Reconfigurable and Traffic-Aware MAC Design for Virtualized Wireless Networks via Reinforcement Learning

Atoosa Dalili Shoaei*, Mahsa Derakhshani†, Tho Le-Ngoc*
*Department of Electrical & Computer Engineering, McGill University, Montreal, QC, Canada
† Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, UK
Email: atoosa.dalilishoaei@mail.mcgill.ca;m.derakhshani@lboro.ac.uk; tho.le-ngoc@mcgill.ca

***Abstract*—In this paper, we present a reconfigurable MAC scheme where the partition between contention-free and contention-based regimes in each frame is adaptive to the network status leveraging reinforcement learning. In particular, to support a virtualized wireless network consisting of multiple slices, each having heterogeneous and unsaturated devices, the proposed scheme aims to configure the partition for maximizing network throughput while maintaining the slice reservations. Applying complementary geometric programming (CGP) and monomial approximations, an iterative algorithm is developed to find the optimal solution. For a large number of devices, a scalable algorithm with lower computational complexity is also proposed. The partitioning algorithm requires the knowledge of the device traffic statistics. In the absence of such knowledge, we develop a learning algorithm employing Thompson sampling to acquire packet arrival probabilities of devices. Furthermore, we model the problem as a thresholding multi-armed bandit (TMAB) and propose a threshold-based reconfigurable MAC algorithm, which is proved to achieve the optimal regret bound.**

I. Introduction

A. Background and Motivation

Modern wireless networks supporting machine-to-machine (M2M) communications experience different traffic characteristics than the traditional networks dedicated to human-to-human communications [1], [2]. In such networks, since there are many M2M applications for data gathering and reporting purposes, the uplink traffic originated from devices to the access point (AP) is heavier. Furthermore, as devices may transmit packets sporadically, the assumption of saturated users is not always valid in these networks. For instance, in a smart metering application, the device only transmits a packet if a power outage happens or in a thermal monitoring application devices measure temperature periodically but only transmit a new packet if a variation has occurred between the last two measurements.

In such a dynamic environment, adopting a fixed medium access control (MAC) protocol cannot meet optimal characteristics along multiple dimensions. In order to improve the channel utilization, the traffic statistics information could be leveraged to efficiently select and configure a MAC protocol adapting to varying conditions. Furthermore, in practice there might not be any prior knowledge of traffic statistics or the statistical parameters might change over time. Thus, employing an appropriate learning algorithm is crucial to acquire the traffic statistics such that the expected total throughput is maximized. In other words, such learning algorithm targets at mitigating the regret defined as the difference between the throughput obtained by the solution for unknown traffic statistics and the achievable throughput with a priori knowledge.

In this paper, we present a learning-based reconfigurable MAC design, modeled as a thresholding multi-armed bandit. The proposed protocol switches from contention-free to contention-based access regime, adaptive to the updated traffic statistics. The logic behind this scheme is assigning devices with high probability of packet transmission to the contention-free regime and allowing the rest of devices to compete in the contention-based regime. In particular, we first propose the optimal scheduler that determines the partition between the two regimes. Then, we develop a *reinforcement learning* algorithm based on Thompson sampling (TS) for scenarios of unknown packet arrival probabilities. We analytically prove that the proposed Thompson sampling-based learning algorithm can efficiently balance the trade-off between exploration and exploitation and achieves the optimal regret bound.

Furthermore, in the emerging networks, heterogeneity is inevitable as devices might belong to different applications and report different events or measurements. This characteristic necessitates a wireless network infrastructure with ability to support multiple concurrent applications. To realize such an infrastructure, a virtualization framework needs to be used allowing multiple services and applications to access deployed network infrastructure and share radio resources. Such framework helps to reduce network deployment expenses and improve resource utilization [3], [4]. The objective of network virtualization is to partition the existent physical network resources in an efficient manner. This partitioning which is also known as resource slicing is a complex research problem in the wireless domain. This paper focuses on two important aspects of slicing: resource allocation and isolation. Slicing implies the allocation of the necessary resources to meet independent service requirements, but, for wireless resources due to the particularities of the wireless medium, assuring slice isolation becomes challenging, even more when quality of service (QoS) constraints come into play [5]. More specifically, in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

this paper, we consider a virtualized wireless network and the proposed MAC protocol aims to preserve isolation taking into account slice reservations [1].

B. Scope of the Paper and Contributions

Generally, in MAC designs, radio resources are divided in time, frequency or code domains. In the contention-free or deterministic assignment (DA) schemes, time/frequency slots or codes are allocated by the AP to the devices, while in the contention-based or random access (RA) schemes, these resources are chosen randomly by each device. In the following MAC design discussions, we consider a time-division multiple access structure for its simplicity in presentation, although the proposed scheme and its results can be easily extended for applications to frequency-division multiple access and code-division multiple access structures.

The main contributions of this paper are four-fold. First, aiming to improve the network efficiency, we design a reconfigurable MAC with optimal contention-free and contention-based partition based on the device packet arrival statistics. To this end, we formulate an optimization problem which is inherently non-convex and suffers from high computational complexity. To tackle this issue, we first show that the problem belongs to the class of complementary geometric programming (CGP). Then, we propose an efficient and tractable iterative approach to solve it. At each iteration, via applying transformation techniques and arithmetic geometric mean approximation (AGMA), we transform the CGP-based formulation into the geometric programming (GP), which can be solved with the softwares such as CVX efficiently [6] [2].

Second, we propose a scalable solution for a large number of devices as in M2M networks with considerably less computational complexity as compared to the proposed CGP-based scheduling. In particular, to overcome the computational burden caused by a large number of devices, the optimization problem is transformed using approximations for RA throughput and airtime. Subsequently, we propose an efficient iterative algorithm to solve the approximated optimization problem, where each iteration is decomposed into two sub-problems: one belongs to the linear-programming category, and the other is of the difference of convex (DC)-programming type.

Third, considering the scenario of unknown device arrival statistics, we develop a *Thompson sampling*-based algorithm to learn packet arrival probabilities efficiently. Furthermore, we propose a simple algorithm in which the DA and RA partition is determined by a threshold. In this algorithm, devices having the expected throughput higher than a certain threshold are considered for DA, while the rest transmits in the RA regime. In particular, we show that the problem to select devices for DA regime can be perfectly matched to a thresholding multi-armed bandit. Thresholding multi-armed bandit (TMAB) is a specific type of combinatorial multi-armed bandits (CMAB), where the learner aims to find the set of arms with the mean rewards exceeding a certain threshold, rather than picking a constant number of arms with the highest mean rewards as in CMABs. In the proposed MAC design, each arm corresponds to a device, and scheduling a device for DA in each frame is equivalent to playing an arm. The goal is to find a device-selection policy that maximizes the cumulative throughput over finite frames. Thompson sampling has been shown to perform well for CMABs [8]. However, its performance for TMABs has not been investigated. In this paper, we show that Thompson sampling is also a proper and efficient algorithm for TMABs.

Finally, to show the efficacy of Thompson sampling algorithm for thresholding multi-armed bandits, we perform the regret analysis. This metric shows the total expected throughput difference between the optimal policy and Thompson sampling algorithm. We prove that Thompson sampling achieves the optimal regret bound for the stochastic TMABs.

C. Related Works

In the literature, there are several works addressing particular requirements of future wireless networks in the MAC layer, e.g., [9]–[16]. In [9], the authors proposed a MAC protocol for M2M networks targeting the scenarios consisting of both periodic and non-periodic traffic. In [10], a hybrid MAC is designed for heterogeneous and massive M2M networks, accounting for traffic statistics. However, the works in [9], [10] assumed known traffic parameters for devices.

The work in [11] focuses on the design of grouping-based MAC protocols for an event-driven scenario, in which a smart meter is attached to each electric vehicle and is used to report the charging parameters to the network. However, the results are derived for a single application in which all devices have the same traffic parameters. Similarly, [12] considers an M2M network where all devices are reporting the same event. In this work, the authors proposed a pure random-based channel access for devices to report the event, in which the optimal transmission probability is dependent on the number of active devices. It is assumed that the number of active devices at each time interval is unknown by the base station (BS). Thus, the BS applies a drift analysis to estimate the number of active devices. The work in [14], also considers a scenario of a single application with an on-off traffic model for the devices. In this work, to reduce the congestion in the large scale network, the access clear bearing (ACB) method is proposed. Similar to [11], [12], considering a network running a single application might be unrealistic for the future networks as different applications may share the same infrastructure.

In [15], an algorithm is proposed to estimate the number of uplink devices to determine the uplink length of each frame. However, the derivation is based on the number of devices while in these networks, devices might be unsaturated. In our previous work [16], a learning algorithm is developed for scenarios of unknown traffic statistics, however, for the scheduling purposes, a heuristic algorithm is proposed which similar to [15] may not lead to the optimal solution.

[1] Note that wireless network virtualization entails other technical challenges to address; however, such implementation challenges are beyond the scope of this paper.

[2] This part of the work has been presented in [7].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

The work in [13], also assumed the scenario of non-periodic traffic with unknown characteristics. In this work, devices with packets for transmission send an access request which leads to the extra overhead for the network. To avoid this overhead, in this paper, we allocate resources to the devices in a proactive manner. This means that instead of allocating separated resources for the access request gathering, the active devices are predicted using the device traffic statistics.

For scenarios of unknown traffic statistics, the efficient access design can be formulated as a MAB problem. In the context of MAB problem, throughout the rounds, there is always a trade-off between exploration and exploitation. On one hand, the learner wants to exploit the past observations by selecting seemingly good arms. On the other hand, there is always a possibility that the other arms have been underestimated, which gives the motivation to pick unexplored arms in order to gather more information. To deal with such trade-off, various approaches have been proposed such as upper confidence bound (UCB), in which a deterministic index is assigned to each arm. This index represents the sample mean reward of the arm (exploitation term) plus an exploration term, which gives a higher chance to underexplored arms. For UCB-type algorithms, strong theoretical guarantees on the regret can be proved. For example, in [17], the regret bound has been derived for the classical UCB algorithm. For CMABs, the authors in [18] perform the regret analysis for linear rewards, while nonlinear reward bandit has been studied in [19].

The index-based policies such as UCB are popular for CMABs, where L arms with largest indices would be selected in each round. However, TMABs are sensitive to the exact value of estimated mean reward associated to each arm (not relative to others as in CMABs) since they would be compared with a threshold. Thus, in index-based policies where the exploration term is added to the sample mean reward, the index may become far from the real mean reward.

For thresholding multi-armed bandits, Bayesian inference can be a better approach where the unknown parameter (i.e., mean reward) is drawn from a prior probability distribution, that would be updated at each round after the distribution is sampled. This approach allows exploration by randomly sampling from a distribution, where the observed value may fluctuate from the true value. However, the more frequently distribution is sampled and updated, the more certainly the observed value approaches the true value of unknown parameter. One of the old heuristic algorithm based on Bayesian ideas is Thompson sampling. For a long time, this algorithm was not of interest due to the lack of theoretical analysis. However, it has received significant attention after some recent studies [8], [20]. It has been revealed that TS has an excellent performance with the optimal regret bound and also could be applicable to a wider class of problems [21], [22]. Moreover, [23] has derived the regret analysis for a case where TS is used in CMABs.

The TMAB setting has been studied in [24], where a pure exploration algorithm is proposed. In this work, it is assumed that the threshold is known and the goal of the learner is to correctly identify the arms whose means are over or under the threshold up to a certain precision. This algorithm, due to its pure exploration-based nature, cannot be applied to a situation where the aim of learner is to maximize the cumulative reward.

D. Structure

We first introduce the system model under consideration in Section II. Section III presents the problem formulation. Subsequently, an iterative CGP-based algorithm along with a scalable reconfigurable MAC scheduling are proposed in Section IV. Section V describes the Thompson sampling-based algorithms for scenarios of unknown packet arrival probabilities. Section VI presents the regret analysis for the proposed Thompson sampling-based approach for thresholding MAC. Section VII provides simulation results. Finally, Section VIII draws the conclusion.

II. System Model

A. Network Model and Frame Structure

We consider a single AP serving N_{d} devices, where all communications are done through the AP. Each device is exclusively subscribed to one service provider (SP) (also referred to as slice) with a specific airtime reservation. There are $\mathcal{S} = \{1, \cdots, S\}$ different slices and $\mathcal{D}_s$ denotes the set of subscribed devices to slice s, where $|\mathcal{D}_s| = N_s$ is the number of devices at slice s.

