliceR is effective in dynamically allocating resources to different slices in proportion to their requirements. Further, Hap-SliceR outperforms NetShare due to the fact that the RL-based slicing strategy learns the optimal size of each slice and, therefore, efficiently meets the requirements of different slices under dynamic environments. The performance gain is up to 30% in more than 80% of the times.

Next, we evaluate the performance of radio resource customization for haptic communications. Fig. 5 shows the CDF of sum rate in the uplink and the downlink for the heuristic algorithm. Based on the proposed algorithm, uplink and downlink schedulers perform joint RB allocation, due to which the achieved sum rate in the uplink is very close to the achieved sum rate in the downlink. To the best of our knowledge, no *joint* uplink/downlink scheduling algorithm for LTE-A networks, explicitly accounting for rate, delay bound, and specific allocation constraints of SC-FDMA and OFDMA, exists in the literature. Therefore, we adopt the classical round robin (RR) and best channel quality indicator (BCQI) resource allocation algorithms as baseline for performance comparison. As shown by the results in Fig. 6, the proposed algorithm outperform both RR and BCQI algorithms in terms of uplink/downlink rate difference by explicitly accounting for the coupled uplink and downlink rate

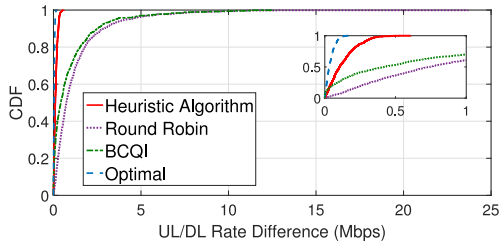

Fig. 6. CDF of uplink/downlink rate difference generated over 100 iterations.

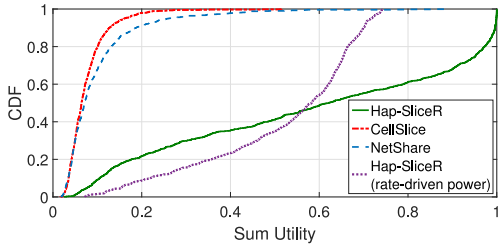

Fig. 7. CDF of sum utility for haptic communications generated over 100 iterations. Only base load scenario has been considered.

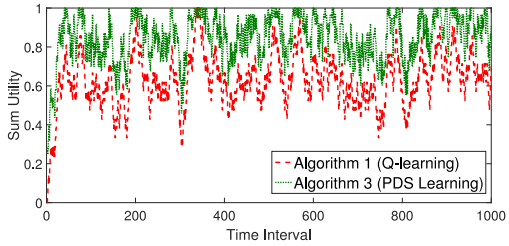

Fig. 8. Comparison of slicing strategies based on Algorithms 1 and 3 in terms of sum utility for haptic communications ($Y_1 = 0.9,\ Y_2 = 0.7,\ Y_3 = 0.5,\ Y_4 = 0.25$).

constraint. Note that the proposed algorithm performs very close to the optimal solution, which is obtained by using exhaustive search.

Next, we evaluate the performance of overall slicing framework in terms of sum utility for haptic communications. The sum utility captures both radio resource slicing and radio resource customization algorithms. As shown by the results in Fig. 7, Hap-SliceR outperforms NetShare in terms of sum utility, by performing nearly 80% better in more than 50% of the times. The performance gain is primarily due to specific customization of radio resources to meet the requirements of haptic communications. Besides, the RL-based slicing strategy dynamically adjusts the size of slice, allocated to haptic communications, as per the traffic demands. We also compare the performance to CellSlice [14] and notice a similar gain in performance. Further, we evaluate the performance of rate-driven power allocation policy as compared to optimal power allocation policy given by Lemma 1. With optimal power allocation, Hap-SliceR achieves much higher overall utility. The results also demonstrate that the optimal power allocation policy is crucial in guaranteeing the delay performance.

Finally, in Fig. 8, we compare the performance of slicing strategies based on Algorithms 1 and 3 in terms of sum utility for haptic communications. We note that in each time

interval, the PDSL-based slicing strategy achieves higher utility than the Q-learning-based slicing strategy. This is primarily due to the fact that the former selects a greedy action in each time step, whereas the later performs random action exploration.

VIII. Concluding Remarks

It is expected that network slicing would be indispensable for 5G network design. This paper proposed Hap-SliceR, which is a radio resource slicing framework, with two-fold contribution, for 5G networks with emerging haptic communications. Hap-SliceR first derives a network-wide radio resource slicing strategy, based on virtualization of radio resources, by adopting an RL approach. The slicing strategy allocates radio resources to different slices while accounting for slice dynamics as well as utility requirements of different vertical applications. The efficiency of the slicing strategy is further enhanced through a PDSL-based approach. Moreover, Hap-SliceR provides novel customization of radio resources, through a low-complexity heuristic resource allocation algorithm, for haptic communications. System-level performance evaluation demonstrates that both the radio resource slicing and the radio resource customization algorithms outperform state-of-the-art solutions. Hap-SliceR provides a viable solution for 5G networks that can be realized in any standard cloud computing infrastructure closer to the RAN edge.

References

[1] Ericsson, "5G systems: Enabling industry and society transformation," White Paper, Jan. 2015. [Online]. Available: http://www.ericsson.com/res/docs/whitepapers/what-is-a-5g-system.pdf

[2] G. Fettweis, "The tactile Internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[3] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh, "Realizing the tactile Internet: Haptic communications over next generation 5G cellular networks," *IEEE Wireless Commun.*, 2015. [Online]. Available: http://arxiv.org/abs/1510.02826

[4] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile Internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016.

[5] E. Steinbach *et al.*, "Haptic communications," *Proc. IEEE*, vol. 100, no. 4, pp. 937–956, Apr. 2012.

