

Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks

Márk Félegyházi, *Student Member, IEEE*, Jean-Pierre Hubaux, *Senior Member, IEEE*, and Levente Buttyán

Abstract—In self-organizing ad hoc networks, all the networking functions rely on the contribution of the participants. As a basic example, nodes have to forward packets for each other in order to enable multihop communication. In recent years, incentive mechanisms have been proposed to give nodes incentive to cooperate, especially in packet forwarding. However, the need for these mechanisms was not formally justified. In this paper, we address the problem of whether cooperation can exist *without* incentive mechanisms. We propose a model based on game theory and graph theory to investigate equilibrium conditions of packet forwarding strategies. We prove theorems about the equilibrium conditions for both cooperative and noncooperative strategies. We perform simulations to estimate the probability that the conditions for a cooperative equilibrium hold in randomly generated network scenarios. As the problem is involved, we deliberately restrict ourselves to a static configuration. We conclude that in static ad hoc networks—where the relationships between the nodes are likely to be stable—cooperation needs to be encouraged.

Index Terms—Ad hoc networks, cooperation, graph theory, game theory, Nash equilibrium.

1 INTRODUCTION

IN multihop wireless ad hoc networks, networking services are provided by the nodes themselves. As a fundamental example, the nodes must make a mutual contribution to packet forwarding in order to ensure an operable network. If the network is under the control of a single authority, as is the case for military networks and rescue operations, the nodes cooperate for the critical purpose of the network. However, if each node is its own authority, cooperation between the nodes cannot be taken for granted; on the contrary, it is reasonable to assume that each node has the goal to maximize its own benefits by enjoying network services and at the same time minimizing its contribution. This selfish behavior can significantly damage network performance [4], [14].

In recent years, researchers have identified the problem of stimulating cooperation in ad hoc networks and proposed several solutions to give nodes incentive to contribute to common network services. These solutions are based on a reputation system [3], [15] or on a virtual currency [5], [27]. All of these solutions are heuristics to provide a reliable cooperation enforcement scheme. However, it has never been formally proven that these techniques are really needed.

Recently, some researchers have claimed that under specific conditions, cooperation may emerge without incentive techniques [21], [23]. However, they have assumed a random connection setup, thus abstracting away the

topology of the network. This paper aims at determining under which conditions such cooperation without incentives can exist, while taking the network topology into account. Indeed, in reality, the interactions between nodes are not random, as they are determined by the network topology and the communication pattern in the network.

We focus on the most basic networking mechanism, namely, packet forwarding. 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 involved, we deliberately restrict ourselves to a static configuration. This work expands our previous publication on the topic [7] and is part of the MICS/Terminodes Project [10].

The remainder of the paper is organized in the following way: In Section 2, we provide an overview of related work. In Section 3, we show how packet forwarding can be modeled in a game theoretic framework. In Section 4, we analyze the model introduced in Section 3. We provide our main results stated as theorems in Section 5. We extend the model in Section 6. Our simulation results are presented in Section 7. In Section 8, we compare our solution to existing approaches. Finally, we conclude the paper in Section 9.

2 STATE-OF-THE-ART

2.1 Incentive Mechanisms in Ad Hoc Networks

The operation of ad hoc networks relies on the contribution of nodes. Several researchers have realized that this cooperation is not obvious and have proposed solutions to give nodes incentive to contribute. There are basically two approaches to motivate nodes: 1) by denying service to misbehaving nodes by means of a reputation mechanism or 2) by remunerating honest nodes, using for example a micropayment scheme. We provide an overview of these approaches below.

- M. Félegyházi and J.-P. Hubaux are with EPFL (Swiss Federal Institute of Technology—Lausanne), EPFL-IC-LCA, 1015, Lausanne, Switzerland. E-mail: {mark.felegyhazi, jean-pierre.hubaux}@epfl.ch.
- L. Buttyán is with the Budapest University of Technology and Economics, BME-HIT, PO Box 91, 1521 Budapest, Hungary. E-mail: buttyan@crysys.hu.

Manuscript received 30 July 2003; revised 18 Feb. 2004; accepted 25 Aug. 2004; published online 15 Mar. 2006.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0115-0703.

Marti et al. [14] consider an ad hoc network where some misbehaving nodes agree to forward packets but then fail to do so. They propose a mechanism, called *watchdog*, in charge of identifying the misbehaving nodes and another mechanism, called *pathrater*, that deflects the traffic around them. The drawback of their solution is that misbehaving nodes are not punished and, thus, there is no incentive for the nodes to cooperate. To overcome this problem, Buchegger and Le Boudec [3] as well as Michiardi and Molva [15] define protocols that are based on a reputation system. In both approaches, the nodes observe the behavior of each other and store this knowledge locally. Additionally, they distribute this information in reputation reports. According to their observations, the nodes are able to behave selectively (e.g., nodes may deny forwarding packets for misbehaving nodes). However, such a scheme requires a reliable authentication scheme. Otherwise, it is vulnerable to the Sybil attack [6]. Note that authentication is an open issue in ad hoc networks and the Sybil attack is proven to be always possible if a central authority is not present in the network.

Other researchers proposed schemes that employ a virtual currency system to motivate cooperation. Zhong et al. [27] present a solution, where an offline central authority collects *receipts* from the nodes that relay packets and remunerates them based on these receipts. They rely on public key cryptography to process each packet. Thus, their solution might be too complex in an ad hoc network. Another solution, presented by Buttyan and Hubaux [4], [5], is based on a virtual currency, called a *nuglet*: If a node wants to send its own packets, it has to pay for it; whereas, if the node forwards a packet for the benefit of another node, it is rewarded. However, some mechanisms of this solution (e.g., the generation of nuglets) still need to be investigated.

2.2 Cooperation without Incentive Mechanisms

The proposals that we have just described were based on heuristics. There was a need for a formal description of the cooperation problem in ad hoc networks.

In [21], Srinivasan et al. 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 and the behavior of each node depends on the energy classes of the participants of each connection. As this paper is the closest to our work, we discuss it in detail in Section 8. Urpi et al. [23] propose a general framework for cooperation without any incentive mechanisms.

In our paper, we analyze the same problem: Is cooperation possible based on the rational behavior of the nodes, or are incentive mechanisms needed for cooperation? However, in contrast to previous approaches, we believe that the network topology and the communication patterns in the network have a significant impact on the existence of spontaneous cooperation.

2.3 Application of Game Theory to Networking

Game theory has been used to solve various problems in ad hoc, fixed, and cellular networks. Qiu and Marbach [18] define a price-based approach for bandwidth allocation in wireless ad hoc networks. Jin and Kesidis [11] propose a

generic mechanism for rate control and study Nash equilibria in a networking game. Alpcan et al. [1] apply game theory for uplink power control in cellular networks. In [26], Xiao et al. describe a utility-based power control framework for a cellular system. In [9], Goodman and Mandayam introduce the concept of network-assisted power control to equalize signal-to-interference ratio between the users. Korilis et al. [12] address the problem of allocating link capacities in routing decisions; in [13], Korilis and Orda suggest a congestion-based pricing scheme. Roughgarden [20] quantifies the worst-possible loss in network performance arising from noncooperative routing behavior. In [25], Yaïche et al. present a game theoretic framework for bandwidth allocation; they study the centralized problem and show that the solution can be distributed in a way that leads to a system-wide optimum.