In the wireless network, slicing can be built on physical radio resources (e.g., transmission point, spectrum, time) or on logical resources abstracted from physical radio resources. Here, we consider airtime as a resource to be virtualized. In the context of wireless virtualization, controlling airtime usage of slices is essential because it can guarantee isolation among slices. Other metrics such as throughput may fail in providing isolation, because the slice with low traffic rate and high outage probabilities needs more airtime to meet its reservation. In other words, it uses more resources to reach the same throughput as other slices, which contradicts the concept of isolation.

Time is divided into fixed-length frames indexed by t. As shown in Figure 1, each frame begins with a beacon issued by the AP followed by the DA regime with the duration of $T_{\mathrm{da}}(t)$ $(\leq T_{\max})$ for scheduled devices, and the RA regime of the length $T_f - T_{\mathrm{da}}(t)$. During the RA regime, carrier sense multiple access (CSMA) protocol runs, where each time-slot with the duration of T_s is divided into backoff units. Such hybrid techniques is inspired by PCF for 802.11, although it is enhanced by proposing dynamic lengths for DA and RA regimes.

Regarding the required airtime per slice, it is assumed that each slice s can reserve time for r_s time-slots per frame. For devices sensitive to the delay, an exclusive time-share in the DA regime could be further reserved to meet their delay requirements.

We consider Bernoulli process to model the packet arrivals of each device [25] since the traffic generated by devices is mainly sporadic with a negligible probability that more than one packet arrive in one frame. More specifically, we assume a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

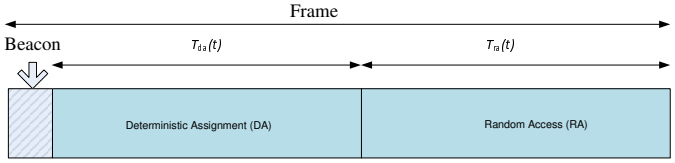

Fig. 1: Frame structure of the proposed reconfigurable MAC for a virtualized wireless network

new packet is generated at device d_s with a probability of a_{d_s} at each frame and it is added to the queue of the device if the current length of the queue is smaller than $Q_{\max}$. Otherwise, the packet is discarded. Furthermore, we first assume that the AP is aware of packet arrival probabilities of devices and it keeps a vector denoted by $\boldsymbol{V}(t) = [v_{d_s}(t)]_{\forall d_s}$, where v_{d_s} denotes the last time that the AP has received a packet from the device d_s. Moreover, each time the device sends a packet, it piggybacks an extra bit (denoted by $q_{d_s}(t)$) telling whether its queue is empty ($q_{d_s}(t) = 0$) or non-empty ($q_{d_s}(t) = 1$, i.e., it has packets backlogged in the queue to transmit). Therefore, at each frame t, the AP updates $\theta_{d_s}(t)$ that indicates the probability of the device d_s having a non-empty queue at t as

$$\theta_{d_s}(t) = \begin{cases} 1 - (1 - a_{d_s})^{t - v_{d_s}(t)}, & \text{if } q_{d_s}(v_{d_s}(t)) = 0 \\ 1 & \text{if } q_{d_s}(v_{d_s}(t)) = 1. \end{cases} \quad (1)$$

Regarding the wireless channel model, we consider path loss and small scale fading. The instantaneous received SNR of device d_s at AP is equal to $GP_t h_{d_s} l_{d_s}^{-\zeta}/\sigma^2$, where P_{t} is the transmission power, σ^2 is the noise power, h_{d_s} is the small-scale Rayleigh fading component of the link from the device d_s to the AP, l_{d_s} is the link distance between device d_s and the AP, ζ is the path-loss exponent, and G is a constant dependent on the frequency and transmitter/receiver antenna gain. For simplicity, without loss of generality, we normalize G to 1 (i.e., $G = 1$) in the following discussions. If the received signal level falls below the receiver threshold, the receiver cannot successfully decode the signal denoting an *outage* event. Thus, the system performance is influenced by the outage probability defined as the probability that the received SNR is less than the receiver threshold υ,

$$\psi_{d_s} = \Pr\left(\frac{P_{\text{t}} h_{d_s} l_{d_s}^{-\zeta}}{\sigma^2} \leq \upsilon\right) = 1 - e^{-\frac{\sigma^2 l_{d_s}^{\zeta} \upsilon}{P_{\text{t}}}}. \quad (2)$$

B. Device Operation

Before the frame t starts, the AP decides on time-slot allocation for the DA regime and notifies the schedule to devices via the beacon. Devices with no allocated time-slot would attempt to transmit in the RA regime if they have a packet, using the p-persistent CSMA protocol as follows. In the RA regime, a device with a non-empty queue performs the channel sensing. If the channel is sensed to be idle, the device will transmit the packet with probability p at the beginning of the next time-slot or defers with probability $(1 - p)$. If a device is unsuccessful in transmission of a packet either in DA or RA regime, it is not allowed to retransmit the packet in that current frame. The reason is that, the retransmissions may affect the airtime of other slices.

C. An Analytical Model for p-persistent CSMA

Here, we model the throughput of p-persistent CSMA protocol in an unsaturated mode. Let P_{idle} be the probability

TABLE I: List of Key Notations

Notations	Description
N_{d}	Number of devices
S	Number of slices
N_s	Number of devices at slice s
$\mathcal{D}_{\text{s}}$	Set of subscribed devices to slice s
t	Index of time-frame
$T_{\text{da}}(t)$	Duration of DA regime at frame t
$T_{\max}$	Maximum length of DA
T_f	Frame length
T_s	Duration of a time slot
$T_{\text{ra}}(t)$	Duration of RA regime at frame t
r_s	Slice s time-slot reservation
d_s	Device belonging to slice s
q_{d_s}	Queue length of device d_s
$Q_{\max}$	Maximum length of queue
a_{d_s}	Packet arrival probability of device d_s
v_{d_s}	Last time AP received a packet from device d_s
$\theta_{d_s}(t)$	Probability that the device d_s has a non-empty queue at t
ψ_{d_s}	Outage probability of device d_s
p_{d_s}	Probability that device d_s transmits a packet over idle channel in the RA regime
ρ_{d_s}	Normalized throughput of device d_s
$\tau_{d_s}(t)$	Total access airtime of the device d_s
$x_{d_s}(t)$	Variable indicating whether a time-slot is allocated to the device d_s at t
M	Number of arms in the MAB
L	Number of arms to be chosen at each round of the MAB problem
T	Number of rounds/time steps in MAB problem
μ_m	Reward expectation of arm m
α, β	Shape parameters of the beta distribution
Δ_m	Regret caused by playing suboptimal arm m
ϑ_{d_s}	Expected throughput of device d_s, when it is selected for DA
γ	Threshold in thresholding reconfigurable MAC
ϕ_m	Thompson sampling index of arm m
R	Regret of TS-TMAB algorithm

TABLE II: List of Key Abbreviations

Abbreviation	Full form of abbreviation
M2M	machine-to-machine
MAC	medium access control
TS	Thompson sampling
DA	deterministic assignment
AP	access point
RA	random access
CGP	complementary geometric programming
DC	difference of convex
TMAB	thresholding multi-armed bandit
CMAB	combinatorial multi-armed bandit

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

that channel is idle in a backoff unit. This probability is calculated as

$$P_{\text{idle}} = \prod_{s\in\mathcal{S}} \prod_{d_s\in\mathcal{D}_s} (1-\theta_{d_s} p_{d_s}), \tag{3}$$

where $\theta_{d_s} p_{d_s}$ represents the transmission probability of device d_s. The key approximation is that we assume that the number of active devices is constant over a frame, however in the proposed scheme, the device transmits at most one packet during the RA regime.

A transmitted packet will be received successfully, if exactly one device transmits on the channel. For device d_s, the probability of *successful* transmission denoted by $P_{\text{succ}}^{d_s}$ is

$$P_{\text{succ}}^{d_s} = \theta_{d_s} p_{d_s} (1-\psi_{d_s}) \prod_{s\in\mathcal{S}} \prod_{d'_s\in\mathcal{D}_s, d'_s\neq d_s} (1-\theta_{d'_s} p_{d'_s}). \tag{4}$$

As introduced in [26], the normalized throughput of device d_s (denoted by ρ_{d_s}) is defined as the fraction of time that the channel is used for its successful transmission,

$$\rho_{d_s} = \frac{P_{\text{succ}}^{d_s} T_s}{P_{\text{idle}}\varrho + (1-P_{\text{idle}})T_s}, \tag{5}$$

where ϱ is the duration of a backoff unit and T_s is the duration of a successful transmission, which includes the data transmission for a fixed time, inter-frame spaces, and signaling overheads. Since signaling and inter-frame spaces are relatively small (in the order of μs) compared with the data transmission length (in the order of ms), we approximately assume that both collided and successful transmissions are of the same size (i.e., T_s). Consequently, the denominator in (5) represents the expected length of a general time-slot.

By introducing a new variable, i.e.,

$$y_{d_s} = \frac{\theta_{d_s} p_{d_s}}{1-\theta_{d_s} p_{d_s}}, \tag{6}$$

we can simplify (5). To this end, first, we rewrite P_{idle} and $P_{\text{succ}}^{d_s}$ in terms of y_{d_s} as

$$P_{\text{idle}} = \frac{1}{\prod_{s\in\mathcal{S}} \prod_{d_s\in\mathcal{D}_s} (1+y_{d_s})}, \tag{7}$$

$$P_{\text{succ}}^{d_s} = \frac{y_{d_s}(1-\psi_{d_s})}{\prod_{s\in\mathcal{S}} \prod_{d_s\in\mathcal{D}_s} (1+y_{d_s})} = y_{d_s}(1-\psi_{d_s}) P_{\text{idle}}. \tag{8}$$

Then, we obtain ρ_{d_s} in terms of y_{d_s} as

$$\rho_{d_s} = \frac{y_{d_s}}{\prod_{s\in\mathcal{S}} \prod_{d_s\in\mathcal{D}_s} (1+y_{d_s}) - t'}, \tag{9}$$

where $t' = \frac{T_s-\varrho}{T_s}$. Furthermore, in the context of virtualized wireless networks, *total access airtime* is considered as another performance metric to measure and preserve isolation. For device d_s, the total access airtime during the RA regime is defined as

$$\tau_{d_s} = \frac{(1-P_{\text{idle}})T_s}{P_{\text{idle}}\varrho + (1-P_{\text{idle}})T_s}, \tag{10}$$

which can also be represented in terms of y_{d_s} as

$$\tau_{d_s} = \frac{y_{d_s} \prod_{s\in\mathcal{S}} \prod_{d'_s\in\mathcal{D}_s, d'_s\neq d_s} (1+y_{d'_s})}{\prod_{s\in\mathcal{S}} \prod_{d_s\in\mathcal{D}_s} (1+y_{d_s}) - t'}. \tag{11}$$

III. Problem Formulation

To enable coexistence of different slices in a shared wireless network, an effective slicing of resources is required with two conflicting objects: maximizing the efficiency and providing isolation between slices. In an unsaturated network, the achievable throughput can be increased by switching between contention-free and contention-based schemes. As a contention-free regime is more efficient for devices with high probabilities of packet transmission, while a contention-based regime has a better performance when devices transmit less frequently. Also, compared to the pure contention-based scheme, splitting devices into two groups, contention-free and contention-based, leads to a lower number of devices in the contention-based period and consequently a less number of collisions, and higher utilization. Note that using the pure contention-free scheme leads to system underutilization as time-slots would be assigned to devices with low traffic demands.

In the proposed reconfigurable MAC, the scheduling algorithm determines the partition between the DA and RA regimes. More specifically, it determines which devices should transmit in the DA, based on the traffic demand of each device and slice reservations. Furthermore, it derives the parameter p for the rest of devices which compete with each other the in the remaining time of the frame using p-persistent CSMA.

Here, we present the formulation for throughput maximization of this reconfigurable MAC scheme, assuming that statistical traffic parameters of devices (i.e., a_{d_s}) are known by the AP.

