NON-COOPERATIVE FORWARDING IN AD-HOC NETWORKS

Eitan Altman*, Arzad A. Kherani*, Pietro Michiardi†, Refik Molva†
* INRIA, 06902 Sophia Antipolis, France. Eitan.Altman, alam@sophia.inria.fr.
† GET/EURECOM, Sophia-Antipolis, France. Pietro.Michiardi, molva@eurecom.fr

Abstract - A wireless Ad-hoc network is expected to be made up of energy aware entities (nodes) interested in their own perceived performance. An important problem in such a scenario is to provide incentives for collaboration among the participating entities. Forwarding packets of other nodes is an example of activity that requires such a collaboration. However, it may not be in interest of a node to always forward the requesting packets. At the same time, not forwarding any packet may adversly affect the network functioning. Assuming that the nodes are rational, i.e., their actions are strictly determined by their self-interest, we view the problem in framework of non-cooperative game theory and provide a simple punishing mechanism considering end-to-end performance objectives of the nodes. We also provide a distributed implementation of the proposed mechanism. This implementation has a small computational and storage complexity hence is suitable for the scenario under consideration.

Keywords - Game theory, Stochastic approximation algorithm.

I. Introduction

In order to maintain connectivity in an Ad-hoc network, mobile terminals should not only spend their resources (battery power) to send their own packets, but also for forwarding packets of other mobiles. Since Ad-hoc networks do not have a centralized base-station that coordinates between them, an important question that has been addressed is to know whether we may indeed expect mobiles to collaborate in such forwarding. If mobiles behave selfishly, they might not be interested in spending their precious transmission power in forwarding of other mobile's traffic. A natural framework to study this problem is noncooperative game theory. As already observed in many papers that consider noncooperative behavior in Ad-hoc networks, if we restrict to simplistic policies in which each mobile determines a fixed probability of forwarding a packet, then this gives rise to the most "aggressive" equilibrium in which no one forwards packets, see e.g. [3, Corollary 1], [4], thus preventing the system to behave as a connected network. The phenomenon of aggressive equilibrium that severely affects performance has also been reported in other noncooperative problems in networking, see e.g. [1] for a flow control context (in which the aggressive equilibrium corresponds to all users sending at their maximum rate).

In order to avoid very aggressive equilibria, we propose strategies based on threats of punishments for misbehaving aggressive mobiles, which is in the spirit of a well established design approach for promoting cooperation in Ad-hoc networks, carried on in many previous works [3], [7]. In all these references, the well known "TIT-FOR-TAT" (TFT) strategy was proposed. This is a strategy in which when a misbehaving node is detected then the reaction of other mobiles is to stop completely forwarding packets during some time; it thus prescribes a threat for very "aggressive" punishment, resulting in an enforcement of a fully cooperative equilibrium in which all mobiles forward all packets they receive (see e.g. [3, Corollary 2]). The authors of [6] also propose use of a variant of TFT in a similar context.

In this work we consider a less agressive punishment policy. We simply assume that if the fraction q' of packets forwarded by a mobile is less than the fraction q forwarded by other mobiles, then this will result in a decrease of the forwarding probability of the other mobiles to the value q'. We shall show that this will indeed lead to non-aggressive equilibria, yet not necessarily to complete cooperation. See [9] for reasons for adopting this milder punishment strategy. As already mentioned, incentive for cooperation in Ad-hoc networks have been studied in several papers, see [3], [4], [6], [7]. Almost all previous papers however only considered utilities related to successful transmission of a mobile's packet to its neighbor. In practice, however, multihop routes may be required for a packet to reach its destination, so the utility corresponding to successful transmission depends on the forwarding behavior of all mobiles along the path. The goal of our paper is therefore to study the forwarding taking into account the multihop topological characteristics of the path.

Most close to our work is the paper [3] which considers a model similar to ours (introduced in Section II below). [3] provides sufficient condition on the network topology under which each node employing the "aggressive" TFT punishment strategy results in a Nash equilibrium. In the present paper, we show that a less aggressive punishment mechanism can also lead to a Nash equilibrium which has a desirable feature that it is less resource consuming in the sense that a node need not accept all the forwarding request. We also provide some results describing the structure of the Nash equilibrium thus obtained (Section V). We then

This work was supported by research grants from EURO-NGI and CEFIPRA.

provide a distributed algorithm which can be used by the nodes to compute their equlibrium strategies and enforce the punishment mechanism using only local information (Section VI). Scetion VII concludes the paper.

