

Random Access Protocols for WLANs Based on Mechanism Design

Man Hon Cheung, A. Hamed Mohsenian-Rad, Vincent W.S. Wong, and Robert Schober

Department of Electrical and Computer Engineering

The University of British Columbia, Vancouver, Canada

E-mail: {mhcheung, hamed, vincentw, rschober}@ece.ubc.ca

Abstract—In wireless local area networks (WLANs), quality of service (QoS) can be provided by mapping applications with different requirements (e.g., delay and throughput) into one of the available access categories (ACs), as is done in the IEEE 802.11e standard. With the increasing programmability of network adapters, a malicious user can strategically declare a higher AC for its application to gain an unfair share of resources. This can drastically degrade the network performance and avoid adequate service distinction among different ACs. In this paper, we use the technique of mechanism design in game theory to tackle this problem in WLANs with random access. We propose to use the Vickrey-Clarke-Groves (VCG) mechanism in order to motivate each station to inform the access point (AP) truthfully, about the required AC of its application. The AP will then inform each station about its persistent probability and the price it needs to pay for the offered service. The result of the allocation of the persistent probabilities can be used for admission control. Simulation results show that the use of mechanism design can lead to a higher aggregate utility and prevents malicious users from gaining an unfair share of the network bandwidth.

I. INTRODUCTION

In wireless local area networks (WLANs), the medium access control (MAC) protocols are important in arbitrating the access of the shared wireless medium. There are two main types of MAC protocols in WLANs: scheduling-based (e.g., point coordination function (PCF)) and contention-based (e.g., distributed coordination function (DCF)). In this paper, we focus on the study of contention-based MACs as they are more scalable and inherently more flexible. In contention-based MACs, users *randomly* access the shared communication channel with certain *persistent probabilities*.

To provide quality of service (QoS) for different applications, the IEEE 802.11e standard [1] classifies applications into four different access categories (ACs). Packets in each AC are placed in a separate queue in the MAC layer. Differentiation in priorities among the four ACs is achieved through the use of different channel access parameters (e.g., contention window), which directly determine the persistent probability for accessing the shared channel by packets in each AC [2]. The IEEE 802.11e protocol was originally designed for a fully *cooperative* network setting, where all stations follow exactly the operations of the protocol. However, by modifying the driver of the network adapter, a user can manipulate the MAC parameters in order to gain an unfair advantage.

Game theory has been shown to be a useful tool in analyzing the selfish behaviors of users in various networking problems.

In [3], Cagalj *et al.* formulated carrier sensing multiple access with collision avoidance (CSMA/CA) using game theory. Both normal-form and repeated-form CSMA/CA games were formulated and the existence of Nash equilibria was shown for each game. In [4], Konorski proposed a game-theoretic strategy called CRISP (Cooperation via Randomized Inclination to Selfish Play) to counteract the selfish behavior of users. In [5], Cui *et al.* proposed a comprehensive framework to study non-cooperative random access games. The Nash equilibria were characterized for various settings and distributed algorithms were proposed. However, the issue of *service differentiation* among different applications was not addressed in [5].

Papers in [3]–[5] address non-cooperative random access from the players' viewpoint, where the key idea is to either propose strategies to counteract malicious users or to analyze how non-cooperative users can degrade network performance. On the other hand, we can take a *proactive* approach from the system designer's point of view and propose proper *mechanisms* to *prevent* players from misbehaving. For example, in [6], Nugehalli *et al.* proposed an incentive mechanism to avoid selfishness. They showed that under certain conditions, the users are encouraged to always be truthful on declaring their ACs. This leads to significantly higher throughput as shown in [6]. Our truthful mechanism design in this paper is closely related to the results in [6]. However, we consider a different class of utility functions and mainly focus on maximizing the aggregate utility across all users aiming to maximize network social welfare. Our proposed mechanism design is also different from that in [6] as it is indeed a special case of the Vickrey-Clarke-Groves (VCG) mechanism [7], [8]. To the best of our knowledge, this work is the first paper to study VCG mechanism design for random access networks by taking into account different ACs for different applications. In summary, the contributions of our work are as follows:

- We consider the problem of assigning persistent probabilities to rational stations in a WLAN to maximize network social welfare, i.e., the aggregate utility across all users. We introduce a class of utility functions that can mathematically model the service requirements for a wide range of network applications.
- We show that in a non-cooperative random access game with selfish users, a Nash equilibrium may not exist and the network has poor performance.

- We formulate the VCG mechanism and its corresponding pricing scheme for random access networks to enforce truthfulness and cooperation among rational users.
- We consider the computational issues in VCG and show that for random access networks, implementing VCG requires solving a complicated *non-convex* optimization problem. To tackle the non-convexity, we propose an enumeration algorithm such that each iteration only solves a *convex* optimization problem. Our algorithm is easy to implement and is guaranteed to reach the optimal solution for any choice of the system parameters.
- Simulation results show that our scheme can ensure achieving maximum aggregate network utility via mechanism design. Moreover, service differentiation and QoS in terms of throughput can be supported.

The rest of the paper is organized as follows. The system model is described in Section II. The non-cooperative random access game when the AP offers different ACs is studied in Section II-C. Our proposed mechanism, which is based on VCG, is formulated in Section III. The computational issues of VCG are discussed in Section III-B. Simulation results are given in Section IV, and the paper is concluded in Section V.