2.4 Cooperation Studies in Other Areas of Science

Most of the studies that consider cooperation of entities use the *Iterated Prisoner's Dilemma (IPD)* game as their underlying model (see e.g., Axelrod [2], Rapaport and Chammah [19], or Trivers [22]). The simplicity of the IPD makes it an attractive model. The *Continuous valued Prisoner's Dilemma (CPD)* game was studied by Wahl and Nowak [24]. In the CPD game, the nodes can choose a degree of cooperation between full cooperation and full defection. In [2], Axelrod identifies *Tit-for-Tat (TFT)* as a robust strategy that performs surprisingly well (in terms of maximizing the player's payoff) in the Prisoner's Dilemma games. TFT begins with cooperation in the first round and then repeats the previous move of the other player. We will see that cooperation based on TFT exists also in the ad hoc networking context. The classical Prisoner's Dilemma game is not appropriate for modeling packet forwarding because it involves only two players that have symmetric roles. Hence, in this paper, we define a multiplayer, asymmetric game that is inspired by the classical Prisoner's Dilemma game, which better suits our purposes.

3 GAME THEORETIC MODEL OF PACKET FORWARDING

3.1 Preliminaries

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.

Communication between two nonneighboring nodes is based on multihop relaying. This means that packets from the source to the destination are forwarded by intermediate nodes. For a given source and destination, the intermediate nodes are those that form the shortest path¹ between the

1. In other words, we abstract away the details of the routing protocol, and we model it as a function that returns the shortest path between the source and the destination. If there are multiple shortest paths, then one of them is selected at random.

source and the destination in the connectivity graph. We call such a chain of nodes (including the source and the destination) a *route*. We call the topology of the network with a given set of communicating nodes a *scenario*.

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 needed to relay a packet from the source to the destination. This means that a node is able to send several packets within one time slot. This allows us to abstract away individual packets and to represent the data traffic in the network with *flows*. We assume CBR flows, which means that a source node sends the same amount of traffic in each time slot. Note, however, that this amount may be different for every source node and every route.

3.2 Forwarding Game

We model the operation of the network 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, whereas 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 $p_i(t)$, which it applies to every route in which it is involved as a forwarder. We prefer to not require the nodes to be able to distinguish the flows 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 at the nodes 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, \dots, f_ℓ . Let us denote by $T_s(r)$ the constant amount of traffic that s wants to send on r in each time slot. 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 packet losses in the network is the noncooperative 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 $T_s(r)$ and the cooperation levels of all intermediate nodes:

$$\tau(r, t) = T_s(r) \cdot \prod_{k=1}^{\ell} p_{f_k}(t). \quad (1)$$

In addition, we define the normalized throughput $\hat{\tau}(r, t)$ as follows:

$$\hat{\tau}(r, t) = \frac{\tau(r, t)}{T_s(r)} = \prod_{k=1}^{\ell} p_{f_k}(t). \quad (2)$$

We will use the normalized throughput later as an input of the strategy function of s .

The payoff $\xi_s(r, t)$ of s on r in t depends on the experienced throughput $\tau(r, t)$. In general, $\xi_s(r, t) = u_s(\tau(r, t))$, where the utility u_s is some nondecreasing function. In this paper, we further assume that u_s is concave, derivable at $T_s(r)$, and $u_s(0) = 0$. We place no other restrictions on u_s . Note that the utility function of different nodes may be different.

The payoff $\eta_{f_j}(r, t)$ of the j th intermediate node f_j on r in t is nonpositive and represents the cost for node f_j to forward packets on route r during time slot t . It is defined as follows:

$$\eta_{f_j}(r, t) = -T_s(r) \cdot c \cdot \hat{\tau}_j(r, t), \quad (3)$$

where c is the cost of forwarding one unit of traffic, and $\hat{\tau}_j(r, t)$ is the normalized throughput on r in t leaving node j . For simplicity, we assume that the nodes have the same, fixed transmission power and, therefore, c is the same for every node in the network, and it is independent from r and t . $\hat{\tau}_j(r, t)$ is computed as the product of the cooperation levels of the intermediate nodes from f_1 up to and including f_j :

$$\hat{\tau}_j(r, t) = \prod_{k=1}^j p_{f_k}(t). \quad (4)$$

In our model, the payoff of the destination is 0. In other words, we assume that only the source benefits if the traffic reaches the destination (information push). However, our model can be applied in the reverse case: All our results also hold when only the destination benefits from receiving traffic. An example of this case is a file download (information pull).

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

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

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

3.3 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 $p_i(t)$ in time slot t based on the normalized throughput it experienced in time slot $t - 1$ on the routes where it was a source:

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

where $[\hat{\tau}(r, t - 1)]_{r \in S_i(t-1)}$ represents the normalized throughput vector for node i in time slot $t - 1$, each element of which is 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.

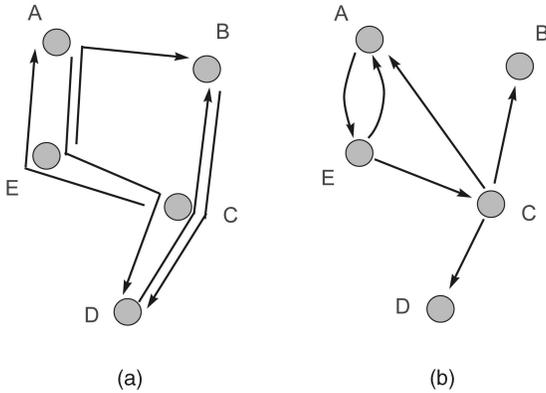


Fig. 1. Representation of a network: (a) A graph showing five routes and (b) the corresponding dependency graph.

There is an infinite number of possible strategies; here, we highlight only a few of them for illustrative purposes. In these examples, we assume that the input of the strategy function is a scalar (i.e., a vector of length 1) denoted by in below:

- *Always Defect (AllD)*: A node playing this strategy defects in the first time slot, and then uses the strategy function $\sigma_i(in) = 0$.
- *Always Cooperate (AllC)*: A node playing this strategy starts with cooperation, and then uses the strategy function $\sigma_i(in) = 1$.
- *Tit-For-Tat (TFT)*: A node playing this strategy starts with cooperation, and then mimics the behavior of its opponent in the previous time slot. The strategy function that corresponds to the TFT strategy is $\sigma_i(in) = in$.
- *Suspicious Tit-For-Tat (S-TFT)*: A node playing this strategy defects in the first time slot, and then applies the strategy function $\sigma_i(in) = in$.
- *Anti Tit-For-Tat (Anti-TFT)*: A node playing this strategy does exactly the opposite of what its opponent does. In other words, after cooperating in the first time slot, it applies the strategy function $\sigma_i(in) = 1 - in$.

If the output of the strategy function is independent of its input, then the strategy is called a *nonreactive strategy* (e.g., AllD or AllC). If the output depends on the input, then the strategy is *reactive* (e.g., TFT or Anti-TFT).

