On Selfish Behavior in CSMA/CA Networks

Mario Čagalj*, Saurabh Ganeriwal†, Imad Aad* and Jean-Pierre Hubaux*

*School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL) , CH-1015 Lausanne, Switzerland
†Electrical Engineering Department, University of California, Los Angeles
{mario.cagalj@epfl.ch, saurabh@ee.ucla.edu, imad.aad@epfl.ch and jean-pierre.hubaux@epfl.ch}

Abstract— CSMA/CA protocols rely on the random deferment of packet transmissions. Like most other protocols, CSMA/CA was designed with the assumption that the nodes would play by the rules. This can be dangerous, since the nodes themselves control their random deferment. Indeed, with the higher programmability of the network adapters, the temptation to tamper with the software or firmware is likely to grow; by doing so, a user could obtain a much larger share of the available bandwidth at the expense of other users.

We use a game-theoretic approach to investigate the problem of the selfish behavior of nodes in CSMA/CA networks, specifically geared towards the most widely accepted protocol in this class of protocols, IEEE 802.11. We characterize two families of Nash equilibria in a single stage game, one of which always results in a network collapse. We argue that this result provides an incentive for cheaters to cooperate with each other. Explicit cooperation among nodes is clearly impractical. By applying the model of dynamic games borrowed from game theory, we derive the conditions for the stable and optimal functioning of a population of cheaters. We use this insight to develop a simple, localized and distributed protocol that successfully guides multiple selfish nodes to a Pareto-optimal Nash equilibrium.

I. Introduction

Carrier-sense multiple-access with collision avoidance (CSMA/CA) protocols rely on the random deferment of packet transmissions for the efficient use of a shared wireless channel among many nodes in a network; this class of MAC protocols is one of the most popular for wireless networks. In general, it is assumed that all nodes respect the rules of the protocol. However, we claim that this assumption is less and less appropriate, because the network adapters are becoming more and more *programmable* [1]. As a result, today a user can modify the behavior of his wireless interface very easily. In this paper, we study the stability and efficiency of wireless networks that contain one or several *selfish* users. By "selfish" we designate the users who are ready to tamper with their wireless interface in order to increase their own share of the common transmission resource; we assume these users to be rational, and not malicious (they are ready to harm other users only if they can derive a benefit from this misbehavior).

More specifically, we consider that a cheater makes use of the easiest (and yet highly rewarding) cheating technique: he deliberately does not respect the random deferment of the transmission of his packets. Although this cheating technique is straightforward, we show that studying its implications is far from trivial. In order to be able to corroborate our simulations with analytical results, we make use of game theory: each node is a player, the throughput it enjoys is its payoff, and the size of its contention window represents its move. By making use of this model and of extensive simulations, we systematically study several problems. First, we consider the simple case of a network with a single cheater. We then assume the presence of several cheaters, and characterize two families of Nash equilibria in a single stage game, one of which always results in a network collapse. We compute the Pareto-optimal point of operation of such a system, and study the equilibria of dynamic games. We introduce the notion of *cooperative players*, namely cheaters who try to continue operating at the Pareto-optimal point of operation. We also propose a detection and a punishment technique against those players who exhibit a non-cooperative behavior. Finally, we explain how the players can collectively search for the optimal point of operation, even if they are unaware of the number of nodes present in the network.

To the best of our knowledge, this paper is the first to provide a systematic analysis of rational cheating in CSMA/CA networks. To make it as concrete as possible, we refer to the most prominent version of CSMA/CA, namely IEEE 802.11; however, the conclusions we derive are valid for any protocol of the CSMA/CA family.

The rest of the paper is organized as follows. The next section addresses the related work. Section III describes the system model considered in the paper. Section IV studies the case of a static game, whereas Section V studies the case of a dynamic game. Section VI shows the deviation detection and penalization mechanisms we use. Section VII analyzes the distributed coordination protocol. Last, Section VIII concludes the paper.

II. Related work

The problem of non-cooperative nodes in wireless (wired) networks has been widely addressed on the network layer, whereas little work has been done on the MAC layer. MacKenzie and Wicker [2] study the problem of selfish users in Aloha from a game-theoretic point of view. They analyze the stability

of the system (Nash equilibrium), and calculate the transmission probabilities that optimize each node's throughput. They assume that all nodes have the same transmission rates and costs. Moreover, every node has an *a priori* knowledge about the total number of nodes in the system. Altman et al. [3] reconsider the same Aloha "game" with partial information, where the transmission probability is adapted according to collision feedback only. They consider two frameworks: team work and non-cooperative game. Jin and Kesidis [4] study non-cooperative equilibria of Aloha networks for heterogeneous users.

For IEEE 802.11, Kyasanur and Vaidya [5] propose that the receiver assigns the backoff value to be used by the sender, so the former can detect any misbehavior of the latter. If the sender deviates from the assigned value, it will be assigned high backoff values on the next round to compensate its deviation. As mentioned by the authors, this mechanism has several limitations such as the possible collusion between sender and receiver, and the fundamental change to the protocol. Konorski [6] proposes a misbehaviour-resilient backoff algorithm; both [5] and [6] exhibit the same drawback: they require to change the current protocol.

Game theory has been applied in the study of optimal routing [7], [8], [9], congestion control [10], power control [11], [12], as well as incentive engineering in wireless access networks [13].

III. SYSTEM MODEL

We consider N wireless nodes that are willing to transmit data to N designated receivers (one per sender). In this study, we assume all the nodes to be within the same communication range (i.e., each node can hear any other node). This is to avoid complications introduced by the *hidden terminal problem*. Nodes use a CSMA/CA based protocol to resolve contention at the MAC layer. In this paper, we will be dealing exclusively with IEEE 802.11 (in DCF mode) [14]; we note that the analysis carried out in this paper can also be extended to other CSMA/CA based protocols. We further assume each node to have an authentic MAC layer identifier. This can be achieved by means of MAC layer authentication. Finally, we assume that the nodes are static and that they always have packets (of the same size) to send. Possible ways to relax some of the above assumptions are discussed in Section VII.

We consider a scenario where out of the N senders, a subset of C nodes deliberately deviate from the IEEE 802.11 protocol. We designate this subset of nodes as *cheaters*. There can be a number of ways in which a node can *cheat*. For example, in violation of the standard protocol, a cheater i initializes his window size to a lower value in order to obtain a higher throughput. We will call this lower value W_i throughout the paper. Moreover, a cheater does not respect the *binary exponential backoff* principle and keeps his contention window size fixed after a collision, i.e. equal to W_i. This mode of cheating is the easiest for potential cheaters, since it does not require changes to be made in the operation of IEEE 802.11

protocol. However, the main conclusions of this paper are applicable to any other cheating technique.

The relevance of these misbehaving techniques becomes even higher with the emerging standards that address the Quality of Service support, such as IEEE 802.11e [15]. The latter gives the users total control of the MAC parameters, therefore enabling them to easily cheat. We assume the cheaters in our model to be rational, i.e., they want to maximize their own benefit. In this particular context, the cheaters want to maximize the average throughput they receive r_i. This problem can easily be modeled in a game theory framework. All the cheater nodes are the *players* in this game. The strategy of each (cheater) player i consists in setting the value of his contention window W_i such that player i's expected payoff (utility) U_i is maximized. In this work, we define a player i's utility to be equal to the enjoyed throughput r_i (i.e., $U_i = r_i$).

IV. STATIC GAME

In this section, we first analyze the problem of misbehaving from the perspective of a single cheater and then consider more complex scenarios with multiple cheaters in the system. We model the cheaters' interaction using a static game [16]. A static game is one in which all players make decisions (or select a strategy) simultaneously, without knowledge of the strategies that are being chosen by other players.

A. *Variation of throughput with* W_i

In [17], Bianchi presented a saturation throughput model for the IEEE 802.11 protocol. Since we assume that a cheater's objective is to maximize his throughput (and we assume he always has a packet to send), he will tend to use the full channel capacity (i.e., the system will operate at the saturation point). Therefore, we make use of the same model as [17]. To estimate the throughput of IEEE 802.11, in a network with no misbehaving nodes, Bianchi [17] used a two-dimensional Markov chain of m backoff stages in which each stage represents the backoff time counter of a node. A transition takes place upon collision and successful transmission, to a higher stage and to the first stage respectively. Considering the stationary distribution of the chain, the channel *access probability* τ of a node can be derived as a function of the number of levels m and the minimum contention window value W_{min}. The throughput enjoyed by a given node, which is the average information payload transmitted in a slot time over the average length of a slot time, can be computed using Bianchi's model as follows:

$$r_i = \frac{P_i^s \overline{L}}{P^s T^s + P^c T^c + P^i T^i} \quad (1)$$

where $P_i^s = \tau_i \prod_{j \neq i}(1 - \tau_j)$ is the probability that an arbitrary station successfully transmits during a random time slot; τ_i is the access probability of station i; $\overline{L}$ is the average size of a packet; $P^s = \sum_k P_k^s$; T^s is the average time needed to transmit a packet of size $\overline{L}$ (including the inter-frame spacing periods [17]); $P^i = \prod_k(1 - \tau_k)$ is the probability of the channel being idle; T^i is the duration of the idle period (a

single slot); $P^c = 1 - P^i - \sum_k P_k^s$ is the probability of collision; T^c is the average time spent in the collision.