II. The Model

Consider an Ad-hoc network described by a directed graph $G = (N, V)$. Along with that network, we consider a set of source-destination pairs O and a given routing between each source s and its corresponding destination d, of the form $\pi(s,d) = (s, n_1, n_2, \ldots, n_k, d)$, where $k = k(s,d)$ is the number of intermediate hops and $n_j = n_j(s,d)$ is the jth intermediate node on path $\pi(s,d)$. We assume that mobile j forwards packets (independently from the source of the packet) with a fixed probabilty γ_j. Let $\underline{\gamma}$ be the vector of forwarding probabilities of all mobiles. We assume however that each source s forwards its own packets with probability one. For a given path $\pi(s,d)$, the probability that a transmitted packet reaches its destination is thus:

$$p(s,d;\underline{\gamma}) = \prod_{j=1}^{k(s,d)} \gamma(n_j(s,d)).$$

If i belongs to a path $\pi(s,d)$ we write $i \in \pi(s,d)$. For a given path $\pi(s,d)$ of the form $(s, n_1, n_2, \ldots, n_k, d)$ and a given mobile $n_j \in \pi(s,d)$, define the set of intermediate nodes before n_j to be the set $S(s,d;n_j) = (n_1, ..., n_{j-1})$. The probability that some node $i \in \pi(s,d)$ receives a packet originating from s with d as its destination is then given by

$$p(s,d;i,\underline{\gamma}) = \prod_{j \in S(s,d;i)} \gamma(j).$$

Note that $p(s,d;d,\underline{\gamma}) = p(s,d;\underline{\gamma})$, the probability that node d receives a packet originating from source s and having d as its destination.

Define $O(i)$ to be all the paths in which a mobile i is an intermediate node. Let the rate at which source s creates packets for destination d be given by some constant λ_{sd}. Then the rate at which packets arrive at node i in order to be forwarded there is given by

$$\xi_i(\underline{\gamma}) = \sum_{\pi(s,d) \in O(i)} \lambda_{sd}\, p(s,d;i,\underline{\gamma}).$$

Let E_f be the total energy needed for forwarding a packet (which includes the energy for its reception and its transmission). Then the utility of mobile i that we consider is

$$\begin{aligned} U_i(\underline{\gamma}) \;\; = & \sum_{n:(i,n)\in O} \lambda_{in} f_i(p(i,n;\underline{\gamma})) \\ & + \sum_{n:(n,i)\in O} \lambda_{ni} g_i(p(n,i;\underline{\gamma})) - aE_f \xi_i(\underline{\gamma}), \end{aligned} \quad (1)$$

where f_i and g_i are utility functions that depend on the success probabilities associated with node i as a source and as a destination respectively and a is some multiplicative constant. We assume that $f_i(\cdot)$ and $g_i(\cdot)$ are nondecreasing concave in their arguments. The objective of mobile i is to choose γ_i that maximizes $U_i(\underline{\gamma})$. We remark here that similar utility function is also considered in [3] with the difference that node's utility does not include its reward as a destination, i.e., they assume that $g_i(\cdot) \equiv 0$.

Definition: *For any choices of strategy $\underline{\gamma}$ for all mobiles, define $(\gamma_i', \underline{\gamma}^{-i})$ to be the strategy obtained when only player i deviates from γ_i to γ_i' and other mobiles maintain their strategies fixed.*

In a noncooperative framework, the solution concept of the optimization problem faced by all players is the following:

Definition: *A Nash equilibrium, is some strategy set $\underline{\gamma}^*$ for all mobiles such that for each mobile i,*

$$U_i(\underline{\gamma}^*) = \max_{\gamma_i'} U_i(\gamma_i', (\underline{\gamma}^*)^{-i}).$$

We call $argmax_{\gamma_i'} U_i(\gamma_i', \underline{\gamma}^{-i})$ the set of optimal responses of player i against other mobiles policy $\underline{\gamma}^{-i}$ (it may be an empty set or have several elements).

In our setting, it is easy to see that for each mobile i and each fixed strategy $\underline{\gamma}^{-i}$ for other players, the best response of mobile i is $\gamma_i = 0$ (unless $O(i) = \emptyset$ in which case, the best response is the whole interval $[0,1]$). Thus the only possible equilibrium is that of $\gamma_i = 0$ for all i. To overcome this problem, we consider the following "punishing mechanism". in order to incite mobiles to cooperate.