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

4 METAMODEL

In this section, we introduce a metamodel in order to formalize the properties of the packet forwarding game defined in the previous section. In the metamodel, we focus on the evolution of the cooperation levels of the nodes; all other details of the model defined earlier (e.g., amounts of traffic, forwarding costs, and utilities) are abstracted away. Unlike in the model, in the metamodel and in the remainder of the paper, we will assume that routes remain unchanged

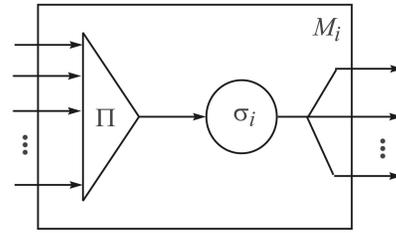


Fig. 2. Internal structure of machine M_i .

during the lifetime of the network. In addition, we assume for the moment that each node is the source of only one route.²

Let us consider a route r . The payoff received by the source on r depends on the cooperation levels of the intermediate nodes on r . We represent this dependency relationship 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.

Now, we define the automaton Θ that will model the unfolding of the forwarding game in the metamodel. 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 multiplication³ gate Π followed by a gate that implements the strategy function σ_i of node i . The multiplication gate Π takes the values on the input 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 .

The automaton Θ works in discrete steps. As an example, Fig. 3 shows the automaton that corresponds to the dependency graph of Fig. 1. 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 $t - 1$.

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 (which, in fact, represent the state of the automaton)

2. We will relax this assumption in Section 6. We emphasize that all of our analytical results hold in the extended case as well.

3. The multiplication comes from the fact that the experienced normalized throughput for the source (which is the input of the strategy function of 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 of only one route.

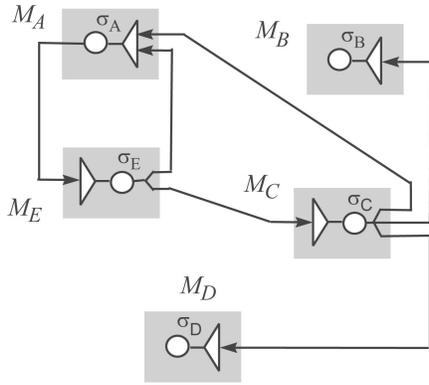


Fig. 3. The automaton that corresponds to the dependency graph of Fig. 1.

on the output links of the machines models the evolution of the cooperation levels of the nodes in the network.

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 Θ . What remains is the automaton without σ_i , which we denote by Θ_{-i} . Θ_{-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 Θ . 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 Θ_{-i} is the sequence $\bar{x}_i = x_i(0), x_i(1), \dots$ of the cooperation levels of i , and its output is the sequence $\bar{y}_i = y_i(0), y_i(1), \dots$ of the normalized throughput values for i .

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

$$\begin{aligned} y_A(0) &= x_C(0) \cdot x_E(0) \\ y_A(1) &= \sigma_C(x_E(0)) \cdot \sigma_E(x_A(0)) \\ y_A(2) &= \sigma_C(\sigma_E(x_A(0))) \cdot \sigma_E(x_A(1)) \\ y_A(3) &= \sigma_C(\sigma_E(x_A(1))) \cdot \sigma_E(x_A(2)) \\ \dots & \quad \dots \end{aligned}$$

A *dependency loop* L of node i is a sequence $(i, v_1), (v_1, v_2), \dots, (v_{l-1}, v_l), (v_l, i)$ of edges in the dependency graph. The length of a dependency loop L is defined as the number of edges in L , and it is denoted by $|L|$. 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. In the example, nodes B and D have no dependency loops.

Every node i has two types of dependency loops; these types depend on the strategies played by the other nodes in the loop. If L is a dependency loop of i and all other nodes $j \neq i$ in L play reactive strategies, then L is said to be a

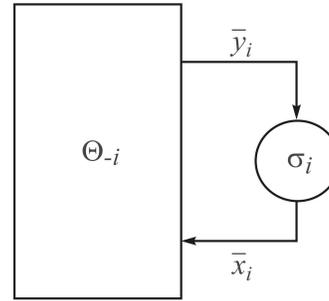


Fig. 4. Model of interaction between node i and the rest of the network represented by the automaton Θ_{-i} .

reactive dependency loop of i . If, on the contrary, there exists at least one node $j \neq i$ in L that plays a nonreactive strategy, then L is called a *nonreactive dependency loop* of i .

5 ANALYTICAL RESULTS

Our goal, in this section, is to find *possible* Nash equilibria⁵ [8], [16], [17] of packet forwarding strategies. In the next section, we will investigate the *probability of fulfillment* of the conditions for possible Nash equilibria in randomly generated scenarios. The existence of a Nash equilibrium based on cooperation would mean that there are cases in which cooperation is “naturally” encouraged, i.e., without using incentive mechanisms. In the following, we use the model and the metamodel that we introduced earlier.

The goal of the nodes is to maximize the payoff that they accumulate over time. However, the end of the game is unpredictable. Thus, we apply the standard technique used in the theory of iterative games [2]. We model the *finite* forwarding game with an unpredictable end as an *infinite* game where future payoffs are *discounted*. The cumulative payoff $\bar{\pi}_i$ of a node i is computed as the weighted sum of the payoffs $\pi_i(t)$ that i obtains in each time slot t :

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

where $0 < \omega < 1$ and, hence, the weights exponentially decrease with t . The *discounting factor* ω 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 for which i is the source and that $F_i(t)$ denotes the set of routes for which i is an intermediate node. As we assume that the routes remain static, meaning that $S_i(t)$ and $F_i(t)$ do not change over time, we will simply write S_i and F_i instead of $S_i(t)$ and $F_i(t)$. In addition, since we assume that each node is a source on exactly one route, S_i is a singleton. We denote the single route in S_i by r_i , and the amount of traffic sent by i on r_i in every time slot by T_i . The cardinality of F_i will be denoted by $|F_i|$. For any route $r \in F_i$, we denote the set of intermediate nodes on r upstream from node i (including node i) by $\Phi(r, i)$. Moreover, $\Phi(r)$ denotes the set of all forwarder nodes on route r , and $src(r)$ denotes the source of route r . Finally, the set of nodes that are forwarders on at least one route is denoted by Φ (i.e., $\Phi = \{i \in N : F_i \neq \emptyset\}$).

5. Recall that in a *Nash equilibrium*, none of the players can increase its utility by unilaterally changing its strategy.

Theorem 1. *If a node i is in Φ and it has no 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 $\bar{\pi}_i$ defined in (7). In our case, $\pi_i(t)$ can be written as:

$$\begin{aligned} \pi_i(t) &= \xi_i(r_i, t) + \sum_{r \in F_i} \eta_i(r, t) \\ &= u_i(T_i \cdot y_i(t)) - \sum_{r \in F_i} T_{src(r)} \cdot c \cdot \prod_{k \in \Phi(r, i)} x_k(t). \end{aligned}$$

Given that i has no dependency loops, $y_i(t)$ is independent of all the previous cooperation levels $x_i(t')$ ($t' < t$) of node i . Thus, $\bar{\pi}_i$ is maximized if $x_i(t') = 0$ for all $t' \geq 0$. \square

Theorem 2. *If a node i is in Φ and it has only nonreactive 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 nonreactive, its experienced normalized throughput \bar{y}_i is independent of its own behavior \bar{x}_i . This implies that its best strategy is full defection. \square

From this theorem, we can easily derive the following corollary:

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.*

If the conditions of Theorems 1 and 2 do not hold, then we cannot 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.