We extend Bianchi's model to describe a network with misbehaving nodes. We therefore consider two separate Markov chains. The first, with a single backoff stage, since cheaters are assumed to fix their contention windows, is used to derive the cheaters' access probabilities τ_i^c. The second chain, similar to the one in [17], is used to derive the access probabilities τ_i^l of well-behaved nodes. The conditional collision probabilities are derived considering both well-behaved and cheating nodes' access probabilities. As mentioned in Section III, cheater i fixes his contention windows to some value W_i. Consequently, cheater i accessing probability calculates as [18]:

$$\tau_i^c = \frac{2}{W_i + 1}. \tag{2}$$

After some algebraic manipulations of equation (1), we obtain the following expression for the throughput of a cheater i [18]:

$$r_i = \frac{\tau_i^c c_1^i}{\tau_i^c c_2^i + c_3^i} \tag{3}$$

where $c_1^i = p_{-i}\overline{L}$; $c_2^i = p_{-i}(T^s - T^i) - s_{-i}(T^s - T^c)$; $c_3^i = (1 - p_{-i} - s_{-i})T^c + s_{-i}T^s + p_{-i}T^i$; with the following substitutions: $p_{-i} = \prod_{j \neq i}(1 - \tau_j^c)\prod_k(1 - \tau_k^l)$; $s_{-i} = \sum_{j \neq i}\tau_j^c\prod_{k,d \neq j,i}(1 - \tau_k^c)(1 - \tau_d^l)$.

It is important to notice here, that the only parameter that a cheating node i has control over is its own contention window, W_i. By varying W_i, node i changes its own access probability τ_i, as well as the access probabilities of the well-behaved nodes. Let us assume that W_i is a continuous variable. Although the accessing probabilities of well-behaved nodes (and thus expressions c_1^i, c_2^i and c_3^i) depend on τ_i^c, we neglect this dependence for a first degree analysis. This approximation allows us to elaborate a closed form expression of the first derivative of equation (3):

$$\frac{\partial r_i}{\partial W_i} = \frac{c_1^i c_3^i}{(\tau_i^c c_2^i + c_3^i)^2}\frac{-2}{(W_i + 1)^2} \tag{4}$$

which, for $T^s \geq T^c$ and $\tau_j^c < 1$, $j \neq i$, is always negative[1]. We conclude that the expected received throughput r_i is a strictly decreasing function of W_i (for $\tau_j^c < 1$, $j \neq i$). Thus, by unilaterally decreasing its own W_i, a node can increase its received throughput (except if $\tau_k^c = 1$, for some player k; we will treat this case later in the text). Note that this conclusion would still hold even if we considered the dependence of c_1, c_2 and c_3 on τ_i^c. In fact, by using this approximation, we actually underestimate the benefits of the cheater (the cheater gets more throughput in reality, as will be shown shortly).

We will now verify this claim by simulations performed in ns-2 [19]. The simulation setup[2] consists of $N = 20$ nodes

---

[1]According to IEEE 802.11 [14] we have the following: $T^s = PHYheader + MACheader + \overline{L} + SIFS + \sigma + ACK + DIFS + \sigma$, $T^c = PHYheader + MACheader + \overline{L} + DIFS + \sigma$, where σ is the propagation delay. From this we conclude that $T^s \geq T^c$ holds.

[2]In the rest of the paper, we will only mention the changes that are made from this reference simulation setup.
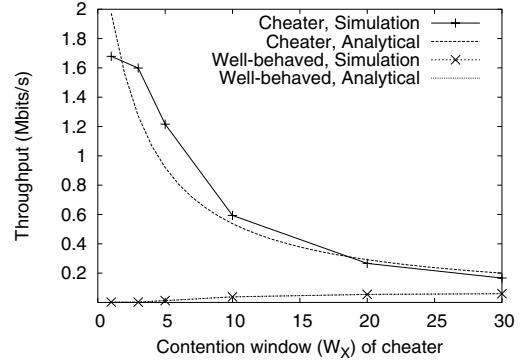

Fig. 1. Throughputs for 20 nodes, out of which one is a cheater

randomly spread over a 100×100m area, all within receive range of each other (no hidden nodes). Node X deliberately fails to adhere to the protocol and tries to misbehave, following the cheating model presented in Section III. Traffic sources are CBR, sending 1050-byte frames every 5 ms. This is enough to saturate the 2 Mbits/s channel (1.6 Mbits/s effective data rate), even when a single node is transmitting. The parameter values for the IEEE 802.11 protocol are chosen according to the IEEE 802.11b standard [14], no RTS/CTS handshake is used. Fig. 1 plots the throughput obtained by cheater X, as well as by each well-behaved node for different values of W_X. Simulation results show a good match with the analytical results. The duration for each simulation run is 50 seconds and the results are averaged over 5 simulation runs. As can be observed from Fig. 1, the cheater can increase his expected payoff (received throughput) at the expense of other nodes by choosing a small value of W_X. Furthermore, the throughput obtained by the cheater increases monotonically with the decrease of W_X.

B. Nash equilibria of a static game

Based on the cheaters' payoff function defined above, in this subsection we study the equilibria of a static game model. The solution concept we will be using to study the static game is Nash equilibrium [16]. In this study we do not consider well-behaved nodes.

We first investigate whether a Nash equilibrium point exists for the system or not. We will study the existence of a Nash equilibrium point by making use of the concept of a player's *best-response function (correspondence)* [16]. Let $S_i \equiv (1, 2, \ldots, W_{max})$, with $W_{max} \in \mathbb{N}$, be the strategy set of player i, i.e., $W_i \in S_i$. Define further $W_{-i} \equiv (W_1, \ldots, W_{i-1}, W_{i+1}, \ldots, W_C)$ and similarly S_{-i}.

Definition 1: We say that player i's best-response correspondence $b_i : S_{-i} \to S_i$, is the correspondence that assigns to each $W_{-i} \in S_{-i}$ the set $b_{W_{-i}} = \{W_i \in S_i : r_i(W_i, W_{-i}) \geq r_i(W_i', W_{-i})$ for all $W_i' \in S_i\}$.

Now, we can characterize a Nash equilibrium as follows. The strategy profile $W = (W_1, \ldots, W_C)$ constitutes a Nash equilibrium of our game if and only if $W_i \in b_{W_{-i}}$ for $i = 1, \ldots, C$ [16].

Proposition 1: For any strategy profile $W = (W_1, \ldots, W_C)$ that constitutes a Nash equilibrium, $\exists i$ s.t. $W_i = 1$.

Proof: Assume $W = (W_1, \ldots, W_C)$ is such that $W_j > 1$, $j = 1, \ldots, C$. Now, take one player, say i, and consider his best-response correspondence $b_{W_{-i}}$. Since r_i is a decreasing function of W_i (equation (4) and assumption $W_j > 1$ ($\tau_j^c < 1$), $j = 1, \ldots, C$), it readily follows that the only value for W_i that satisfies $r_i(W_i, W_{-i}) \geq r_i(W_i', W_{-i})$ for all $W_i' \in S_i$, is unity (i.e., $b_{W_{-i}} = \{1\}$). Finally, assuming that $\tau_k^c = 1$ for some player k, the proposition follows trivially from equation (2). ∎

Proposition 2: Our static game admits exactly $W_{max}^C - (W_{max} - 1)^C$ Nash equilibria, where C is the number of cheaters and W_{max} is the maximum value of the cheaters' contention window.

Proof: Assume that for player i we have $W_i = 1$. Then his access probability $\tau_i^c = 1$ and consequently for all players $j \neq i$, $p_{-j} = (1 - \tau_i^c) \prod_{k \neq j,i}(1 - \tau_k^c) = 0$. From equation (3) it follows that $r_j = 0$ ($j \neq i$) for any value of W_j. Thus for any value of W_j we have $W_j \in b_{W_{-j}}$. This is clearly true for any number of players who have their contention window set to 1. Combining this with Proposition 1, we obtain the following characterization of Nash equilibria: *at any Nash equilibrium we have one or more cheaters who set their contention window to unity and the other cheaters play any value greater than 1*. The proposition follows by observing that out of the total of W_{max}^C different vectors $W = (W_1, \ldots, W_C)$, exactly $(W_{max} - 1)^C$ do not contain any unity element. ∎

Propositions 1 and 2 give us an insight into the characteristics of Nash equilibria points. It is interesting to notice that the equilibria can be classified in the two families. To describe these, we define a set $\mathcal{D} \equiv \{i : W_i = 1, i = 1, \ldots, C\}$.

- **1st family:** $|\mathcal{D}| = 1$, that is, there is only one selfish node i who plays $W_i = 1$ and receives a non-null throughput $r_i > 0$, and for all other nodes j we have $r_j = 0$;
- **2nd family:** $|\mathcal{D}| > 1$, that is, we have more than one selfish node who set their contention windows to unity, in which case $r_i = 0$, for $i = 1, \ldots, C$.

However, since we assume that each cheater strives to attain the most of the channel capacity, the most likely equilibrium is the one at which more than one cheater sets $W_i = 1$, therefore:

Corollary 1: In the presence of more than one cheater, each player's payoff most likely equals zero ($r_i = 0$, $i = 1, \ldots, C$). This result is know as the *tragedy of the commons* in economics.

C. Fairness and Pareto-optimality

According to the analysis of the earlier section, there exist two families of Nash equilibrium points. In one family, we have great unfairness (only a single cheater gets some positive payoff). In the second family, we have highly undesirable equilibria resulting in a zero payoff for every cheater. Therefore, we look for an alternative solution to the CSMA/CA game. A desirable solution of the CSMA/CA game should exhibit the

following three properties: (i) the solution should be unique, (ii) the solution should result in a fair distribution of the system throughput (and each cheater should achieve a strictly positive payoff[3]), and (iii) the solution should result in system optimal allocation of the available capacity.