[6] P. Marsch, "5G radio access network design—a brief overview on the 5G-PPP Project METIS-II," in *Proc. Special Session Eur. Conf. Netw. Commun.*, Jun. 2015, pp. 676–680.

[7] N. Johansson, Y.-P. E. Wang, E. Eriksson, and M. Hessler, "Radio access for ultra-reliable and low-latency 5G communications," in *Proc. IEEE Int. Conf. Commun. Workshop*, Jun. 2015, pp. 1184–1189.

[8] O. N. C. Yilmaz, Y. -P. E.Wang, N. A. Johansson, N. Brahmi, S. A. Ashraf, and J. Sachs, "Analysis of ultra-reliable and low-latency 5G communication for a factory automation use case," in *Proc. IEEE Int. Conf. Commun. Workshop*, Jun. 2015, pp. 1190–1195.

[9] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, Firstquarter 2015.

[10] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A substrate for virtualizing wireless resources in cellular networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1333–1346, Oct. 2012.

[11] G. Bhanage, R. Daya, I. Seskar, and D. Raychaudhuri, "VNTS: A virtual network traffic shaper for air time fairness in 802.16e systems," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–6.

[12] L. Zhao, M. Li, Y Zaki, A. Timm-Giel, and C. Görg, "LTE virtualization: From theoretical gain to practical solution," in *Proc. 23rd Int. Teletraffic Congr.*, Sep. 2011, pp. 71–78.

[13] R. Mahindra, M. Khojastepour, H. Zhang, and S. Rangarajan, "Radio access network sharing in cellular networks," in *Proc. IEEE Int. Conf. Netw. Protocols*, Oct. 2013, pp. 1–10.

[14] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "CellSlice: Cellular wireless resource slicing for active RAN sharing," in *Proc. IEEE 5th Int. Conf. Commun. Syst. Netw.*, Jan. 2013, pp. 1–10.

[15] J. He and W. Song, "AppRAN: Application-oriented radio access network sharing in mobile networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 3788–3794.

[16] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT, 1998. [Online]. Available: https://webdocs.cs.ualberta.ca/ sutton/book/ebook/the-book.html

[17] C. Watkins and P. Dayan, "Technical note: Q-learning," *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, May 1992.

[18] N. Mastronarde and M. Schaar, "Fast reinforcement learning for energy-efficient wireless communication," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6262–6266, Dec. 2011.

[19] Y. Zaki, Liang Zhao, C. Goerg, and A. Timm-Giel, "LTE wireless virtualization and spectrum management," in *Proc. 3rd Joint IFIP Wireless Mobile Netw. Conf.*, Oct. 2010, pp. 1–6.

[20] M. Kalil, A. Shami, and Y. Ye, "Wireless resources virtualization in LTE systems," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2014, pp. 363–368.

[21] F. Fu and U. Kozat, "Stochastic game for wireless network virtualization," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 84–97, Feb. 2013.

[22] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *Proc. 2nd ACM Workshop Hot Topics Softw. Defined Netw.*, 2013, pp. 25–30.

[23] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, Inc., 1994.

[24] S. Hirche and M. Buss, "Human-oriented control for haptic teleoperation," *Proc. IEEE*, vol. 100, no. 3, pp. 623–647, Mar. 2012.

[25] H. Myung, J. Lim, and D. Goodman, "Single carrier FDMA for uplink wireless transmission," *IEEE Veh. Tech. Mag.*, vol. 1, no. 3, pp. 30–38, Sep. 2006.

[26] R. Giuliano and F. Mazzenga, "Exponential effective SINR approximations for OFDM/OFDMA-based cellular system planning," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4434–4439, Sep. 2009.

[27] J. Tang and X. Zhang, "Quality-of-service driven power and rate adaptation over wireless links," *IEEE Trans. Wireless Commun.*, vol. 6, no. 8, pp. 3058–3068, Aug. 2007.

[28] F. Calabrese, "Scheduling and link adaptation for uplink SC-FDMA systems," Ph.D. Dissertation, Aalborg Univ., Aalborg, Denmark, 2009.

[29] F. Kelly, S. Zachary, and I. Ziedins, *Stochastic Networks: Theory and Applications*, vol. 4. London, U.K.: Oxford Univ., 1996.

[30] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, Jul. 2003.

[31] J. Tang and X. Zhang, "Cross-layer-model based adaptive resource allocation for statistical QoS guarantees in mobile wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2318–2328, Jun. 2008.

[32] I. Wong, O. Oteri, and W. Mccoy, "Optimal resource allocation in uplink SC-FDMA systems," *IEEE Trans. Wireless Commun.*, vol. 8, no. 5, pp. 2161–2165, 2009.

[33] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*. New York: Academic, Oct. 2013.

[34] M. Mu, A. Mauthe, and F. Garcia, "A utility-based QoS model for emerging multimedia applications," in *Proc. 2nd Int. Conf. Next Gener. Mobile Appl., Serv., Technol.*, Sep. 2008, pp. 521–528.

Adnan Aijaz (M'14) received the B.E. degree in electrical (telecom) engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2008, and the Ph.D. degree in telecommunications engineering from King's College London (KCL), London, U.K., in 2014.

After a postdoctoral year at KCL, he moved to Toshiba Research Europe, Ltd., where he is currently a Senior Research Engineer. Prior to joining KCL, he was involved with the cellular industry for nearly 2.5 years in the areas of network performance management, optimization, and quality assurance. His primary research interests include wireless networking, with emphasis on 5G cellular networks, cognitive radio networks, sensor networks, 802.11-based WLANs, Internet-of-Things, and, more recently, the Tactile Internet. His publications in these areas have been featured in internationally renowned conferences and journals.