Now, we will show that, under certain conditions, cooperative equilibria do exist in the network. In order to do so, we first prove the following lemma:

Lemma 1. *Let us assume that node i is in Φ , and let us consider a route $r \in F_i$. In addition, let us assume that there exists a dependency loop L of i that contains the edge $(i, src(r))$. If all nodes in L (other than i) play the TFT strategy, then the following holds:*

$$y_i(t + \delta) \leq \prod_{k \in \Phi(r, i)} x_k(t), \quad (8)$$

where $\delta = |L| - 1$.

Proof. Let L be the following sequence of edges in the dependency graph: $(v_0, v_1), (v_1, v_2), \dots, (v_\delta, v_{\delta+1})$, where $v_{\delta+1} = v_0 = i$ and $v_1 = src(r)$. We know that each node is the source of a single route; let us denote by r_{v_j} ($0 < j \leq \delta + 1$) the route, on which v_j is the source. It follows that $r_{v_1} = r$. In addition, we know that the existence of edge (v_j, v_{j+1}) ($0 \leq j \leq \delta$) in the dependency graph means that v_j is a forwarder on $r_{v_{j+1}}$. The following holds for every node v_j ($0 \leq j \leq \delta$):

$$x_{v_j}(t) \geq \prod_{k \in \Phi(r_{v_{j+1}}, v_j)} x_k(t) \geq \prod_{k \in \Phi(r_{v_{j+1}})} x_k(t) = y_{v_{j+1}}(t). \quad (9)$$

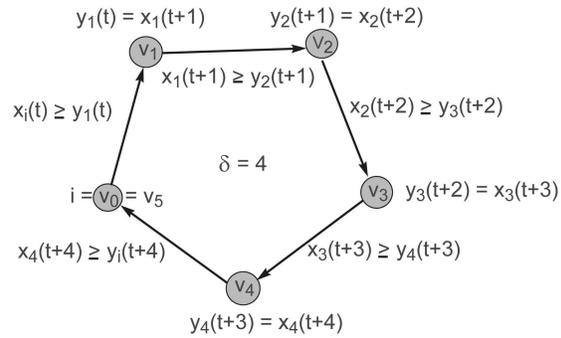


Fig. 5. Example to illustrate the propagation of behavior as expressed formally in Lemma 1.

Furthermore, since every node except for $v_0 = v_{\delta+1} = i$ plays TFT, we have the following for every $0 < j \leq \delta$:

$$x_{v_j}(t + 1) = y_{v_j}(t). \quad (10)$$

Using (9) and (10) in an alternating order, we get the following:

$$\begin{aligned} x_{v_0}(t) &\geq \prod_{k \in \Phi(r_{v_1}, v_0)} x_k(t) \geq y_{v_1}(t) \\ &= x_{v_1}(t + 1) \geq y_{v_2}(t + 1) \\ &= x_{v_2}(t + 2) \geq \dots \geq y_{v_{\delta+1}}(t + \delta). \end{aligned} \quad (11)$$

By substituting i for v_0 and $v_{\delta+1}$, and r for r_{v_1} , we get the statement of the lemma:

$$x_i(t) \geq \prod_{k \in \Phi(r, i)} x_k(t) \geq \dots \geq y_i(t + \delta). \quad (12)$$

\square

As an example, let us consider Fig. 5, which illustrates a dependency loop of length 5 (i.e., $\delta = 4$). According to Lemma 1, if nodes v_1, v_2, v_3 , and v_4 play TFT, then the normalized throughput enjoyed by node i in time slot $t + 4$ is upper bounded by its own cooperation level in time slot t . Intuitively, this means that if node i does not cooperate, then this defection “propagates back” to it on the dependency loop. The delay of this effect is given by the length of the dependency loop.

Theorem 3. *Assuming that node i is in Φ , the best strategy for i is full cooperation in each time slot, if the following set of conditions holds:*

1. For every $r \in F_i$, there exists a dependency loop $L_{i, src(r)}$ that contains the edge $(i, src(r))$.
2. For every $r \in F_i$,

$$\frac{u'_i(T_i) \cdot T_i \cdot \omega^{\delta_{i, src(r)}}}{|F_i|} > T_{src(r)} \cdot c, \quad (13)$$

where $u'_i(T_i)$ is the value of the derivative⁶ of $u_i(\tau)$ at $\tau = T_i$, and $\delta_{i, src(r)} = |L_{i, src(r)}| - 1$.

3. Every node in Φ (other than i) plays the TFT strategy.

The proof of Theorem 3 is provided in the Appendix.

6. Recall the assumption that u_i is derivable at T_i .

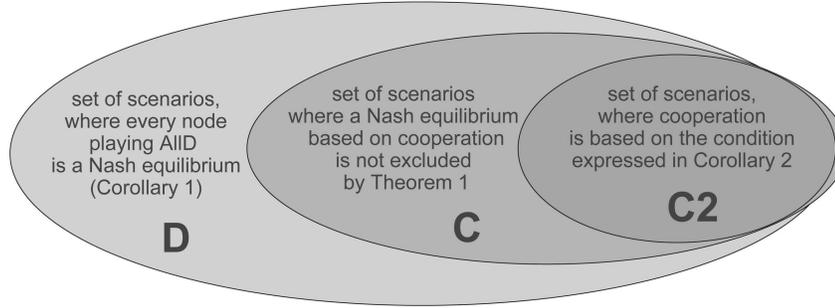


Fig. 6. Classification of scenarios defined by our analytical results.

We have derived *necessary* conditions for spontaneous cooperation from Theorems 1 and 2. The fulfillment of the three conditions of Theorem 3 is *sufficient* for cooperation to be the best strategy for node i . We now discuss these three conditions one by one. *Condition 1* requires that node i has a dependency loop with all of the sources for which it forwards packets. *Condition 2* means that the maximum forwarding cost for node i on every route where i is a forwarder must be smaller than its possible future benefit averaged over the number of routes where i is a forwarder. Finally, *Condition 3* requires that all forwarding nodes in the network (other than node i) play TFT. This implies that all the dependency loops of node i are reactive. We note that the reactivity of the dependency loops can be based on other reactive strategies, different from TFT (for example, Anti-TFT), but in that case the analysis becomes very complex. The analysis of the case when every node plays TFT is made possible by the simplicity of the strategy function $\sigma(x) = x$, which belongs to the TFT strategy. If all three conditions of Theorem 3 are satisfied, then node i has an incentive to cooperate, since otherwise its defective behavior will negatively affect its own payoff. However, as we will show in Section 7, *Condition 1* is a very strong requirement that is virtually never satisfied in randomly generated scenarios.