To derive such a solution we use the Nash bargaining framework from cooperative game theory [20]. The Nash bargaining framework is used to model a situation in which the players negotiate on which point of the set of joint feasible payoffs R they will agree upon. In our case, $R \equiv \{r = (r_1, \ldots, r_C) : r_i = f_i(W), \ i = 1, \ldots, C; \ W \in S_1 \times S_2 \times \ldots \times S_C\}$, where the functions f_i are derived from equations (2) and (3). An important element of the Nash bargaining framework is a fixed disagreement vector $r^0 = (r_1^0, \ldots, r_C^0)$, where usually $r_i^0 \equiv \max_i \min_{-i} r_i$. The role of the disagreement point is to provide an incentive for the agreement point to take effect; in case negotiations break down, the outcome becomes $r^0 = (r_1^0, \ldots, r_C^0)$. In the case of our CSMA/CA game, $r_i^0 = \max_i \min_{-i} r_i = 0$ (i.e., each node can simply obstruct the traffic of any other node) and hence the disagreement vector is $r^0 = (0, \ldots, 0)$. Note here that the disagreement vector corresponds to the Nash equilibrium of the static game, which gives high credibility to the disagreement vector. Thus the whole bargaining problem can be described conveniently by the pair (R, r^0). To solve this problem, Nash took an axiomatic approach and proposed a solution to (R, r^0) that results in a unique vector $r^* = (r_1^*, \ldots, r_C^*)$ satisfying a certain set of properties [20]. In this work, we are interested in the following properties of the Nash solution: (i) uniqueness, (ii) fairness, and (iii) system optimality (i.e., Pareto-optimality).

A well-known result states that if the set R is compact and convex, and there exists at least one $r \in R$ such that $r > r^0$, then the unique solution to the bargaining problem (R, r^0) corresponds to the unique solution of the following optimization problem [20]:

$$\max \prod_{i=1}^{C}(r_i - r_i^0) \text{ subject to } r \in R, \ r \geq r^0. \quad (5)$$

The Nash bargaining framework has already been proposed for fair bandwidth allocation for elastic services in wired networks by Yaïche et. al. [10]. The important difference between the Nash bargaining framework (the framework used in [10]) and our CSMA/CA game is that the set of feasible payoffs R is neither compact nor convex in our CSMA/CA game. Nevertheless, we will show that the optimization problem (5) with R being the (non-convex and non-compact) set of feasible payoffs of our CSMA/CA game has a unique solution exhibiting Pareto-optimality and fairness properties (similar to the properties of the Nash bargaining solution of the problem (5) where R is a convex and compact set [20]). By taking the logarithm of the objective function of (5) we

---

[3] Assuming that the number of cheaters is not "too high".

obtain the equivalent optimization problem $\boldsymbol{\Pi_1}$ [20], [10]:

$$\boldsymbol{\Pi_1}: \quad \begin{aligned} z_1 = & \max \sum_{i=1}^{C} \log(r_i - r_i^0) \\ \text{s.t.} \quad & r = f(W) \\ & r \geq r^0 \\ & W \in S_1 \times S_2 \times \ldots \times S_C \ . \end{aligned}$$

Define a set $\Sigma \equiv \{\Sigma_k : \Sigma_k = \sum_{i \in \mathcal{C}} f_i(W), \ W \in S_1 \times S_2 \times \ldots \times S_C, \ k = 1, 2, \ldots, W_{max}^C\}$. Note that some $\Sigma_k \in \Sigma$ will have the same value (e.g., for $C = 3$, vectors $W = (2, 3, 5)$ and $\hat{W} = (5, 3, 2)$ are equivalent with respect to the operator $\sum_{i \in \mathcal{C}} f_i(\cdot)$). Next we relax the constraints in the problem $\boldsymbol{\Pi_1}$ to obtain the following optimization problem $\boldsymbol{\Pi_2}$:

$$\boldsymbol{\Pi_2}: \quad \begin{aligned} z_2 = & \max \sum_{i=1}^{C} \log(r_i - r_i^0) \\ \text{s.t.} \quad & \sum_{i=1}^{C} r_i \in \Sigma \\ & r \leq \mathbf{B} \\ & r \geq r^0 \ , \end{aligned}$$

where $r_i, \forall i \in \mathcal{C}$, are now continuous variables and $\mathbf{B}$ is a vector containing C elements, all of which are equal to B (where B is the maximum data rate achievable by any cheater). We observe the following relationship between the optimal values of $\boldsymbol{\Pi_1}$ and $\boldsymbol{\Pi_2}$: $z_2 \geq z_1$. This follows from the fact that $\boldsymbol{\Pi_2}$ is a relaxed version of $\boldsymbol{\Pi_1}$. A possible way to solve $\boldsymbol{\Pi_2}$ is to solve one instance of it for each $\Sigma_k \in \Sigma$ and then simply pick the instance that maximizes the corresponding objective function. Thus, for an arbitrary instance (an arbitrary $\Sigma_k \in \Sigma$) the first constraint simplifies to $\sum_{i=1}^{C} r_i = \Sigma_k$. To solve $\boldsymbol{\Pi_2}$ we define the corresponding Lagrangian as follows:

$$\begin{aligned} L(r, \lambda, \alpha, \beta) = & \\ = & \sum_{i=1}^{C} \log(r_i - r_i^0) - \lambda \Big(\sum_{i=1}^{C} r_i - \Sigma_k \Big) - \\ & - \sum_{i=1}^{C} \alpha_i(r_i^0 - r_i) - \sum_{i=1}^{C} \beta_i(r_i - B) \ . \end{aligned}$$

From the Karush-Kuhn-Tucker first-order necessary conditions [21], we have:

$$\frac{\partial L}{\partial r_i} = \frac{1}{r_i - r_i^0} - \lambda + \alpha_i - \beta_i = 0, \quad i = 1, \ldots, C \ ,$$

and

$$\begin{aligned} & \lambda \Big(\sum_{i=1}^{C} r_i - \Sigma_k \Big) = 0, \quad \lambda \geq 0 \\ & \alpha_i(r_i^0 - r_i) = 0, \quad\quad\quad \alpha_i \geq 0, \quad i = 1, \ldots, C \\ & \beta_i(r_i - B) = 0, \quad\quad\quad \beta_i \geq 0, \quad i = 1, \ldots, C \ . \end{aligned}$$

Now, it is easily seen that $\alpha_i = \beta_i = 0, \ i = 1, \ldots, C$. This follows from the fact that there exists a feasible vector W such that that the optimal value of the equivalent to $\boldsymbol{\Pi_2}$, i.e., $\prod_{i=1}^{C}(r_i - r_i^0)$ is strictly positive. Likewise, $r_i = B$ implies $r_j = r_j^0(= 0), \ \forall j \neq i$, which in turn implies $\prod_{i=1}^{C}(r_i - r_i^0) = 0$. Then, the first-order Kuhn-Tucker conditions reduce to:

$$\frac{1}{r_i - r_i^0} - \lambda = 0, \ i = 1, \ldots, C \ . \tag{6}$$

Plugging (6) into the first condition of the corresponding instance of $\boldsymbol{\Pi_2}$, i.e., $\sum_{i=1}^{C} r_i = \Sigma_k$, we finally obtain:

$$r_i = r_i^0 + \frac{\Sigma_k - \sum_{i=1}^{C} r_i^0}{C}, \ i = 1, \ldots, C \ . \tag{7}$$
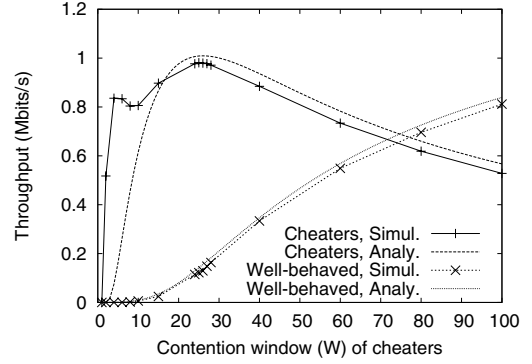

Fig. 2. Throughput vs. contention window size of the cheaters (20 nodes, out of which 10 cheaters)

Therefore, any optimal solution to the problem $\boldsymbol{\Pi_2}$ is characterized by the following **fairness property**: *Each cheater should receive r_i^0 (zero in the case of the CSMA/CA game) and the remaining capacity $\Sigma_k - \sum_{i=1}^{C} r_i^0 = \sum_{i=1}^{C}(f_i(W) - r_i^0)$, for a given W, is split equally among C cheaters.* In our game, this means that each cheater should receive equal share of $\sum_{i=1}^{C} f_i(W^*)$, where W^* designates any optimal solution. In other words, any optimal solution of $\boldsymbol{\Pi_2}$ satisfies: $W_i^* = W_j^*, \ i, j = 1, \ldots, C$.