II. SYSTEM MODEL

Consider a WLAN with a single AP and N stations. The set of stations is denoted by $\mathcal{N} = \{1, 2, \dots, N\}$. In this paper, we use terms “stations” and “users” interchangeably. All stations are one-hop neighbors to the AP. We only consider the uplink scenario¹ where each station $i \in \mathcal{N}$ can access the shared medium with a persistent probability p_i . Let p_i^{succ} denote the probability of successful transmission by station i . We have

$$p_i^{\text{succ}} = \gamma_i(\mathbf{p}) = p_i \prod_{j \in \mathcal{N} \setminus \{i\}} (1 - p_j), \quad \forall i \in \mathcal{N}, \quad (1)$$

where γ_i determines the probability of successful transmission of station i when $\mathbf{p} = (p_i, i \in \mathcal{N})$ is given. Given the nominal data rate φ (e.g., 11 Mbps), the *average* data rate for user i is equal to $\varphi p_i^{\text{succ}}$. Stations are assumed to run different types of applications, each of which may have distinct QoS requirements. For the ease of exposition, we limit our study to the case where each station may only run one application at a time. The proposed framework can be extended to the case when each station runs multiple applications simultaneously.

A. Utility Function

Let $u_i(p_i^{\text{succ}})$ denote the level of satisfaction station i achieves when it experiences success probability p_i^{succ} , which itself depends on all persistent probabilities $\mathbf{p} = (p_i, i \in \mathcal{N})$. We refer to u_i as the *utility function* corresponding to the application running on station i . Clearly, utility functions depend on the QoS requirements of the running applications. For example, many voice and video applications require a *minimum* level of available bandwidth. If the available bandwidth drops below the required bandwidth, then the connections

¹We notice that malicious users may only affect the performance of uplink transmissions. In fact, when it comes to the downlink transmissions, the AP can simply perform *scheduling* with adequate QoS provisioning.

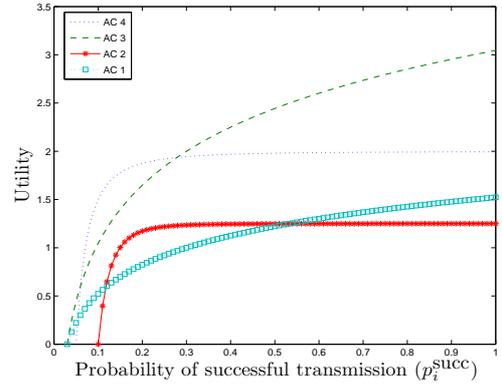


Fig. 1. Sample utility functions for four ACs using (2) and (3) with parameters: $K_1 = 1$, $\alpha_1 = 1$, $p_1^{\text{critical}} = 0.03$; $K_2 = 0.0005$, $\alpha_2 = 5$, $p_2^{\text{critical}} = 0.1$; $K_3 = 2$, $\alpha_3 = 1$, $p_3^{\text{critical}} = 0.03$; $K_4 = 0.01$, $\alpha_4 = 3$, $p_4^{\text{critical}} = 0.05$.

should be dropped, leading to zero utilities. In this paper, the utility function u_i for user $i \in \mathcal{N}$ depends on p_i^{succ} , as well as $\alpha_i > 0$, $p_i^{\text{critical}} \in (0, 1)$, and $K_i > 1$, where p_i^{critical} refers to the minimum required p_i^{succ} for the application to run properly in station i . It is used to mathematically model various applications (such as voice and video streaming) which cannot operate if the minimum required data rate is not provided. K_i and α_i determine the amplitude and curvature of the utility functions, respectively. For utility parameter $\alpha_i \neq 1$, we have

$$u_i(p_i^{\text{succ}}, \alpha_i, p_i^{\text{critical}}, K_i) = \begin{cases} K_i(1 - \alpha_i)^{-1} \left[p_i^{\text{succ}(1 - \alpha_i)} - p_i^{\text{critical}(1 - \alpha_i)} \right], & \text{if } p_i^{\text{succ}} \geq p_i^{\text{critical}}, \\ 0, & \text{if } p_i^{\text{succ}} < p_i^{\text{critical}}, \end{cases} \quad (2)$$

and for $\alpha_i = 1$, we have

$$u_i(p_i^{\text{succ}}, 1, p_i^{\text{critical}}, K_i) = \begin{cases} K_i \log \left(\frac{p_i^{\text{succ}}}{p_i^{\text{critical}}} \right), & \text{if } p_i^{\text{succ}} \geq p_i^{\text{critical}}, \\ 0, & \text{if } p_i^{\text{succ}} < p_i^{\text{critical}}. \end{cases} \quad (3)$$

When $K_i = 1$ and $p_i^{\text{critical}} = 0$ in (2) (or $p_i^{\text{critical}} = 1$ in (3)), the above utility functions reduce to the well-known α -fair utility functions [9]. Sample utility functions based on different choices of parameters are shown in Fig. 1. Notice that each utility represents a distinct AC.

