Equilibrium Analysis of Packet Forwarding Strategies in Wireless Ad Hoc Networks – the Static Case[*]

Márk Félegyházi[1], Levente Buttyán[2], Jean-Pierre Hubaux[1]

[1] Laboratory of Computer Communications and Applications,
Swiss Federal Institute of Technology – Lausanne
[2] Department of Telecommunications, Budapest University of Technology and
Economics

Abstract. In multi-hop wireless networks, every node is expected to forward packets for the benefit of other nodes. Yet, if each node is its own authority, then it may selfishly deny packet forwarding in order to save its own resources. Some researchers have proposed to introduce an incentive mechanism in the network that motivates the nodes to cooperate. In this paper, we address the question of whether such an incentive mechanism is necessary or cooperation between the nodes exists in the absence of it. We define a model in a game theoretic framework and identify the conditions under which cooperative strategies can form an equilibrium. As the problem is somewhat involved, we deliberately restrict ourselves to a static configuration.

1   Introduction

By definition, in multi-hop wireless networks, every node is expected to forward packets for the benefit of other nodes. This is probably the most important difference of this family of networks with respect to those that are more conventional, such as the Internet or cellular networks.

As long as the whole network is under the control of a single authority, as is usually the case in military networks or for rescue operations, this situation is not problematic because the interest of the mission by far exceeds the vested interest of each participant. However, if each node is its own authority, the situation changes dramatically: In this case, the most reasonable assumption is to consider that each node will try to maximize its benefits by exploiting the network, even if this means adopting a selfish behavior. This selfishness can mean not participating in the unfolding of mechanisms of common interest (e.g., route setup, packet forwarding, or mobility management), notably to spare resources, including batteries.

Over the last few years, several researchers have proposed incentive techniques to encourage nodes to collaborate, be it by circumventing misbehaving nodes [MGLB00], by making use of a reputation system [BL02], or by relating the right to benefit from the

network to the sense of the common interest demonstrated so far [BH03]. To the best of our knowledge the only paper that focuses on the conditions of cooperation without any incentive mechanism is [SNCR03], where the authors build up a theoretical framework based on different energy classes.

In this paper, we study the problem of cooperation for the most basic mechanism, namely packet forwarding. In order to do so, we define a model in a game theoretic framework and identify the conditions under which an equilibrium based on cooperation exists. As the problem is somewhat involved, we deliberately restrict ourselves to a static configuration.

The rest of the paper is organized in the following way. In Section 2, we show how packet forwarding can be modelled in a game theoretic framework. In Section 3, we analyze the model introduced in Section 2. We provide our main results, stated as theorems in Section 4 along with our simulation results. In Section 5, we present the related work. Finally, we conclude the paper in Section 6.

2   Modelling packet forwarding as a game

Connectivity graph. Let us consider an ad hoc network of n nodes. Let us denote the set of all nodes by N. Each node has a given power range and two nodes are said to be neighbors if they reside within the power range of each other. We represent the neighbor relationship between the nodes with an undirected graph, which we call the connectivity graph. Each vertex of the connectivity graph corresponds to a node in the network, and two vertices are connected with an edge if the corresponding nodes are neighbors.

Routes. Communication between two non-neighboring nodes is based on multi-hop relaying. This means that packets from the source to the destination are forwarded by intermediate nodes. Such a chain of nodes (including the source and the destination) is represented by a path in the connectivity graph, which we call a route. We assume that one of the shortest paths is chosen randomly as a route for each source-destination pair.

Time. We use a discrete model of time where time is divided into slots. We assume that both the connectivity graph and the set of existing routes remain unchanged during a time slot, whereas changes may happen at the end of each time slot. We assume that the duration of the time slot is much longer than the time of a packet relaying from the source to the destination. This means that a node is able to send several packets within one time slot.

Forwarding game. We model the packet forwarding operation as a game, which we call the forwarding game. The players of the forwarding game are the nodes. In each time slot t, each node i chooses a cooperation level $p_i(t) \in [0, 1]$, where 0 and 1 represent full defection and full cooperation, respectively. Here, defection means that the node does not forward traffic for the benefit of other nodes and cooperation means that it does. Thus, $p_i(t)$ represents the fraction of the traffic routed through i in t that i actually forwards. Note that i has a single cooperation level $p_i(t)$ that it applies to every route in which it is involved as a forwarder. We preferred not to require the nodes to be able to distinguish the packets that belong to different routes, because this would require identifying the source-destination pairs and applying a different cooperation level to each of them. This would probably increase the computation significantly.