We next show that problem $\boldsymbol{\Pi_2}$ admits a unique solution. In [17], Bianchi showed that given $W_i = W_j, \ i, j = 1, \ldots, C$, there exists a unique contention window size maximizing the system throughput. A similar phenomenon can be observed in our case: in Fig. 2 we plot the average aggregated throughput (the system throughput) obtained by 10 cheaters, all of which use the same contention window size; the simulation setup and the analytical model used here are described in Section IV-A. Therefore, we conclude that problem $\boldsymbol{\Pi_2}$ admits a **unique solution** $W_i^* = W_j^*, \ i, j = 1, \ldots, C$. It follows immediately that this is also the unique solution of problem $\boldsymbol{\Pi_1}$, that is, we simply set $r_i = f_i(W^*)$ in $\boldsymbol{\Pi_1}$ to get $z_1 = z_2$.

Finally, observe that W^* is a unique Pareto-optimal point, given the condition $W_i^* = W_j^*, \ i, j = 1, \ldots, C$; this follows readily from Fig. 2. Thus, the point W^* obtained as the unique solution to problem $\boldsymbol{\Pi_1}$ exhibits all the properties of a desirable point of operation in the CSMA/CA game. In our context, this is significant since: (i) W^* is not a Nash equilibrium point (i.e., $W_i^* > 1, \ i = 1, \ldots, C$), but it is Pareto optimal; (ii) $W_i = 1 \ (i = 1 \ldots C)$ is a Nash equilibrium point but is not Pareto optimal (efficient). This creates a dilemma for a system with multiple cheaters, where they look for a Nash equilibrium point of operation which is also Pareto optimal, fair and efficient.

V. DYNAMIC GAME

Having determined the Pareto-optimal point W^*, we now intend to devise a strategy allowing the cheaters to converge to this point. For this purpose, we make use of the theory of *dynamic and repeated games* [22], [16]. Using the dynamic game model, we devise a simple distributed algorithm that

cheaters can use to converge to a desired Nash equilibrium point. Again, we assume W_i (i.e., τ_i) to be a continuous variable.

A. Game formulation

We extend the game theory model introduced in Section IV-A to a dynamic game model in which the players are allowed to make their decisions based on previous actions and system states. We assume that our dynamic game is played infinitely long. We also assume all the nodes to be cheaters, i.e., $C = N$. Let $\mathcal{C}$ denote the set comprising all the cheaters. In the new game theoretic model the cheater's (player's) utility function J_i takes the following form:

$$J_i = U_i - P_i, \qquad (8)$$

where P_i denotes a penalty function. Let us assume, for the moment, that P_i is defined to be:

$$P_i = k_i(\tau_i - \underline{\tau}), \quad k_i \geq 0, \quad \underline{\tau} \in (0,1), \qquad (9)$$

with k_i and $\underline{\tau}$ are constants that we impose on player i. Note that we assume P_i can be negative; we will show in Section V-C that with an appropriate selection of k_i we always have $P_i \geq 0$ (i.e., $\tau_i \geq \underline{\tau}$). In this section, the definition of P_i, (as well as k_i and $\underline{\tau}$), is used as a mathematical tool to derive optimality conditions for the players; later in Sections V-C and VI-A, we discuss practical interpretations of values k_i and $\underline{\tau}$. Combining equations (8), (9) and (3) (definition of U_i) we can define the following non-cooperative game:

$$\max_{0 \leq \tau_i \leq 1} J_i = \frac{\tau_i c_1^i}{\tau_i c_2^i + c_3^i} - k_i(\tau_i - \underline{\tau}), \quad \forall i \in \mathcal{C} . \qquad (10)$$

In order to solve the maximization problem (10), we define the Lagrangian function $L(\tau_i, \lambda_i)$ for each player i as $L(\tau_i, \lambda_i) = J_i(\tau_i) + \lambda_i(1 - \tau_i)$ where $\lambda_i \geq 0$ is a Lagrangian multiplier. Based on the Lagrangian function $L(\tau_i, \lambda_i)$, we can obtain necessary and sufficient conditions for J_i to be maximized for each player $i \in \mathcal{C}$. It is known from the convex optimization theory (with inequality constraints) that in order for the Karush-Kuhn-Tucker first order conditions to be sufficient for J_i to be maximized, J_i has to be a concave function, whereas constraint $\tau_i \leq 1$ has to be convex, quasiconvex or linear [21]. Indeed, in our case $\tau_i \leq 1$ is a linear constraint in τ_i. Furthermore, using the approximation $T^s = T^c$ (see footnote 1, Section IV-A), J_i is a concave function in τ_i i.e., $\frac{\partial^2 J_i}{\partial \tau_i^2} = -\frac{2c_1^i c_2^i c_3^i}{(\tau_i c_2^i + c_3^i)^3} \leq 0$. Now, for each player i, the sufficient conditions for J_i to be maximized are:

$$\begin{aligned}
\tau_i\left(\frac{\partial J_i}{\partial \tau_i} - \lambda_i\right) &= 0 \quad \text{and} \quad \frac{\partial J_i}{\partial \tau_i} - \lambda_i \leq 0, \quad \tau_i \geq 0 \\
\lambda_i(\tau_i - 1) &= 0 \quad \text{and} \quad \lambda_i \geq 0, \qquad\qquad \tau_i \leq 1 .
\end{aligned} \qquad (11)$$

Solving system (11), we obtain the following optimality conditions for player i: (i) $\tau_i = 0$ if $\frac{\partial J_i}{\partial \tau_i} \leq 0$, (ii) $\tau_i \in (0,1)$ if $\frac{\partial J_i}{\partial \tau_i} = 0$, and (iii) $\tau_i = 1$ if $\frac{\partial J_i}{\partial \tau_i} \geq 0$. Let γ_i^* denote the solution to equation $\frac{\partial J_i}{\partial \tau_i} = 0$, i.e.,

$$\gamma_i^* = \frac{1}{c_2^i}\left(\sqrt{\frac{c_1^i c_3^i}{k_i}} - c_3^i\right) . \qquad (12)$$

Lemma 1: Assume $T^s = T^c$. Then for each player i the strategy prescribing: (i) play $\tau_i = 0$ if $\gamma_i^* < 0$, (ii) play $\tau_i = \gamma_i^*$ if $0 \leq \gamma_i^* \leq 1$, and (iii) play $\tau_i = 1$ if $\gamma_i^* > 1$ is a unique Nash equilibrium strategy for the game (10).

It is interesting to note that even if $T^s > T^c$, it still can be shown that by an appropriate selection of penalty constant k_i the optimality conditions of Lemma 1 still hold.

B. Equilibria of a dynamic game

Lemma 1 reveals an interesting point about our game (10): If we can find conditions for which $\gamma_i^* \in (0,1)$ for all players $i \in \mathcal{C}$, then no player i will ever play $\tau_i = 1$ (i.e., the network will not collapse). We extend this observation by the following proposition:

Proposition 3: Any profile $\tau = (\tau_1, \tau_2, \ldots, \tau_C)$, with $\tau_i \in (0,1)$, $\forall i \in \mathcal{C}$, can be made a Nash equilibrium point of the game (10).

The proof is provided in [18]. This result is similar to the *Nash Folk theorem* [16].

C. Reaching a Nash equilibrium point

Let us assume that we want to convert $\tau_i = \underline{\tau} \in (0,1)$, $\forall i \in \mathcal{C}$ to a unique Nash equilibrium point. Assume first that $T^s = T^c$. We know from Lemma 1 that at a Nash equilibrium, τ_i should satisfy equation (12). As we want to make $\underline{\tau}$ a Nash equilibrium, we simply set $\gamma^* = \underline{\tau}$. Then we rewrite (12) with $\gamma^* = \underline{\tau}$ and obtain:

$$k_i = \frac{c_1^i c_3^i}{(\underline{\tau} c_2^i + c_3^i)^2} . \qquad (13)$$

Thus, player i's payoff function becomes:

$$J_i = \frac{\tau_i c_1^i}{\tau_i c_2^i + c_3^i} - \frac{c_1^i c_3^i}{(\underline{\tau} c_2^i + c_3^i)^2}(\tau_i - \underline{\tau}) . \qquad (14)$$

Note that at the Nash equilibrium point $\tau_i = \underline{\tau}$, $\forall i \in \mathcal{C}$, the following holds: $J_i(\underline{\tau}) = r_i(\underline{\tau})$. It is interesting to note that even if P_i takes negative values for $\tau_i < \underline{\tau}$, in which case P_i can be seen as a reward function, the best choice for the node is still $\tau_i = \underline{\tau}$, where $P_i = 0$. This suggests the following redefinition of player i's payoff function J_i:

$$J_i = r_i - k_i \times \begin{cases} (\tau_i - \underline{\tau}), & \tau_i > \underline{\tau}; \\ 0, & \tau_i \leq \underline{\tau} . \end{cases} \qquad (15)$$

Therefore, by an appropriate selection of k_i, we have made $\underline{\tau}$ a unique Nash equilibrium for all players $i \in \mathcal{C}$. Next we derive an algorithm that leads the players to the unique Nash equilibrium point $\tau_i = \underline{\tau}$, $\forall i \in \mathcal{C}$. The first issue to be resolved is the following: How do the players agree on the Nash equilibrium point $\tau_i = \underline{\tau}$, $\forall i \in \mathcal{C}$, to which they all should converge? A simple way to resolve this is to define:

$$\underline{\tau} \equiv \min_{i \in \mathcal{C}} \tau_i, \qquad (16)$$

in which case P_i can be seen as the penalty that the player with the lowest accessing probability (the highest contention window) inflicts on player i. Let k denote an arbitrary player such that $\tau_k = \underline{\tau}$, that is, $k \in \{\arg\min_{i \in \mathcal{C}} \tau_i\}$. Observe that