B. Network Utility Maximization

Given complete knowledge of all parameters and centralized control of the network, an efficient choice of all persistent probabilities, i.e., $\mathbf{p} = (p_i, \forall i \in \mathcal{N})$, is characterized as an optimal solution of the *network utility maximization* problem:

$$\max_{\mathbf{p} \in \mathcal{P}} \sum_{i \in \mathcal{N}} u_i(\gamma_i(\mathbf{p}), \theta_i), \quad (4)$$

where

$$\theta_i = [\alpha_i, p_i^{\text{critical}}, K_i], \quad \forall i \in \mathcal{N}, \quad (5)$$

and

$$\mathcal{P} = \{\mathbf{p} : 0 \leq p_i \leq 1, \forall i \in \mathcal{N}\} \quad (6)$$

represents the set of feasible persistent probabilities. The objective function in (4) is also called *network social welfare*. We notice that since the utilities are *not* available to the AP as they are local to the stations, the AP may solve problem (4) only after each station $i \in \mathcal{N}$ declares its *type* θ_i . Clearly, if all stations are truthful, then the obtained vector of optimal persistent probabilities leads to optimal network performance. However, if a user $i \in \mathcal{N}$ is malicious, then it may declare its *type* to be $\hat{\theta}_i \neq \theta_i$. In that case, the obtained persistent probabilities cannot be optimal. In fact, the network performance can be quite poor, as we will see in Section IV.

C. Non-cooperative Random Access Game with Different ACs

Using non-cooperative game theory, we first formulate the described N -user random access system as a finite N -person non-cooperative normal-form game $(\mathcal{N}, \Theta, \mathbf{u})$, where \mathcal{N} is the set of stations (i.e., players), $\Theta = \theta_1 \times \theta_2 \times \dots \times \theta_N$ is the set of action profiles, and $\mathbf{u} = (u_1, u_2, \dots, u_N)$ is the vector of utility functions for all stations. The action of each station $i \in \mathcal{N}$ is to *strategically* select its declared type $\hat{\theta}_i$ (not necessarily the same as its true type θ_i) to maximize its utility. In other words, given $\hat{\theta}_{-i}$ as the vector of declared types for all stations *other than* station i , station i selects $\hat{\theta}_i$ to solve the following *local* problem:

$$\max_{\hat{\theta}_i} u_i(\gamma_i(\mathbf{p}(\hat{\theta}_i, \hat{\theta}_{-i})), \theta_i), \quad (7)$$

where

$$\mathbf{p}(\hat{\theta}_i, \hat{\theta}_{-i}) = \arg \max_{\mathbf{p} \in \mathcal{P}} \sum_{i \in \mathcal{N}} u_i(\gamma_i(\mathbf{p}), \hat{\theta}_i). \quad (8)$$

Notice that each station $i \in \mathcal{N}$ is already aware that the AP will select the vector of persistent probabilities by solving problem (8). The complete analysis of game $(\mathcal{N}, \Theta, \mathbf{u})$ is not easy in general. However, we can show the following key result:

Theorem 1: Game $(\mathcal{N}, \Theta, \mathbf{u})$ has *no* Nash equilibrium.

The proof of Theorem 1 is given in Appendix A. From Theorem 1, the non-cooperative game $(\mathcal{N}, \Theta, \mathbf{u})$, with no mechanism design, leads to an unstable network. It is also important to notice that in this game, stations always have incentive to *cheat* and declare their types to be different from their true types. This can be seen by a simple example as in Fig. 1 with four ACs. In this example, any station which is running an application of type corresponding to AC 1 has an incentive to declare its type as AC 3. We further study this example in detail in Section IV.

III. TRUTHFUL MECHANISM DESIGN FOR WLANs

From the results in Section II-C, it is required to use a scheme to force the stations be *truthful*. In this section, we consider *mechanism design* for this purpose. Mechanism design is a sub-field in microeconomics and game theory that studies the implementation of an optimal system allocation with self-interested players, who aim to maximize their own payoffs. Mechanisms are responsible for the allocation of resources and incur *payment* to the players, so as to provide them with the *incentives* to declare their private information (i.e.,

their type) *truthfully*. Groves mechanism and its subfamily named Vickrey-Clarke-Groves (VCG) mechanism are among the most efficient mechanisms that not only tackle dishonesty, but also guarantee achieving maximum social welfare. The latter implies achieving the optimum of the network utility maximization problem in (4).

A. VCG Mechanism Design for Random Access WLANs

For the VCG mechanism [8], given the declared types of all players $\hat{\theta} = (\hat{\theta}_1, \dots, \hat{\theta}_N)$, the AP selects the vector of persistent probabilities according to the optimal solution of problem (8), and also the *payment* $t_i(\hat{\theta})$ for each station $i \in \mathcal{N}$ given by

$$t_i(\hat{\theta}) = \sum_{j \in \mathcal{N} \setminus \{i\}} u_j(\gamma_j(\tilde{\mathbf{p}}(\hat{\theta})), \hat{\theta}_j) - \sum_{j \in \mathcal{N} \setminus \{i\}} u_j(\gamma_j(\mathbf{p}(\hat{\theta})), \hat{\theta}_j), \quad (9)$$

where

$$\tilde{\mathbf{p}}(\hat{\theta}) = \arg \max_{\mathbf{p} \in \mathcal{P}, p_i=0} \sum_{j \in \mathcal{N} \setminus \{i\}} u_j(\gamma_j(\mathbf{p}), \hat{\theta}_j). \quad (10)$$

Given the vector of payment rules $\mathbf{t} = (t_1, \dots, t_N)$, each station needs to pay t_i to the AP for relaying its transmitted packets. Intuitively, VCG selects the payment values such that it is the *best* choice for the users to be honest and declare the correct types. Notice that since each user needs to pay for the packets it transmits to the AP, the payoff function for each user $i \in \mathcal{N}$ is indeed its own *surplus*

$$s_i(\hat{\theta}, \theta_i) = u_i(\gamma_i(\mathbf{p}(\hat{\theta})), \theta_i) - t_i(\hat{\theta}), \quad \forall i \in \mathcal{N}, \quad (11)$$

i.e., its utility minus its payment. Notice that since VCG forces all users to be honest, we indeed have $\hat{\theta}_i = \theta_i$ for all users $i \in \mathcal{N}$. Thus, solving problem (8) based on the declared types suffices to achieve optimal network performance, i.e., maximum aggregate network utility.