Both the AllC and TFT strategies result in full cooperation if the conditions of Theorem 3 hold. However, node i should not choose AllC because AllC is a nonreactive strategy, and this might cause other nodes to change their strategies to AllD, as we will show in Section 7. Hence, we can derive the following corollary for cooperative Nash equilibria.

Corollary 2. *If the first two conditions of Theorem 3 hold for every node in Φ , then all nodes playing TFT is a Nash equilibrium.*

In Section 7, we study *Condition 1* of Theorem 3, more specifically, the probability that it is satisfied for all nodes in randomly generated scenarios. Now, we briefly comment on *Condition 2*. As it can be seen, the following factors make *Condition 2* easier to satisfy:

- *Steep utility functions.* The steeper the utility function of node i is, the larger the value of its derivative is at $\tau = T_i$, which, in turn, makes the left side of (13) larger.

- *Short dependency loops.* In *Condition 2*, $\delta_{i,src(r)} + 1$ is the length of *any* dependency loop of node i that contains the edge $(i, src(r))$. Clearly, we are interested in the shortest of such loops, because the smaller $\delta_{i,src(r)}$ is, the larger the value of $\omega^{\delta_{i,src(r)}}$ is, which, in turn, makes the left side of (13) larger. It is similarly advantageous if ω is close to 1, which means, in general, that the probability that the game will continue is higher and thus possible future payoffs count more.
- *Small extent of involvement in forwarding.* The left side of (13) is increased if the cardinality of F_i is decreased. In other words, if node i is a forwarder on a smaller number of routes, then *Condition 2* is easier to satisfy for i .

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 always defects, then this is a Nash equilibrium. Theorem 3 leads to Corollary 2, which shows the existence of a cooperative equilibrium (each node playing TFT) under certain conditions.

Fig. 6 shows a classification of scenarios from the cooperation perspective. In the figure, set D denotes the set of all possible scenarios; indeed, we know from Corollary 1 that all nodes playing AllD is a Nash equilibrium in any possible scenario. Set $C2$ contains those scenarios where the conditions of Corollary 2 hold. Hence, all nodes playing TFT is a Nash equilibrium in every scenario in $C2$. Finally, set C contains those scenarios, where the condition of Theorem 1 does not hold for any of the nodes in Φ , or in other words, where every node in Φ has at least one dependency loop. Determining the Nash equilibria in the scenarios that belong to set $C \setminus C2$ is still an open research problem. In Section 7, we will describe our simulation results that quantify the size of the above sets.

6 EXTENSION OF THE MODEL: ALLOWING NODES TO BE A SOURCE OF SEVERAL ROUTES

In this section, we show that the assumption that each node is a source on only one route can be relaxed. For this, we slightly modify our initial model illustrated in Fig. 2, and we define another machine representation for the node. This new representation is illustrated in Fig. 7. The main novelty is that the outputs of the multiplication gates that correspond

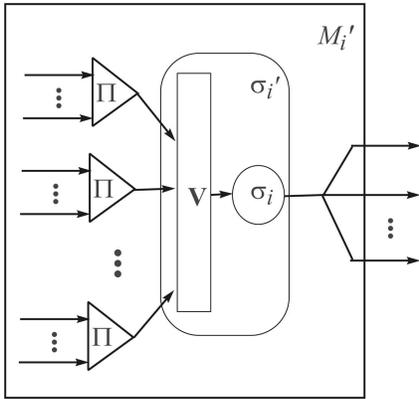


Fig. 7. Modified machine for a node (M_i'). σ_i' represents the modified strategy function for node i .

to the various routes for which the node is the source are transformed into a single value using a function V .

Theorems 1 and 2 still hold for an arbitrary V because the existence of dependency loops does not depend on the internal structure of the machine representing the nodes. Moreover, assuming that V has the property that $V(y_1, y_2, \dots, y_{|S_i|}) \leq y_k$ for all $1 \leq k \leq |S_i|$, we can prove a theorem similar to Theorem 3, where *Condition 1* and *Condition 3* are unchanged, and *Condition 2* has a slightly different form:

Theorem 4. *Assuming that node i is in Φ , the best strategy for i is full cooperation in each time slot, if the following set of conditions holds:*

1. For every $r \in F_i$, there exists a dependency loop $L_{i,src(r)}$ that contains the edge $(i, src(r))$.
2. For every $r \in F_i$,

$$\frac{\omega^{\delta_{i,src(r)}}}{|F_i|} \cdot \sum_{q \in S_i} u_i'(T_i(q)) \cdot T_i(q) > T_{src(r)} \cdot c,$$

where $T_i(q)$ denotes the amount of traffic that node i sends as a source on route q , while u_i' and $\delta_{i,src(r)}$ have the same meaning as in Theorem 3.

3. Every node in Φ (other than i) plays the TFT strategy.

The proof of Theorem 4 is analogous to that of Theorem 3 and, therefore, we omit it.

An example for such a function V is the minimum function (which we will use in our simulations). This choice of V represents a pessimistic perception for the node: It considers the minimum normalized throughput it receives on the different routes as an aggregate view of all its routes. Similarly to Corollary 2, if the (modified) conditions of Theorem 3 hold for every node that is a forwarder on at least one route, then all nodes playing TFT is a Nash equilibrium.

7 SIMULATION RESULTS

We have run a set of simulations to determine the probability that the conditions of our theorems and their corollaries hold. In particular, our goal is to estimate the probability that the first condition of Theorem 3 holds for

TABLE 1
Parameter Values for the Simulation

| Parameter | Value |
|---------------------------------|------------------------------------|
| Number of nodes | 100, 150, 200 |
| Distribution of the nodes | random uniform |
| Area type | Torus |
| Area size | 1500x1500m, 1850x1850m, 2150x2150m |
| Radio range | 200 m |
| Number of destinations per node | 1-10 |
| Route selection | shortest path |

every node in randomly generated scenarios.⁷ In addition, we also estimate the probability that the condition of Theorem 1 does not hold for any of the nodes in randomly generated scenarios. These probabilities quantify the size of sets $C'2$ and C , respectively.

In our simulations, we randomly place nodes on a toroid⁸ area. Then, for each node, we randomly choose a number of destinations and we determine a route to these destinations using a shortest path algorithm. If several shortest paths existed to a given destination, then we randomly choose a single one. From the routes, we build up the dependency graph of the network. The simulation parameters are summarized in Table 1.

Note that we increase the network size and the simulation area in parallel in order to keep the node density at a constant level. All the presented results are the mean values of 1,000 simulation runs.

In the first set of simulations, we investigate the probability that the first condition of Theorem 3 holds for every node (the size of the set $C'2$ in Fig. 6). Among the 1,000 scenarios that we generated randomly, we observed that there was not a single scenario in which the first condition of Theorem 3 was satisfied for all nodes. Thus, we conclude that the probability of a Nash equilibrium based on TFT as defined in Corollary 2 is very small.