$\frac{\partial r_k}{\partial \tau_i} < 0$, $\forall i \neq k$ (assuming that $\tau_i < 1$, $\forall i \neq k$). This property and the fact that $k \in \{\arg\min_{i \in \mathcal{C}} \tau_i\}$ inspire the following joint optimization problem $\mathbf{\Pi_3}$ for all $i \neq k$:

$$\mathbf{\Pi_3}: \quad \text{s.t.} \quad \begin{array}{c} \max r_k(\tau_k, \tau_{-k}) \\ \underline{\tau} \leq \tau_i \leq 1, \ \forall i \in \mathcal{C}\backslash\{k\} \\ \tau_k = \underline{\tau} \end{array} \quad (17)$$

It can easily be proven that by replacing the objective function r_k in $\mathbf{\Pi_3}$ with $\prod_{i \in \mathcal{C}\backslash\{k\}}(1 - \tau_i)$, we obtain the equivalent problem, i.e., optimal τ_i's (where $i \neq k$) for the one problem are also optimal for the other problem. This is because: (i) $r_k = (\tau_k \prod_{i \in \mathcal{C}\backslash\{k\}}(1 - \tau_i)\overline{L})/(\tau_k c_2^k + c_3^k)$ and $\prod_{i \in \mathcal{C}\backslash\{k\}}(1 - \tau_i)$ exhibit similar behaviors in τ_i ($i \neq k$) (i.e., both are strictly decreasing in $\tau_i < 1$ ($i \neq k$) and zero if $\exists j$ such that $\tau_j = 1$), and (ii) τ_k is a constant equal to $\underline{\tau}$. Furthermore, by taking the logarithm of $\prod_{i \in \mathcal{C}\backslash\{k\}}(1 - \tau_i)$ (as in Section IV-C) we arrive at the equivalent optimization problem $\mathbf{\Pi_4}$:

$$\mathbf{\Pi_4}: \quad \text{s.t.} \quad \begin{array}{c} \max \sum_{i \in \mathcal{C}\backslash\{k\}} \log(1 - \tau_i) \\ \underline{\tau} \leq \tau_i \leq 1, \ \forall i \in \mathcal{C}\backslash\{k\} \end{array} \quad (18)$$

We next write the Lagrangian for $\mathbf{\Pi_4}$ as follows:

$$L(\tau_{-k}, \alpha, \beta) = $$
$$= \sum_{i \in \mathcal{C}\backslash\{k\}} \left\{ \log(1 - \tau_i) - \alpha_i(\underline{\tau} - \tau_i) - \beta_i(\tau_i - 1) \right\} .$$

From the Karush-Kuhn-Tucker first order necessary conditions we have $\beta_i(\tau_i - 1) = 0$, $\beta_i \geq 0$, $\forall i \in \mathcal{C}\backslash\{k\}$. This implies $\beta_i = 0$, $\forall i \in \mathcal{C}\backslash\{k\}$, since $\tau_i < 1$, $\forall i \in \mathcal{C}\backslash\{k\}$ (i.e., there exists strictly positive solutions to $\mathbf{\Pi_4}$). Thus, the above Lagrangian reduces to $L(\tau_{-k}, \alpha) = \sum_{i \in \mathcal{C}\backslash\{k\}} l(\tau_i)$, where $l(\tau_i) \equiv \log(1 - \tau_i) - \alpha_i(\underline{\tau} - \tau_i)$. Since $l(\tau_i)$ is strictly concave and twice continuously differentiable, $l(\tau_i)$ admits a unique solution obtained from: $\frac{dl(\tau_i)}{d\tau_i} = 0$. Thus we have:

$$\tau_i^{opt} = 1 - \frac{1}{\alpha_i}, \ \forall i \in \mathcal{C}\backslash\{k\} . \quad (19)$$

In order to obtain a distributed algorithm that leads the players to the Nash equilibrium $\underline{\tau}$, we finally consider the dual problem $\mathbf{\Pi_5}$ of $\mathbf{\Pi_4}$ as follows:

$$\mathbf{\Pi_5}: \quad \min_{\alpha \geq \mathbf{0}} L(\tau_{-j}^{opt}, \alpha) .$$

Now we can solve the dual problem $\mathbf{\Pi_5}$ by using the following gradient-based method [23]:

$$\begin{aligned} \alpha_i(t+1) &= \alpha_i(t) - \gamma \frac{\partial L(\tau_{-j}^{opt}, \alpha)}{\partial \lambda_i} \\ &= \alpha_i(t) - \gamma(\tau_i^{opt}(t) - \underline{\tau}), \ \forall i \in \mathcal{C}\backslash\{k\}, (20) \end{aligned}$$

where $\gamma > 0$ is a step size, and where the differentiability of $L(\tau_{-j}^{opt}, \lambda)$ follows from the fact that the objective function of problem $\mathbf{\Pi_4}$ is strictly concave[4] [23].

The gradient method based algorithm (20) works as follows. Whenever for some player $i \neq k$ we have $\tau_i^{opt}(t) > \underline{\tau}$ or $\tau_i^{opt}(t) < \underline{\tau}$, the corresponding $\alpha_i(t)$ is updated to $\alpha_i(t+1)$ and communicated to player i. Having received $\alpha_i(t+1)$,

---

[4]Strict concavity does not hold if $\exists i \neq k$ such that $\tau_i = 1$, however, we know that $\tau_i^{opt} < 1$ for all $i \neq k$.

player i recalculates his optimal access probability as follows: $\tau_i^{opt}(t+1) = 1 - 1/\alpha_i(t+1)$. Here, we assumed that $\alpha_i(t)$ for all $i \neq k$ (and for all $t \geq 0$) are controlled by some *oracle*. Later in this section, we will show how the role of the oracle can be delegated to the players themselves. We next study the convergence of algorithm (19)-(20).

Proposition 4: Assume $\underline{\tau} \in [\epsilon, 1 - \hat{\epsilon}]$, for some $0 < \epsilon, \hat{\epsilon} < 1$ and $1 - \hat{\epsilon} > \epsilon$. Then for $0 < \gamma < 1$, algorithm (19)-(20) converges to the unique Nash equilibrium $\tau_i = \underline{\tau}$, $\forall i \neq k$, for any initial access probability point $\epsilon \leq \tau_i \leq 1 - \hat{\epsilon}$, $\forall i \neq k$ (with $\alpha_i(0) = \frac{1}{1 - \tau_i(0)}$, for all $i \neq k$).
The proof is included in the Appendix.

We saw that the two roles of the oracle are: (i) to correctly estimate the $\alpha_i(t)$, $\forall i \neq k$, and (ii) to communicate $\alpha_i(t)$ afterwards. The first task requires a *detection mechanism* that is capable of detecting *deviations* from $\underline{\tau}$. As far as the second task is concerned, it is easily seen that it takes only an appropriate *penalization mechanism*. Indeed, by appropriately choosing the step size that governs updates of values $\alpha_i(t)$ and $\tau_i(t)$ in (19)-(20), we can focus solely on deviations where $\tau_i > \underline{\tau}$ (i.e., where a penalty is needed; in Section VII-A we implement such an *adaptive strategy*). Note, however, that a deviating player i should still receive penalty P_i; in order to make $\underline{\tau}$ a unique maximizer of player i's payoff function J_i. We describe and study a possible implementation of the two mechanisms in Section VI.

As can be seen from the payoff function (14), in order to evaluate the correct penalty to be inflicted on some node i, another node j has to calculate (estimate) the values c_k^i, $k = 1, 2, 3$, effectively meaning that the node j has to estimate, at the same time, the access probabilities of all the other nodes. In Section VI-A, we show how to alleviate this requirement by introducing a simpler penalty function that preserves all the important properties of the penalty function given by (9).

We now address the remaining challenge of how to achieve the most efficient $\underline{\tau}$, i.e., the Pareto-optimal point in the sense of Section IV-C.

D. Moving Nash equilibrium

To make the Pareto-optimal point W^* a Nash equilibrium, we use the fact that W^* is the only point satisfying the first-order necessary conditions (see Fig. 2). Let τ^* denote the corresponding optimal access probability vector.

Initially, cheaters set their access probabilities to some value close to 1 (but strictly smaller than 1), i.e., $\tau(0) < \mathbf{1}$. The cheaters also have to make sure that $\tau^* \leq \tau(0)$. The cheaters then start running algorithm (19)-(20) until they stabilize at some Nash equilibrium point $\tau_i^n(0) = \min_{i \in \mathcal{C}} \tau_i(0)$, $\forall i \in \mathcal{C}$. One player (cheater), say i, will eventually decrease its current τ_i by some small value ϵ, i.e., $\tau_i(1) = \tau_i^n(1) - \epsilon$. This will in turn trigger the penalizing mechanism of player i and pull the other players to a new Nash equilibrium point, since they also run algorithm (19)-(20). At this stage, each player j will compare its current payoff $J_j(1)$ (at the new Nash equilibrium point) to the payoff $J_j(0)$ achieved at the previous Nash equilibrium point. If $|J_j(1) - J_j(0)| < \delta$, where $\delta > 0$ is some

appropriately chosen small value, the players terminate their search, since they have reached close proximity of the Pareto-optimal point τ^* (i.e., W^*). Otherwise, the players repeat the previous steps. Since every point at which the players stabilize is a Nash equilibrium point (Proposition 4), we have achieved our goal of making the Pareto-optimal point W^* (or some point close to W^*) a Nash equilibrium point.