We are now ready to propose our VCG-based mechanism for QoS provisioning in WLANs with random access. It includes the following four key steps:

- 1) *Type declaration:* Before each station $i \in \mathcal{N}$ starts transmission, it informs the AP about the type $\hat{\theta}_i$.
- 2) *VCG mechanism:* Given the declared types $\hat{\theta}$, the AP calculates the persistent probability as in (8). It also calculates the payments \mathbf{t} as in (9).
- 3) *Resource allocation and payment:* The obtained vectors of persistent probabilities and payments are *broadcast* by the AP to all stations. Stations may only transmit based on the persistent probabilities assigned by the AP.
- 4) *Outcome enforcement:* If a station $i \in \mathcal{N}$ is detected² to transmit with a persistent probability higher than the persistent probability it is assigned, it will be punished (e.g., forbid to use the system for a certain amount of time to offset the advantage it gains from cheating).

The block diagram of the our proposed protocol using VCG mechanism design is shown in Fig. 2.

²It is easy for the AP to check whether the stations are indeed transmitting according to the assigned persistent probabilities by listening to the shared communication medium as explained in [10].

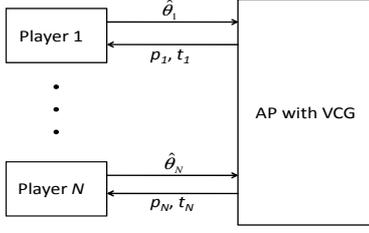


Fig. 2. Truthful mechanism design for WLANs with random access.

B. VCG Computations

In order to perform the VCG mechanism, we need to compute $\mathbf{p}(\hat{\theta})$ in (8) and $\tilde{\mathbf{p}}(\hat{\theta})$ in (10), which are not easy to solve in general due to *non-convexity* of the product forms in (1). We also notice that both problems (8) and (10) are *non-differentiable*. In this section, we propose an algorithm to compute the optimal solutions of problems (8) and (10) efficiently when the utility functions are as in (2) and (3).

First, consider the optimization problem in (8). We can write this optimization problem in the following form:

$$\begin{aligned} \max_{\mathbf{p}} \quad & \sum_{i \in \mathcal{N}} u_i(\gamma_i(\mathbf{p}), \hat{\theta}_i) \\ \text{s.t.} \quad & 0 \leq p_i \leq 1, \quad \forall i \in \mathcal{N}. \end{aligned} \quad (12)$$

Let $\mathbf{p}^* = \mathbf{p}(\hat{\theta})$. Also let $p_i^{\text{succ}*} = p_i^* \prod_{j \in \mathcal{N} \setminus \{i\}} (1 - p_j^*)$ denote the corresponding optimal success probability for station i . We can show the following lemma which provides us with a way to compute the allocation of persistent probabilities.

Lemma 1: At any optimal solution of problem (12), for all $i \in \mathcal{N}$, we have either $p_i^{\text{succ}*} > p_i^{\text{critical}}(\hat{\theta}_i)$ or $p_i^{\text{succ}*} = 0$.

The proof of Lemma 1 is given in Appendix B. From Lemma 1, VCG allocates the vector of persistent probabilities \mathbf{p} such that for each station $i \in \mathcal{N}$, we have either $p_i^{\text{succ}} > p_i^{\text{critical}}(\hat{\theta}_i)$ or $p_i^{\text{succ}} = 0$. We notice that the utility function $u_i(\gamma_i(\mathbf{p}), \hat{\theta}_i)$ is concave for any $p_i^{\text{succ}} \in (p_i^{\text{critical}}(\hat{\theta}_i), 1]$. Thus, we can reformulate problem (12) to be as follows:

$$\begin{aligned} \max_{\mathbf{p} \in \mathcal{P}, \mathcal{M} \subseteq \mathcal{N}} \quad & \sum_{i \in \mathcal{M}} u_i(\gamma_i(\mathbf{p}), \hat{\theta}_i) \\ \text{s.t.} \quad & p_i^{\text{succ}} > p_i^{\text{critical}}, \quad \forall i \in \mathcal{M}, \\ & p_i = 0, \quad \forall i \in \mathcal{N} \setminus \mathcal{M}. \end{aligned} \quad (13)$$

In problem (13), we simply divide the set of stations \mathcal{N} into two subsets: subset \mathcal{M} and subset $\mathcal{N} \setminus \mathcal{M}$. For each station $i \in \mathcal{M}$, we include the constraint $p_i^{\text{succ}} > p_i^{\text{critical}}$. On the other hand, for each $i \in \mathcal{N} \setminus \mathcal{M}$, we include the constraint $p_i = 0$. In fact, set \mathcal{M} acts as an *auxiliary* variable to model *admission control*. Here, \mathcal{M} denotes the set of those stations which are admitted to the system. Constraint $p_i^{\text{succ}} > p_i^{\text{critical}}$ implies that all admitted stations $i \in \mathcal{M}$ should achieve their required minimum bandwidth. On the other hand, those stations which are not admitted to the system are simply assigned zero success probability and zero data rate. From Lemma 1, optimization problems (12) and (13) are indeed *equivalent*.

