

Learning-based Hybrid TDMA-CSMA MAC Protocol for Virtualized 802.11 WLANs

Atoosa Dalili Shoaie*, Mahsa Derakhshani[†], Saeedeh Parsaeefard*, Tho Le-Ngoc*

*Department of Electrical & Computer Engineering, McGill University, Montreal, QC, Canada

[†]Department of Electrical & Computer Engineering, University of Toronto, Toronto, ON, Canada

Email: atoosa.dalilishoaie@mail.mcgill.ca; mahsa.derakhshani@utoronto.ca;

saeideh.parsaeifard@mcgill.ca; tho.le-ngoc@mcgill.ca

Abstract—This paper presents an adaptive hybrid TDMA-CSMA MAC protocol to improve network performance and isolation among service providers (SPs) in a virtualized 802.11 network. Aiming to increase network efficiency, wireless virtualization provides the means to slice available resources among different SPs, with an urge to keep different slices isolated. Hybrid TDMA-CSMA can be a proper MAC candidate in such scenario benefiting from both the TDMA isolation power and the CSMA opportunistic nature. In this paper, we propose a dynamic MAC that schedules high-traffic users in the TDMA phase with variable size to be determined. Then, the rest of active users compete to access the channel through CSMA. The objective is to search for a scheduling that maximizes the expected sum throughput subject to SP reservations. In the absence of arrival traffic statistics, this scheduling is modeled as a multi-armed bandit (MAB) problem, in which each arm corresponds to a possible scheduling. Due to the dependency between the arms, existing policies are not directly applicable in this problem. Thus, we present an index-based policy where we update and decide based on learning indexes assigned to each user instead of each arm. To update the indexes, in addition to TDMA information, observations from CSMA phase are used, which adds a new exploration phase for the proposed MAB problem. Throughput and isolation performance of the proposed self-exploration-aided index-based policy (SIP) are evaluated by numerical results.

I. INTRODUCTION

In the next-generation wireless networks, improving resource utilization across the network is essential due to the tremendous growth in wireless traffic and services. One of the promising solutions is wireless network virtualization, which enables sharing physical resources among different service providers (SPs). Virtualizing resources into different slices, each for a SP, would result in higher utilization and lower implementation costs [1]–[5].

In 802.11-based WLANs, administrative virtualization is well established where a single physical access point (AP) can advertise multiple service set identifiers (SSID), and thus can be shared by different SPs (also referred to as slices). Although such administrative virtualization can differentiate flows from different SPs, it cannot further ensure isolation among SPs. However, the success of wireless virtualization implementation depends on how well SPs can be kept isolated from each other. In other words, it should be guaranteed that any change in one slice due to user activities (e.g., mobility of users and varying channel conditions) could rarely impact the resource utilization of other slices [6].

To provide isolation among SPs, the quality-of-service (QoS) requirements of each slice should be provisioned under any condition. One approach is a strict isolation where the deterministic resource allocation techniques are applied for the sake of QoS provisioning. For instance, different slices can completely be kept isolated by applying a time division multiple access (TDMA)-based scheme in which an exclusive timeshare can be reserved for each slice based on its requirement. However, such allocation could lead to underutilization since the reserved timeshare of a slice might partly be left unused in the presence of inactive users of each slice. This is in contrast with the wireless network provider’s goal to maximize utilization of resources by keeping its resources as busy as possible.

On the other hand, random access protocols such as carrier sense multiple access with collision avoidance (CSMA/CA) have more dynamic and opportunistic nature and thus capable of adaptively managing the timeshare of each slice according to the number of active users. Therefore, they can overcome the utilization inefficiency. But, they suffer from isolation issues because of unavoidable collisions, which couple flows of different slices. Using such protocols, the total time used by each slice should be defined as time spent for both successful and unsuccessful transmissions. Therefore, each slice would affect the performance of other slices by increasing the collision time [7]. These two major practical issues with TDMA and CSMA-based protocols call for developing novel multiple access protocol to ensure QoS provisioning for each slice in addition to increasing resource utilization efficiency which is the main goal of this paper.

In order to achieve the mentioned goals, we propose a hybrid TDMA-CSMA MAC protocol. In the proposed MAC, AP schedules high-traffic users from different slices in the TDMA phase (with variable length to be optimized), while providing the transmission chance for low-traffic users in a CSMA phase. Thus, this scheme could benefit from both the TDMA isolation power and the CSMA opportunistic nature.