One important point about the procedure just described is that it requires the players to deviate from a Nash equilibrium point to reach the Pareto equilibrium point W^*. At first sight this may look confusing: Why should a player deviate from a Nash equilibrium point? The answer is simple. The game in our new model is played infinitely long (an infinitely repeated game). For this reason, the players can afford any finite number of deviations from a Nash equilibrium point (i.e., for $0 < k < \infty$, $\lim_{T \to \infty} \sum_{t=k}^{T} J_i(t)/T = \lim_{T \to \infty} \sum_{t=0}^{T} J_i(t)/T$).

VI. DEVIATION DETECTION AND PENALIZATION

In this section, we propose a distributed and practical implementation of the two key building blocks for the model of dynamic games, namely, the penalization mechanism and the detection mechanism. Before describing the two mechanisms, we introduce (and analyze) a penalty function simpler than the one given by (9), and preserve its most important property (notably, the uniqueness of the maximizer).

A. Practical interpretation of the penalty function P_i

Let $\tau_e \in (0, 1)$ denote a desirable equilibrium value (point) and let $p_i(\tau) > 0$ be some positive continuously differentiable function. Let us consider the following function $\Phi_i(\tau)$:

$$\Phi_i = \begin{cases} p_i, & \text{if } \tau_i > \tau_e \\ 0, & \text{otherwise .} \end{cases} \quad (21)$$

Proposition 5 (Sufficient condition): Let the access probabilities of cheaters other than i (τ_{-i}) be fixed and strictly lower than 1 ($\tau_{-i} < 1$). If $\frac{\partial p_i}{\partial \tau_i} > \frac{\partial r_i}{\partial \tau_i}$, then the payoff function $J_i(\tau_i) = r_i(\tau_i) - \Phi_i(\tau_i)$ has a unique maximizer $\tau_i^* = \tau_e$.

Proof: Since $\tau_{-i} < 1$, we have $\partial r_i / \partial \tau_i > 0$ for $\tau_i \in [0, 1]$. Therefore, on the interval $[0, \tau_e]$, $\tau_i = \tau_e$ is the unique maximizer of the utility function $J_i(\tau_i) = r_i(\tau_i)$. We conclude the proof by observing that for the remaining interval $(\tau_e, 1]$ we have $\partial J_i / \partial \tau_i = \partial r_i / \partial \tau_i - \partial p_i / \partial \tau_i < 0$, i.e., $J_i(\tau_i)$ is a strictly decreasing function in $\tau_i \in (\tau_e, 1]$. ∎

Consider node j that calculates the penalty P_i to be inflicted on node i as follows:

$$P_i = \Phi_i, \text{ with } p_i = r_i - r_j \text{ and } \tau_e = \tau_j . \quad (22)$$

Assume $\tau_{-i} < 1$. Then $\partial p_i / \partial \tau_i = \partial r_i / \partial \tau_i - \partial r_j / \partial \tau_i$. Since $\partial r_i / \partial \tau_i > 0$ and $\partial r_j / \partial \tau_i < 0$, it follows that $\partial p_i / \partial \tau_i > \partial r_i / \partial \tau_i$. We thus see that the penalty function P_i given by (22) satisfies the sufficient condition of Proposition 5. Therefore, for fixed $\tau_{-i} < 1$, $\tau_i = \tau_j$ is a unique maximizer of the player i's utility function $J_i(\tau_i) = r_i(\tau_i) - P_i(\tau_i)$. Thus, similarly to the the penalty function (9), the new penalty function (22) allows us to make any point $\underline{\tau} = \min_{k \in \mathcal{C}} \tau_k$

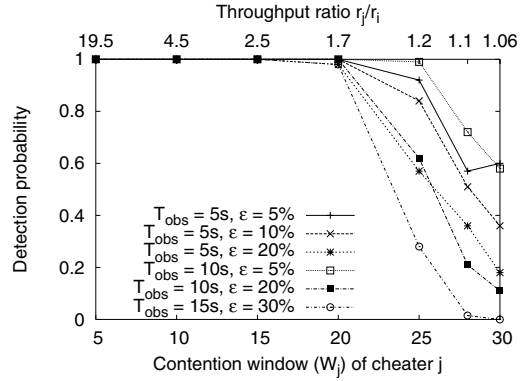

Fig. 3. Performance of cheating detection based on throughput measurements

a Nash equilibrium point (i.e., for each player i set $\tau_e = \underline{\tau}$ and $p_i = r_i - r_j$, $j \in \{\arg\min_{k \in \mathcal{C}} \tau_k\}$).

In the context of two cheaters i and j, a very important property of the new penalty function is that a penalty has to be such that it results in the same throughputs for both cheater i and cheater j. Indeed, $J_i = r_i - P_i = r_j = J_j$, where J_i and J_j represent the resulting throughputs. We will see shortly (in Section VI-C) that this property of "throughput equalization" requires far less information at the side of the cheater who inflicts a penalty, when compared to the penalty function (9).

In what follows, we first describe the detection mechanism helping each cheater j to assess if some other cheater i is receiving a higher throughput (Section VI-B). If this is the case, we say that cheater i has *deviated* from cheater j's point of view. Then, in Section VI-C, we describe the penalizing mechanism that the cheaters use against the deviating cheater(s) such that their throughputs become equal.

B. Detection mechanism

In our approach, each cheater measures the throughput of each other node[5], including itself. This is indeed feasible due to the broadcast nature of the wireless medium. If a cheater observes a difference in throughput with some other node, it characterizes that node as a deviating cheater. Let r_i and r_j be the measured throughput of nodes i and j, respectively. Due to the inherent short-time unfairness of the IEEE 802.11 MAC protocol [24], and in order to increase the efficiency of the detection mechanism, we introduce two parameters: the observation time-window size T_{obs}, in seconds, and the tolerance margin ϵ, in percentage of throughput. After measuring the throughput of each node for T_{obs} seconds, node i concludes that node j is deviating whenever the throughput of node j exceeds the throughput of node i i.e., $r_j / r_i > 1 + \epsilon$.

We have implemented this detection mechanism in *ns-2*, with $N = C = 30$ nodes. We vary the contention window size (W_j) of a single node j, and set others' contention window sizes to 30 ($W_{-j} = 30$). Fig. 3 shows the performance of the detection mechanism for different values of T_{obs} and ϵ. The

---

[5]We deliberately use the word *node*, since in reality well-behaved nodes may be present as well (even though we neglect them in the analysis).

probability of false positives corresponds to the probability of detection with $W_j = 30$; at this point, node j uses a contention window value equal to that of node i, but still gets a higher throughput, $r_j/r_i = 1.06$, due to the IEEE 802.11 unfairness. Therefore, node j gets detected as deviating with positive detection probability. To reduce the false positives (at contention window size 30), one can consider large ϵ values ($> 10\%$). However, this comes at the expense of lower detection probabilities if cheater j uses contention window sizes slightly lower than 30. Similarly, large T_{obs} values ($\geq 15s$) will reduce the effect of the inherent IEEE 802.11 unfairness, and therefore the corresponding false positives. This also comes at the expense of lower detection probabilities if cheater j uses contention window sizes slightly lower than 30. Therefore, choosing appropriate values for T_{obs} and ϵ is crucial to our detection mechanism. For very low contention window sizes of cheater j ($W_j \leq 20$), the throughput ratio r_j/r_i is much larger than $1+\epsilon$, making detection of the cheater j's deviation easy.

C. Penalizing mechanism

The action taken by cheaters in response to non-cooperation (deviation) by another cheater is termed as a penalizing mechanism. In Section V, we expressed the cost of this penalizing scheme (imposed by another cheater) on a cheater by the penalty function. We note that a penalizing scheme should be designed so that it does not bring any performance degradation to penalizing cheaters. This can be achieved by *selectively jamming* the deviating cheater's packets. Indeed in CSMA/CA networks, we can single out a deviating cheater for punishment, since only one station (within the same collision domain) can transmit at a time. In game theory, this property is know as *full dimensionality* [16].

We have designed a simple penalizing scheme, in which the packets of the non-cooperative cheater are selectively jammed for a short duration of time, T_{jam}, by the other cheaters in the system. Suppose cheater i detects the presence of a non-cooperative (deviating) cheater j. Thereafter, if cheater i listens to a transmitted packet corresponding to cheater j, it switches to transmission mode and jams enough bits so that the packet cannot be properly recovered at the receiver. Meanwhile, all other cheaters in the system should be able to read the header of the jammed frame and properly update their NAV (net allocation vector). This is to avoid waiting for EIFS [14], which would reduce the system's efficiency. Therefore, jamming should be done on frame payloads rather than frame headers. This is indeed possible since the transceiver's turnaround time, which is on the order of 5 μs [14], is much shorter than the data frame transmission time, which is of the order of 700 μs (depending on the data rate and the packet size).