Next, assume that we fix set \mathcal{M} and have the persistent probabilities as the only optimization variables in problem

Algorithm 1 Algorithm to solve (8) or (12) for the modified α -fair utility functions defined in (2) and (3).

- 1: (Initialization) $s = -\infty$, $\mathbf{p}^* = \mathbf{0}$, and $\mathcal{M}^* = \phi$.
 - 2: **for all** subset \mathcal{M} of \mathcal{N} **do**
 - 3: $M = |\mathcal{M}|$.
 - 4: **if** $\sqrt[M-1]{\prod_{i=1}^M p_i^{\text{critical}}} \leq 1 - \sum_{i=1}^M p_i^{\text{critical}}$ **then**
 - 5: Solve (16) for \mathbf{p} .
 - 6: **if** $\sum_{i \in \mathcal{M}} u_i(\gamma_i(\mathbf{p}), \hat{\theta}_i) > s$, **then**
 - 7: $s = \sum_{i \in \mathcal{M}} u_i(\gamma_i(\mathbf{p}), \hat{\theta}_i)$, $\mathbf{p}^* = \mathbf{p}$, and $\mathcal{M}^* = \mathcal{M}$.
 - 8: **end for**
-

(13). In other words, given set \mathcal{M} with size $M = |\mathcal{M}|$, we solve the following optimization problem:

$$\begin{aligned} \max_{\mathbf{x}, \mathbf{p} \in \mathcal{P}} \quad & \sum_{i \in \mathcal{M}} u_i(x_i, \hat{\theta}_i) \\ \text{s.t.} \quad & p_i^{\text{succ}} \geq x_i \geq 0, \quad \forall i \in \mathcal{M}, \\ & p_i^{\text{succ}} > p_i^{\text{critical}}, \quad \forall i \in \mathcal{M}, \\ & p_i = 0, \quad \forall i \in \mathcal{N} \setminus \mathcal{M}. \end{aligned} \quad (14)$$

Notice that there are 2^N possible choices for set \mathcal{M} , i.e., all possible subsets of set \mathcal{N} . In some cases, (e.g., when too many users are admitted to the network while the required *critical* probabilities are high), problem (14) may become *infeasible*.

Lemma 2: Given set \mathcal{M} and the declared vector $\mathbf{p}^{\text{critical}} = (p_i^{\text{critical}}, \forall i \in \mathcal{M})$, problem (14) is feasible *only if*:

$$\sqrt[M-1]{\prod_{i=1}^M p_i^{\text{critical}}} \leq 1 - \sum_{i=1}^M p_i^{\text{critical}}. \quad (15)$$

The proof of Lemma 2 is in Appendix C. Notice that the condition in (15) is a *necessary condition* for feasibility. Next, by taking the logarithm of both sides of the first constraint in problem (14) and a log change of variables $u'_i(x'_i, \hat{\theta}_i) = u_i(e^{x'_i}, \hat{\theta}_i)$ and $x'_i = \log x_i$, we reformulate it into the following equivalent optimization problem for any utility parameter $\alpha_i \geq 1$:

$$\begin{aligned} \max_{\mathbf{x}', \mathbf{p} \in \mathcal{P}} \quad & \sum_{i \in \mathcal{M}} u'_i(x'_i, \hat{\theta}_i) \\ \text{s.t.} \quad & \log p_i + \sum_{j \in \mathcal{N} \setminus \{i\}} \log(1 - p_j) - x'_i \geq 0, \quad \forall i \in \mathcal{M}, \\ & \log p_i + \sum_{j \in \mathcal{N} \setminus \{i\}} \log(1 - p_j) > \log p_i^{\text{critical}}, \quad \forall i \in \mathcal{M}, \\ & p_i = 0, \quad \forall i \in \mathcal{N} \setminus \mathcal{M}. \end{aligned} \quad (16)$$

It is easy to verify that problem (16) is *convex*. From this, together with the results from Lemmas 1 and 2, we are now ready to propose our algorithm to find the exact global optimal solution of problem (8). The algorithm is given in Algorithm 1. A similar algorithm can be given to solve problem (10).