In the context of conventional wireless networks, the idea of using hybrid TDMA-CSMA MAC has been studied. For instance, in [8], a hybrid TDMA-CSMA approach is proposed in which only request packets are transmitted during the CSMA phase. Then, the coordinator allocates time-slots in the TDMA phase to the users based on the received request packets.

In other words, CSMA phase is only used for information gathering, which can lead to throughput loss. Another problem with this approach is that it adds delay since data packets can be transmitted only after delivery of request packets. In another work [9], data packets can also be transmitted in the CSMA phase with piggyback requests. Although this approach suffers less from overheads, it needs much modification in the packet frame structure.

Different from [8], [9], in this paper, we propose an adaptive hybrid TDMA-CSMA in which both TDMA and CSMA phases can be used for data transmission. Also, we consider a *virtualized* network where the predetermined QoS of each slice should be met instead of QoS per user as in conventional networks. The main contribution of this paper is to design a scheduling algorithm for the hybrid MAC when *arrival traffic statistics* are *unknown*, considering *per-slice reservations*.

Assuming that AP is not aware of traffic profile of each user, we formulate the scheduling problem as a multi-armed bandit (MAB) model, in which each arm corresponds to a possible scheduling. Due to the dependency between the arms, existing policies are not directly applicable in this problem. Thus, we present an index-based policy where we update and decide based on learning indexes assigned to each user instead of each arm. To update the indexes, in addition to TDMA information, observations from CSMA phase are used, which adds a new exploration phase for the proposed MAB problem.

The remainder of this paper is organized as follows. We first introduce our system model in Section II. Section III presents the problem formulation. Subsequently, assuming that user traffic statistics are unknown, an iterative index-based policy is proposed to learn the optimal solution with low complexity in Section IV. Section V presents the numerical results. Finally, we conclude the paper in Section VI.

II. SYSTEM MODEL

A. Network Model and Frame Structure

We consider an IEEE 802.11-based WLAN with a single AP. The AP carries the traffic belonging to K different SPs (also referred to as slices). The set of all slices is denoted by \mathcal{K} . Also, the set of subscribed users to the slice k is denoted by \mathcal{N}_k , where $|\mathcal{N}_k| = N_k$ is the number of users at slice k . Thus, the total number of all users over all slices is equal to N_u where $N_u = \sum_{k=1}^K N_k$.

Time is divided into fixed-length superframes, which are indexed by t . Each superframe consists of three phases: beacon transmission, TDMA, and CSMA. As shown in Figure 1, the superframe begins with a beacon frame issued by AP to the users. Following the beacon transmission, TDMA phase starts during which transmissions can happen by scheduled users. In each superframe, $D(t)$ time-slots are allocated to the TDMA phase. Let D_{\max} be the maximum allowable length of the TDMA phase. The CSMA phase follows the TDMA phase. Assuming that the total length of a superframe is T_f time-slots excluding the beacon frame, the length of CSMA phase is equal to $T_f - D(t)$ time-slots. During this phase, CSMA protocol runs with each time-slot divided into T_s backoff units.

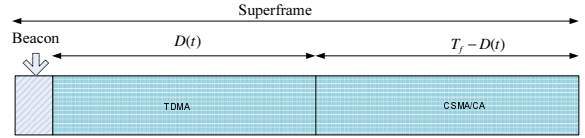


Fig. 1: Superframe structure of the proposed MAC for a virtualized 802.11 WLAN

Regarding the required QoS per slice, it is assumed that each SP k can reserve r_k time-slots per superframe.

We assume that all the data packets have the same size and one data packet can be sent in one time-slot. The state of each user (e.g., user n_k from slice k) at superframe t is denoted by $s_{n_k}(t) \in \{0, 1\}$; 0 indicates the user has no packets to transmit and 1 represents that user has a packet to transmit. $s_{n_k}(t)$ evolves as a Bernoulli random process over time with probability θ_{n_k} to have a packet for transmission. The vector to represent such probabilities for all users is denoted by $\Theta = [\Theta_k]_{\forall k}$ where $[\Theta_k] = [\theta_{n_k}]_{\forall n_k}$ which is called traffic profile and assumed to be unknown by AP.

B. User Operation

Before superframe t starts, the AP decides on time-slot allocation for TDMA phase and notifies the schedule to the users via the beacon. Users with no allocated time-slot would attempt to transmit in the CSMA phase if they have a packet, using the CSMA/CA protocol as follows.