Let the throughput obtained by the two considered cheaters over the last observation window, T_{obs}, be r_i and r_j, respectively, where $r_j/r_i > 1 + \epsilon$. As we saw in Section VI-A, the penalizing mechanism is aimed at making the throughputs received by cheaters i and j equal. Thus, the throughput
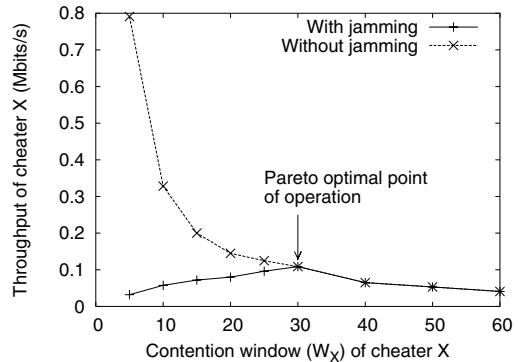

Fig. 4. Unilateral deviation by a cheater before and after the introduction of the penalizing mechanism

received by cheaters i and j should be the same over the total time duration of $T_{obs} + T_{jam}$, i.e., $r_i T_{obs} + r_i T_{jam} = r_j T_{obs} + 0 T_{jam}$. From this, cheater i calculates T_{jam} as follows:

$$T_{jam} = \left(\frac{r_j}{r_i} - 1\right) T_{obs} . \qquad (23)$$

To avoid situations in which $T_{jam} = \infty$, in practice, we can define $T_{jam} = \min\{\overline{T}_{jam}, (r_j/r_i - 1)T_{obs}\}$, where $\overline{T}_{jam} > 0$ is a sufficiently large value to bring r_j close to $r_i = 0$. In this way, we give a chance to the deviating cheater j to properly adjust its contention window W_j and thus its throughput r_j.

We have implemented the jamming mechanism in *ns-2*. The simulation setup is the same as in Section IV-A ($N = 20$, $C = 10$). We randomly pick up a cheater, designated as cheater X, and fix his contention window size to be 10. The contention window size for all the other cheaters in the system is fixed to the Pareto-optimal (W^*) value of 30. We use the observation window size, T_{obs}, of 20 seconds, and the tolerance margin, ϵ, of 5% in the detection mechanism. Fig. 4 plots the average throughput obtained by cheater X, when it unilaterally deviates from the Pareto-optimal point of operation ($W^* = 30$). The results are averaged over a duration of 1000 seconds. As can be observed from Fig. 4, after the introduction of the detection and penalizing mechanism, cheater X achieves maximum throughput by operating at the Pareto-optimal point of operation, which is consistent with the result of Proposition 5 applied to the penalty function (22). Thus, any unilateral deviation from this point, $W_X < W^*$ or $W_X > W^*$, brings less payoff (throughput) for cheater X. Therefore, no cheater will have any incentive for deviating unilaterally from the Pareto-optimal point of operation and hence it is at Nash equilibrium.

VII. DISTRIBUTED COORDINATION PROTOCOL

Based on the two building blocks described in the previous section, we will now build a comprehensive and distributed protocol that guides multiple selfish nodes to the Pareto-optimal Nash equilibrium point W^*.

	Strategy	
	Non-adaptive	Adaptive
Cheater X	7650	11577
Other cheaters	7826	11448
Well-behaved nodes	1286	2318

Fig. 5. System performance with varying levels of misbehavior

	Strategy	
	Non-adaptive	Adaptive
Cheater X	2843	10356
Cheater Y	2686	10185
Cheater Z	2565	10239
Other cheaters	2544	10172
Well-behaved nodes	270	1981

A. Adaptive strategy

Inspired by the algorithm (19)-(20), we have implemented the following adaptive strategy. When cheater i observes that he is being jammed (penalized) during some period Δ, he gradually increases his contention window by steps of size γ. Note that a cheater can easily decide whether he is being jammed by observing his own throughput. The choice of Δ determines the efficiency of the system. For example, choosing a small value of Δ might magnify the effect of misdetection by unnecessarily causing a cheater to increase his contention window size. This will eventually lead the whole system towards an inefficient point of operation. The choice of the step size, γ, offers a tradeoff between convergence time and efficiency. If we increase the contention window in large steps, though the system will stabilize in less time, the point of operation might be far away from the Pareto-optimal point (W^*), resulting in an inefficient system and vice-versa.

We have implemented this adaptive strategy in *ns-2*. The simulation setup is the same as in the previous section ($N = 20, C = 10, W^* = 30$). We randomly pick up a cheater, designated as node X, and fix his contention window size to 10. The contention window size for all the other cheaters in the system is fixed to W^*. We fix Δ to be 5 seconds and γ to be 5. One can observe (cf. [18]) how node X adapts its contention window size by following the adaptive strategy and eventually converging to a window size of 30, equal to W^*. Thus the other cheaters in the system are successful in guiding the misbehaving cheater to reach the optimal point of operation. Table I summarizes the throughput averages obtained by different nodes over a time interval of 1000 seconds. As can be observed from Table I, the jamming and detection mechanism combined with the adaptive strategy, besides being fair to all the cheaters in the system, is also the most efficient. Note that even the introduction of the adaptive strategy does not encourage the abuse of jamming. Cheater X might try to unnecessarily jam other cheaters, hoping that an increase in the contention window sizes by all the cheaters (following the adaptive strategy) will get him more throughput. However, eventually cheater X will be identified as a misbehaving cheater, because of the throughput difference, by the other cheaters in the system. In turn, cheater X will be forced to increase his own contention window size, due to the penalizing mechanism. As a result, every cheater i, including cheater X, will now be operating at an inefficient point of operation ($W_i > W^*$). Thus cheaters have no incentive to
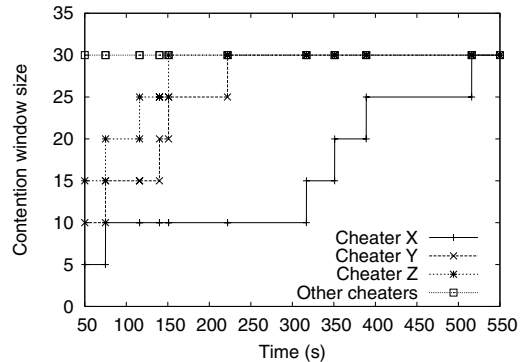
over-jam other cheaters in the system.

Finally, we evaluate the performance of our protocol in a scenario consisting of multiple levels of misbehavior in the system. The simulation setup is the same as in the previous section ($N = 20, C = 10, W^* = 30$). We randomly pick up three cheaters, designated as node X, Y and Z. We fix their contention window sizes to be 5, 10 and 15, respectively. The contention window size for all the other cheaters in the system is fixed to W^*. Fig. 5 plots the evolution of the contention window sizes of the different cheaters over time. Since each deviating cheater is punished in proportion to its misbehavior, each of them eventually converges to W^* and the system continues to operate at this stable point of operation. Thus, our protocol self-adapts to the different levels of misbehavior in the system. As can be observed from Table II, the jamming mechanism combined with the adaptive strategy results in the optimal and fair performance, even with multiple levels of misbehavior in the system.

B. Reaching the Pareto-optimal point

An accurate implementation of the detection, penalizing and adaptive strategy will lead the nodes to reach an equilibrium point, $\underline{W}$. However, the intention is to reach the Pareto point of operation, W^*. As we described in Section V-D, this can be achieved by using a simple gradient climbing algorithm. At the onset of the system, assume $W_i = W_i^{in}$ for all cheaters. Every cheater sets up a random timer (in our simulations this corresponds to a random value between 0 and 20 seconds) to increase his contention window by a step of size γ. One of
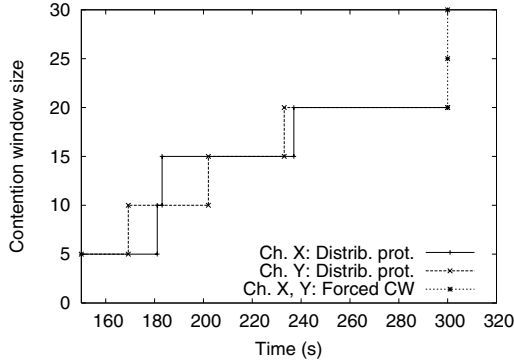
(a) Evolution of the contention windows
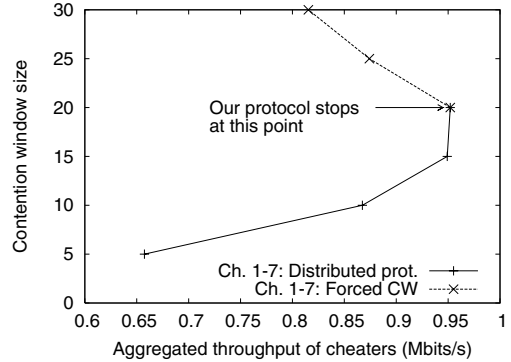
(b) Contention window vs. Average throughput

Fig. 6. Performance of the distributed coordination protocol, with $N = 20$ and $C = 7$. The axes in (b) are swapped for the convenience of matching them with (a)

the cheaters, say X, will eventually increase his contention window size to $W_X^{in} + \gamma$. Based on the detection mechanism, node X will conclude that all other cheaters in the system are deviating and will begin penalizing them. If a cheater observes that he is being penalized, he will disable the timer. Eventually the system will stabilize, when $W_i = W_i^{in} + \gamma$ for all cheaters. The cheaters realize that they have reached a new stable point of operation, when they all begin enjoying the same throughput (in our implementation, the cheaters remain at this stable point for 20 seconds before continuing the search for W^*). At this point in time, a cheater i compares his throughput at $W_i = W_i^{in} + \gamma$ with the throughput at $W_i = W_i^{in}$; if he observes a decrease in his throughput, he will terminate the search for W^*. Otherwise he again sets up the random timer to increase his contention window size by γ. Note that even if only one cheater observes an increase in his throughput, eventually the whole system will move towards a new point of operation.