IV. PERFORMANCE EVALUATION

In this section, we assess the performance of our proposed VCG-based mechanism design using MATLAB. We first consider a simple WLAN with one AP and four stations. The AP supports four ACs. For each $i = 1, \dots, 4$, the i^{th} station runs an application which belongs to the i^{th} AC. The utility functions for the four applications are as in Fig. 1. With the use of the VCG mechanism, we plot the utilities, payments, and surpluses of the four players in Fig. 3. We consider two cases. In Case I,

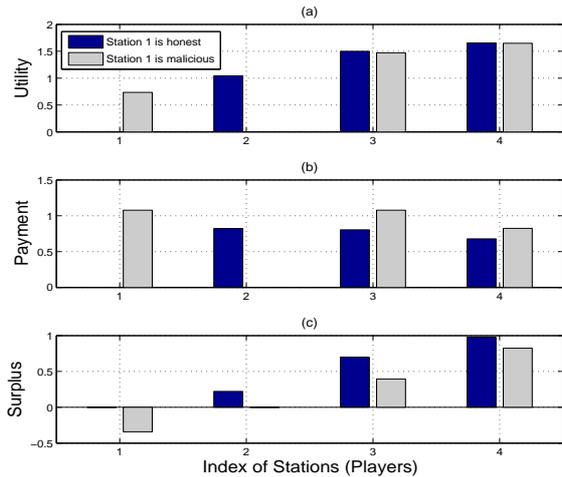


Fig. 3. Results in a sample network with four stations and four ACs: (a) Utilities, (b) Payments, and (c) Surpluses of all four stations, using the proposed VCG-based mechanism in Algorithm 1.

station 1 honestly declares that it supports applications in AC 1 while all other stations are also honest. In Case II, station 1 *maliciously* declares that it supports applications in AC 4 while other stations are still honest. The results are shown in Fig. 3, where Cases I and II are represented by the two bars (with grey and blue colors) at the index of each station. When all the stations are honest, only AC 2 to AC 4 are given admission to the system. If station 1 lies about its AC and declares it as AC 3, station 1 but not station 2 is admitted into the system. The obtained utility of station 1 increases, while the utilities of the other stations decrease. However, due to the VCG mechanism, station 1 is punished with a higher payment when it lies, as shown in Fig. 3(b). Considering both utility and payment, the *surplus* of player 1 indeed decreases if it lies, as shown in Fig. 3(c). Clearly, this forces user 1 to be truthful about its type.

Next, we evaluate the performance of our proposed scheme in a larger network with 10 stations. We assume that there are two ACs available: AC 1 and AC 2. For AC 1, the utilities are characterized by $K_i = 1$, $\alpha_i = 1$, and $p_i^{\text{critical}} = 0.01$. For AC 2 the utilities are characterized by $K_i = 2$, $\alpha_i = 1$, and $p_i^{\text{critical}} = 0.012$. Assuming that five stations run AC 1 applications and five stations run AC 2 applications, the throughput of the two ACs with different numbers of malicious stations with AC 1 (i.e., lower AC) are shown in Fig. 4(a). Here we assume the nominal data rate φ to be 11Mbps. We can see that, without the use of mechanism design, malicious stations running AC 1 applications will indeed declare themselves as running AC 2 in order to obtain a higher utility. Thus, the AP is unable to provide differentiated QoS as it was initially intended to do. On the other hand, when our proposed VCG-based mechanism is being used, differentiated QoS support is indeed guaranteed and AC 2 applications are offered better throughput.

We also compare the aggregate network utility (i.e., the objective function in (4) with different numbers of malicious stations.) We use $K_i = 1$ for AC 1, $K_i = 30$ for AC 2, and the other parameters are the same as above. Results, when the number of malicious stations (which are running AC 1

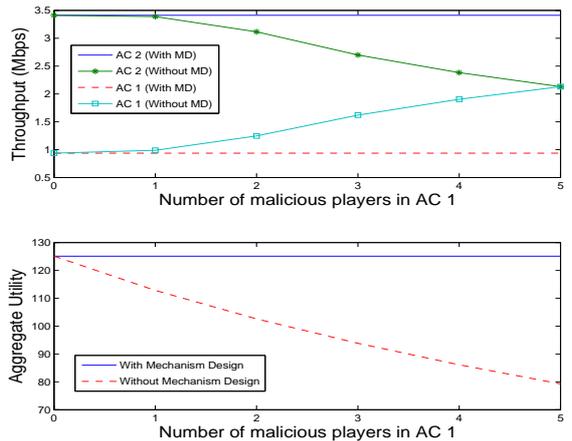


Fig. 4. Impact of using VCG-based mechanism design in a network with one AP, ten stations, and two ACs: (a) Aggregate throughput of AC 1 and AC 2; (b) Aggregate utility, with and without mechanism design (MD). We assume that the five stations which are running AC 2 applications are indeed honest, and then we vary the number of malicious players among those five stations which are running AC 1 applications.

applications) varies from 0 to 5, are shown in Fig. 4(b). Recall that dishonest stations declare themselves as running AC 2 applications, rather than AC 1 applications. As we can see, this causes deviation from optimal aggregate network utility. The performance reduction becomes more severe as the number of malicious stations increase, e.g., resulting in more than 36% efficiency loss in the presence of five malicious stations. Thus, using a VCG-based mechanism as in Algorithm 1 is indeed resulting in a significantly better network performance.

V. CONCLUSION

In this paper, we have studied the problem that a station can maliciously declare the AC of application that it is supporting in order to gain a higher utility. This can result in a drastic degradation of network performance and avoid adequate service distinction among different ACs. We have applied the VCG mechanism in our random access protocol to motivate the stations to declare truthfully their ACs of the applications. The AP then performs admission control, and informs the stations about their persistent probabilities and the required payments. We have also studied the computational issue of the VCG mechanism. The utility functions of the stations are in the form of modified α -fair utility functions, which can represent the utilities of most applications in general, but are non-convex and non-differentiable. An algorithm is proposed to compute the persistent probability and the set of admitted stations for these utility functions. With the use of mechanism design, our results show that a higher aggregate utility can be achieved and the QoS in terms of throughput can be supported.