In the CSMA/CA protocol, each active user first needs to backoff for a random number of backoff units (called backoff value), which is uniformly chosen from the range of $(0, 2^B - 1)$ where B is the backoff stage [10]. The user waits for this time and then performs the channel sensing for two backoff units in a row. If the channel is sensed to be idle, the user starts its transmission. Otherwise, B and a counter C are incremented by 1 up to B_{\max} and C_{\max} , respectively. For each packet transmission, B and C start with B_{\min} and 0, respectively. If C is still lower than C_{\max} , the user keeps trying to transmit by starting another backoff process. Otherwise, the user would drop that packet.

III. PROBLEM FORMULATION

Our goal is to develop a MAC protocol for a virtualized 802.11 WLAN, which maximizes the system utility in terms of the total number of transmitted packets. On the other hand, we aim to provide isolation between slices over each superframe as much as it is possible. In other words, we try to minimize the impacts of activities in one slice on other slices.

In order to increase the overall network throughput, we propose a hybrid TDMA-CSMA MAC protocol. In this scheme, a scheduling algorithm dynamically reserves a variable number of time-slots in each superframe based on traffic demand of each user and required reservation of each slice. Then, in the remaining time-slots of a superframe, the unscheduled users would take their chance to transmit (if they have any packet) in a contention-based phase using CSMA/CA.

This approach can benefit from strengths of both protocols.

TDMA protocol performs well for users with high probability of packet arrival and CSMA is more efficient for users with low values of θ_{n_k} . Also, compared to pure CSMA, splitting users into two groups, TDMA and CSMA, leads to lower number of users in the CSMA, which ends up to the less number of collisions and consequently higher utilization. Note that using pure TDMA leads to system under utilization due to idle left time-slots assigned to users with low traffic demand.

Here, we explain what is the framework to dynamically decide on the length of the TDMA period and specifically which users would be assigned TDMA time-slots in each superframe. A policy π is an algorithm that schedules users for the TDMA phase at the superframe t , based on the whole history of the scheduling and instantaneous user traffic demands. Then, we denote by $\mathbf{X}^\pi(t) = [x_{n_k}^\pi(t)]$ the vector representing the time-slot allocation of all users at superframe t according to π . More specifically, $x_{n_k}^\pi$ is a binary variable indicating whether a time-slot is allocated to the user n_k in the current superframe (i.e., $x_{n_k}^\pi = 1$) or not (i.e., $x_{n_k}^\pi = 0$). The instantaneous reward (in terms of number of transmitted packets) associated with the TDMA phase is

$$R_{td}(t) = \sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} s_{n_k}(t) x_{n_k}^\pi(t). \quad (1)$$

Unlike TDMA, computation of average throughput for CSMA phase is not straightforward. Generally, it is known that CSMA throughput is a function of length of CSMA phase and number of active users in this phase. Let $f(b, c)$ denote the average number of packets transmitted by CSMA/CA under the condition that the number of users participating in the CSMA period is b and the length of the CSMA period is c . Then, the instantaneous reward (in terms of number of transmitted packets) associated with the CSMA phase is

$$R_{cs}(t) = f\left(\sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} s_{n_k}(t) (1 - x_{n_k}^\pi(t)), T_f - \sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} x_{n_k}^\pi(t)\right). \quad (2)$$

It should be noted that $f(b, c)$ is calculated offline by simulation here. Now, we define the expected instantaneous reward at superframe t as

$$\begin{aligned} R_{\Theta}(\mathbf{X}^\pi(t)) &= \mathbb{E}\left(R_{td}(t) + R_{cs}(t) - \beta \sum_{k \in \mathcal{K}} \left[\sum_{n_k \in \mathcal{N}_k} x_{n_k}^\pi(t) + p_k^\pi(t) - r_k \right]^+\right) \\ &= \sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} \theta_{n_k} x_{n_k}^\pi(t) \\ &+ f\left(\sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} \theta_{n_k} (1 - x_{n_k}^\pi(t)), T_f - \sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} x_{n_k}^\pi(t)\right) \\ &- \beta \sum_{k \in \mathcal{K}} \left[\sum_{n_k \in \mathcal{N}_k} x_{n_k}^\pi(t) + p_k^\pi(t) - r_k \right]^+, \end{aligned} \quad (3)$$

which consists of three components. The first two terms represent the numbers of packets transmitted during the TDMA and CSMA phases. Moreover, in order to provide isolation, the third term is included, which works as a pricing mechanism to avoid scheduling a user whose slice has already been