We have implemented this protocol in *ns-2*. The simulation setup consists of 20 nodes and 7 cheaters ($N = 20, C = 7$). The cheaters initialize their contention window size to 5 ($W_i^{in} = 5$). The cheaters continue their search for W^* only if they see an increase of 10% or more in their throughput from the last stable point of operation. Fig. 6(a) plots a sample evolution of the contention window for 2 arbitrary chosen cheaters, X and Y, in the system. Note that all of the cheaters follow a similar pattern and eventually converge to a window size of 20. We are unable to show their evolution in the same plot as it simply generates overlapping lines. Fig. 6(b) plots the aggregated throughput obtained by all the cheaters at different contention window sizes. As can be seen from Fig. 6(b), the throughput is maximized at $W_i = 20$. In reality, the cheaters will stop at W_i equal to 20 and the system will continue to operate at this point of operation. For completeness, we obtain the "dotted" curve in Fig. 6(b) by deliberately forcing the cheaters to go beyond $W_i = 20$.

C. Discussion

As can be seen from Table II (Section VII-A), well-behaved nodes, which continue to follow the IEEE 802.11 protocol, obtain negligible throughput from the system. Thus, in the presence of cheaters in the system, we speculate that eventually all the nodes in the system will start to behave as cheaters. As they strive for optimal contention window size, even in such a scenario, our protocol will be at least as efficient as the normal IEEE 802.11 protocol. Note that our goal in this paper is prevent network collapse, in an efficient way, rather than finding the optimal contention window for a network of $N = C$ nodes [25], [17]. However, our adaptive cheating algorithm leads to the same optimal point, without explicit knowledge of the number of contending nodes. As mentioned earlier, we adopted throughput-based detection to simplify the presentation. However, the use of more adequate detection mechanisms, such as *backoff-based* detection [1] (i.e., comparing the nodes' average backoffs) is needed in general, for example in the following two scenarios.

Hidden terminal problem. A challenge here is that a node belonging to two different clusters, hidden from each other, will suffer unfair shares in both clusters. To cope with this problem, we can take a similar approach as in the present work. Thus, we first define (axiomatically) a desirable point of operation that will exhibit a reasonable fairness metric, and then we find a distributed algorithm that reaches this point. In general topology networks, it may be plausible to use backoff-based detection mechanisms [1].

Different traffic constraints. Since CSMA/CA is time-fair rather than throughput-fair, flows with different constraints (e.g., different packet lengths) will result in different throughputs, even without cheating. Hence, throughput-based detection mechanisms cannot be applied since nodes are not supposed to know each other's traffic constraints. Again, backoff-based detection mechanisms are more appropriate.

VIII. Conclusions

In this paper we have addressed the problem of cheating in CSMA/CA networks. For this purpose, we have developed a game-theoretical model and verified our findings by appropriate simulations. We have made several contributions. First, we have provided a formalism for the systematic study of rational cheating. Second, we have studied the simple cases (i) of a single cheater and (ii) of several cheaters acting without restraint. Third, we have identified the Pareto-optimal point of operation of a network with multiple cheaters. Fourth, we have shown how it is possible to transform this Pareto-optimal point into a Nash equilibrium. Fifth, we have shown that smart cheaters can collectively find this point. We believe these contributions to be very relevant in wireless networks.

In terms of future work, we intend to study in more detail general topology networks (involving hidden terminals). We will also try to define a punishment technique that is less intrusive than jamming. Finally, we intend to adapt this approach to problems beyond CSMA/CA networks: a possible direction is to study how smart cheating could become a technique to collectively fine-tune the behavior of a protocol.

Acknowledgment

References

[1] M. Raya, J.-P. Hubaux, and I. Aad, "DOMINO: A system to detect greedy behavior in IEEE 802.11 hotspots," in *ACM MobiSys*, June 2004.

[2] A. B. MacKenzie and S. B. Wicker, "Stability of multipacket slotted Aloha with selfish users and perfect information," in *Proceedings of IEEE INFOCOM*, 2003.

[3] E. Altman, R. El Azouzi, and T. Jiménez, "Slotted Aloha as a stochastic game with partial information," in *Proceedings of WiOpt*, 2002.

[4] Y. Jin and G. Kesidis, "Equilibria of a noncooperative game for heterogeneous users of an ALOHA network," *IEEE Comm. Letters*, vol. 6, 2002.

[5] Kyasanur, P. and Vaidya, N., "Detection and handling of MAC layer misbehavior in wireless networks," in *Dependable Systems and Networks*, June 2003.

[6] J. Konorski, "Multiple access in ad hoc wireless LANs with noncooperative stations," in *NETWORKING*, Springer, 2002, vol. 2345 of LNCS.

[7] A. Orda, R. Rom, and N. Shimkin, "Competitive routing in multi-user communication networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 5, pp. 510–521, 1993.

[8] Y. A. Korilis, A. A. Lazar, and A. Orda, "Architecting noncooperative networks," *IEEE Journal of Selected Areas in Communications*, vol. 13, no. 7, pp. 1241–1251, 1995.

[9] R. J. La and V. Anantharam, "Optimal routing control: repeated game approach," *IEEE Transactions on Automatic Control*, March 2002.

[10] H. Yaïche, R. Mazumdar, and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, 2000.

[11] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 291–303, February 2002.

[12] T. Alpcan, T. Basar, R. Srikant, and E. Altman, "CDMA uplink power control as a noncooperative game," in *Proceedengs of 40th IEEE Conference on Decision and Control*, 2001.

[13] R.-F. Liao, Rita H. Wouhaybi, and A.T. Campbell, "Incentive engineering in wireless LAN based access networks," in *Proceedings of 10th International Conference on Network Protocols (ICNP 2002)*, 2002.

[14] LAN/MAN Standards Committee, *ANSI/IEEE Std 802.11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Computer Society, 1999.

[15] IEEE 802.11 WG, *ANSI/IEEE Std 802.11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS) IEEE 802.11/D2.0*, IEEE, 2001.

[16] D. Fudenberg and J. Tirole, *Game Theory*, The MIT Press, 1991.

[17] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal of Selected Areas in Communications*, vol. 18, 2000.

[18] M. Čagalj, S. Ganeriwal, I. Aad, and J.-P. Hubaux, "On cheating in CSMA/CA ad hoc networks," Tech. Rep., EPFL, February 2004.

[19] *Network Simulator (NS)*, http://www.isi.edu/nsnam/ns/.

[20] F. Forgó, J. Szép, and F. Szidarovszky, *Introduction to the Theory of Games: Concepts, Methods, Applications*, Kluwer Academic Publishers, 1999.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[22] T. Başar and G.J. Olsder, *Dynamic Noncooperative Game Theory*, SIAM, 1999.

[23] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 1997.

[24] C. E. Koksal, H. Kassab, and H. Balakrishnan, "An analysis of short-term fairness in wireless media access protocols," in *Proceedings of ACM Sigmetrics*, 2000.

[25] F. Calì, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, 2000.

Appendix
Proof of Proposition 4 (sketch)

To prove the convergence of the iterative procedure described by equations (19) and (20), we will show that function $f(\alpha_i(t)) \equiv \alpha_i(t) - \gamma\left(\tau_i^{opt}(t) - \underline{\tau}\right)$ is a contraction mapping for the appropriately chosen step size γ. Thus, we have to show that the following holds (for brevity, we drop index i from $\alpha_i(t)$ and $\tau_i^{opt}(t)$, as well as *opt* from $\tau_i^{opt}(t)$):

$$\frac{|f(\alpha(t+1)) - f(\alpha(t))|}{|\alpha(t+1) - \alpha(t)|} < 1, \text{ for all } t \geq 0 .$$

By substituting $f(\alpha_i(t))$ and in turn $\tau(t) = 1 - 1/\alpha(t)$, and after some reordering we obtain that the following condition has to be met for $f(\alpha_i(t))$ to be a contraction mapping:

$$|1 - \gamma/\alpha(t)\alpha(t+1)| < 1 . \tag{24}$$

From (24) we have $0 < \gamma < 2\alpha(t)\alpha(t+1)$. Thus, if we can show that $\alpha(t) > 0$ for all $t \geq 0$, then there exists (an arbitrary small) $\gamma > 0$ such that $f(\alpha_i(t))$ is a contraction mapping. We observe that at $t = 0$, $\alpha(t) > 1$ (i.e., by assumption $1 > \tau(0) > \epsilon > 0$). We will show that $\alpha(t) > 1$ for all $t > 0$. By definition $\alpha(1) = \alpha(0) - \gamma(\tau(0) - \underline{\tau})$. Consider $\tau(0) > \underline{\tau}$ (otherwise, $\alpha(1)$ increases with respect to $\alpha(0)$). Let $0 < \gamma < 1$. Hence, $\alpha(1) > \alpha(0) - (\tau(0) - \underline{\tau}) = 1/(1 - \tau(0)) - (\tau(0) - \underline{\tau}) = \{1 - (1 - \tau(0))(\tau(0) - \underline{\tau})\}/(1 - \tau(0))$. Now, since $\tau(0), \underline{\tau} \in (0, 1)$ (by assumption), we have $(1 - \tau(0))(\tau(0) - \underline{\tau}) < (\tau(0) - \underline{\tau})$. Therefore, $\alpha(1) > 1 + \frac{\underline{\tau}}{1 - \tau(0)} > 1$. By applying mathematical induction and similar arguments as above we can prove that $\alpha(t) > 1$ for all $t \geq 0$. ∎