Let us assume that in time slot t there exists a route r with source node s and ℓ intermediate nodes $f_1, f_2, \ldots, f_\ell$. The normalized value of the throughput $\tau(r,t)$ experienced by the source s on r in t is defined as the fraction of the traffic sent by s on r in t that is delivered to the destination. Since we are studying cooperation in packet forwarding, we assume that the main reason for dropping packets in the network is the non-cooperative behavior of the nodes. In other words, we assume that the network is not congested and that the number of packets dropped because of the limited capacity of the nodes and the links is negligible. Hence, $\tau(r,t)$ can be computed as the product of the cooperation levels of all intermediate nodes:

$$\tau(r,t) = \prod_{k=1}^{\ell} p_{f_k}(t) \tag{1}$$

The payoff $\xi_s(r,t)$ of s on r in t depends on the normalized throughput $\tau(r,t)$. In general, $\xi_s(r,t) = u(\tau(r,t))$, where u is some non-decreasing function. In this paper, we assume that u is linear and has the following form: $u(x) = (G+L) \cdot x - L = G \cdot x - L \cdot (1-x)$, where $G = u(1) > 0$ represents the gain that s obtains if all its traffic sent on r in t is delivered to the destination, and $-L = u(0) < 0$ represents the loss that s suffers if nothing is delivered in t to the destination. Thus, we get that

$$\xi_s(r,t) = (G+L) \cdot \tau(r,t) - L \tag{2}$$

For simplicity, we assume that every node in the network uses the same function u and the same parameters G and L, which we assume to be independent of r and t.

The payoff $\eta_{f_j}(r,t)$ of the j-th intermediate node f_j on r in t is non-positive and represents the cost for node f_j to forward packets on route r during time slot t. It depends on the fraction $\tau_j(r,t)$ of the traffic sent by the source on r in t that is forwarded by f_j. The value $\tau_j(r,t)$ is the normalized throughput on r in t leaving node j, and it is computed as the product of the cooperation levels of the intermediate nodes from f_1 up to and including f_j:

$$\tau_j(r,t) = \prod_{k=1}^{j} p_{f_k}(t) \tag{3}$$

Let $-C < 0$ represent the forwarding cost for the first forwarder if it forwards all the traffic sent by the source on r in t. The payoff $\eta_{f_j}(r,t)$ of the j-th intermediate node is then defined as follows:

$$\eta_{f_j}(r,t) = -C \cdot \tau_j(r,t) \tag{4}$$

Again, for simplicity, we assume that forwarding one unit of traffic costs the same for every node in the network and that the parameter C is independent from r and t.

By definition, the payoff of the destination is 0. In other words, we assume that only the source has a benefit if the traffic reaches the destination (information pushing). We note however, that our model can be applied and all of our results hold in the reverse case when only the destination benefits from receiving traffic; an example for this scenario would be a file download.

The total payoff $\pi_i(t)$ of node i in time slot t is then computed as

$$\pi_i(t) = \sum_{r \in S_i(t)} \xi_i(r, t) + \sum_{r \in F_i(t)} \eta_i(r, t) \qquad (5)$$

where $S_i(t)$ is the set of routes in t where i is source, and $F_i(t)$ is the set of routes in t where i is an intermediate node.

Strategy space. In every time slot, each node i updates its cooperation level using a strategy function σ_i. In general, i could choose a cooperation level to be used in time slot t based on the information it obtained in *all* preceding time slots. In order to make the analysis feasible, we assume that i uses only information that it obtained in the previous time slot. More specifically, we assume that i chooses its cooperation level in time slot t based on the normalized throughput it experienced in time slot $t-1$ on the routes where it was a source. Formally:

$$p_i(t) = \sigma_i([\tau(r, t-1)]_{r \in S_i(t-1)}) \qquad (6)$$

where $[\tau(r, t-1)]_{r \in S_i(t-1)}$ represents the normalized throughput vector for node i in time slot $t-1$, each element of which being the normalized throughput experienced by i on a route where it was source in $t-1$. The strategy of a node i is then defined by its strategy function σ_i and its initial cooperation level $p_i(0)$.

Note that σ_i takes as input the normalized throughput and not the total payoff received by i in the previous time slot. The rationale is that i should react to the behavior of the rest of the network, which is represented by the normalized throughput in our model.

There are an infinite number of strategy functions; we highlight three special cases (in Table 1) that we will encounter later in the analysis [Axe84]. If the output of the strategy function is independent of the input, then the strategy is called a *non-reactive strategy* (e.g., always defecting and always cooperating in Table 1). If the output depends on the input, then the strategy is *reactive* (e.g., Tit-For-Tat in Table 1).