In the second set of simulations, we investigate the proportion of random scenarios, where cooperation of all nodes is not excluded by Theorem 1. Fig. 8 shows the proportion of scenarios, where each node in Φ has at least one dependency loop (the scenarios in set C in Fig. 6) as a function of the number of routes originating at each node. We can observe that for an increasing number of routes originating at each node, the proportion of scenarios, where each node has at least one dependency loop, increases as well. Intuitively, as more routes are introduced in the network, more edges are added to the dependency graph. Hence, the probability that a dependency loop exists for

7. The second condition of Theorem 3 is a numerical one. Whether it is fulfilled or not very much depends on the actual utility functions and parameter values (e.g., amount of traffic and discounting factor) used. Since, by appropriately setting these parameters, the second condition of Theorem 3 can always be satisfied, in our analysis, we make the optimistic assumption that this condition holds for every node in Φ .

8. We use 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.

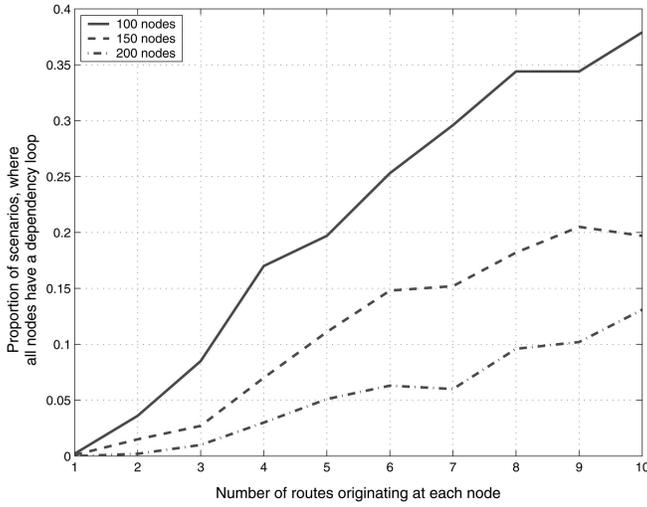


Fig. 8. Proportion of scenarios, where each node that is a forwarder has at least one dependency loop.

each node increases. Furthermore, we can observe that the proportion of scenarios in which each node has at least one dependency loop decreases, as the network size increases. This is due to the following reason: the probability that there exists at least one node for which the condition of Theorem 1 holds increases as the number of nodes increases.

Fig. 8 shows that the proportion of scenarios, where cooperation of all nodes is not excluded by Theorem 1 (set C) becomes significant (with respect to set D) only for cases in which each node is a source of a large number of routes. This implies that the necessary condition expressed by Theorem 1 is a strong requirement for cooperation in realistic settings (i.e., for a reasonably low number of routes per node).

Now, let us consider the case, in which the nodes for which Theorem 1 holds begin to play AllD. This non-cooperative behavior can lead to an “avalanche effect” if the nodes iteratively optimize their strategies: nodes that defect can cause the defection of other nodes. We examine this avalanche effect in a simulation setting as follows.

Let us assume that each node is a source on one route. First, we identify the nodes in the set of forwarders Φ that have AllD as the best strategy due to Theorem 1. We denote the set of these defectors by Z_0 . Then, we search for sources that are dependent on the nodes in Z_0 . We denote the set of these sources by Z_0^+ . Since the normalized throughput of the nodes in Z_0^+ is less than or equal to the cooperation level of any of their forwarders (including the nodes in Z_0), their best strategy becomes AllD, as well, due to Theorem 2. Therefore, we extend the set Z_0 of defectors and obtain $Z_1 = Z_0 \cup Z_0^+$. We extend the set Z_k of defectors iteratively in this way until no new sources are affected (i.e., $Z_k \cup Z_k^+ = Z_k$). The remaining set $\Phi \setminus Z_k$ of nodes is not affected by the behavior of the nodes in Z_k (and, hence, the nodes in Z_0); this means that they are potential cooperators. Similarly, we can investigate the avalanche effect when the nodes are sources of several routes. In that case, we take the pessimistic assumption that the defection of a forwarder causes the defection of its sources. Then, we can iterate the search for the nodes that are affected by defection in the same way as above.

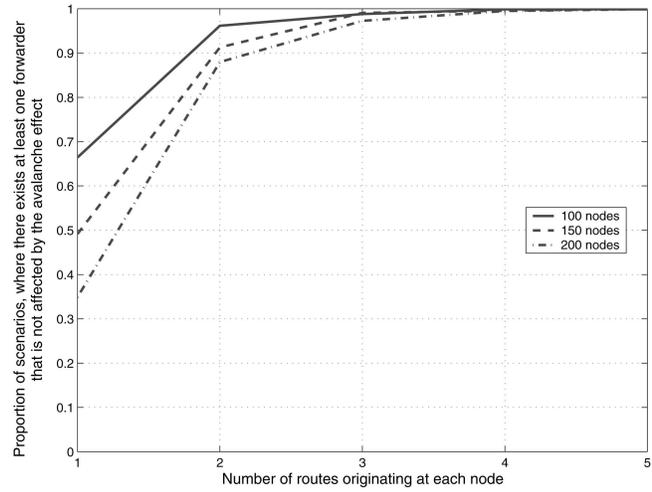


Fig. 9. Proportion of scenarios, where at least one node is not affected by the defective behavior of the initial nodes.

In Fig. 9, we present the proportion of scenarios, where there exists a subset of nodes that are not affected by the defective behavior of the initial AllD players. We can see that this proportion converges rapidly to 1 as the number of routes originating at each node increases. The intuitive explanation is that increasing the number of routes per source (i.e., adding edges to the dependency graph) decreases the probability that Theorem 1 holds for a given node. Thus, as the number of routes per sources increases the number of forwarders that begin to play AllD decreases and so does the number of nodes affected by the avalanche effect.

Additionally, we present in Fig. 10 the proportion of forwarder nodes that are not affected by the avalanche effect. The results show that, if we increase the number of routes originating at each node, the average number of unaffected nodes increases rapidly. For a higher number of routes per node, this increase slows down, but we can observe that the majority of the nodes are not affected by the defective behavior of the initial AllD players.

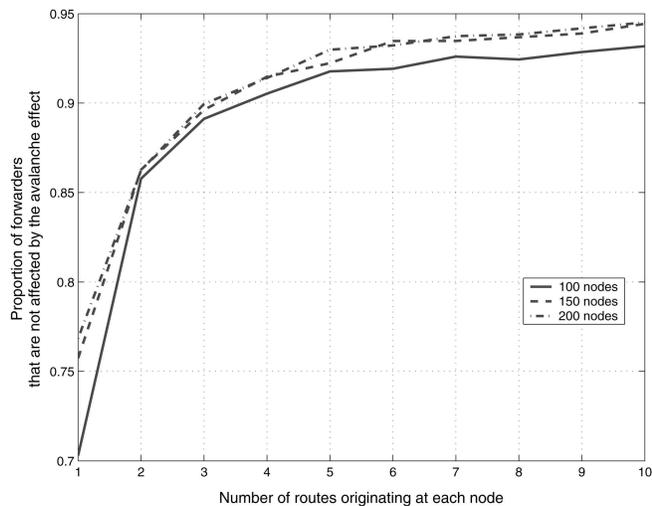


Fig. 10. Average proportion of forwarder nodes that are not affected by the avalanche effect.