APPENDIX

A. Proof of Theorem 1

We prove by contradiction. We first notice that the payoff function for each user $i \in \mathcal{N}$ is as in (7), where $p(\hat{\theta}_i, \hat{\theta}_{-i})$ is indeed as in (8). If there exists any Nash equilibrium $\hat{\theta}^* =$

$(\hat{\theta}_1^*, \dots, \hat{\theta}_N^*)$, then for each station $i \in \mathcal{N}$, we should have:

$$u_i(\gamma_i(\mathbf{p}(\hat{\theta}_i^*, \hat{\theta}_{-i}^*)), \theta_i) \geq u_i(\gamma_i(\mathbf{p}(\hat{\theta}_i, \hat{\theta}_{-i}^*)), \theta_i), \quad \forall \hat{\theta}_i \in \Theta_i. \quad (17)$$

In that case, we can define another vector $\hat{\theta}^*$ such that $\hat{\theta}_i^* = \hat{\theta}_i^*$ for all $i \in \mathcal{N} \setminus \{1\}$ and we have: $\hat{\theta}_1^* = (\hat{\alpha}_1^*, \hat{p}_1^{\text{critical}*}, \hat{K}_1^*) = (\hat{\alpha}_1^*, \hat{p}_1^{\text{critical}*}, \beta \hat{K}_1^*)$ where $\beta \gg 1$. It is easy to verify that for the utility functions in (2) and (3), if $\beta \rightarrow \infty$, then the optimal solution of problem (8) becomes $(1, 0, \dots, 0)$. This indicates that $p_1^{\text{succ}} \rightarrow 1$ and $p_i^{\text{succ}} \rightarrow 0$ for all $i \in \mathcal{N} \setminus \{1\}$. Since the utility for each user is an increasing function of the probability of successful transmission of each user, we have:

$$u_1(\gamma_1(\mathbf{p}(\hat{\theta}_1^*, \hat{\theta}_{-1}^*)), \theta_1) < u_1(\gamma_1(\mathbf{p}(\hat{\theta}_1, \hat{\theta}_{-1}^*)), \theta_1). \quad (18)$$

Therefore, (17) does *not* hold. This contradicts the assumption that $\hat{\theta}^*$ is a Nash equilibrium. Since we considered any arbitrary Nash equilibrium, this implies that the non-cooperative random access game does not have a Nash equilibrium. In other words, given *any* choice of types for the stations, each station $i \in \mathcal{N}$ has an incentive to unilaterally *deviate* to another type with higher K_i value to be paid off better. ■

B. Proof of Lemma 1

We first fix $i \in \mathcal{N}$. Then, we compare the sum of utilities of the players with the two vectors of persistent probabilities \mathbf{p}' and \mathbf{p}'' , where $p'_j = p''_j, \forall j \in \mathcal{N} \setminus \{i\}$, $p'_i = 0$, and $p''_i > 0$ such that $0 \leq p_i^{\text{succ}'} \leq p_i^{\text{succ}''}(\hat{\theta}_i)$.

First, we consider the case that $\exists j \in \mathcal{N} \setminus \{i\}$ such that $p'_j = p''_j = 1$. Then, we have $p_i^{\text{succ}'} = p_i^{\text{succ}''} = 0, \forall i \in \mathcal{N} \setminus \{j\}$, which implies that $u_i(\gamma_i(\mathbf{p}'), \hat{\theta}_i) = u_i(\gamma_i(\mathbf{p}''), \hat{\theta}_i) = 0, \forall i \in \mathcal{N} \setminus \{j\}$. From \mathbf{p}' and \mathbf{p}'' , we have $p_j^{\text{succ}'} \geq p_j^{\text{succ}''}$. Because u_j is an increasing function of p_j^{succ} , we have $u_j(\gamma_j(\mathbf{p}'), \hat{\theta}_j) \geq u_j(\gamma_j(\mathbf{p}''), \hat{\theta}_j)$. As a whole, we have $\sum_{j \in \mathcal{N}} u_j(\gamma_j(\mathbf{p}'), \hat{\theta}_j) \geq \sum_{j \in \mathcal{N}} u_j(\gamma_j(\mathbf{p}''), \hat{\theta}_j)$.

Then, we consider the case $p'_j = p''_j \neq 1, \forall j \in \mathcal{N} \setminus \{i\}$. In this way, we have $p''_i > 0$ such that $0 < p_i^{\text{succ}''} \leq p_i^{\text{critical}}(\hat{\theta}_i)$. From the definition of the utility functions in (2) and (3), we have $u_i(\gamma_i(\mathbf{p}'), \hat{\theta}_i) = u_i(\gamma_i(\mathbf{p}''), \hat{\theta}_i) = 0$. From \mathbf{p}' and \mathbf{p}'' , we have $p_j^{\text{succ}'} \geq p_j^{\text{succ}''}, \forall j \in \mathcal{N} \setminus \{i\}$. Because u_j is an increasing function of p_j^{succ} , we have $u_j(\gamma_j(\mathbf{p}'), \hat{\theta}_j) \geq u_j(\gamma_j(\mathbf{p}''), \hat{\theta}_j), \forall j \in \mathcal{N} \setminus \{i\}$. As a whole, we have $\sum_{j \in \mathcal{N}} u_j(\gamma_j(\mathbf{p}'), \hat{\theta}_j) \geq \sum_{j \in \mathcal{N}} u_j(\gamma_j(\mathbf{p}''), \hat{\theta}_j)$.