Definition: *Consider a given set of policies $\underline{\gamma} = (\gamma, \gamma, \gamma, ...)$. If some mobile deviates and uses some $\gamma' < \gamma$, we define the punishing policy $\kappa(\gamma', \gamma)$ as the policy in which all mobiles decrease their forwarding probability to γ'.*

When this punishing mechanism is enforced, then the best strategy of a mobile i when all other mobiles use strategy γ is γ' that achieves

$$J(\gamma) := \max_{\gamma' \leq \gamma} U_i(\underline{\gamma'}) \quad (2)$$

where $\underline{\gamma'} = (\gamma', \gamma', \gamma', ...)$.

Definition: *If some γ^* achieves the minimum in (2) we call the vector $\underline{\gamma}^* = (\gamma^*, \gamma^*, \gamma^*, ...)$ the equilibrium strategy (for the forwarding problem) under threats. $J(\gamma)$ is called the corresponding value.*

Remark: Note that $\gamma^* = 0$ is still a Nash equilibrium, a fact that will be used frequently in Section V where we obtain some structural properties of equilibrium strategy under threats.

III. Utilities for Symmetrical Topologies

By symmetrical topology we mean the case where f_i, g_i and ξ_i are independent of i. This implies that for any source-destination pair (s,d), there are two nodes s' and d' such that the source-destination pairs (s', s) and (d, d') are identical to (s,d) in the sense that there view of the network is similar to that of (s,d). This implies that, under the punishment mechanism where all nodes have same forwarding probability, we have $p(s,d;\underline{\gamma}) = p(s',s;\underline{\gamma})$. Thus we can replace the rewards $f_i + g_i$ by another function that we denote $f(\cdot)$.

Consider $\underline{\gamma}$ where all entries are the same and equal to γ, except for that of mobile i. For a path $\pi(s,d)$ containing n intermediate nodes, we have $p(s,d;\underline{\gamma}) = \gamma^n$. Also, if a mobile i is $n+1$ hops away from a source, $n = 1,2,3,...$, and is on the path from this source to a destination (but is not itself the destination), then $p(s,d;i,\underline{\gamma}) = \gamma^n$. We call the source an "effective source" for forwarding to mobile i since it potentially has packets to be forwarded by mobile i. Let $h(n)$ be the rate at which all effective sources located $n+1$ hops away from mobile i transmit packets that should use mobile i for forwarding (we assume that h is the same for all nodes). Let $\lambda^{(n)}$ denote the rate at which a source s creates packets to all destinations that are $n+1$ hops away from it. Then we have

$$U_i(\underline{\gamma}) = \sum_{n=1}^{\infty} \lambda^{(n)} f(\gamma^n) - aE_f \sum_{n=1}^{\infty} h(n)\gamma^n. \qquad (3)$$

The equilibrium strategy under threat is then the value of γ that maximizes the r.h.s.

Remark: If we denote by $\Lambda(z) = \sum_{n=1}^{\infty} z^n \lambda^{(n)}$ the generating function of $\lambda^{(n)}$ and $H(z) := \sum_{n=1}^{\infty} z^n h(n)$ the generating function of h. Then

$$\max_{\gamma} \Big(\Lambda(\gamma) - aE_f H(\gamma) \Big)$$

is the value of the problem with threats in the case that f is the identity function.

IV. Examples

In this section we present, by means of two examples, the effect of imposing the proposed punishment mechanism.

A. An Asymmetric Network

Consider the network shown in Figure 1. For this case nodes 1 and 4 have no traffic to forward. Note also that if we assume that $g_3(\cdot) \equiv 0$ in Equation 1 then node 3 has no incentive even to invoke the punishment mechanism for node 2. This will result in no cooperation in the network. Assume for the time being that $f_2(x) = g_3(x) = x$, i.e., f_2 and g_3 are identity functions. In this case it is seen that the utility functions for nodes 2 and 3 are, assuming $\lambda_{13} = \lambda_{24} = 1$, $U_2(\gamma_2,\gamma_3) = \gamma_3 - aE_f\gamma_2$ and $U_3(\gamma_2,\gamma_3) = \gamma_2 - aE_f\gamma_3$. When we impose the punishment mechanism, it turns out that the equilibrium strategy for the two nodes is to always cooperate, i.e., $\gamma_2 = \gamma_3$. This is to be compared with the TFT strategy of [3] which would imply $\gamma_2 = \gamma_3 = 0$.