8 DISCUSSION

In this section, we give a detailed discussion of the paper written by Srinivasan et al. [21] as this is the work that is the closest to our framework.

In [21], the authors propose a game theoretic model that considers cooperation from the energy efficiency point of view. They consider a maximal battery level and an expected lifetime for each node, and they group the nodes into *energy classes* according to this information. They derive the energy class for a connection as the minimum of the energy classes of the participants. The energy class is a novel idea that allows the authors to express the heterogeneity of devices. They define time slots as a unit of operation for the nodes as we also do in our framework. However, in contrast to our approach, they do not take into account the topology of the network and the existing communication flows. Instead, they assume that a single communication session with random participants is generated in each time slot. Based on the random session generation, they show that cooperation emerges within the network because, by the nature of the random participation in the sessions, nodes have a symmetric interaction pattern. However, in reality, the interactions between nodes are likely to be asymmetric; this is practically true in the extreme case of a static network. In this paper, we have shown that spontaneous cooperation exists only if the interaction between the nodes is balanced and we have also shown that this property does not hold in general. Our conclusion justifies the need for incentive mechanisms, that should reestablish the balance between the utilities of nodes, for example by remunerating nodes that contribute more.

The authors of [21] provide a framework that relies on two mechanisms: The first communicates energy class information while the second enables the relays of a session to communicate their decision to the source (accept or refuse relaying). These mechanisms are needed to optimize the nodes' contribution with respect to energy conditions. From the security point of view, however, these mechanisms are vulnerable. This is an important issue since the whole analysis is about selfish nodes that want to maximize their utility, even if it means disobeying the network protocols. Cheating can be done as follows: First, a high-energy node could use its own identity when sending its own packets and pretend to be a low energy node when asked to forward packets. By doing this, it could decrease its load in terms of packet forwarding. This kind of selfish behavior could be detected using an appropriate authentication scheme, combined with a cheating detection mechanism. Second, in [21], it is assumed that once nodes agree to relay packets in a session, they do so. But, there is no guarantee that a node really complies to its promise. Thus, an additional mechanism should be applied to punish nodes whenever it is necessary. Although far from perfect, our model relies on the *real* behavior of the nodes (and not on their declared behavior) and does not require any form of authentication.

A major contribution of [21] is the investigation of both the existence and emergence of cooperation in wireless ad hoc networks; in this paper, we focus only on the existence of cooperative equilibria. Another important result of [21] is the proof that the emerging cooperative Nash equilibrium is Pareto-efficient (thus, it is a *desired* outcome of the packet forwarding game).

9 CONCLUSION

In this paper, we have presented a game theoretic model to investigate the conditions for cooperation in wireless ad hoc networks, in the absence of incentive mechanisms. Because of the complexity of the problem, we have restricted ourselves to a static network scenario. We have then derived conditions for cooperation from the topology of the network and the existing communication routes. We have introduced the concept of dependency graph, based on which we have been able to prove several theorems. As one of the results, we have proven that cooperation solely based on the self-interest of the nodes can *in theory* exist. However, our simulation results show that, *in practice*, the conditions of such cooperation are virtually never satisfied. We conclude that with a very high probability, there will be some nodes that have AllD as their best strategy and, therefore, these nodes need an incentive to cooperate. In this paper, we have also shown that the behavior of these defectors affects only a fraction of the nodes in the network; hence, local subsets of cooperating nodes are not excluded.

To the best of our knowledge, this work on cooperation in ad hoc networks is the first that takes the topology of the network into account. It is important to notice that our approach does not require a node to keep track of the behavior of other nodes. Thus, our solution does not require any node authentication.

In terms of future work, we intend to investigate cooperation in *mobile* ad hoc networks. Mobility increases mutual dependencies between the nodes; thus, it gives a natural incentive for cooperation. We also want to extend our model by including the effect of battery in the utility of the nodes. In this paper, we have studied the conditions for possible equilibria; a distributed solution that drives the system to the most desirable equilibrium state is also part of our future work.

APPENDIX A

PROOF OF THEOREM 3

In this proof, we will express the maximum possible value of the total payoff for node i in general. Then we will show that the maximum corresponds to the case in which node i fully cooperates. First, we introduce the linear function $f(\tau) = u'_i(T_i) \cdot \tau + u_i(T_i) - u'_i(T_i) \cdot T_i$. Function f is the tangent of function u_i at $\tau = T_i$. Note that due to the fact that u_i is nondecreasing and concave, we have that $f(\tau) \geq u_i(\tau)$ for all τ ; in addition, we have equality at $\tau = T_i$ (i.e., $f(T_i) = u_i(T_i)$).

By definition, the total payoff $\bar{\pi}_i$ of node i is the following:

$$\begin{aligned} \bar{\pi}_i &= \sum_{t=0}^{\infty} \left[\xi_i(r_i, t) + \sum_{r \in F_i} \eta_i(r, t) \right] \omega^t \\ &= \sum_{t=0}^{\infty} \left[u_i(T_i \cdot y_i(t)) - \sum_{r \in F_i} T_{src(r)} \cdot c \cdot \prod_{k \in \Phi(r, i)} x_k(t) \right] \omega^t. \end{aligned} \quad (14)$$

Because of *Condition 1* and *Condition 3*, we can use Lemma 1 to obtain the following inequality for every $r \in F_i$:

$$\prod_{k \in \Phi(r,i)} x_k(t) \geq y_i(t + \delta_{i,src(r)}) \quad (15)$$

which leads to the following upper bound on $\bar{\pi}_i$:

$$\bar{\pi}_i \leq \sum_{t=0}^{\infty} \left[u_i(T_i \cdot y_i(t)) - \sum_{r \in F_i} T_{src(r)} \cdot c \cdot y_i(t + \delta_{i,src(r)}) \right] \omega^t. \quad (16)$$

Since the first term of the right side of (16), $u_i(T_i \cdot y_i(t))$, is independent of r , the following holds:

$$u_i(T_i \cdot y_i(t)) = \sum_{r \in F_i} \frac{u_i(T_i \cdot y_i(t))}{|F_i|}. \quad (17)$$

By substituting the right side of (17) into (16), we get the following:

$$\begin{aligned} \bar{\pi}_i &\leq \sum_{t=0}^{\infty} \left[\sum_{r \in F_i} \frac{u_i(T_i \cdot y_i(t))}{|F_i|} - \sum_{r \in F_i} T_{src(r)} \cdot c \cdot y_i(t + \delta_{i,src(r)}) \right] \omega^t \\ &= \sum_{r \in F_i} \left[\sum_{t=0}^{\infty} \frac{u_i(T_i \cdot y_i(t))}{|F_i|} \cdot \omega^t \right. \\ &\quad \left. - \sum_{t=0}^{\infty} T_{src(r)} \cdot c \cdot y_i(t + \delta_{i,src(r)}) \cdot \omega^t \right]. \end{aligned} \quad (18)$$