served by a sufficient number of time-slots according to its reservation. β is a positive scalar and represents a pricing factor. Furthermore, $[y]^+ = \max\{0, y\}$. Moreover, $p_k^\pi(t)$ denotes the time spent by slice k during CSMA phase at superframe t . Proportional to the ratio of the CSMA traffic load of slice k (i.e., $\sum_{n_k \in \mathcal{N}_k} \theta_{n_k} (1 - x_{n_k}^\pi(t))$) to the total CSMA throughput, we approximately calculate the share of slice k in the total CSMA time as

$$p_k^\pi(t) = \frac{\left(T_f - \sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} x_{n_k}^\pi(t)\right) \times \sum_{n_k \in \mathcal{N}_k} \theta_{n_k} (1 - x_{n_k}^\pi(t))}{f\left(\sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} \theta_{n_k} (1 - x_{n_k}^\pi(t)), T_f - \sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} x_{n_k}^\pi(t)\right)}. \quad (4)$$

The objective is to find a policy π that chooses $\mathbf{X}^\pi(t)$, $t = 1, \dots, T$, which maximizes the total expected accumulated reward over a time horizon T . This problem can be expressed as

$$\max_{\pi} \sum_{t=1}^T R_{\Theta}(\mathbf{X}^\pi(t)) \quad (5a)$$

$$\text{s.t. : } \sum_{k \in \mathcal{K}} \sum_{n_k \in \mathcal{N}_k} x_{n_k}^\pi(t) \leq D_{\max}, \quad t = 1, \dots, T. \quad (5b)$$

Assuming that AP is not a priori aware of Θ , a learning policy is required to solve the optimization problem in (5). Such algorithm enables sequential estimation of the traffic profile based on the observations of the instantaneous rewards. Let $\hat{\Theta}(t)$ be the estimate of Θ at superframe t . To learn Θ , we model the proposed problem in (5) as the multi-armed bandit (MAB) and propose an index-based policy to address the exploration versus exploitation tradeoff and reach to the optimal solution over time.

IV. POLICY DESIGN

Here, first, we briefly review MAB problem and the existing policy designs proposed for it. Then, we propose an index-based policy customized for the proposed problem (5) where a new exploration phase is added to benefit from CSMA observations in addition to the acquired information from TDMA phase.

A. Mathematical Background

Generally, an infinite horizon non-Bayesian multi-armed bandit (MAB) is applied to solve sequential decision making problems with the aim to make a selection among multiple choices, each leading to stochastic rewards with partial knowledge of the system [11]. For the proposed problem of this paper, the decision-maker is AP aiming to schedule users for TDMA transmission with unknown Θ . Following the MAB terminology, each possible TDMA scheduling corresponds to one arm. At each superframe t , one arm is selected. Via this approach, the achieved rewards for $t = 1, \dots, T$ are i.i.d. random variables with unknown means. The objective of MAB problem is to select the arms such that the accumulated expected reward is maximized over time.

The expected reward of each arm is estimated, based on its instantaneous reward observations. The accuracy of this estimation directly depends on the number of times that each arm can be selected. Clearly, for a sufficiently large number of choosing each arm, more precise estimation is obtained. This process to estimate the reward of each arm is referred to as the exploration, which inherently time-consuming and the total expected reward is not maximized. In contrast, when the arms with higher expected rewards are frequently pulled, the expected accumulated reward is increased. This is called exploitation of known arms (i.e., maximizing the rewards). Consequently, there is always a trade-off between the exploration and exploitation.

If the expected reward of each arm is known a priori, the optimal action is to choose the arm with the highest expected reward. For the case of unknown reward, the main question is what would be the best policy to choose an arm. A metric to evaluate a policy is *regret*, which is the difference between the expected reward obtained by always choosing the optimal arm, and that obtained by the selected policy.

In [12], it is shown that the minimum regret grows at a logarithmic order under certain regularity conditions. Regarding the policy design, [13] has developed an index-based arm selection approach, called the upper confidence bounds (UCB). In UCB, the arm f^* would be chosen if

$$f^* = \max_{f=1, \dots, F} \hat{\phi}_f(t) + \sqrt{\frac{2 \ln(t)}{M_f(t)}},$$

where F is the total number of arms, $\hat{\phi}_f$ is the observed reward of arm f , and M_f is the number of times that arm f has been played. The main drawback of UCB is that it ignores the underlying dependencies among arms. Thus, its storage and regret scale linearly with the number of arms. However, in the problems with dependent arms (e.g., proposed problem of this paper), the correlation among arms needs to be taken into account, so that the required storage and regret can be reduced [14].