Table 1. Three special strategies. Here, we assume that σ_i takes as input a one dimensional vector (i.e., a scalar)

Strategy	Initial move	Function
AllD (always defect)	0	$\sigma_i(x) = 0$
AllC (always cooperate)	1	$\sigma_i(x) = 1$
TFT (Tit-For-Tat)	1	$\sigma_i(x) = x$

Our model requires that each source be able to observe the throughput in a given time slot on every route. We assume that this is made possible with sufficiently high precision by using some higher level control protocol above the network layer.

3 Meta-model

In this section, we introduce a meta-model in order to formalize the properties of the packet forwarding game defined in the previous section. Thus, we study the behavior of the model through the analysis of the meta-model (i.e., a model of the model). Unlike in the model presented in the previous section, in the meta-model and in the rest of the paper, we will assume that routes remain unchanged during the lifetime of the network. In addition, we also assume that each node is a source on only one route.

Dependency graph. Let us consider a route r. The normalized throughput experienced (hence the payoff received) by the source on r depends on the cooperation levels of the intermediate nodes on r. We represent this dependency relation between the nodes with a directed graph, which we call the dependency graph. Each vertex of the dependency graph corresponds to a network node. There is a directed edge from vertex i to vertex j, denoted by the ordered pair (i,j), if there exists a route where i is an intermediate node and j is the source. Intuitively, an edge (i,j) means that the behavior (cooperation level) of i has an effect on j. The concept of dependency graph is illustrated in Fig. 1.

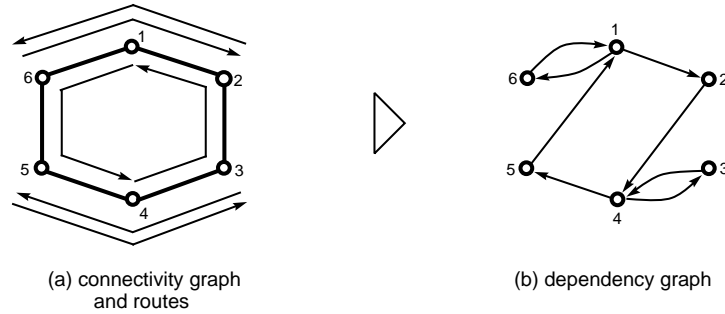
(a) connectivity graph
and routes

(b) dependency graph

Fig. 1. Example of dependency graph In (a), the arrows represent the routes (e.g., there is a route between nodes 1 and 4 passing through nodes 6 and 5). The corresponding dependency graph is shown in (b)

Game automaton. Now we define the automaton A that will model the unfolding of the forwarding game. The automaton is built on the dependency graph. We assign a machine M_i to every vertex i of the dependency graph and interpret the edges of the dependency graph as links that connect the machines assigned to the vertices. Each machine M_i thus has some input and some (possibly 0) output links.

The internal structure of the machine is illustrated in Fig. 2. Each machine M_i consists of a multiplication gate $\prod$ followed by a gate that implements the strategy

function σ_i of node i. The multiplication gate $\prod$ takes the values on the input[3] links and passes their product to the strategy function gate[4]. Finally, the output of the strategy function gate is passed to each output link of M_i.
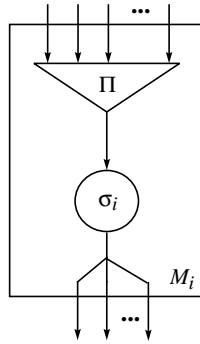

Fig. 2. Internal structure of machine M_i

The automaton A works in discrete steps. Initially, in step 0, each machine M_i outputs some initial value $x_i(0)$. Then, in step $t > 0$, each machine computes its output $x_i(t)$ by taking the values that appear on its input links in step $t - 1$.

Formally, the operation of A can be described by a set of equations, where each equation corresponds to a machine M_i. As an example, let us consider the automaton in Fig. 3, which is the automaton that corresponds to the dependency graph of Fig. 1. Its operation is described by the following set of equations:

$$x_1(t) = \sigma_1(x_5(t - 1) \cdot x_6(t - 1))$$
$$x_2(t) = \sigma_2(x_1(t - 1))$$
$$x_3(t) = \sigma_3(x_4(t - 1))$$
$$x_4(t) = \sigma_4(x_2(t - 1) \cdot x_3(t - 1))$$
$$x_5(t) = \sigma_5(x_4(t - 1))$$
$$x_6(t) = \sigma_6(x_1(t - 1))$$

It is easy to see that by iteratively substituting the appropriate equations into each other, the value of any output in any step can be expressed as a function of the initial values

---

[3] The multiplication comes from the fact that the experienced throughput for the source is the product of the cooperation levels of the forwarders on its route.