The expected throughput associated with the DA regime is

$$S_{\text{da}}(t) = \sum\nolimits_{s\in\mathcal{S}} \sum\nolimits_{d_s\in\mathcal{D}_s} \theta_{d_s}(t)(1-\psi_{d_s}) x_{d_s}(t), \tag{12}$$

where $x_{d_s}(t)$ is a binary variable indicating whether a time-slot is allocated to the device d_s in the frame t (i.e., $x_{d_s}(t)=1$) or not (i.e., $x_{d_s}(t)=0$) and $\boldsymbol{X}(t) = [x_{d_s}(t)]_{\forall d_s}$. Moreover, the RA-regime throughput can be computed as

$$S_{\text{ra}}(t) = T_{\text{ra}}(t) \sum\nolimits_{s\in\mathcal{S}} \sum\nolimits_{d_s\in\mathcal{D}_s} \rho_{d_s}, \tag{13}$$

where $T_{\text{ra}}(t)$ denotes the duration of RA regime in the frame t. Taking into account the number of scheduled devices for DA regime, $T_{\text{ra}}(t)$ can be represented as

$$T_{\text{ra}}(t) = T_f - T_s \sum\nolimits_{s\in\mathcal{S}} \sum\nolimits_{d_s\in\mathcal{D}_k} x_{d_s}(t). \tag{14}$$

Furthermore, the instantaneous expected total access airtime for slice s can be obtained as

$$\tau_s(t) = \sum\nolimits_{d_s\in\mathcal{D}_s} \left[T_s x_{d_s}(t) + T_{\text{ra}}(t)\tau_{d_s}(t)\right], \tag{15}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

where the first term represents the time assigned to devices belonging to the slice s during the DA regime and the second term indicates the average total access time of devices of the slice s in the RA regime. Finally, at the frame t, the AP should solve the following optimization problem in order to obtain $\boldsymbol{X}$ and $\boldsymbol{Y}$.

$$\max_{\boldsymbol{X},\boldsymbol{Y}} \quad S_{\text{da}}(t) + S_{\text{ra}}(t), \quad \text{subject to,} \tag{16}$$

$$\text{C16.1: } \tau_s(t) \geq r_s, \qquad \forall s \in \mathcal{S}$$

$$\text{C16.2: } x_{d_s} p_{d_s} = 0, \qquad \forall s \in \mathcal{S}, \forall d_s \in \mathcal{D}_s$$

$$\text{C16.3: } T_s \sum\nolimits_{s\in\mathcal{S}} \sum\nolimits_{d_s\in\mathcal{D}_s} x_{d_s} \leq T_{\max},$$

$$\text{C16.4: } p_{d_s} \leq 1, \qquad \forall s \in \mathcal{S}, \forall d_s \in \mathcal{D}_s,$$

$$\text{C16.5: } x_{d_s} \in \{0,1\}, \qquad \forall s \in \mathcal{S}, \forall d_s \in \mathcal{D}_s.$$

In this optimization problem, the objective function represents the total network throughput in both DA and RA regimes for the frame t. The constraint C16.1 is to guarantee that the reservation of each slice is met. Moreover, C16.2 ensures that the device d_s is only selected for either DA or RA. C16.3 limits the number of devices that could transmit in the DA regime. Finally, C16.4 indicates that p_{d_s} should be less than or equal to one and C16.5 states that x_{d_s} is a binary variable. In the rest of the paper, t is omitted in all equations for the sake of simplicity. Substituting (12), (13), (14), and (15) in (16), the optimization problem can be written as

$$\max_{\boldsymbol{X},\boldsymbol{Y}} \sum_{s\in\mathcal{S}} \sum_{d_s\in\mathcal{D}_s} (1-\psi_{d_s}) \times \tag{17}$$

$$\left[\theta_{d_s} x_{d_s} + \frac{y_{d_s}(T_f - T_s \sum_{s\in\mathcal{S}} \sum_{d_s\in\mathcal{D}_s} x_{d_s})}{\prod_{s\in\mathcal{S}} \prod_{d_s\in\mathcal{D}_s} (1+y_{d_s}) - t'}\right], \quad \text{subject to,}$$

$$\text{C17.1: } \sum_{d_s\in\mathcal{D}_s} \Big[T_s x_{d_s} + \frac{y_{d_s} \prod_{s\in\mathcal{S}} \prod_{d'_s\in\mathcal{D}_s, \neq d_s} (1+y_{d'_s})}{\prod_{s\in\mathcal{S}} \prod_{d_s\in\mathcal{D}_s} (1+y_{d_s}) - t'} \times$$

$$(T_f - T_s \sum\nolimits_{s\in\mathcal{S}} \sum\nolimits_{d_s\in\mathcal{D}_s} x_{d_s})\Big] \geq r_s, \ \forall s \in \mathcal{S}$$

$$\text{C17.2: } x_{d_s} y_{d_s} = 0, \qquad \forall s \in \mathcal{S}, \forall d_s \in \mathcal{D}_s$$

$$\text{C17.3: } T_s \sum\nolimits_{s\in\mathcal{S}} \sum\nolimits_{d_s\in\mathcal{D}_s} x_{d_s} \leq T_{\max},$$

$$\text{C17.4: } \frac{y_{d_s}}{\theta_{d_s}(1+y_{d_s})} \leq 1, \qquad \forall s \in \mathcal{S}, \forall d_s \in \mathcal{D}_s,$$

$$\text{C17.5: } x_{d_s} \in \{0,1\}, \qquad \forall s \in \mathcal{S}, \forall d_s \in \mathcal{D}_s.$$

It is clear that the optimization problem in (17) has a non-convex objective function due to the couplings in the RA throughput and involves non-linear constraints with the combination of continuous and binary variables, i.e., y_{d_s} ($y_{d_s} \geq 0$) and x_{d_s} ($x_{d_s} \in \{0,1\}$). Consequently, (16) is a non-convex mixed-integer, NP-hard optimization problem. Therefore, an efficient algorithm with reasonable computational complexity is needed to solve this scheduling problem.

IV. Reconfigurable MAC Scheduling with Traffic Knowledge

To solve the scheduling problem (17), we first formulate it as a CGP. Then, for scenarios of large number of devices, the optimization problem is transformed by using approximations and solved by using a two-step decomposition method. We discuss these algorithms in the following.

A. Reconfigurable MAC Scheduling via CGP

The formulated problem in (17) is non-convex and thus intractable to solve. To reduce the complexity, we relax the binary variable x_{d_s} into a continuous one in the interval of [0,1]. The induced problem potentially looks like a CGP problem. Based on successive convex approximation, a computationally tractable iterative algorithm can be developed to solve a CGP problem. More specifically, a CGP problem can be transformed to a GP by monomial approximations and then a series of GPs can be solved iteratively to obtain the solution.

Here, we describe how to transform the problem (17) into a CGP form and then solve it iteratively by applying monomial approximations as discussed in Appendix A. First, we can maximize the objective function, by minimizing its negative. However, in CGP the objective function should be positive, and this can be done by adding a sufficiently large constant H. Moreover, we introduce three auxiliary variables $z_{d_s} = 1 + y_{d_s}$, $b = \prod_{s\in\mathcal{S}} \prod_{d_s\in\mathcal{D}_s} z_{d_s} - t'$ and $T_{\text{ra}} = T_f - T_s \sum_{s\in\mathcal{S}} \sum_{d_s\in\mathcal{D}_s} x_{d_s}$. By replacing these auxiliary variables with their corresponding terms and applying the aforementioned changes in the objective function, the problem becomes

$$\min_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z},T_{\text{ra}},b} H - \sum_{s\in\mathcal{S}} \sum_{d_s\in\mathcal{D}_s} (1-\psi_{d_s}) \big[\theta_{d_s} x_{d_s} + T_{\text{ra}} y_{d_s} b^{-1}\big]$$

subject to: (18)

$$\text{C18.1: } \sum\nolimits_{d_s\in\mathcal{D}_s} \big[T_s x_{d_s} +$$

$$T_{\text{ra}} y_{d_s} b^{-1} \prod\nolimits_{\forall s\in\mathcal{S}} \prod\nolimits_{d'_s\in\mathcal{D}_s, \neq d_s} z_{d'_s}\big] \geq r_s, \quad \forall s \in \mathcal{S}$$

$$\text{C18.2: } x_{d_s} y_{d_s} = 0, \ \forall s \in \mathcal{S}, \ \forall d_s \in \mathcal{D}_s$$

$$\text{C18.3: } T_s \sum\nolimits_{s\in\mathcal{S}} \sum\nolimits_{d_s\in\mathcal{D}_k} x_{d_s} \leq T_{\max}$$

$$\text{C18.4: } z_{d_s} = 1 + y_{d_s}, \ \forall s \in \mathcal{S}, \ \forall d_s \in \mathcal{D}_s$$

$$\text{C18.5: } b = \prod\nolimits_{s\in\mathcal{S}} \prod\nolimits_{d_s\in\mathcal{D}_k} z_{d_s} - t'$$

$$\text{C18.6: } T_{\text{ra}} = T_f - T_s \sum\nolimits_{s\in\mathcal{K}} \sum\nolimits_{d_s\in\mathcal{D}_s} x_{d_s},$$

$$\text{C18.7: } \frac{y_{d_s}}{\theta_{d_s} z_{d_s}} \leq 1, \qquad \forall s \in \mathcal{S}, \forall d_s \in \mathcal{D}_s,$$

$$\text{C18.8: } x_{d_s} \leq 1, \qquad \forall s \in \mathcal{S}, \forall d_s \in \mathcal{D}_s.$$

In (18), the objective function is not posynomial because of the negative multiplicative in the second term. This can be handled by introducing and minimizing a new auxiliary variable x_0 in addition to guaranteeing the following constraint C19.7. The resulting optimization problem is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

$$\min_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z},T_{\rm ra},b,x_0} x_0, \text{ subject to:} \quad (19)$$

$$\text{C19.1:} \frac{r_s}{\sum_{d_s\in\mathcal{D}_s}\left[T_s x_{d_s}+T_{\rm ra}y_{d_s}b^{-1}\prod_{\forall s\in\mathcal{S}}\prod_{d'_s\in\mathcal{D}_s,\neq d_s} z_{d'_s}\right]}\le 1, \quad \forall s\in\mathcal{S}$$

$$\text{C19.2: } \frac{1}{1+x_{d_s}y_{d_s}}=1 \ \ ,\forall s\in\mathcal{S}, d_s\in\mathcal{D}_s$$

$$\text{C19.3: } T_sT_{\max}^{-1}\sum\nolimits_{s\in\mathcal{S}}\sum\nolimits_{d_s\in\mathcal{D}_s} x_{d_s}\le 1$$

$$\text{C19.4: } \frac{z_{d_s}}{1+y_{d_s}}=1 \ \ ,\forall s\in\mathcal{S}, d_s\in\mathcal{D}_s$$

$$\text{C19.5: } \frac{\prod_{s\in\mathcal{S}}\prod_{d_s\in\mathcal{D}_s} z_{d_s}}{t'+d}=1 \ \ ,$$

$$\text{C19.6: } \frac{T_f}{T_{\rm ra}+T_s\sum_{s\in\mathcal{S}}\sum_{d_s\in\mathcal{D}_s} x_{d_s}}=1 \ \ ,$$

$$\text{C19.7: } \frac{H}{x_0+\sum_{s\in\mathcal{S}}\sum_{d_s\in\mathcal{D}_s}(1-\psi_{d_s})[\theta_{d_s}x_{d_s}+T_{\rm ra}y_{d_s}b^{-1}]}\le 1,$$

$$\text{C19.8: } \frac{y_{d_s}}{\theta_{d_s}z_{d_s}}\le 1, \qquad \forall s\in\mathcal{S},\forall d_s\in\mathcal{D}_s,$$

$$\text{C19.9: } x_{d_s}\le 1, \qquad \forall s\in\mathcal{S},\forall d_s\in\mathcal{D}_s.$$

In this optimization problem, all inequality constraints are in the form of a ratio between two posynomials and equality constraints are in the form of a ratio between a monomial and a posynomial, as in a CGP problem. As discussed in Appendix A, the algorithm to deal with CGP consists of monomial approximations and solving a sequence of resulting GP problems until convergence happens. The proposed algorithm to solve (19) is described in Algorithm 1.

At each iteration, the resulting GP problem needs to be solved by transforming it to a convex optimization problem. It has been shown that the worst-case computational complexity of this approach is $\mathcal{O}(n_t n_v^3)$, where n_v denotes the number of variables and n_t is the total number of terms in all the monomials and posynomials in the optimization problem [27]. Since n_t and n_v grow linearly with $N_{\rm d}$ in (19), each iteration of reconfigurable MAC scheduling problem has computational complexity of $\mathcal{O}({N_{\rm d}}^4)$.