This motivates us to present a policy in which observations are efficiently stored per user rather than per scheduling. Since an arbitrary combination of users makes a scheduling strategy, this allows exploitation of information stored per user to make decisions about scheduling strategies. In this way, we can reduce the required storage from $\mathcal{O}(2^{N_u})$ to $\mathcal{O}(N_u)$ and improve the performance in terms of regret accordingly.

B. Proposed Self-exploration-aided Index-based Policy (SIP)

Here, we present an index-based policy design customized for the proposed problem in this paper. In this design, we update and decide, based on learning indexes assigned to each user instead of each arm. To update the indexes, in addition to TDMA information, observations from CSMA phase are used, which adds a new exploration phase for the proposed MAB problem. The proposed policy design consists of two phases, *coarse-grained* and *fine-grained* scheduling, described in the following.

Algorithm 1 Coarse-grained scheduling

Input: $\hat{\Theta}$
 $\rho \leftarrow 0$, $\mathbf{X} = [x_{n_k}] \leftarrow \mathbf{0}$, $\mathbf{\Omega} = [\omega_k]_{k=1}^K \leftarrow \mathbf{1}$
while $\mathbf{\Omega}^\top > 0$ and $\mathbf{X}^\top \leq D_{\max}$ **do**
 $n_k \leftarrow \operatorname{argmax}_{\omega_k \neq 0} \hat{\Theta}$
 $x_{n_k} \leftarrow 1$
 if $R_{\hat{\Theta}}(\mathbf{X}) > \rho$ **then**
 $\rho \leftarrow R_{\hat{\Theta}}(\mathbf{X})$
 else
 $\omega_k \leftarrow 0$
 $x_{n_k} \leftarrow 0$
 end if
end while
Output: $\mathbf{D} = [D_k]_{k=1}^K$, where $D_k = \sum_{n_k \in \mathcal{N}_k} x_{n_k}$, $\forall k \in \mathcal{N}_k$.

1) *Coarse-grained scheduling:* In each superframe, first, the TDMA share of each slice is determined in terms of number of time-slots. Nevertheless, the decision on which specific users would be scheduled for the TDMA phase is postponed to the fine-grained scheduling. The key objective of this coarse-grained scheduling is to ensure isolation among different slices, while optimizing the overall throughput. Thus, at the beginning of superframe t , this can be accomplished by looking for the arm that maximizes the reward $R_{\hat{\Theta}}(\mathbf{X}^\pi(t))$ given $\hat{\Theta}(t-1)$.

In order to solve this maximization problem, exhaustive search is a prohibitively expensive solution since the total number of available arms is equal to $\sum_{j=0}^{D_{\max}} C(N_u, j)$, where $C(N_u, j)$ denotes the number of j combinations of N_u elements. We therefore propose a subroutine presented in Algorithm 1 to solve the relevant combinatorial optimization.

In this algorithm, generally, it is desired to allocate TDMA time-slots to high-traffic users, aiming to increase throughput. Nevertheless, since throughput is not the only consideration in this problem, we have to consider slice reservations, which has been reflected in the reward function. To this end, in each iteration, Algorithm 1 assigns one time-slot to the user n_k having the highest $\hat{\theta}_{n_k}$ among the users with no assigned time-slot. However, the slot is allocated only if it leads to a reward increment. Otherwise, no more users from slice k (including n_k) would be allocated a time-slot in the current and subsequent iterations.

The procedure continues as long as reward increment can be obtained. The output of this algorithm is represented by $\mathbf{D} = [D_k]_{k=1}^K$, where D_k is the number of time-slots allocated to slice k in the TDMA phase.

2) *Fine-grained scheduling:* The role of fine-grained scheduling is to pick users for TDMA access by considering the TDMA shares of different slices (which is already determined by coarse-grained scheduling).

To this end, first, an index is assigned to each user to facilitate the decision making. The larger is the value of an index the higher is the chance of choosing the corresponding user. Thus, in the proposed algorithm, we simply choose $D_k(t)$ users from slice k with the largest indexes.

Algorithm 2 Fine-grained scheduling

INITIALIZATION

Schedule each user at least once.

Update $\hat{\theta}_{n_k}$ & $M_{n_k}^T$ based on (7) & (8), $\forall n_k \in \mathcal{N}_k, \forall k \in \mathcal{K}$.