[4] Note that here σ_i takes a single real number as input, instead of a vector of real numbers as we defined earlier, because we assume that each node is source on only one route.
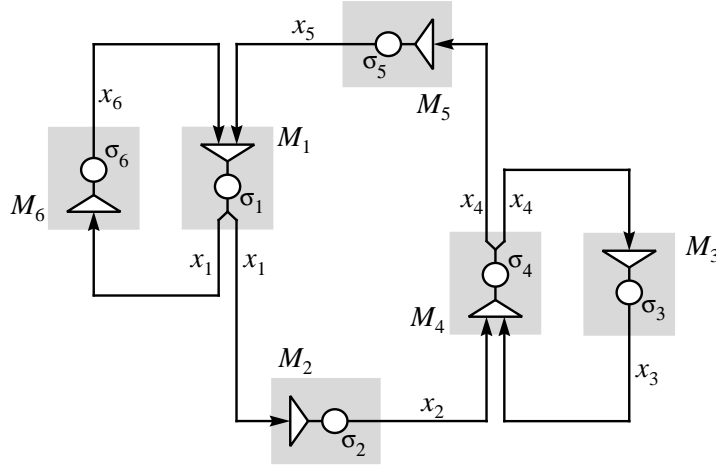
Fig. 3. The automaton that corresponds to the dependency graph of Fig. 1

$x_i(0)$. For instance:

$$x_1(3) = \sigma_1(x_5(2) \cdot x_6(2))$$
$$= \sigma_1(\sigma_5(x_4(1)) \cdot \sigma_6(x_1(1)))$$
$$= \sigma_1(\sigma_5(\sigma_4(x_2(0) \cdot x_3(0))) \cdot \sigma_6(\sigma_1(x_5(0) \cdot x_6(0))))$$

Note that if $x_i(0) = p_i(0)$ for all i, then in step t, each machine M_i will output the cooperation level of node i in time slot t (i.e., $x_i(t) = p_i(t)$), as we assumed that the set of routes (and hence the dependency graph) remains unchanged in every time slot. Therefore, the evolution of the values on the output links of the machines (which, in fact, represent the state of the automaton) models the evolution of the cooperation levels of the nodes in the network. It also follows that the cooperation level $p_i(t)$ of any node i in any time slot t can be expressed as a function of the initial cooperation levels of the nodes. For instance:

$$p_1(3) = x_1(3)$$
$$= \sigma_1(\sigma_5(\sigma_4(x_2(0) \cdot x_3(0))) \cdot \sigma_6(\sigma_1(x_5(0) \cdot x_6(0))))$$
$$= \sigma_1(\sigma_5(\sigma_4(p_2(0) \cdot p_3(0))) \cdot \sigma_6(\sigma_1(p_5(0) \cdot p_6(0))))$$

In order to study the interaction of node i with the rest of the network, we extract the gate that implements the strategy function σ_i from the automaton A. What remains is the automaton without σ_i, which we denote by A_{-i}. A_{-i} has an input and an output link; if we connect these to the output and the input, respectively, of σ_i (as illustrated in Fig. 4) then we get back the original automaton A. In other words, the automaton in Fig. 4 is another representation of the automaton in Fig. 3, which captures the fact that

from the viewpoint of node i, the rest of the network behaves like an automaton: The input of A_{-i} is the sequence $\overline{x}_i = x_i(0), x_i(1), \ldots$ of the cooperation levels of i, and its output is the sequence $\overline{y}_i = y_i(0), y_i(1), \ldots$ of the normalized throughput values for i.
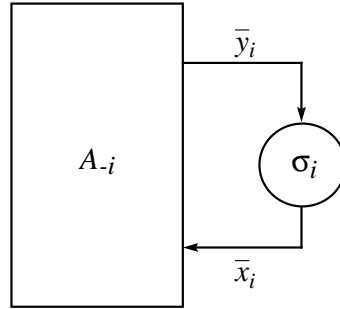

Fig. 4. Model of interaction between node i and the rest of the network represented by the automaton A_{-i}

By using the system of equations that describe the operation of A, one can easily express any element $y_i(t)$ of sequence $\overline{y}_i$ as some function of the preceding elements $x_i(t-1), x_i(t-2), \ldots, x_i(0)$ of sequence $\overline{x}_i$ and the initial values $x_j(0)$ $(j \neq i)$ of the machines within A_{-i}. We call such an expression of $y_i(t)$ the t-th *input/output formula* or shortly the t-th *i/o formula* of A_{-i}. It is important to note that the i/o formulae of A_{-i} may involve any strategy function σ_j where $j \neq i$, but they never involve σ_i. Taking the automaton of Fig. 3 again, and extracting, for instance, σ_6, we can determine the first few i/o formulae of A_{-6} as follows:

$$y_6(0) = x_1(0)$$
$$y_6(1) = \sigma_1(x_5(0) \cdot x_6(0))$$
$$y_6(2) = \sigma_1(\sigma_5(x_4(0)) \cdot x_6(1))$$
$$y_6(3) = \sigma_1(\sigma_5(\sigma_4(x_2(0) \cdot x_3(0))) \cdot x_6(2))$$
$$\cdots \qquad \cdots$$