B. Scalable Reconfigurable MAC for Dense Networks

Although the CGP-based algorithm has polynomial complexity, for a massive number of devices an algorithm with less computational complexity is needed. To this end, assuming $N_{\rm d} >> 1$, we first approximate the RA throughput as

$$\rho_{d_s}=\frac{y_{d_s}(1-\psi_{d_s})}{\prod_{s\in\mathcal{S}}\prod_{d_s\in\mathcal{D}_s}(1+y_{d_s})-t'}\approx\frac{y_{d_s}(1-\psi_{d_s})}{\prod_{s\in\mathcal{S}}\prod_{d_s\in\mathcal{D}_s}(1+y_{d_s})}$$
$$\approx y_{d_s}(1-\vartheta_{d_s})(1-\sum\nolimits_{s\in\mathcal{S}}\sum\nolimits_{d_s\in\mathcal{D}_s} y_{d_s}). \quad (20)$$

Moreover, the RA airtime can be approximated as

Algorithm 1 Reconfigurable MAC scheduling via CGP

Input: $\boldsymbol{\Theta}$,$\boldsymbol{\Psi}$, $T_{\max}$, ς, r_s $\forall s\in\mathcal{S}$
Initialization: Set initial value to ($\boldsymbol{X}$,$\boldsymbol{Y}$,$\boldsymbol{Z}$,$T_{\rm ra}$,b,x_0)
repeat
Step 1: Monomial approximation
1) Compute ζ^p for denominators of C19.1 and C19.7
2) Use (33) to approximate the posynomials
3) Compute ζ^q for denominators of C19.2, C19.4, C19.5, C19.6
4) Use (34) to approximate the posynomials
Step 2: Solve the transformed GP problem
1) replace denominators of (19) with obtained monomial terms in **Step 1**
2) $(\boldsymbol{X'},\boldsymbol{Y'},\boldsymbol{Z'},T'_{\rm ra},b',x'_0)\leftarrow$ solve (19)
until $|x_0-x'_0|<\varsigma$
$p_{d_s}\leftarrow\frac{y_{d_s}}{\theta_{d_s}(1+y_{d_s})}$
Set $x_{d_s}=1$ if it is in the sum(X_s) highest value of X, otherwise set $x_{d_s}=0$
Output: $\boldsymbol{X}$, $\boldsymbol{P}$

$$\tau_{d_s}=\frac{y_{d_s}\prod_{s\in\mathcal{S}}\prod_{d'_s\in\mathcal{D}_s,d'_s\neq d_s}(1+y_{d'_s})}{\prod_{s\in\mathcal{S}}\prod_{d_s\in\mathcal{D}_s}(1+y_{d_s})-t'}\approx\frac{y_{d_s}}{1+y_{d_s}}. \quad (21)$$

Replacing (20) and (21) into (17), the optimization problem can be expressed as

$$\max_{\boldsymbol{X},\boldsymbol{Y}}\sum_{s\in\mathcal{S}}\sum_{d_s\in\mathcal{D}_s}(1-\psi_{d_s})\left[\theta_{d_s}x_{d_s}+y_{d_s}(1-\sum_{s\in\mathcal{S}}\sum_{d_s\in\mathcal{D}_s}y_{d_s})\right]$$
subject to: (22)

$$\text{C22.1: } \sum_{d_s\in\mathcal{D}_s}\Big[T_sx_{d_s}+\frac{y_{d_s}}{1+y_{d_s}}\times (T_f-T_s\sum\nolimits_{s\in\mathcal{S}}\sum\nolimits_{d_s\in\mathcal{D}_s}x_{d_s})\Big]\ge r_s,\ \forall s\in\mathcal{S}$$

$$\text{C22.2: } x_{d_s}p_{d_s}=0, \qquad \forall s\in\mathcal{S},\forall d_s\in\mathcal{D}_s$$

$$\text{C22.3: } T_s\sum_{s\in\mathcal{S}}\sum_{d_s\in\mathcal{D}_s}x_{d_s}\le T_{\max}.$$

To solve this optimization problem, we employ an iterative approach, in which each iteration consists of two steps. At each step, we solve the optimization problem over one variable, while for the other variable we use the value obtained from the last iteration. That is, for scalable reconfigurable MAC, we first maximize over $\boldsymbol{X}$ for fixed $\boldsymbol{Y}$, then we maximize over $\boldsymbol{Y}$ for fixed $\boldsymbol{X}$. The iteration continues until convergence happens between the results of two last rounds. In the following, details of this algorithm are presented.

The first step of the algorithm is to solve the optimization problem over $\boldsymbol{X}$. In fact, for a fixed value of $\boldsymbol{Y}$, the optimization problem becomes

$$\max_{\boldsymbol{X}}\sum_{s\in\mathcal{S}}\sum_{d_s\in\mathcal{D}_s}(1-\psi_{d_s})\left[\theta_{d_s}x_{d_s}+y_{d_s}(1-\sum_{s\in\mathcal{S}}\sum_{d_s\in\mathcal{D}_s}y_{d_s})\right]$$
subject to: C22.1 & C22.3 (23)

Clearly, this optimization problem is linear due to its linear objective function and constraints with respect to $\boldsymbol{X}$. In the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

Algorithm 2 Scalable Reconfigurable MAC

Input: $\boldsymbol{\Theta}$,$\boldsymbol{\Psi}$, $T_{\max}$, r_s $\forall s \in \mathcal{S}$
Initialization: Set initial value to $(\boldsymbol{X},\boldsymbol{Y})$
repeat
$(\boldsymbol{X'},\boldsymbol{Y'}) \leftarrow (\boldsymbol{X},\boldsymbol{Y})$
Step 1: Find $\boldsymbol{X}$
- $\boldsymbol{X} \leftarrow$ Solve the optimization problem (23) for fixed $\boldsymbol{Y}$

Step 2: Find $\boldsymbol{Y}$
- $\boldsymbol{Y} \leftarrow$ Solve the optimization problem (25) for fixed $\boldsymbol{X}$

until $|(\boldsymbol{X'},\boldsymbol{Y'}) - (\boldsymbol{X},\boldsymbol{Y})| < \varsigma'$
$p_{d_s} \leftarrow \frac{y_{d_s}}{\theta d_s(1+y_{d_s})}$
Set $x_{d_s} = 1$ if it is in the sum(X_s) highest value of X, otherwise set $x_{d_s} = 0$
Output: $\boldsymbol{X}$, $\boldsymbol{P}$

second step of the algorithm, we maximize the optimization problem over $\boldsymbol{Y}$ for a fixed $\boldsymbol{X}$. Here, the optimization problem is

$$\max_{\boldsymbol{Y}} \sum_{s\in\mathcal{S}} \sum_{d_s\in\mathcal{D}_s} (1-\psi_{d_s}) y_{d_s} \Big(1 - \sum_{s\in\mathcal{S}} \sum_{d_s\in\mathcal{D}_s} y_{d_s}\Big)$$
$$\text{subject to: C22.1} \tag{24}$$

This problem has a concave constraint but non-concave objective function, thus it is a non-convex optimization problem. By doing a simple manipulation, we rewrite the objective function as difference of two concave functions as follows

$$\max_{\boldsymbol{Y}} \sum_{s\in\mathcal{S}} \sum_{d_s\in\mathcal{D}_s} (1-\psi_{d_s})$$
$$\Bigg[(y_{d_s} - y_{d_s}^2) - \sum_{s'\in\mathcal{S}} \sum_{d'_s\in\mathcal{D}_s, d'_s\neq d_s} \frac{y_{d_s}^2 + y_{d_s}'^2}{2}\Bigg] -$$
$$\sum_{s\in\mathcal{S}} \sum_{d_s\in\mathcal{D}_s} \sum_{s'\in\mathcal{S}} \sum_{d'_s\in\mathcal{D}_s, d'_s\neq d_s} -[(1-\psi_{d_s}) \frac{(y_{d_s} - y'_{d_s})^2}{2}],$$
$$\text{subject to: C22.1} \tag{25}$$

With this reformulation, the problem falls into the category of difference of convex functions (DC) programming. To solve this optimization problem, we use an iterative approach, wherein at each iteration the second term of the objective function is linearized by Taylor expansion. The details of this algorithm can be found in Algorithm 2.

At each iteration, the computational complexity of Algorithm 2 consists of two steps: 1) solving a linear optimization problem and 2) obtaining the DC-programming results. The linear programming can be efficiently solved by using existing methods, since it has only $\mathcal{S}+1$ constraints. The second step, i.e. DC programming is solved in an iterative manner. Each iteration of this algorithm consists of a convex optimization problem, which due to the low number of constraints, i.e. $\mathcal{S}$, can be solved in an efficient manner. Based on the simulation results, the average number of DC iterations does not vary much over number of devices. Furthermore, the outer iteration terminates after a few rounds.

V. Learning-based Reconfigurable MAC via Thompson Sampling

For scenarios of unknown packet arrival probabilities, the input information for Algorithm 1 is not available. One approach is to apply a simple passive learning that uses the empirical mean of packet arrival probabilities as an estimator. In this algorithm, each time a device sends a packet over DA, its estimated packet arrival probability is updated. The problem is that the devices having higher empirical mean may obtain a higher chance for DA transmission than the ones that have smaller empirical mean in the past but may show higher mean in the future. Thus, this approach may lead to a huge performance loss over time.

In other words, if we only rely on the exploitation that uses empirical mean as an estimator, we may take the chance from the high-traffic devices that showed low arrival rates in the past. On the other hand, if we assign time-slots to the devices with low empirical mean (exploration), the performance might be decreased because a device that shows low arrival probability in the past might actually be a low traffic device. Therefore, a proper trade-off between exploration and exploitation is needed. As Thompson sampling is able to provide this balance, we use Thompson sampling indices as the inputs for the Algorithm 1 in which, instead of using empirical mean, indices are sampled from a beta distribution with mean equal to the empirical mean of the device.

In the following, we first describe the Thompson sampling for classical CMABs. After that, we provide the details of the proposed Thompson-sampling-based algorithm. Then, we develop a thresholding algorithm for the scheduling, model it as a TMAB, and apply TS for learning.

A. Thompson Sampling: A Brief Overview

In a classical CMAB setting, there is a system of M arms, each having a Bernouli reward distribution with an unknown mean. At each round, $L < M$ arms are chosen to be played. Let $\boldsymbol{\mu} = (\mu_1, \mu_2, ..., \mu_M)$ be the vector of expectations of all arms, which is unknown to the player. The goal is to repeatedly play these arms in multiple rounds such that the total expected reward over T time steps is maximized.

In order to select the arms, at each round, the TS algorithm assigns a score to each arm. This score is randomly generated based on a prior distribution. One convenient choice of priors for Bernoulli rewards is the beta distribution, which is a family of continuous probability distributions defined in the interval of $[0, 1]$. Furthermore, it is the conjugate distribution of the Bernoulli distribution, i.e., assuming beta distribution as prior, the posterior distribution is also from the same family [8]. The probability distribution function (pdf) of the beta distribution is denoted by $\mathrm{beta}(\alpha, \beta)$, where $\alpha > 0$ and $\beta > 0$ are the shape parameters. The mean of $\mathrm{beta}(\alpha, \beta)$ is equal to $\frac{\alpha}{\alpha+\beta}$; and apparently from the pdf, the higher are the α and β, the narrower is the concentration of $\mathrm{beta}(\alpha, \beta)$ around the mean. For Bernoulli rewards, after playing each arm, the shape parameters are updated as following. If the reward obtained by playing arm i is 1, α is incremented by 1 otherwise we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

have $\beta = \beta + 1$. In other words, the posterior distribution is simply $\mathrm{beta}(\alpha+1, \beta)$ or $\mathrm{beta}(\alpha, \beta+1)$, depending on whether the reward is 1 or 0, respectively. The Thompson sampling algorithm initially assumes the arm m to have prior $\mathrm{beta}(1, 1)$ on μ_m, which is natural because $\mathrm{beta}(1, 1)$ is the uniform distribution on $[0, 1]$. At each time step, the TS algorithm samples from these posterior distributions of the μ_m's, and plays the arms which have the L largest scores [21]. Thus, the computational complexity of TS is $\mathcal{O}(M)$.

B. Thompson Sampling for Reconfigurable MAC

Here, we develop a Thompson sampling-based algorithm for the reconfigurable MAC. The proposed algorithm helps to learn the unknown packet arrival probabilities. To this end, at each round Thompson sampling indices of devices are passed as inputs to the optimal scheduler presented in Section IV. The optimal scheduler indicates the time-slot allocation for devices, i.e. which arms are chosen to be played. Once the DA regime terminates, packet arrival probabilities of devices are updated. In the following, we describe how the update process is performed.

In the proposed system model, each device has a queue in which a packet generated at a certain time-slot could be maintained in the queue, when the device d_s is chosen for transmission it may transmit the packet which was generated in the previous frames. Consequently, this could result in a biased shape parameter update in the Thompson sampling, and therefore, a biased estimation of the packet arrival probabilities over a long run.