From the above two cases, we see that the vector \mathbf{p}'' should never be allocated, because the system is performing the allocation of persistent probability according to (8). In this way, we can never have $0 < p_i^{\text{succ}} < p_i^{\text{critical}}(\hat{\theta}_i)$ at optimality of optimization problem (12). ■

C. Proof of Lemma 2

Problem (13) is feasible if and only if we have

$$p_i \left(\prod_{j \in \mathcal{M} \setminus \{i\}} (1 - p_j) \right) \geq p_i^{\text{critical}}, \quad \forall i \in \mathcal{M}. \quad (19)$$

By manipulating (19), for each user $i \in \mathcal{M}$, we have

$$1 - p_i \leq \frac{\prod_{j \in \mathcal{M}} (1 - p_j)}{p_i^{\text{critical}} + \prod_{j \in \mathcal{M}} (1 - p_j)}. \quad (20)$$

Multiplying both sides of the inequality above for all users, we can finally come up with the following key condition:

$$\prod_{i \in \mathcal{M}} (1 - p_i) \leq \prod_{i \in \mathcal{M}} \left(\frac{\prod_{j \in \mathcal{M}} (1 - p_j)}{p_i^{\text{critical}} + \prod_{j \in \mathcal{M}} (1 - p_j)} \right) \quad (21)$$

We define: $A(\mathbf{p}) = \prod_{j \in \mathcal{M}} (1 - p_j)$. Clearly, $0 \leq A(\mathbf{p}) \leq 1$ is the probability of experiencing an *idle* time slot. That is, the probability that *no* user transmits any packet. Replacing $A(\mathbf{p})$ in (21), problem (13) is feasible if there *exists* any value A between zero and one such that we have:

$$\prod_{i \in \mathcal{M}} (p_i^{\text{critical}} + A) \leq A^{M-1}. \quad (22)$$

For the rest of the proof, we show that condition (15) is a *necessary* condition for existence of any A such that (22) holds. We first notice that from (22), we need to have

$$A^{M-1} \sqrt[M-1]{\prod_{i \in \mathcal{M}} p_i^{\text{critical}}} \leq A. \quad (23)$$

Condition (22) can be written as the following extended form:

$$A^M + \left(\sum_{i=1}^M p_i^{\text{critical}} - 1 \right) A^{M-1} + \Gamma(A) \leq 0, \quad (24)$$

where $\Gamma(A)$ is a polynomial in A with degree $M-2$ and only non-negative multipliers. Clearly, $\Gamma(A) \geq 0$. Thus, from (24) we also need to have

$$A \leq 1 - \sum_{i=1}^M p_i^{\text{critical}}. \quad (25)$$

Putting the *lower-bound* in (23) and the *upper-bound* in (25) together, optimization problem (13) is feasible *only if* there exists any $0 \leq A \leq 1$ such that the following holds:

$$A^{M-1} \sqrt[M-1]{\prod_{i \in \mathcal{M}} p_i^{\text{critical}}} \leq A \leq 1 - \sum_{i=1}^M p_i^{\text{critical}}. \quad (26)$$

Clearly, the above condition holds as long as the upper bound is greater than or equal to the lower bound. This directly results in condition (15). ■

REFERENCES

- [1] IEEE 802.11e WG, "Wireless LAN MAC and PHY specifications amendment 8: MAC quality of service enhancements," Sept. 2005.
- [2] J. W. Lee, A. Tang, J. Huang, M. Chiang, and A. R. Calderbank, "Reverse-engineering MAC: A non-cooperative game model," *IEEE J. Select. Areas Commun.*, vol. 25, no. 6, pp. 1135–1147, Aug. 2007.
- [3] M. Cagalj, S. Ganerwal, I. Aad, and J. P. Hubaux, "On selfish behavior in CSMA/CA networks," in *Proc. of IEEE INFOCOM*, Miami, FL, Mar. 2005.
- [4] J. Konorski, "A game-theoretic study of CSMA/CA under a backoff attack," *IEEE/ACM Trans. on Networking*, vol. 14, no. 6, pp. 1167–1178, Dec. 2006.
- [5] T. Cui, L. Chen, and S. Low, "A game-theoretic framework for medium access control," *IEEE J. Select. Areas Commun.*, vol. 26, no. 7, pp. 1116–1127, Sept. 2008.
- [6] P. Nuggehalli, M. Sarker, and R. Rao, "QoS and selfish users: A MAC layer perspective," in *Proc. IEEE GLOBECOM*, Washington, DC, Nov. 2007.
- [7] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, Massachusetts: MIT Press, 1991.
- [8] D. C. Parkes, "Iterative combinatorial auctions: Achieving economic and computational efficiency," Ph.D. dissertation, University of Pennsylvania, May 2001.
- [9] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. on Networking*, vol. 8, no. 5, pp. 556–567, Oct. 2000.
- [10] A. H. Mohsenian Rad, J. Huang, M. Chiang, and V. Wong, "Utility-optimal random access without message passing," *IEEE Trans. Wireless Commun.* (accepted), 2009.