Dependency loops. Node i has a dependency loop if there exists a sequence $(i, v_1), (v_1, v_2), \ldots, (v_{\ell-1}, v_\ell), (v_\ell, i)$ of edges in the dependency graph. The existence of dependency loops is important: If node i has no dependency loops, then the cooperation level chosen by i in a given time slot has no effect on the normalized throughput experienced by i in future time slots.

We define two types of dependency loops depending on the strategies played by the nodes on the dependency loop. If all nodes j (where $j \neq i$) on the dependency loop of i play reactive strategies, then we talk about a *reactive dependency loop* of i. If, on the

other hand, there exists at least one node j (where $j \neq i$) on the dependency loop of i that plays a non-reactive strategy, then the loop will be a *non-reactive dependency loop* of i.

4 Results

Our goal, in this section, is to find possible Nash equilibria [Nash50] and investigate the conditions for them. The existence of a Nash equilibrium based on cooperation would mean that there are cases in which cooperation is "naturally" encouraged, meaning without using incentive mechanisms. In the following, we use the model presented in Fig. 4. In Section 4.1, we present analytical results that define the best strategies for a node under specific network conditions. In Section 4.2, we present the results of simulations that we performed to estimate the probability that these conditions are satisfied by the network.

4.1 Analytical Results

The aim of the nodes is to maximize the payoff that they accumulate over time. We compute this cumulative payoff by applying the standard technique used in the theory of iterative games [Axe84]: The cumulative payoff $\overline{\pi}_i$ of a node i is computed as the weighted sum of the payoffs $\pi_i(t)$ that i obtains in each time slot t:

$$\overline{\pi}_i = \sum_{t=0}^{\infty} [\pi_i(t) \cdot \omega^t] \tag{7}$$

where $0 < \omega < 1$, and hence, the weights exponentially decrease with t. The value ω is often called the *discounting factor*, as it represents the degree to which the payoff of each time slot is discounted relative to the previous time slot.

Recall that $S_i(t)$ denotes the set of routes where i is the source, and $F_i(t)$ denotes the set of routes where i is an intermediate node. Since we assumed that the routes remain static, $S_i(t)$ and $F_i(t)$ do not change in time, and we will simply write S_i and F_i instead of $S_i(t)$ and $F_i(t)$. In addition, since we assumed that each node is source on exactly one route, S_i is a singleton. We denote the single route in S_i by r_i. The cardinality of F_i will be denoted by $|F_i|$. Let us assume that node i is an intermediate node on route r. We denote the set of intermediate nodes on r upstream from node i including node i in time slot t by $\Phi(r, i)$. If node i has a reactive dependency loop with some of its sources, then we consider the shortest loop for each source. We denote the length of the longest of such reactive dependency loops minus one by Δ_i.

Theorem 1. *If a node i does not have any dependency loops, then its best strategy is AllD (i.e., to choose cooperation level 0 in every time slot).*

Proof. Node i wants to maximize its cumulative payoff $\overline{\pi}_i$ defined in (7). In our case, $\pi_i(t)$ can be written as:

$$\pi_i(t) = \xi_i(r_i, t) + \sum_{r \in F_i} \eta_i(r, t)$$

$$= [(G + L) \cdot y_i(t) - L] - C \cdot \sum_{r \in F_i} \prod_{k \in \Phi(r,i)} x_k(t)$$

Given that i has no dependency loops, $y_i(t)$ is independent of all the previous cooperation levels of node i. Thus, $\overline{\pi}_i$ is maximized if $x_i(t) = 0$ for all $t \geq 0$. $\quad\square$

Theorem 2. *If a node i has only non-reactive dependency loops, then its best strategy is AllD.*

Proof. The proof is similar to the proof of Theorem 1. Since all dependency loops of i are non-reactive, its experienced normalized throughput $\overline{y}_i$ is independent of its own behavior $\overline{x}_i$. This implies that its best strategy is not to forward. $\quad\square$

Corollary 1. *If every node j ($j \neq i$) plays AllD, then the best response of i to this is AllD. Hence, every node playing AllD is a Nash equilibrium.*