To avoid a biased estimation, the proposed algorithm takes advantage of the piggybacked extra bit with any transmission which indicates whether the corresponding device has still any packets in its queue or not (i.e., $q_{d_s} = 1$ or $q_{d_s} = 0$). More specifically, the values of α_{d_s} and β_{d_s} would be updated only when $q_{d_s} = 0$. In other words, whenever a device transmits a packet with $q_{d_s} = 1$, the scheduler keeps assigning time-slots to that device in the subsequent frames until it has no more packets in the queue. At this point, the scheduler updates the values of α_{d_s} and β_{d_s}, where α_{d_s} is increased by the number of packets that are successfully transmitted during the last sequence of device d_s's transmission denoted by $w_{d_s}(t)$ and β_{d_s} is increased by $t - v_{d_s}(t) - w_{d_s}(t)$. Note that the increment in α_{d_s} also contains the successful transmission over RA, in case the first packet of this transmission sequence was started by transmitting a packet over RA. Thus, in this approach, even RA observations can be used to update the empirical mean, while, in a pure CSMA scheme, this information cannot be used to update these parameters. Let assume that, in a pure CSMA scheme, the AP updates α_{d_s} whenever it receives a packet, and updates β_{d_s} whenever it does not receive a packet from the device. The problem is that if the AP does not receive a packet, it does not mean that the device did not have a packet for transmission. The device might have a packet for transmission, but it does not get a chance to transmit or its sent packet might be collided. Thus, the shape parameters cannot be updated in a correct manner. The details of the proposed algorithm can be found in Algorithm 3.

Algorithm 3 TS algorithm for reconfigurable MAC

Input: $T_{\max}$, r_s $\forall s \in \mathcal{S}$
Initialization: $\boldsymbol{\alpha} = \mathbf{1}$, $\boldsymbol{\beta} = \mathbf{1}$, $t = 1$, $\boldsymbol{W} = \mathbf{1}$, $\boldsymbol{V} = \mathbf{1}$
repeat
  Step 1: Sample $\boldsymbol{\phi}(t) \sim \mathrm{beta}(\boldsymbol{\alpha}, \boldsymbol{\beta})$
  Step 2: Run Algorithm 1 with $\boldsymbol{\phi}$
  Step 3: Update $\boldsymbol{\alpha}, \boldsymbol{\beta}$
  if $x_{d_s}(t) = 1$ & $q_{d_s}(t) = 0$ **then**
    $\alpha_{d_s}(t) = \alpha_{d_s}(t-1) + w_{d_s}(t)$
    $\beta_{d_s}(t) = \beta_{d_s}(t-1) + t - v_{d_s}(t) - w_{d_s}(t)$
  end if
  if $x_{d_s}(t) = 1$ & $q_{d_s}(t) = 1$ **then**
    $\alpha_{d_s}(t) = \alpha_{d_s}(t-1)$
    $\beta_{d_s}(t) = \beta_{d_s}(t-1)$
  end if
  $t = t + 1$
until $t < T$

Note that for scenarios of time-varying packet arrival probabilities, the calculation of Thompson sampling indices needs modifications since they are randomly chosen from beta distribution with mean equal to the sample mean of the packet arrival probabilities. As the mean varies over time, the information obtained from previous observations should be carefully used and some perturbation should be introduced to enable the tracking of its variations over time. For example, one approach is to use a weighted averaging method to update the parameters of Beta distribution, in which larger weights are assigned to recent observations.

C. Thompson Sampling for Thresholding Multi-Armed Bandits

In the proposed reconfigurable MAC, the aim is to maximize the network throughput. To reach this goal, it separates devices into two groups, by allocating devices with higher packet arrival probabilities to the DA regime and the rest to the RA regime. Another interpretation is that devices with mean throughput of larger than a certain threshold are proper candidates for DA, while it is more efficient not to assign any time-slots for the rest. In the following theorem, we analytically prove this fact. But, first, we define $\vartheta_{d_s} = (1 - \psi_{d_s}) \times \theta_{d_s}$, which represents the expected throughput of device d_s, when it is selected for DA.

Theorem 1. *If $\boldsymbol{X}^*$ is the solution of the optimization problem (16), then*

$$\vartheta_{d_s} > \gamma^*, \quad \forall d_s \in \mathcal{N}_{da}. \tag{26}$$

when all devices have distinct ϑ_{d_s} and there is only one slice. In (26), $\gamma^* = \max\{\vartheta_{d_s}\}$ for all $d_s \in \mathcal{N}_{ra}$, $\mathcal{N}_{ra}$ is the set of all devices assigned to the RA regime (i.e., $\forall d_s \in \mathcal{D}_k$:$x^*_{d_s} = 0$), and $\mathcal{N}_{da}$ is the set of all devices assigned to the DA regime (i.e., $\forall d_s : x^*_{d_s} = 1$).

Proof. See Appendix B. □

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

Algorithm 4 TS-TMAB algorithm for thresholding reconfigurable MAC

Initialization: $\boldsymbol{\alpha} = \mathbf{1}$, $\boldsymbol{\beta} = \mathbf{1}$
for $t = 1 : T$ **do**
  for arm $m = 1 : M$ **do**
    sample $\phi_m(t) \sim \text{beta}(\alpha_m, \beta_m)$
  end for
  Set $\mathcal{M}(t) = \{m \mid \phi_m(t) > \gamma\}$
  if Size of $\mathcal{M}(t)$ is $> T_{\max}$ **then**
    Keep only arms with $T_{\max}$ largest values of $\phi_m(t)$
  end if
  for $m \in \mathcal{M}(t)$ **do**
    Play arm m
    Update α_m and β_m
  end for
end for

Remark 1. *For multiple slices, the proposed scheduling is like having multiple TMABs on different slices with distinct thresholds.*

Remark 2. *Considering cases where some devices have the same θ_{d_s}, Theorem 1 still holds when devices have the same expectations larger or smaller than $\gamma*$, which is the breaking point.*

Based on Theorem 1, an alternative algorithm for reconfigurable MAC is a threshold-based algorithm in which devices having mean throughput larger than certain threshold are chosen for DA while the rest are considered for RA. Assuming that packet arrival probabilities are unknown, this problem can be modeled as a TMAB wherein each arm corresponds to a device.

Thresholding multi-armed bandit is a specific class of CMABs, in which the arm is worth playing if its expected reward is larger than a certain threshold (denoted by γ). As a result, when the player has an option to choose from M arms, the optimal number of arms which should be played is dependent on the number of arms for which we have $\mu_m > \gamma$. Then, since the expected rewards of different arms are unknown, the number of arms which gives the highest expected reward is not known either. Therefore, index-based policies which choose the L arms with the highest indices are not applicable for TMABs. However, in the TS approach, the scores are randomized around the estimated mean of the arms, thus they are more proper to be compared against the threshold. The TS-TMAB algorithm is presented in Algorithm 4.

VI. Regret Analysis

In this section, we study the regret bound in DA regime of thresholding reconfigurable MAC when the network consists of one slice, $Q_{\max} = 0$ (i.e., there is no queue to store the packets) and $T_{\max} = \Gamma$, where Γ indicates the number of optimal arms.

Notations: $\mathbf{1}\{\mathcal{A}\}$ is an indicator function which is equal to 1 if event $\mathcal{A}$ holds and 0 otherwise. $d(p,q) = p\log(p/q) + (1-p)\log((1-p)/(1-q))$ is the Kullback-Leibler divergence between two Bernoulli distributions with means p and q.

For the TS-TMAB algorithm as explained in Algorithm 4, in Theorem 2, we prove that the TS-TMAB algorithm for binary rewards achieves an optimal regret bound.

Theorem 2. *The regret of TS-TMAB Algorithm is upper bounded by*

$$\mathbb{E}[R(T)] \leqslant \sum_{m\in\mathbb{\Gamma}^-} \frac{\Delta_m}{d(\mu_m^+, \gamma^-)}\log(T) + \mathcal{O}\left(\frac{1}{\delta^2}\right) \tag{27}$$

where Δ_m represents the regret caused by playing suboptimal arm m and can be upperbouded by $\max_{i\in M}\mu_i - \mu_m$. Also, $\mu_m^+ = \mu_m + \delta$, $\gamma^- = \gamma - \delta$ for $\delta > 0$ and $\mathbb{\Gamma}^-$ is the set of arms for which $\mu_m < \gamma$.

Proof. Let first define arm m as suboptimal, if $\mu_m < \gamma$. Different from CMABs, the regret of a TMAB is not only dependent on the number of times that a suboptimal arm is chosen. It is also affected by the number of times that only a sub-set of all optimal arms is played. Suppose $\mathbb{\Gamma}^+ = \{m \mid \mu_m > \gamma\}$ as the set of all optimal arms and Γ as the number of optimal arms in $\mathbb{\Gamma}^+$.

Lemma 1. *The regret of TS-TMAB algorithm can be decomposed as*

$$R(T) = R_u(T) + \sum_{m\in\mathbb{\Gamma}^-} R_m(T), \tag{28}$$

where $R_u(T)$ represents the regret caused when the number of optimal arms played is less than Γ and $R_m(T)$ indicates the regret caused by playing the suboptimal arm $m \in \mathbb{\Gamma}^-$.

According to Lemma 1, in order to derive an upper bound for R, we could separately study regret bounds for R_u and R_m.

To find an upper bound for R_u, let us first define an event $\mathcal{U}(t) = \{\varphi^* > \gamma^-\}$, where φ^* represents the Γ-th largest element of the vector $\Phi = (1-\Psi)\times\Phi$. The complement of this event $\neg\mathcal{U}(t)$ represents the situation that the number of selected arms is less than Γ, which implies that at least one of the optimal arms is underestimated and not played.

The regret caused by event $\neg\mathcal{U}(t)$ depends on the number of optimal arms, which are not played, as well as which ones exactly. The worst-case scenario is when no optimal arm is selected. For this case, the regret is upper bounded by Γ, since for each arm m we have $\mu_m \leq 1$. Consequently, for any case, it can be concluded that the regret caused by $\neg\mathcal{U}(t)$ is always smaller than $\Gamma - 1$. Having an upper bound for the instantaneous event of $\neg\mathcal{U}(t)$, in the next lemma, we calculate an upper bound on the number of occurrences of $\neg\mathcal{U}(t)$ over T, aiming to find $R_u(T)$.

Lemma 2. *The regret $R_u(T)$ is upper bounded by*

$$R_u(T) = \Gamma\sum_{t=1}^{T}\mathbf{1}\{\neg\mathcal{U}(t)\} \leq \mathcal{O}\left(\frac{1}{(\gamma-\gamma^-)^2}\right) = \mathcal{O}\left(\frac{1}{\delta^2}\right) \tag{29}$$

Proof. The proof is provided in [23]. □

Here, we continue by studying the regret incurred by choosing a sub-optimal arm $m \in \mathbb{\Gamma}^-$. More specifically, Lemma 3 presents an upper bound on $R_m(T)$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

Lemma 3. *The regret of each suboptimal arm m is upper bounded as*

$$R_m(T) \leq \frac{\Delta_m}{d(\mu_m^+, \gamma^-)} \log(T) + \mathcal{O}\left(\frac{1}{\delta^2}\right) \tag{30}$$

Proof. See Appendix C. □

According to Lemmas 1, 2 and 3, the regret bound in Theorem 2 can be concluded. □

According to Theorems 1 and 2, in the following corollaries, we discuss the regret bounds that can be achieved by applying the proposed TS-TMAB-based algorithm for S slices.

Corollary 1. *Considering that there are S different slices, according to Remark 2, the reconfigurable MAC behaves like multiple TMABs with distinct thresholds. Thus, the regret of this problem also can be upperbounded by the aggregate regrets of each TMAB as in the worst case.*

VII. Illustrative Results

The simulation is done in MATLAB and GP problems are solved using CVX [6]. For evaluation, we study the throughput, defined as the number of packets successfully transmitted in a frame (pckt/f) and delay which represents the number of frames between the time that a packet is generated until it is received by the AP. Results are compared with following methods:

- *p-persistent CSMA*: In this scheme, all devices compete with each other by performing p-persistent CSMA. Parameter p is the same for all devices and it is set to 0.05.
- *Random Hybrid DA-RA*: In this scheme, no traffic arrival statistic is taken into account; at each frame, $T_{\max}$ time-slots are assigned to the devices randomly, while the rest of devices compete in the CSMA regime with $p = 0.05$. The reason that we use this algorithm is to show how considering traffic parameters can enhance the network performance.
- *Distributed queuing (DQ)*: In this scheme, the frame structure is divided into three parts: i) C sub-slots for collision resolution, ii) one slot for data transmission and iii) one sub-slot for transmission of feedback information from AP to devices [28]. The MAC scheme works based on two queues: contention resolution queue (CRQ) and data transmission queue (DTQ). At each frame, the devices at the front of CRQ, randomly choose one of the C contention sub-slots or backoff units to transmit an access request sequence (ARS). Thus, the status of each sub-slot can be 1) idle (no ARS is transmitted), 2) successful (only one ARS is transmitted), and 3) busy (more than one ARS is transmitted). The AP broadcasts these information at the end of the frame in a feed-back slot. Devices with successful ARS transmission are added to the DTQ, while colliding devices over each sub-slot are added to CRQ. Furthermore, at each frame, during the data transmission phase, the device at the front of DTQ transmits its packet. It should be noted that each device can compute its position in each queue based on the feedback information sent by the AP.