MAIN LOOP

for $t = \lceil N_u/T_f \rceil + 1 : T$ **do**Obtain $\mathbf{D}(t) = [D_k]$ using Algorithm 1 with $\hat{\Theta}(t-1)$ Compute the indexes of all users (i.e., $\mathbf{I}(t)$) based on (6).**for** $k = 1 : K$ **do** $\mathbf{Z} = [z_j]_{j=1}^{N_k} \leftarrow \text{sort } \mathbf{I}_k(t) = [I_{n_k}]$ descendingly.**for** $i = 1 : D_k(t)$ **do**Set ν_k as the corresponding user to z_i . $x_{\nu_k}^\pi(t) \leftarrow 1$.**end for****end for**Run TDMA and CSMA phases based on $\mathbf{X}^\pi(t)$ Update $\hat{\Theta}(t)$, $\mathbf{M}^T(t)$, $\mathbf{M}^S(t)$, and $\mathbf{M}^C(t)$ based on (7), (8), (9), and (10).**end for**

To construct indexes, aiming to reflect the tradeoff between exploitation and exploration, we define the index for the user n_k as

$$I_{n_k}(t+1) = \hat{\theta}_{n_k}(t) + \sqrt{\frac{M_{n_k}^S(t)}{M_{n_k}^C(t)}} \times \sqrt{\frac{\ln(t)}{M_{n_k}^T(t)}}. \quad (6)$$

In (6), the first term (which is associated with exploitation) is the estimated value of traffic profile of user n_k . The second term promotes exploration, which is built in two factors. The first factor, $\sqrt{M_{n_k}^S(t)/M_{n_k}^C(t)}$, indicates the ratio of the number of times that user n_k has successfully transmitted in the CSMA phase (i.e., $M_{n_k}^S(t)$) to the number of times that user n_k is not assigned times slot (i.e., $M_{n_k}^C(t)$). This term is multiplied by $\sqrt{\ln(t)/M_{n_k}^T(t)}$, where $M_{n_k}^T(t)$ denotes the number of times that user n_k has been chosen for TDMA phase. Using CSMA observations, we give a higher weight to a user index if it shows a successful CSMA experience. Since a high value of $M_{n_k}^S(t)/M_{n_k}^C(t)$ can be interpreted as being a high-traffic user.

At the end of each superframe, based on observations from both TDMA and CSMA phases, the values of $\hat{\Theta}$, \mathbf{M}^T , \mathbf{M}^C and \mathbf{M}^S should be updated. There are two ways to update values for $\hat{\Theta}$: using information from the TDMA phase and/or the CSMA phase. Using information from TDMA phase is more straightforward and leads to the exact values during the long run. However, this is not true for CSMA phase since all users who have packets to transmit cannot succeed during this phase. Therefore, observations can be considered as imperfect/noisy. Thus, we use only the information from TDMA phase to update the traffic profile estimate.

More specifically, at the end of superframe t , we observe the user states (i.e., $S_{n_k}(t)$) for the users who are scheduled for TDMA access (i.e., $x_{n_k}^\pi = 1$). Then, values of $\hat{\Theta}$ and \mathbf{M}^T

are updated as

$$\hat{\theta}_{n_k}(t+1) = \begin{cases} \frac{\hat{\theta}_{n_k}(t)M_{n_k}^T(t)+S_{n_k}(t)}{M_{n_k}^T(t)+1} & \text{if } x_{n_k}^\pi = 1, \\ \hat{\theta}_{n_k}(t) & \text{Otherwise,} \end{cases} \quad (7)$$

and

$$M_{n_k}^T(t+1) = \begin{cases} M_{n_k}^T(t) + 1 & \text{if } x_{n_k}^\pi = 1, \\ M_{n_k}^T(t) & \text{Otherwise.} \end{cases} \quad (8)$$

Furthermore, based on the observations from the CSMA phase, \mathbf{M}^S and \mathbf{M}^C are updated as, for all n_k ,

$$M_{n_k}^S(t+1) = \begin{cases} M_{n_k}^S(t) + 1 & \text{For successful CSMA} \\ & \text{transmission of } n_k, \\ M_{n_k}^S(t) & \text{Otherwise,} \end{cases} \quad (9)$$

and

$$M_{n_k}^C(t+1) = \begin{cases} M_{n_k}^C(t) + 1 & \text{if } x_{n_k}^\pi = 0, \\ M_{n_k}^C(t) & \text{Otherwise.} \end{cases} \quad (10)$$