In order to illustrate the case in which some nodes have no dependency loops, we modify the network of Fig. 1. In the modified network, shown in Fig. 5, nodes 2 and 5 have no dependency loops.
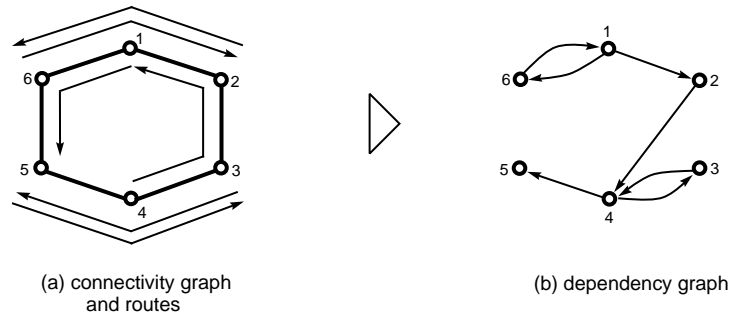

(a) connectivity graph
and routes

(b) dependency graph

Fig. 5. A network with routes where nodes 2 and 5 have no dependency loops

In the case when the conditions of Theorems 1 and 2 do not hold, we could not determine the best strategy of a node i in general, because it very much depends on the particular scenario (dependency graph) in question and the strategies played by the other nodes. However, in certain cases, the best strategy can be determined, as shown by the following theorem:

Theorem 3. *The best strategy for node i is TFT, if:*

1. *Node i has a reactive dependency loop with at least one of its sources. In addition, for each of its sources s, either node i has a reactive dependency loop with s or i is the first forwarder for s,*
2. *every other node j ($j \neq i$) plays TFT, and*
3. $(G + L) \cdot \omega^{\Delta_i} > |F_i| \cdot C.$

The expression $(G + L) \cdot \omega^{\Delta_i} > |F_i| \cdot C$ means that the possible benefit for node i must be greater than its maximal forwarding cost considering every source node with which node i has a reactive dependency loop.

Because of space limitations we only provide the idea of the proof: We give an upper bound for the total payoff of any node using the concept of dependency graph. Then, we show that the node playing TFT reaches the maximum payoff defined by the upper bound. The complete proof is provided in [FBH03].

Corollary 2. *If conditions 1 and 3 of Theorem 3 holds for every node, then all nodes playing TFT is a Nash equilibrium.*

As an example, let us consider Fig. 1 again. If the third condition of Theorem 3 holds for each node, then all nodes playing TFT is a Nash equilibrium for this network.

The first two theorems state that if the behavior of node i has no effect on its experienced normalized throughput, then defection is the best choice for i. In addition, Corollary 1 says that if every node defects constantly, it is a Nash equilibrium. The condition of Theorem 2 implies that it would be a naïve approach to program all devices with the strategy that always forwards (AllC), because this would make the network exploitable by selfish participants. Instead, manufacturers should program a reactive strategy into the devices. Theorem 3 and its corollary mentions the existence of a cooperative equilibrium (each node playing TFT). Yet, the conditions for this existence are extremely stringent.

4.2 Simulation Results

In this subsection our goal is to determine the likelihood that a cooperative equilibrium exists (i.e., the conditions of Theorem 3 hold). To this end, we performed simulations in randomly generated scenarios. We summarize the parameters in Table 2.

Table 2. Parameter values for the simulation

Parameter	Value
Number of nodes	100
Area type	Torus
Area size	1500 m^2
Radio range	250 m
Route length	4 hops

In our simulations, we randomly placed nodes in a toroid[5] area. Then, for each node, we randomly chose a destination at a fixed distance (in hops) from the node and we determined a route to this destination using any shortest path algorithm. If several routes existed, then we randomly chose a single one. From the routes, we built up the dependency graph of the network. On this dependency graph, we performed a search for reactive dependency loops while varying the fraction of nodes that played non-reactive strategies. All the results presented in the following subsections are the mean values of 100 simulations with a 95% confidence interval.

Figure 6 shows the proportion of nodes that (a) can play reactive strategies (i.e. are not programmed to play non-reactive strategies), (b) are forwarders at least on one route, (c) have at least one reactive dependency loop and (d) fulfill the first condition of Theorem 3.
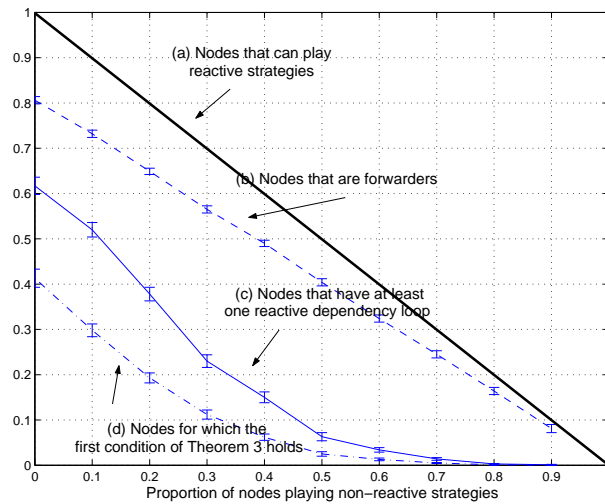