B. A Symmetric Network: Circular Network with Fixed Length of Paths

We consider here equally spaced mobile nodes on a circle and assume that each node i is a source of traffic to a node located L hops to the right, i.e. to the node $i+L$.

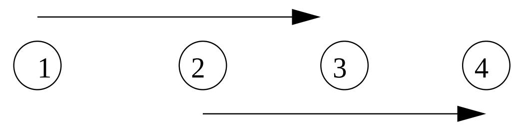

Fig. 1
An asymmetric network.

Let the rate of traffic generated from a source be λ. For this case, $h(n) = \lambda I_{\{n \le L-1\}}$. Also, $\lambda^{(n)} = \lambda I_{\{n=L\}}$, for some λ. It follows from Equation 3 that the utility function for mobile i is

$$U_i(\underline{\gamma}) = \lambda f(\gamma^{L-1}) - aE_f \lambda \sum_{n=0}^{L-2} \gamma^n.$$

For $f(\cdot)$ an identity function, we see that $U_i(\underline{\gamma}) = \lambda\left[\gamma^{L-1} - aE_f(\gamma^{L-2} + \gamma^{L-3} + \ldots + \gamma + 1)\right]$. Note that if $L = 2$ and $a = \frac{1}{E_f}$, the utility function is independent of γ hence in this case the equilibrium strategy is any value of forwarding probability. Also, if $aE_f \ge 1$, the equilibrium strategy is $\gamma = 0$. We will have more to say on this in th next section where we study the structure of equilibrium strategy for symmetric network.

V. Structure of Equilibrium Strategy for Symmetric Network

In this section we undertake the study of dependence of the equilibrium strategy on the various system parameters. We restrict ourselves to the case of symmetric topologies. Symmetry of the problem along with the imposed punishment mechanism implies that the equilibrium strategy (the forwarding probabilities) will be same for all the nodes in the network. We denote this probability by γ^*.

This is to be understood as follows. When a node i computes its equilibrium strategy γ_i, it must consider the fact that the other nodes will respond with a punishing mechanism to its strategy. Thus, the problem faced by node i is *not* that of optimizing Equation 3 with respect to γ_i considering γ^{-i} fixed (which will lead to the trivial solution of $\gamma_i = 0$ as seen before). Owing to the punishment mechanism, node i should apriori assume that all the forwarding probabilities are same, i.e., $\gamma^{-i} = (\gamma_i, \ldots, \gamma_i)$. This makes the problem faced by node i a single variable optimization problem.

Though $f(\cdot)$ is concave in its argument ($p(\gamma)$, which is a polynomial in γ), $f(p(\gamma))$ may not be concave as a function of γ. For example, in the case of circular network above, $f(p(\gamma)) = p(\gamma) = \gamma^{L-1}$, convex in γ. Thus obtaining a direct structural result for γ^* seems to be hard for general $f(\cdot)$ and $p(\cdot)$. We can get some interesting insights using some approximations; this is the aim of present section. In particular, we study how γ^* depends on the *system paramaters*, L, $f(\cdot)$, $p(\cdot)$, a and E_f.

It is clear from the expression of the utility function that γ^* will depend on a and E_f only through their product. Let us introduce the notation $K := aE_f$.

It is also clear from the definition of utility function ($U_i(\gamma)$) that if either K or L is *large*, the equilibrium strategy of the game is at *smaller* γ. It is also intuitive that for *small* values of K (or L), a node may forward most of the requesting packets. In the following we characterize what value of K or L can be considered as *large* or *small*. Clearly this characterization will depend on $f(\cdot)$ and the network, i.e., $p(\cdot)$ and $H(\cdot)$. If we fix K and increase the hop-length L, it is intuitive that γ^* will eventually start decreasing as a function of L. This is established in [9]. The effect of varying K for a general network is also presented in [9]. For this case we obtain the following,

Result 5.1: For a fixed L, if the network topology and $f(\cdot)$ satisfy $f'(0)p'(1) \leq Kh(1)$, then equilibrium strategy is $\gamma^* = 0$.

Remark: Above result shows that if K, i.e., the energy spent in reception and transmission is larger than a threshold (say $K^* = f'(0)p'(1)(h(1))^{-1}$), it is best for the nodes to not forward packets at all *even under the punishing mechanism*. This is to be compared with the fact proved below that $\gamma = 0$ is *always* a *local* maximum for $L > 2$ for the circular network. Thus the above result gives a criteria when $\gamma = 0$ is also a *global* maximum.

The effect of varying K for the special case of circular network is presented in [9] where existence of a similar threshold is established.