TABLE III: List of Simulation Parameters

Simulation Parameter	Value
T_f	16 time-slots
T_s	12 backoff units
r_s	6 time-slots
$Q_{\max}$	10
$T_{\max}$	10 time-slots
ζ	3
υ	0 dB
$\frac{P_t}{\sigma^2}$	20 dB

A. Reconfigurable MAC: Known Statistics

We consider a network of an AP and two slices with all devices within the communication range. The reason to choose two slices is to better demonstrate the dynamics of direct effects of change in one slice on another. We assume that each frame is divided into 16 time-slots and the length of each time-slot, $T_{\rm s}$, is equal to 12 backoff units. The simulation time is set to 100 frames and each simulation is repeated 10 times. We also set the reservation of each slice equal to $r_s = 6$, the maximum length of each queue equal to $Q_{\max} = 10$, $T_{\max} = 10$ and the convergence parameter $\varsigma = 0.05$. Furthermore, we assume that channel parameters are as following: path loss exponent $\zeta = 3$, receiver threshold $\upsilon = 0$ dB and $\frac{P_t}{\sigma^2} = 20$ dB. The list of these parameters is provided in Table III.

1) Medium-size Scenario: First, we consider a medium-size network, where the packet arrival probabilities of two slices are set as $\boldsymbol{A}_1 = \{[0.8]_5, [0.4]_8\}$ and $\boldsymbol{A}_2 = \{[0.8]_5, [0.4]_{N_2-4}\}$. The notation $[f]_{c_f}$ indicates that there are c_f devices having the same packet arrival probability of f. Devices are randomly located in a circular area following a uniform distribution. The radii of the circular areas are 2m and 5m for devices with packet arrival probability of 0.8, and the rest, respectively. Furthermore, the parameter C of the DQ MAC is set to 4 backoff units. To study how well the isolation among slices can be protected in the presence of a variation in one slice, the throughput of both slices is plotted for different numbers of devices in slice 2, while no parameter has changed in slice 1. As shown in Figure 2, by increasing N_2, the throughput of slice 1 degrades slightly, while its reservation is still met. The reason is that larger values of p_{n_1} are assigned to the devices of this slice to keep it isolated from any variation in slice 2. However, the throughput of slice 2 increases since more packets are generated in this slice and therefore assigned time-slots to this slice are left idle with a lower probability. However, by using p-persistent CSMA, DQ, and random hybrid MAC, the throughput of slice 1 degrades as N_2 increases since the devices have less chance to transmit their packets. The network throughput also decreases for the pure p-persistent scheme due to a larger number of collisions but remains almost the same for the DQ scheme (since its

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

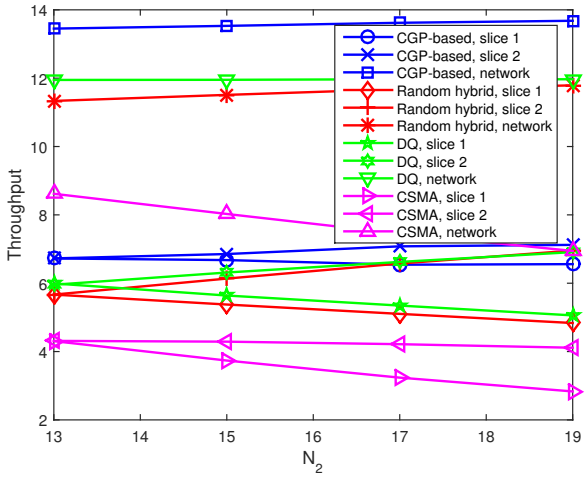

Fig. 2: Throughput versus N_2

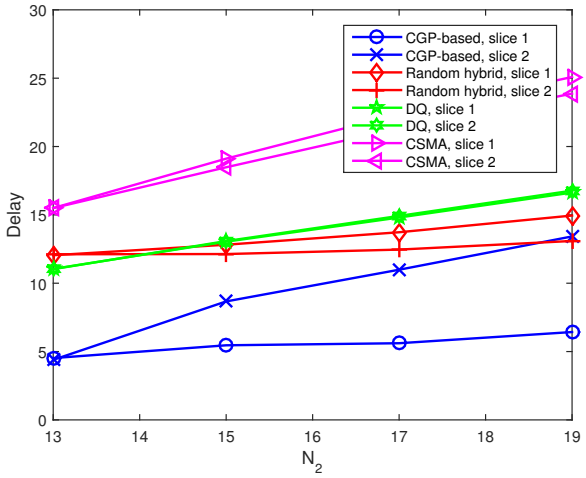

Fig. 3: Delay versus N_2

throughput is mainly dependent on the ratio of the duration of the contention resolution and data transmission phase which is constant for this setup), and the random hybrid scheme (due to the increment in DA throughput). Although generally increasing the number of devices can affect the successful transmission probability of ARS which consequently impacts the throughput, the effect is not noticeable in this scenario.

Delay results in terms of number of frames are shown in Figure 3. As observed, the CGP-based algorithm almost outperforms other schemes. This is due to the fact that with piggyback mechanism, a device can request for a time-slot. As a result, the probability that a device may have packet for transmission is updated at each frame and based on that free time-slots are assigned to the devices. This prevents starvation or long delay when a device has a packet for transmission. Since in CSMA, devices compete with each other and a device having a packet for transmission may fail or does not get a chance to transmit after several time frames. In general, delay increases with increasing N_2. For the other three schemes, average delay is quite similar for both slices 1 and 2. On the other hand, for the proposed CGP-based scheme, a noticeably larger delay increase with N_2 in slice 2 as compared to slice 1. This indicates a much better slice isolation offered by the proposed scheme, i.e., change in slice 2 (i.e., increasing N_2) affects the QoS (i.e., delay) of slice 2 but has a much lower effect on the QoS (i.e., delay) of slice 1.

We also obtain the results for higher numbers of slices. Figure 4 shows the average throughput for $S = 2, 4, 6$ and 8. In these simulations, all slices have the same traffic parameters, while the last slice has a larger number of devices than the rest. The traffics parameter setting for different slices is as follows: $\{[0.8]_{\lfloor 10/S \rfloor}, [0.4]_{\lfloor 16/S \rfloor}\}$, for $s < S$, $S \in \{2, 4, 8\}$. The last slice, i.e., $s = S$ when $S \in \{2, 4, 8\}$ has additional devices compared to the other slices and its traffic parameter setting is $\{[0.8]_{\lfloor 10/S \rfloor + 10 \mod S}, [0.4]_{\lfloor 16/S \rfloor + (16 \mod S) + 6}\}$, where $a \mod b$ indicates the remainder of a divided by b. For $S = 6$,

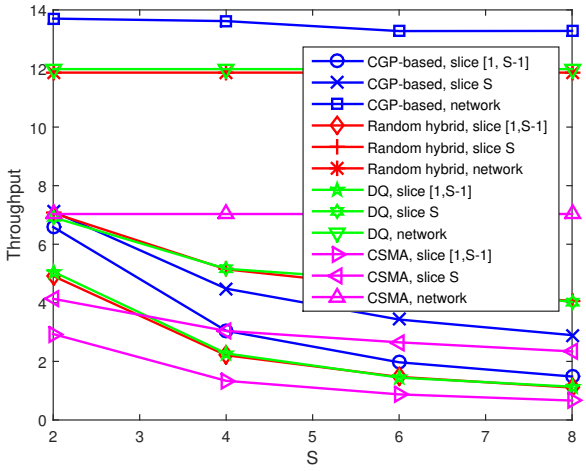

Fig. 4: Throughput versus S

the traffic parameters of slices are as follows: $\{[0.8]_1, [0.4]_3\}$ for $s < 5$, and $\{[0.8]_5, [0.4]_7\}$ for $s = 6$. Furthermore, the reservation per slice is set to $r_s = 12/S$. As observed, the results show that the throughput of each slice is equal or greater than its airtime reservation, while other schemes fail to provide the isolation. Furthermore, with increasing S, the total throughput decreases. The reason is that in this scenario, the average traffic of each slice is close to its airtime reservation (except the last slice); therefore, to be able to satisfy the airtime reservation of the slice s, a device belonging to this slice with lower expected throughput might be assigned a time-slot, while a device with higher expected throughput from the last slice is not allocated a time-slot.

2) Dense Virtulazed Wireless Network: For this scenario, we consider a network consisting of two slices with traffic parameters as follows: $\boldsymbol{A}_1 = \{[0.95]_5, [0.85]_5, [0.75]_5, [0.65]_5, [0.55]_5, [0.45]_5, [0.35]_5, [0.3]_5, [0.25]_5, [0.15]_5\}$ and $\boldsymbol{A}_2 =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

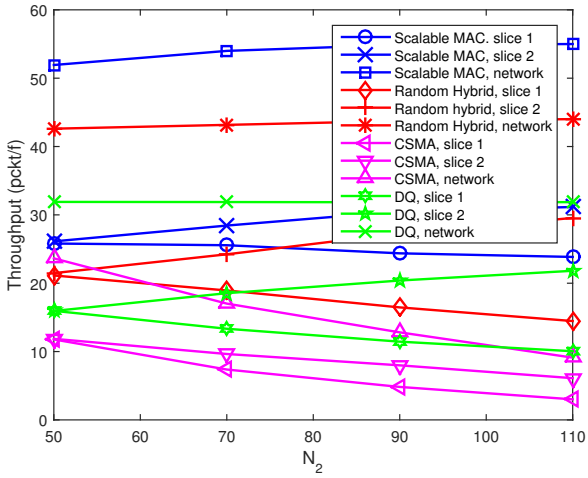

Fig. 5: Throughput versus N_2

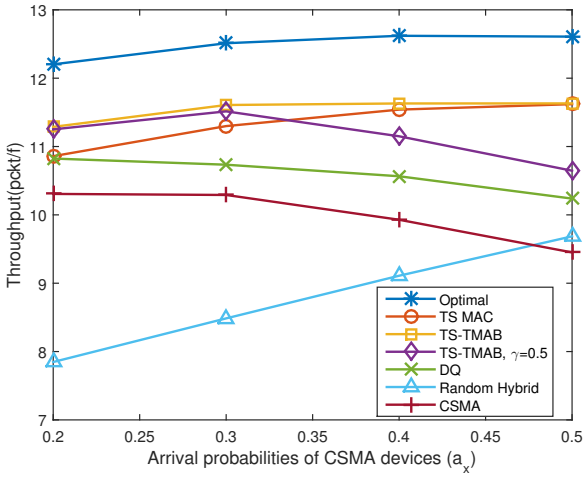

Fig. 6: Throughput versus a_x

$\{[0.95]_5, [0.85]_5, [0.75]_5, [0.65]_5, [0.55]_5, [0.45]_5, [0.35]_5, [0.25]_5, [0.15]_5, [0.3]_{N_2-45}\}$[3]. For this scenario, we consider shorter packet sizes each with length of 3 backoff units. Similarly, the length of each time-slot, T_s is 3 backoff units and T_f is 64 time-slots. Furthermore, $T_{\max}$ and r_s are set to 50 and 24 time-slots, respectively. Both the inner and outer convergence parameters are set to 0.01. For the DQ, we consider $C = 3$ backoff units.

To investigate the algorithm performance in terms of isolation, we fix the number of devices in slice 1 and increase the number of devices of the other slice, N_2. Results in the plotted Figure 5 show that the proposed scalable MAC outperforms other schemes and slice 1 throughput is almost not affected by increasing N_2. The reason is that the scheduling algorithm takes into account the reservation of each slice, thus isolation is achieved. On the other hand, its throughput slightly increases by increasing N_2 since as the number of packets grows, overall DA throughput increases while RA throughput is controlled by adjusting the p parameter. The random hybrid and CSMA schemes which employ CSMA with fixed p parameter, thus by increasing N_2, less number of devices of slice 1 get a chance to transmit their packets and furthermore, in case of transmission, there is a higher probability of collision. The reason that the overall throughput of random hybrid MAC does not drop is that increasing N_2 leads to more congested network meaning that DA time-slot left idle with lower probability which here almost compensates the CSMA throughput reduction. DQ offers a much lower network throughput than the proposed scheme since for these scenarios devices have small packet sizes and consequently the amount of time devoted for the contention resolution is large compared to the data transmission.