Fig. 6. The proportion of nodes that (a) can play reactive strategies (i.e. are not programmed to play non-reactive strategies), (b) are forwarders at least on one route, (c) have at least one reactive dependency loop and (d) fulfill the first condition of Theorem 3

The nodes that cannot play reactive strategies are irrelevant to our investigations because they follow a fixed packet forwarding strategy, thus they do not make a decision about packet forwarding at all. Additionally, there is a fraction of nodes that are not forwarders in any of the routes, thus they have no effect on other nodes. The nodes in this situation are also irrelevant to our study. In Fig. 7 we show the same results as in Fig. 6 excluding these two types of nodes. We can see that if initially every node can play reactive strategy, the proportion of nodes that fulfill the first condition of Theorem 3

_____

[5] We used this area type to avoid border effects. In a realistic scenario, the toroid area can be considered as an inner part of a large network.

is about 50 percent. This proportion decreases as we increase the proportion of nodes that play constantly non-reactive strategies.
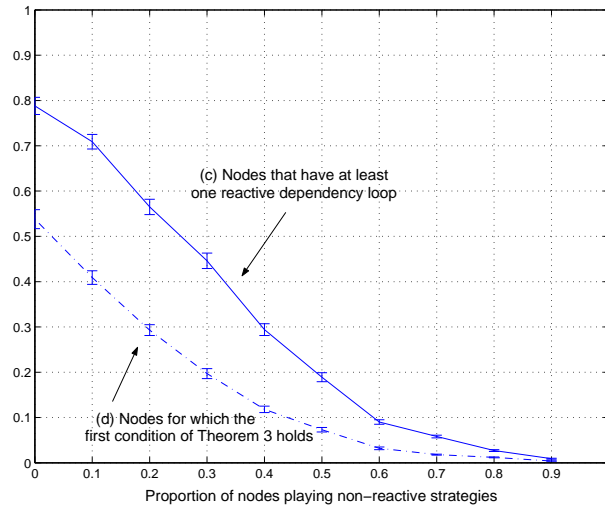

Fig. 7. The proportion of nodes (c) have at least one reactive dependency loop and (d) fulfill the first condition of Theorem 3

If there exists at least one node that needs an incentive mechanism to cooperate, then we say that an incentive mechanism must be applied for the whole network. Since the proportion of such nodes is always around or above 50 percent, we can conclude that in general an incentive mechanism is almost always needed under the assumption we have considered.

5 Related Work

Energy-efficient cooperation in ad hoc networks. Srinivasan *et al.* [SNCR03] provide a mathematical framework for cooperation in ad hoc networks, which focuses on the energy-efficient aspects of cooperation. In their solution, the nodes are classified in different energy classes. The nodes differentiate between the connections based on the energy classes of the participants and apply different behavior according to the type of the connection. This framework relies on an ideal mechanism that distributes class information. We introduce a game theoretic model that does not rely on an additional mechanism, thus our investigations are more generic. Similarly to our framework, Srinivasan *et al.* define time slots as an operating unit. But, they generate only one communication session in the network in each time slot. They choose the participating nodes for this session uniformly. In our model we take the constitution of the network also

into account, which is in our opinion an important factor. Finally, we develop a model for the nodes where they can make decisions based only on local information.

Incentive mechanism in ad hoc networks. Marti *et al.* [MGLB00] consider an ad hoc network where some misbehaving nodes agree to forward packets but then fail to do so. However, in their solution misbehaving nodes are not punished, and thus there is no motivation for the nodes to cooperate. To overcome this problem, Buchegger and Le Boudec [BL02] as well as Michiardi and Molva [MM02] define protocols that are based on a reputation system. Another solution presented by Buttyan and Hubaux [BH03] that gives an incentive for cooperation is based on a virtual currency, called *nuglets or credits*. Plaggemeier *et al.* [PWL03] study the impact of incentive mechanisms in general.

Application of game theory to networking. Game theory has been used to solve problems both in fixed and cellular networks. Researchers addressed problems like power control (e.g. Ji and Huang [JH98]), resource allocation (e.g. Korilis *et al.* [KLO95]) and pricing (e.g. Yaïche *et al.* [YMR00]).