V. SIMULATION RESULTS

We use MATLAB simulation to evaluate the performance of the proposed policy for different system parameters. We consider a network with 2 slices versus different system parameters. The superframe length, T_f , is set to 16 time-slots where each time-slot consists of 12 backoff units. Each slice has reservation of 8 time-slots (i.e., $r_k = 8$) and $\beta = 2$. CSMA parameters are set as following: $B_{\min} = 3$, $B_{\max} = 5$, and $C_{\max} = 4$. In the first scenario, we plot the results for two cases of packet arrival probability and two different values of $D_{\max} = \{8, 12\}$.

To study the regret of learning algorithm, we consider two cases based on the traffic profiles of users of slices: 1) Case 1 or homogeneous slices where $\Theta_1 = \Theta_2 = \{[0.9]_4, [0.5]_8, 0.4\}$ in which $[x]_y$ is the vector $1 \times y$ with entries x . For this case, $N_1 = N_2 = 13$ and $N_u^{\text{Case 1}} = 26$; 2) Case 2 or heterogeneous slices where $\Theta_1 = \{[0.95]_2, [0.9]_2, [0.85]_2, [0.8]_2, [0.5]_2\}$ and $\Theta_2 = \{[0.6]_4, [0.5]_4, [0.4]_4, [0.3]_4, [0.2]_4\}$ in which $N_1 = 10$ and $N_2 = 20$ and $N_u^{\text{Case 2}} = 30$.

Figure 2 shows that the regret of Case 1 is lower than the regret of Case 2. Also, with decreasing D_{\max} , the regret is decreased. The main reason is with decreasing both N_u and D_{\max} , the number of arms is decreased, leading to higher performance of the proposed learning algorithm.

To show how increasing the number of users of one slice affects on the performance of users of the other slice, we study the total network throughput and isolation factor in Figures 3 and 4. Total throughput is the average number of packets transmitted from both slices for each superframe. For the isolation factor, we apply Jain's fairness index on packet delivery ratios of different slices, which is defined as

$$\rho_k(t) = \frac{\min((\mathbf{S}_k(t)\mathbf{X}_k(t))^T + \mathbf{M}_k^S(t))}{\min(\mathbf{S}_k\mathbf{1}^T, r_k), 1}, \quad (11)$$

where $\mathbf{S}_k(t)\mathbf{X}_k(t)^T + \mathbf{M}_k^S(t)$ represents the number of transmitted packets in both TDMA and CSMA phases for slice k ,

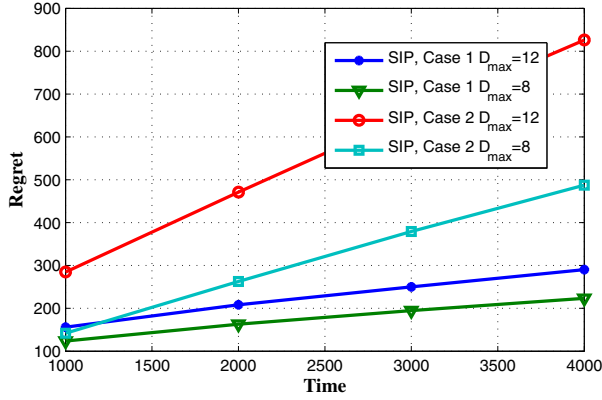


Fig. 2: Regret versus time

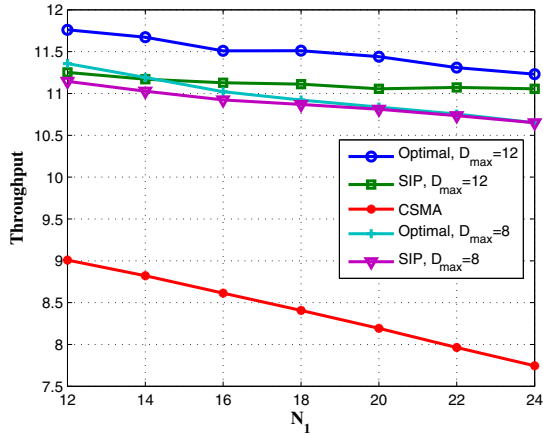


Fig. 3: Total throughput versus N_1

and $\theta_k \mathbf{1}^T$ denotes the number of generated packets from users belonging to slice k at superframe t . Note that we limit the maximum number of generated packets of SP k that can be potentially served, to its reservation r_k . This is the reason that $\min(\mathbf{S}_k \mathbf{1}^T, r_k)$ is used in the denominator in (11).