[3]Such parameters are chosen to include both high load and low load devices as in practice devices with heterogeneous traffic parameters exist.

B. Reconfigurable MAC: Unknown Statistics

Here, we focus on the performance of TS algorithms. We consider the scenarios with $T_f = 16$ time-slots, $T_s = 12$ backoff units, $T_{\max} = 10$ time-slots, and C of DQ is 4 backoff units.

1) Effect of Suboptimal Arms Statistic: Here, we consider a scenario in which packet arrival probabilities of suboptimal arms, i.e., CSMA devices, are increased. The motivation behind defining this scenario is to show how the performance of the TS-TMAB is dependent on the closeness of mean reward of suboptimal arms to the optimal arms. In Figure 6, the simulation results are obtained for T =100 frames, where we set $\boldsymbol{A}_1 = \{[0.9]_3, [0.8]_2, [a_x]_{15}\}$, $\boldsymbol{A}_2 = \{[0.95]_3, [0.85]_2, [a_x]_{15}\}$ and a_x represents the packet arrival probabilities of suboptimal arms. Furthermore, p is set to 0.05 for all CSMA devices. As observed, the TS-TMAB algorithm with $\gamma = \gamma^*$ achieves better performance compared to the other schemes except the optimal algorithm. The reason is that, for $\gamma = \gamma^*$, although arrival probabilities are unknown, however as γ is set to the optimal value, the performance of this scheme becomes closer to the optimal algorithm. However, the performance of the algorithm for $\gamma = 0.5$ drops. Because, for lower threshold, the probability that CSMA devices are chosen for DA increases especially for larger a_x. Therefore, for $\gamma = 0.5$, the TS-TMAB scheme has the lowest performance at $a_x = 0.5$. Furthermore for $a_x = 0.2$, TS MAC has the largest regret. The reason is that in this case suboptimal devices have larger throughput difference with optimal devices. Therefore, in case they are chosen for DA, lower throughput will be achieved, leading to the larger regret. Furthermore, it is shown the throughput of the random hybrid scheme increases by increasing a_x, since time-slots left with a lower probability. For the CSMA and DQ schemes, increasing a_x leads to throughput decrement since the number of collisions increases.

2) Effect of Time: Here, we investigate how the performance of TS-TMAB varies over T. As time grows, more

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

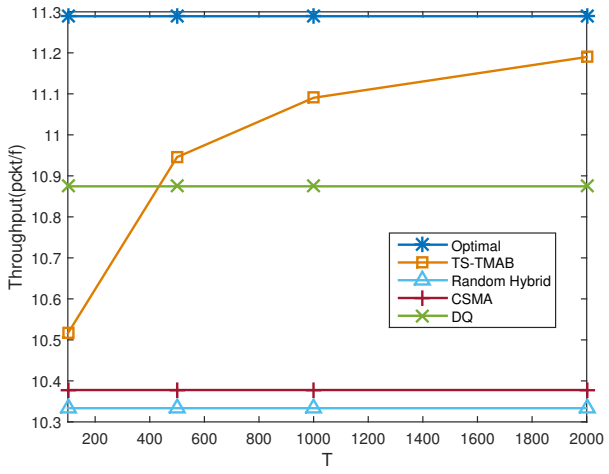

Fig. 7: Throughput versus T (High Load)

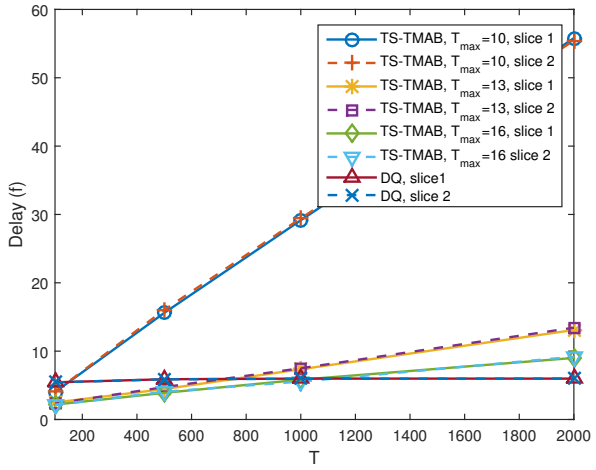

Fig. 9: Delay versus T (High Load)

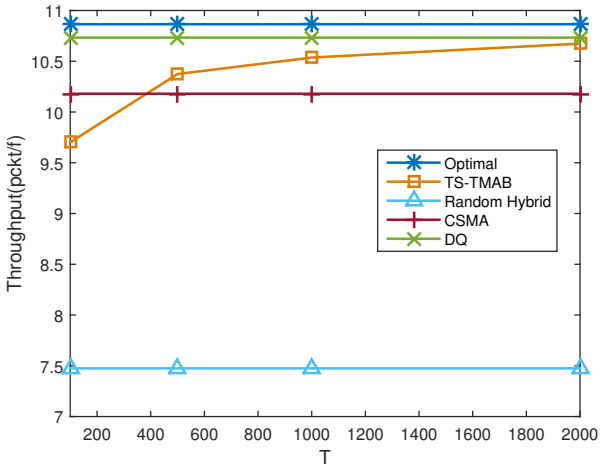

Fig. 8: Throughput versus T (Low Load)

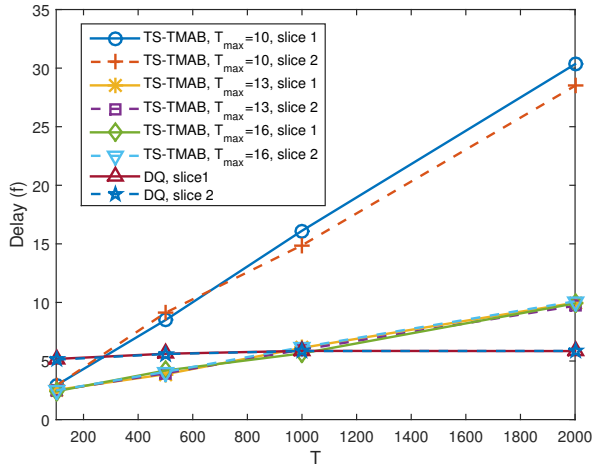

Fig. 10: Delay versus T (low Load)

observations on device activities are obtained and more precise estimation for packet arrival probabilities can be achieved. Thus, we obtain the numerical results versus T in order to observe the learning performance over time and demonstrate the effectiveness of learning packet arrival probabilities to achieve better performance in terms of throughput and delay. We provide the results for two settings: high traffic devices and low traffic devices versus T. In the first scenario, all devices have high packet arrival probabilities; $\boldsymbol{A}_1 = \boldsymbol{A}_2 = \{[0.9]_2, [0.8]_2, [0.6]_2, [0.55]_2, [0.5]_3\}$. In the second scenario, devices have lower packet arrival probabilities; $\boldsymbol{A}_1 = \boldsymbol{A}_2 = \{[0.8]_2, [0.7]_2, [0.6]_2, [0.2]_{12}\}$. As observed in Figures 7 and 8, by increasing T the TS-TMAB performance gets closer to the optimum in both scenarios. However, for the random hybrid, CSMA and DQ schemes, the performance is not dependent on T. We further obtain the delay results for these two scenarios, where $Q_{\max} = 4$, shown in Figures 9, 10. In these figures, we omit the results for CSMA and random hybrid schemes due to their large delay. As observed the delay of TS-TMAB depends on the size of $T_{\max}$. For these parameter settings, larger $T_{\max}$ leads to a lower delay. Since at each frame, more devices are scheduled in the DA, more packets with shorter delay are transmitted. Furthermore, DQ achieves low delay since it applies a queuing strategy which can prevent packets from experiencing long delay.

VIII. Conclusion

This paper presents a reconfigurable MAC, where DA and RA are used for devices with high and low packet transmission probabilities, respectively. This scheduling is formulated as an optimization problem with the objective to maximize the network throughput subject to constraints on slice reservations. To solve this problem, we show that it belongs to the class of CGP, which can be efficiently solved by applying approximations and solving the sequence of resulting GP problems. Furthermore, a scalable algorithm is developed for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

dense networks. In the proposed scheme, to maximize the throughput, packet arrival statistics are taken into account. However, in practice this information may not be known in prior. For these scenarios, two Thompson sampling-based algorithms are proposed. Furthermore, the regret analysis is provided for performance evaluation of TS-TMAB algorithm. Finally, using simulation results, we show the effectiveness of the proposed algorithms for both known and unknown packet arrival statistics.

REFERENCES

[1] F. Ghavimi and H.-H. Chen, "M2M communications in 3GPP LTE/LTE-A networks: architectures, service requirements, challenges, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 525–549, 2015.

[2] E. Soltanmohammadi, K. Ghavami, and M. Naraghi-Pour, "A survey of traffic issues in machine-to-machine communications over LTE," *IEEE Internet of Things J.*, vol. 3, no. 6, pp. 865–884, 2016.

[3] H. Wen, P. K. Tiwary, and T. Le-Ngoc, *Wireless Virtualization*. Springer, 2013.

[4] A. Dalili Shoaei, M. Derakhshani, S. Parsaeifard, and T. Le-Ngoc, "MDP-based MAC design with deterministic backoffs in virtualized 802.11 WLANs," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7754–7759, 2016.

[5] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Trans. on Network and Service Management*, vol. 13, no. 3, pp. 462–476, 2016.

[6] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[7] A. Dalili Shoaei, M. Derakhshani, S. Parsaeefard, and T. Le-Ngoc, "Efficient and fair hybrid TDMA-CSMA for virtualized green wireless networks," *Proc. IEEE Veh. Tech. Conf. (VTC)*, 2016.

[8] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in *Advances in Neural Information Process. Systems*, 2011, pp. 2249–2257.

[9] G. C. Madueño, Č. Stefanović, and P. Popovski, "Reliable and efficient access for alarm-initiated and regular M2M traffic in IEEE 802.11 ah systems," *IEEE Internet of Things J.*, vol. 3, no. 5, pp. 673–682, 2016.

[10] Y. Liu, C. Yuen, X. Cao, N. U. Hassan, and J. Chen, "Design of a scalable hybrid MAC protocol for heterogeneous M2M networks," *IEEE Internet of Things J.*, vol. 1, no. 1, pp. 99–111, 2014.

[11] Y. Yang and S. Roy, "Grouping-based MAC protocols for EV charging data transmission in smart metering network," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 7, pp. 1328–1343, 2014.

[12] H. Wu, C. Zhu, R. J. La, X. Liu, and Y. Zhang, "Fast adaptive S-ALOHA scheme for event-driven machine-to-machine communications," in *Proc. IEEE Veh. Tech. Conf. (VTC)*. IEEE, 2012, pp. 1–5.

[13] P. Si, J. Yang, S. Chen, and H. Xi, "Adaptive massive access management for QoS guarantees in M2M communications," *IEEE Trans. Veh. Technol.*, vol. 64, no. 7, pp. 3152–3166, 2015.

[14] H. He, Q. Du, H. Song, W. Li, Y. Wang, and P. Ren, "Traffic-aware ACB scheme for massive access in machine-to-machine networks," in *Proc. IEEE Intl. Conf. Commun. (ICC)*. IEEE, 2015, pp. 617–622.

[15] C. W. Park, D. Hwang, and T.-J. Lee, "Enhancement of IEEE 802.11 ah MAC for M2M communications," *IEEE Commun. Letters*, vol. 18, no. 7, pp. 1151–1154, 2014.

[16] A. Dalili Shoaei, M. Derakhshani, S. Parsaeefard, and T. Le-Ngoc, "Learning-based hybrid TDMA-CSMA MAC protocol for virtualized 802.11 WLANs," in *Proc. IEEE Intl. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC)*. IEEE, 2015, pp. 1861–1866.

[17] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[18] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1466–1478, 2012.

[19] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework, results and applications," in *Proc. Intl. Conf. on Machine Learning*, 2013, pp. 151–159.

[20] S. L. Scott, "A modern Bayesian look at the multi-armed bandit," *Applied Stochastic Models in Business and Industry*, vol. 26, no. 6, pp. 639–658, 2010.

[21] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," in *COLT*, 2012, pp. 1–39.

[22] S. Agrawal and N. Goyal, "Further optimal regret bounds for Thompson sampling," in *Artificial Intelligence and Statistics*, 2013, pp. 99–107.

[23] J. Komiyama, J. Honda, and H. Nakagawa, "Optimal regret analysis of Thompson sampling in stochastic multi-armed bandit problem with multiple plays," in *Intl. Conf. on Machine Learning*, 2015, pp. 1152–1161.

[24] A. Locatelli, M. Gutzeit, and A. Carpentier, "An optimal algorithm for the thresholding bandit problem," *arXiv preprint arXiv:1605.08671*, 2016.