VI. Algorithm for Computing the Equilibrium Strategy in a Distributed Manner

It is interesting to design distributed algorithms which can be used by the mobiles to compute the equilibrium strategy and simultaneously enforce the proposed punishment mechanism. The obvious desirable features of such an algorithm are that it should be decentralised, distributed scalability and should be able to adapt to changes in network.

We propose such an algorithm in this section. We present it, for ease of notation, for the case of symmetric network. Assume for the moment that $f(\cdot)$ is the identity function. In this case each node has to solve the equation (recall the notation of Section III)

$$U'(\gamma) = \Lambda'(\gamma) - KH'(\gamma) = 0, \tag{4}$$

where the primes denote the derivatives with respect to γ. In general this equation will be nontrivial to solve directly. For the case of more general network, one needs to compute the derivative of the utility function of Equation 1, the rest of procedure that follows is similar.

Note that in the above expression we first assume that the forwarding probabilities of all the nodes in the network are same (say γ) and then compute the derivative with respect to this common γ. This is because in the node must take the effect of punishment mechanism into account while computing its own optimal forwarding probability, i.e., a node should assume that all the other nodes will use the same forwarding probability that it computes.

Thus, solving Equation 4 is reduced to a single variable optimization problem. Since the actual problem from which we get Equation 4 is a maximization problem, a node does a gradient *ascent* to compute its optimal forwarding probability. Thus, in its n^{th} computation, a node i uses the iteration

$$\gamma_i^{(n+1)} = \gamma_i^{(n)} + a(n)(\Lambda'(\gamma_i^{(n)}) - KH'(\gamma_i^{(n)})), \tag{5}$$

where $a(n)$ is a sequence of positive numbers satisfying the usual conditions imposed on the learning parameters in stochastic approximation algorithms [8], i.e., $\sum_n a(n) = \infty$ and $\sum_n a(n)^2 < \infty$.

The relation to stochastic approximation algorithm here is seen as follows: the network topology can be randomly changing with time owing to node failures/mobility et cetera. Thus a node needs to appropriately modify the functions $\Lambda(\cdot)$ and $H(\cdot)$ based on its most recent view of the network (this dependence of $\Lambda(\cdot)$ and $H(\cdot)$ on n is suppressed in the above expression).

Though the above is a simple stochastic approximation algorithm, it requires a node to know the topology of the part of network around itself. This information is actually trivially available to a node since it can extract the required information from the packets requesting forwarding or using a neighbour discovery mechanism. However, in case of any change in the network, there will typically be some delay till a node completely recognizes the change. This transient error in a node's knowledge about the network whenever the network changes is ensured to die out ultimately owing to the assumption of finite second moment for the learning parameters.

It is known by the o.d.e. approach to stochastic approximation algorithm that the above algorithm will asymptotically track the o.d.e. [8]:

$$\dot{\gamma}_i(t) = \Lambda'(\gamma_i(t)) - KH'(\gamma_i(t)), \tag{6}$$

and will converge to one of the *stable* critical points of o.d.e. of Equation 6. It is easily seen that a local maximum of the utility function forms a stable critical point of Equation 6 while any local minimum forms an unstable critical point. Thus the above algorithm inherently makes the system converge to a local maximum and avoids a local minimum.

However, it is possible that different nodes settle to different local maxima. The imposed punishment mechanism then ensures that all the nodes settle to the one which corresponds to the lowest values of γ. This is a desirable feature of the algorithm that it inherently avoids multiple simultaneous operating points. An implementation of the punishment mechanism is described next.

A. Distributed Implementation of the punishment mechanism

An implementation of punishment mechanism proposed in Section II requires, in general, a node to know about the misbehaving node in the network, if any. Here we propose a simple implementation of the punishment mechanism which requires only local information for its implementation.

Let $\mathcal{N}(i)$ be the set of neighbours of node i. Every node computes its forwarding policy in a distrubuted manner using the above mentioned stochastic approximation algorithm. However, as soon as a neighboring node is detected to misbehave by a node, the node computes its forwarding policy as follows:

$$\gamma_i^* = \min\{\gamma_i, \min_{j \in \mathcal{N}(i)} \hat{\gamma}_j\} \tag{7}$$

where γ_i and $\hat{\gamma}_j$ represents, respectively, the forwarding policy adopted by node i and the estimate of node j's forwarding probability available to node i. γ_i^* represents the new policy selected by node i. Note here that γ_i is still computed using iteration of Equation 5. We are also assuming here that a node can differentiate between a misbehaving neighbouring node and the failure/mobility of a neighbouring node.