Cooperation studies in other areas of science. The emergence of cooperation has also been previously studied in an economic context using the *Iterated Prisoner's Dilemma (IPD)* game as an underlying model(see e.g. Axelrod [Axe84,CRA99]). Wahl and Nowak [WN99] study the *Continuous valued Prisoner's Dilemma (CPD)* game, where the nodes can choose a degree of cooperation between full cooperation and full defection. Our model is also based on a continuous valued game.

6 Conclusion

In this paper, we have proposed a theoretical framework to study the possible existence of cooperation in packet forwarding in multi-hop wireless networks. We have introduced the concept of dependency graphs, based on which we were able to prove several theorems. As a main result, we proved the existence of a cooperative equilibrium of packet forwarding strategies. But, our simulation results show that, in general, the conditions for such a cooperative equilibrium are satisfied for at most half of the nodes. Thus, the likelihood that the condition for Corollary 2 holds (i.e., every node has TFT as the best strategy) is extremely small.

It is important to notice that our approach does not require a node to keep track of the individual behavior of other nodes: indeed, the considered node is served simply in a way that reflects the way it served others. As we have shown, this concept is captured by the game in which a given node perceives itself as playing with the rest of the network. In this way, the node does not even need to know the identity of the nodes forwarding (or deciding to not forward) its packets.

In this paper, we have assumed the routes to be static; this drastic decision was motivated by the complexity of the problem and by our willingness to provide formal results. In terms of future work, we intend to study the impact of mobility; we expect mobility to have beneficial effects, as it will increase the dependency between the nodes. In this work, we focused on the possible *existence* of equilibria; in the future, we will also investigate the possible *emergence* of such equilibria. Finally, we intend to relax the assumption of linearity of the utility function.

References

[Axe84]   Robert Axelrod, *The Evolution of Cooperation*, Basic Books, New York, 1984.

[BL02]   S. Buchegger, J-Y. Le Boudec, "Performance Analysis of the CONFIDANT Proto-col (Cooperation Of Nodes–Fairness In Dynamic Ad-hoc NeTworks)," *In Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mo-biHoc'02)*, Lausanne, Switzerland, pp. 80-91, June 9-11, 2002.

[BH03]   L. Buttyán and J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," to appear in *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Mobile Ad Hoc Networks*, Vol. 8 No. 5, October 2003.

[CRA99]   M. C. Cohen, R. L. Riolo, R.Axelrod, "The Emergence of Social Organization in the Prisoner's Dilemma: How Context-Preservation and Other Factors Promote Cooper-ation," Santa Fe Institute *Technical report Nr. 99-01-002*, submitted to *Nature*

[FBH03]   M. Félegyházi, L. Buttyán, J.-P. Hubaux, "Equilibrium Analysis of Packet Forward-ing Strategies in Wireless Ad Hoc Networks – the Static Case (extended version)," Technical Report Nr. IC/2003/33

[JH98]   H. Ji, C-Y. Huang, "Non-Cooperative Uplink Power Control in Cellular Radio Sys-tems," *Wireless Networks (WINET)*, vol. 4, pp 233-240, Apr. 1998

[KLO95]   Y. Korilis, A. Lazar, A. Orda, "Architecting Noncooperative Networks," *IEEE Jour-nal on Selecte Areas in Communication*, vol. 13, No. 8, 1995

[MGLB00]   S. Marti, T.J. Giuli, K. Lai, M. Baker, "Mitigating Routing Misbehavior in Movile Ad Hoc Networks," *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'00)*, pp. 255-265, 2000.

[MM02]   P. Michiardi, R. Molva, "Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks," *Communication and Multimedia Se-curity 2002*, Portoroz, Slovenia, September 26-27, 2002.

[Nash50]   J. Nash, "Equilibrium Points in N-person Games," *Proceedings of the National Academy of Sciences*, 36:48-49, 1950.

[PWL03]   M. Plaggemeier, D. Westhoff, B. Lamparter, "About the Impact of Co-operation Ap-proaches for Ad Hoc Networks," Poster presentation at *ACM International Sympo-sium on Mobile Ad Hoc Networking and Computing (MobiHoc'03)*, Annapolis, MD, USA, June 1-3, 2003.

[SNCR03]   V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, R. R. Rao, "Cooperation in Wireless Ad Hoc Networks," *In Proceedings of IEEE INFOCOM'03*, San Francisco, Mar 30 - Apr 3, 2003

[WN99]   L. M. Wahl, M. A. Nowak, "The Continuous Prisoner's Dilemma: I. Linear Reactive Strategies," *Journal of Theoretical Biology* vol. 200, 307-321., 1999.

[YMR00]   H. Yaïche, R. R. Mazumdar, C. Rosenberg, "A Game Theoretical Framework for Bandwidth Allocation and Pricing in Broadband Networks," *IEEE/ACM Transactions on Networking*, vol. 8, No. 5, October 2000.