[25] O. Dementev, O. Galinina, M. Gerasimenko, T. Tirronen, J. Torsner, S. Andreev, and Y. Koucheryavy, "Analyzing the overload of 3GPP LTE system by diverse classes of connected-mode MTC devices," in *Internet of Things (WF-IoT)*. IEEE, 2014, pp. 309–3012.

[26] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[27] M. del Mar Hershenson, S. P. Boyd, T. H. Lee, "Optimal design of a CMOS op-amp via geometric programming," *IEEE Trans. on CAD of ICs and Systems*, vol. 20, no. 1, pp. 1–21, 2001.

[28] A. Laya, C. Kalalas, F. Vazquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Goodbye, Aloha!" *IEEE Access*, vol. 4, pp. 2029–2044, 2016.

[29] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

APPENDIX

A. A brief overview of complementary geometric programming

A geometric programming (GP) is an optimization problem of the form

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.}: \quad & f_i(\boldsymbol{x}) \leq 1, \ i = 1, 2, ..., I \\ & g_j(\boldsymbol{x}) = 1, \ j = 1, 2, ..., J, \end{aligned} \tag{31}$$

where $\boldsymbol{x} = [x_1, , ..., x_N]$ is a non-negative vector of optimization variables, $g_j(\boldsymbol{x}) = c_i \prod_{n=1}^{N} x_n^{b_{i,n}}$ for all j are monomial functions, and $f_i(\boldsymbol{x}) = \sum_{k=1}^{K_j} c_{j,k} \prod_{n=1}^{N} x_n^{b_{j,k,n}}$ are posynomial functions for $i = 0, ..., I$, where coefficients are positive (i.e., $c_i, c_{j,k} > 0$) and $b_{i,n}, b_{j,k,n} \in \boldsymbol{R}$.

There is a class of optimization problems called complementary geometric programming (CGP), which potentially looks like an extension of GP. In particular, a CGP is presented as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & P_0(\boldsymbol{x}) \\ \text{s.t.}: \quad & P_i(\boldsymbol{x}) \leq 1, \ i = 1, ..., I, \\ & Q_j(\boldsymbol{x}) = 1, \ j = 1, ..., J, \end{aligned} \tag{32}$$

where $P_0(\boldsymbol{x})$ is a posynomial and $P_i(\boldsymbol{x}) = \frac{p_i(\boldsymbol{x})}{p_i^+(\boldsymbol{x})}$, in which $p_i(\boldsymbol{x})$ and $p_i^+(\boldsymbol{x})$ are posynomials. Moreover, $Q_j(\boldsymbol{x}) = \frac{q_j(\boldsymbol{x})}{q_j^+(\boldsymbol{x})}$, in which $q_j(\boldsymbol{x})$ are monomials and $q_j^+(\boldsymbol{x})$ are posynomials. By approximating $p_i^+(\boldsymbol{x})$ for all i and $q_i^+(\boldsymbol{x})$ for all j with monomials, a CGP can be turned into a standard form of GP. Let $p_i^+(\boldsymbol{x}) = \sum_{k=1}^{K_i} h_{i,k}^p(\boldsymbol{x})$ and $q_j^+(\boldsymbol{x}) = \sum_{k=1}^{K_j} h_{j,k}^q(\boldsymbol{x})$, where $h_{i,k}^p$ and $h_{j,k}^q$ are monomials. Using AGMA, at iteration l, $p_i^+(\boldsymbol{x})$ and $q_j^+(\boldsymbol{x})$ can be approximated as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

$$\widetilde{p}_i^{+}(\boldsymbol{x}(l)) = \prod_{k=1}^{K_i} \left(\frac{h_{i,k}^{p}(\boldsymbol{x}(l))}{\zeta_{i,k}^{p}(\boldsymbol{x}(l))} \right)^{\zeta_{i,k}^{p}(\boldsymbol{x}(l))}, \tag{33}$$

$$\widetilde{q}_j^{+}(\boldsymbol{x}(l)) = \prod_{k=1}^{K_j} \left(\frac{h_{j,k}^{q}(\boldsymbol{x}(l))}{\zeta_{j,k}^{q}(\boldsymbol{x}(l))} \right)^{\zeta_{j,k}^{q}(\boldsymbol{x}(l))}, \tag{34}$$

The parameters $\zeta_{i,k}^{p}(\boldsymbol{x}(l))$ and $\zeta_{j,k}^{q}(\boldsymbol{x}(l))$ can be computed as

$$\zeta_{i,k}^{p}(\boldsymbol{x}(l)) = \frac{h_{i,k}^{p}((\boldsymbol{x}(l-1))}{p_i^{+}((\boldsymbol{x}(l-1))}, \qquad \forall i,\ k, \tag{35}$$

$$\zeta_{j,k}^{q}(\boldsymbol{x}(l)) = \frac{h_{j,k}^{q}((\boldsymbol{x}(l-1))}{q_j^{+}((\boldsymbol{x}(l-1))}, \qquad \forall k,\ j, \tag{36}$$

where $\boldsymbol{x}(l-1)$ is the last-round solution of the optimization problem. It is proved that AGMA gives the best local monomial approximation for a posynomial function [29].

B. Proof of Theorem 1

Proof by contradiction: Let assume that

$$\exists n' : x_{n'}^{*} = 1 \text{ and } \ \vartheta_{n'} < \gamma^{*}. \tag{37}$$

Also, let $\boldsymbol{x'}$ be a suboptimal solution for the optimization problem in which $x'_{d_s} = x^{*}_{d_s},\ \forall d_s \in \mathcal{D}_k$ except for $d_s = n'$ and $\text{argmax}(\theta_{d_s})$ for all $d_s \in \mathcal{N}_{\text{cs}}$. Therefore, denoting $\hat{n} = \text{argmax}(\theta_{d_s})$ for all $d_s \in \mathcal{N}_{\text{cs}}$, it is clear that $x'_{n'} = 0$ and $x'_{\hat{n}} = 1$, while $x^{*}_{n'} = 1$ and $x^{*}_{\hat{n}} = 0$. Thus, we have $\boldsymbol{X^{*}\Theta} < \boldsymbol{X'\Theta}$, since $\vartheta_{n'} < \vartheta_{\hat{n}} = \gamma^{*}$ according to (37). On the other hand, RA throughput is dependent on the length of the RA regime and the number of devices competing in this regime. Since these two parameters are the same in the optimal and suboptimal solutions, the RA throughput stays the same for these solutions. In other words, the same throughput can be achieved by setting $y'_{n'} = y^{*}_{\hat{n}}$. This means that the total throughput of $(\boldsymbol{X'}, \boldsymbol{Y'})$ is larger than $(\boldsymbol{X}^{*}, \boldsymbol{Y}^{*})$, which contradicts the fact that $(\boldsymbol{X}^{*}, \boldsymbol{Y}^{*})$ is the optimal solution.

C. Proof of Lemma 3

The regret incurred by playing the suboptimal arm m over T is

$$R_m(T) = \sum_{t=1}^{T} \mathbf{1}\{\varphi_m(t) > \gamma\} \Delta_m. \tag{38}$$

To present an upper bound for $R_m(T)$, let us first decompose the event $\mathcal{A}_m = \{\varphi_m(t) > \gamma\}$ into two complementary sub-events $\mathcal{B}_m = \{\varphi_m(t) > \gamma\ , \hat{\mu}_m(t) > \mu_m^{-}\}$ and $\{\mathcal{C}_m = \varphi_m(t) > \gamma\ , \hat{\mu}_m \leq \mu_m^{-}\}$, where $\hat{\mu}_m$ is the empirical mean of arm m. Thus, $R_m(T)$ can be found as

$$R_m(T) = \sum_{t=1}^{T} \mathbf{1}\{\mathcal{B}_m\} \Delta_m + \sum_{t=1}^{T} \mathbf{1}\{\mathcal{C}_m\} \Delta_m \tag{39}$$

Here, we calculate the regret bounds for both $\mathcal{B}_m$ and $\mathcal{C}_m$. $\sum_{t=1}^{T} \mathbf{1}\{\mathcal{B}_m\}$ can be bounded as

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{B}_m\} \leq \sum_{t=1}^{T} \mathbf{1}\{\hat{\mu}_m(t) > \mu_m^{-}\} \tag{40}$$

According to [22], we have

$$\sum_{t=1}^{T} \mathbf{1}\{\hat{\mu}_m(t) > \mu_m^{-}\} \leq 1 + \frac{1}{d(\mu_m, \mu_m^{-})} = \mathcal{O}\left(\frac{1}{\delta^2}\right) \tag{41}$$

As a result, from (40) and (41), we have

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{B}_m\} \leq \mathcal{O}\left(\frac{1}{\delta^2}\right) \tag{42}$$

To calculate $\sum_{t=1}^{T} \mathbf{1}\{\mathcal{C}_m\}$, we first define $N_m^{\text{suf}}(T) = \log(T)/d(\mu_m^{+}, \gamma^{-})$, which intuitively is the sufficient number of explorations to make sure that arm m is not worth playing. Then, we continue by decomposing $\mathcal{C}_m$ into two complementary sub-events, $\mathcal{D}_m = \{\varphi_m(t) > \gamma\ , \hat{\mu}_m \leq \mu_m^{-}, N_m(t) \leq N_m^{\text{suf}}(T)\}$ and $\mathcal{E}_m = \{\varphi_m(t) > \gamma\ , \hat{\mu}_m \leq \mu_m^{-}, N_m(t) > N_m^{\text{suf}}(T)\}$, where $N_m(t)$ represents the number times arm m has been played until round t. Consequently,

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{C}_m\} = \sum_{t=1}^{T} \mathbf{1}\{\mathcal{D}_m\} + \sum_{t=1}^{T} \mathbf{1}\{\mathcal{E}_m\} \tag{43}$$

Simply, $\sum_{t=1}^{T} \mathbf{1}\{\mathcal{D}_m\}$ can be upper bounded by

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{D}_m\} \leq N_m^{\text{suf}}(T) = \frac{\log(T)}{d(\mu_m^{+}, \gamma^{-})} \tag{44}$$

Then, for $\sum_{t=1}^{T} \mathbf{1}\{\mathcal{E}_m\}$, since $N_m(t) > N_m^{\text{suf}}(T)$, we have

$$\sum\nolimits_{t=1}^{T} \mathbf{1}\{\mathcal{E}_m\} \leq \sum_{t=1}^{T} \sum_{n=N_m^{\text{suf}}(T)+1}^{T} \Pr\{\varphi_m(t) > \gamma \ | \hat{\mu}_m(t) \leq \mu_m^{-}, N_m(t) = n\} \tag{45}$$

When $N_m(t) = n$, ϕ_m is sampled from

$$\text{beta}(\hat{\mu}_m(t)(N_m^{\text{suf}}(T)+1), (1-\hat{\mu}_m(t))(N_m^{\text{suf}}(T)+1)).$$

Considering this fact, based on the Chernoff-Hoeffding inequality bound, it has been proved that

$$\Pr\{\varphi_m(t) > \frac{\gamma}{1-\psi_m)} \ | \hat{\mu}_m(t) \leq \mu_m^{-}, N_m(t) = n\} \leq e^{-d(\frac{\gamma}{1-\psi_m}, \mu_m^{-})n} \tag{46}$$

Consequently,

$$\sum\nolimits_{t=1}^{T} \mathbf{1}\{\mathcal{E}_m\} \leq \sum_{t=1}^{T} \sum_{n=N_m^{\text{suf}}(T)+1}^{T} e^{-d(\frac{\gamma}{1-\psi_m}, \mu_m^{-})n}$$

By Chernoff bound and Pinsker's inequality, it can be shown that $\sum_{n=N_m^{\text{suf}}(T)+1}^{T} e^{-d(\gamma, \mu_m^{-})n}$ is an order of $\mathcal{O}(1/T)$. Subse-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2913413, IEEE Transactions on Communications

quently,

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{E}_m\} = \sum_{t=1}^{T} \mathcal{O}(1/T) = \mathcal{O}(1). \tag{47}$$

Finally, considering (42), (44) and (47), it can be concluded that

$$R_m(T) = \\ (\sum_{t=1}^{T} \mathbf{1}\{\mathcal{B}_m\} + \sum_{t=1}^{T} \mathbf{1}\{\mathcal{D}_m\} + \sum_{t=1}^{T} \mathbf{1}\{\mathcal{E}_m\})\Delta_m \leq \\ \frac{\Delta_m \log(T)}{d(\mu_m^+, \gamma^-)} + \mathcal{O}\left(\frac{1}{\delta^2}\right). \tag{48}$$