This punishment propagates in the network until all the nodes in the network settle to the common forwarding probability (corresponding to that of the misbehaving node). In particular, the effect of this punishment will be seen by the misbehaving ndoe as a degradation in its own utility. Suppose now that the misbehaving node, say n_i, decides to change to a cooperative behavior: at that point, it will detect and punish its neighbors because of the propagation of the punishment that induced its neighbouring nodes to decrease their forwarding policy. Thus, the intial punishment introduces a negative loop and the forwarding policy of every node of the network collapses to the forwarding policy selected by the misbehaving node. Since now every node in the network has same value of forwarding probability, none of the nodes will be able to increase its forwarding probability even if none of the node is misbehaving now.

An example of this phenomenon can be seen from the network of Figure 1. Assume that $\gamma_2 = \gamma_3 = \gamma$ and now node 2 reduces γ_2 to a smaller value γ'. Owing to the punishment mechanism, node 3 will respond with $\gamma_3 = \gamma'$. This will result in a reduced utility for node 2 which would then like to increase γ_2. But, since $\gamma_3 = \gamma'$, the punishing mechanism would imply that $\gamma_2 = \gamma'$ as well. This *lock-in* problem is avoided by the solution proposed below.

We modify our algorithm to account for the above mentioned effect. Our solution is based on timers of a fixed duration. When a node enters in the punishing phase (starts punishing some of its neighbour) the local timer for that node is set and the forwarding policy is selected as in equation 7. When the timer expires, the punishing node evaluates its forwarding policy as if there were no misbehaving nodes, then uses some of standard mechanism to detect any persistent misbehavior (this also helps distinguishing between a misbehaving node and a failed/moved node). In the case no misbehaviors are detected, depending on the choice of the learning parameter of the stochastic apporximation algotithm, the forwarding policy of the network eventually returns to the optimal value for the network. If the neighboring node continues to misbehave, the timer is set again and the punishment mechanism is re-iterated. We assume that the sequence of learning parameters by a node is restarted each time the timer is set.

See [9] for remark on computational and storage complexity of the above algorithm. Some numerical results from an implementation of the proposed algorithm are also presented in [9].

VII. Conclusion

We use the framework of non-cooperative game theory to provide incentives for collaboration in the case of wireless Ad-hoc networks. The incentive proposed in the paper is based on a simple punishment mechanism that can be implemented in a completely distributed manner with very small computational complexity. The advantage of the proposed strategy is that it results in a less "aggressive" equilibrium in the sense that it does not result in a degenerate scenario where a node either forwards all the requested traffic or does not forward any of the request.

References

[1] D. Dutta, A. Goel and J. Heidemann, "Oblivious AQM and Nash Equilibria", IEEE Infocom, 2003.

[2] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring. Modelling incentives for collaboration in mobile Ad-hoc networks. In *Proceedings of WiOpt'03*, Sophia-Antipolis, France, 3-5, March 2003.

[3] M. Félegyházi, L. Buttyán and J. P. Hubaux, "Equilibrium analysis of packet forwarding strategies in wireless Ad-hoc entworks – the static case", PWC 2003 Personal Wireless Communications, Sept. 2003, Venice, Italy.

[4] P. Michiardi and R. Molva. A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile Ad-hoc networks. In *Proceedings of WiOpt'03*, Sophia-Antipolis, France, 3-5, March 2003.

[5] L. Samuelson, "Subgame Perfection: An Introduction," in John Creedy, Jeff Borland and Jrgen Eichberger, eds., Recent Developments in Game Theory, Edgar Elgar Publishing, 1992, 1-42.

[6] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini and R. R. Rao, "Cooperation in wireless Ad-hoc networks", *Proceedings of IEEE Infocom*, 2003.

[7] A. Urpi, M. Bonuccelli, and S. Giordano. Modelinig cooperation in mobile Ad-hoc networks: a formal description of selfishness. In *Proceedings of WiOpt'03*, Sophia-Antipolis, France, 3-5, March 2003.

[8] H. J. Kushner and G. Yin, "Stochastic Approximation Algorithms and Applications," Springer-Verlag, 1997.

[9] E. Altman, A. A. Kherani, P. Michiardi, and R. Molva. Non-cooperative Forwarding in Ad-hoc Networks. Technical Report INRIA Report No. RR-5116, 2004.