For these simulations, when $N_1 = N_2 = 12$, $\Theta_1 = \Theta_2 = \{[0.9]_2, [0.8]_2, [0.7]_2, [0.6]_2, [0.5]_4\}$. For $N_1 > 12$, the traffic profile of each new user is 0.5. From Figures 3 and 4, the throughput and isolation factor of CSMA are decreased with increasing the number of users (i.e., overloaded network), due to the increasing number of collisions. However, the performance of SIP in terms of both throughput and isolation are close to the optimal solution. Also, SIP holds the isolation factors close to 1. From these two figures, we can conclude that the proposed hybrid TDMA-CSMA algorithm maintains isolation of slices while increasing the system throughput.

VI. CONCLUSIONS

In this paper, we have proposed an adaptive hybrid TDMA-CSMA protocol to balance the tradeoff between throughput and slice isolation in a virtualized WLAN. When the traffic arrival statistics of users are unknown, a reinforcement learning algorithm has been applied to dynamically decide

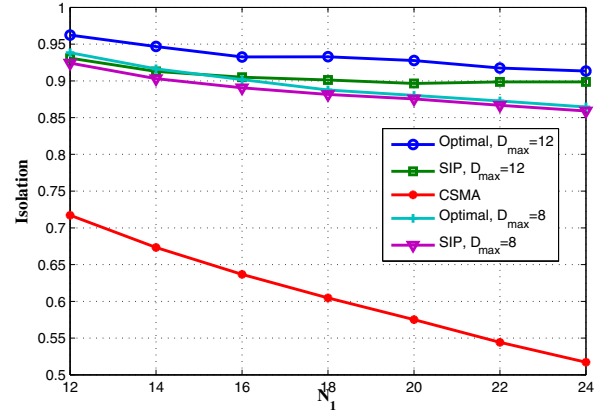


Fig. 4: Isolation factor versus N_1

on how many, from which slice, and which users should be scheduled for TDMA access. The proposed algorithm directs high-traffic users from different slices to the TDMA phase (with variable length) by considering slice reservations, while providing the transmission chance for low-traffic users over CSMA/CA. Simulation results show the effectiveness of the proposed policy to keep the isolation among slices.

REFERENCES

- [1] C. Liang and F. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surv. Tuts.*, 2014.
- [2] A. Alexiou, "Wireless world 2020: Radio interface challenges and technology enablers," *IEEE Veh. Tech. Mag.*, vol. 9, no. 1, pp. 46–53, Mar. 2014.
- [3] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Commun. Surv. Tuts.*, vol. 16, no. 4, pp. 2181–2206, 4th Q. 2014.
- [4] H. Wen, P. K. Tiwary, and T. Le-Ngoc, *Wireless Virtualization*. Springer, 2013.
- [5] M. Rahman, C. Despins, and S. Affes, "Analysis of CAPEX and OPEX benefits of wireless access virtualization," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Jun. 2013, pp. 436–440.
- [6] R. Kokku *et al.*, "NVS: a virtualization substrate for WiMAX networks," in *Proc. ACM conf. Mobile comput. netw.*, 2010.
- [7] A. Checco and D. J. Leith, "Fair virtualization of 802.11 networks," *IEEE/ACM Trans. Netw.*, Dec. 2013.
- [8] B. Shrestha, K. W. Choi, and E. Hossain, "A dynamic time slot allocation scheme for hybrid CSMA/TDMA MAC protocol," *IEEE Wireless Commun. Lett.*, vol. 2, no. 5, pp. 535–538, Oct. 2013.
- [9] B. Shrestha, E. Hossain, and K. W. Choi, "Distributed and centralized hybrid CSMA/CA-TDMA schemes for single-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 4050–4065, Jul. 2014.
- [10] M. Derakhshani and T. Le-Ngoc, "Cognitive MAC designs: Background," in *Cognitive MAC Designs for OSA Networks*. Springer International Publishing, 2014, pp. 15–31.
- [11] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, pp. 1–122, 2012.
- [12] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied math.*, vol. 6, no. 1, pp. 4–22, 1985.
- [13] 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.
- [14] Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *IEEE Symposium on New Frontiers in Dynamic Spectrum*, Apr. 2010, pp. 1–9.