Let us consider the first term of (18). We will now split up the summation that goes from $t = 0$ to ∞ into two summations such that one goes from $t = 0$ to $\delta_{i,src(r)} - 1$, and the other goes from $t = \delta_{i,src(r)}$ to ∞ . Then, we shift the index in the second sum in such a way that the summation goes from $t = 0$ to ∞ again:

$$\begin{aligned} &\sum_{t=0}^{\infty} \frac{u_i(T_i \cdot y_i(t))}{|F_i|} \cdot \omega^t \\ &= \sum_{t=0}^{\delta_{i,src(r)}-1} \frac{u_i(T_i \cdot y_i(t))}{|F_i|} \cdot \omega^t + \sum_{t=\delta_{i,src(r)}}^{\infty} \frac{u_i(T_i \cdot y_i(t))}{|F_i|} \cdot \omega^t \\ &= \sum_{t=0}^{\delta_{i,src(r)}-1} \frac{u_i(T_i \cdot y_i(t))}{|F_i|} \cdot \omega^t + \sum_{t=0}^{\infty} \frac{u_i(T_i \cdot y_i(t + \delta_{i,src(r)}))}{|F_i|} \cdot \omega^{t+\delta_{i,src(r)}}. \end{aligned} \quad (19)$$

By writing (19) back into (18), we get the following:

$$\begin{aligned} \bar{\pi}_i &\leq \sum_{r \in F_i} \left[\sum_{t=0}^{\delta_{i,src(r)}-1} \frac{u_i(T_i \cdot y_i(t))}{|F_i|} \cdot \omega^t \right. \\ &\quad \left. + \sum_{t=0}^{\infty} \left[\frac{u_i(T_i \cdot y_i(t + \delta_{i,src(r)}))}{|F_i|} \cdot \omega^{\delta_{i,src(r)}} \right. \right. \\ &\quad \left. \left. - T_{src(r)} \cdot c \cdot y_i(t + \delta_{i,src(r)}) \right] \cdot \omega^t \right]. \end{aligned} \quad (20)$$

Let us consider the first term of (20). Since the utility function u_i is nondecreasing and $y_i(t) \leq 1$, we get the following:

$$\begin{aligned} \sum_{t=0}^{\delta_{i,src(r)}-1} \frac{u_i(T_i \cdot y_i(t))}{|F_i|} \cdot \omega^t &\leq \sum_{t=0}^{\delta_{i,src(r)}-1} \frac{u_i(T_i)}{|F_i|} \cdot \omega^t \\ &= \frac{u_i(T_i)}{|F_i|} \cdot \frac{1 - \omega^{\delta_{i,src(r)}}}{1 - \omega}. \end{aligned} \quad (21)$$

Now, let us consider the second term of (20). By using the fact that $f(\tau) \geq u_i(\tau)$ for all τ , we get the following:

$$\begin{aligned} &\sum_{t=0}^{\infty} \left[\frac{u_i(T_i \cdot y_i(t + \delta_{i,src(r)}))}{|F_i|} \cdot \omega^{\delta_{i,src(r)}} - T_{src(r)} \cdot c \cdot y_i(t + \delta_{i,src(r)}) \right] \cdot \omega^t \\ &\leq \sum_{t=0}^{\infty} \left[\frac{f(T_i \cdot y_i(t + \delta_{i,src(r)}))}{|F_i|} \cdot \omega^{\delta_{i,src(r)}} - T_{src(r)} \cdot c \cdot y_i(t + \delta_{i,src(r)}) \right] \cdot \omega^t \\ &= \sum_{t=0}^{\infty} \left[\frac{u'_i(T_i) \cdot T_i \cdot y_i(t + \delta_{i,src(r)}) + u_i(T_i) - u'_i(T_i) \cdot T_i}{|F_i|} \cdot \omega^{\delta_{i,src(r)}} - T_{src(r)} \cdot c \cdot y_i(t + \delta_{i,src(r)}) \right] \cdot \omega^t \\ &= \sum_{t=0}^{\infty} \left[\frac{u_i(T_i) - u'_i(T_i) \cdot T_i}{|F_i|} \cdot \omega^{\delta_{i,src(r)}} \right. \\ &\quad \left. + \left(\frac{u'_i(T_i) \cdot T_i \cdot \omega^{\delta_{i,src(r)}}}{|F_i|} - T_{src(r)} \cdot c \right) \cdot y_i(t + \delta_{i,src(r)}) \right] \cdot \omega^t \\ &= \frac{u_i(T_i) - u'_i(T_i) \cdot T_i}{|F_i|} \cdot \frac{\omega^{\delta_{i,src(r)}}}{1 - \omega} \\ &\quad + \sum_{t=0}^{\infty} \left(\frac{u'_i(T_i) \cdot T_i \cdot \omega^{\delta_{i,src(r)}}}{|F_i|} - T_{src(r)} \cdot c \right) \cdot y_i(t + \delta_{i,src(r)}) \cdot \omega^t \end{aligned} \quad (22)$$

$$\begin{aligned} &\leq \frac{u_i(T_i) - u'_i(T_i) \cdot T_i}{|F_i|} \cdot \frac{\omega^{\delta_{i,src(r)}}}{1 - \omega} \\ &\quad + \sum_{t=0}^{\infty} \left(\frac{u'_i(T_i) \cdot T_i \cdot \omega^{\delta_{i,src(r)}}}{|F_i|} - T_{src(r)} \cdot c \right) \cdot \omega^t \end{aligned} \quad (23)$$

$$\begin{aligned} &= \frac{u_i(T_i) - u'_i(T_i) \cdot T_i}{|F_i|} \cdot \frac{\omega^{\delta_{i,src(r)}}}{1 - \omega} \\ &\quad + \left(\frac{u'_i(T_i) \cdot T_i \cdot \omega^{\delta_{i,src(r)}}}{|F_i|} - T_{src(r)} \cdot c \right) \cdot \frac{1}{1 - \omega} \\ &= \frac{u_i(T_i)}{|F_i|} \cdot \frac{\omega^{\delta_{i,src(r)}}}{1 - \omega} - \frac{T_{src(r)} \cdot c}{1 - \omega}, \end{aligned} \quad (24)$$

where, in the transition from (22) to (23), we used *Condition 2* and the fact that $y_i(t + \delta_{i,src(r)}) \leq 1$. By using (21) and (24) in (20), we get the following:

$$\begin{aligned} \bar{\pi}_i &\leq \sum_{r \in F_i} \left[\frac{u_i(T_i)}{|F_i|} \cdot \frac{1 - \omega^{\delta_{i,src(r)}}}{1 - \omega} + \frac{u_i(T_i)}{|F_i|} \cdot \frac{\omega^{\delta_{i,src(r)}}}{1 - \omega} - \frac{T_{src(r)} \cdot c}{1 - \omega} \right] \\ &= \frac{1}{1 - \omega} \cdot \sum_{r \in F_i} \left[\frac{u_i(T_i)}{|F_i|} - T_{src(r)} \cdot c \right] \\ &= \frac{1}{1 - \omega} \cdot \left(u_i(T_i) - c \cdot \sum_{r \in F_i} T_{src(r)} \right). \end{aligned} \quad (25)$$

Now, let us consider what payoff is achieved by node i if it fully cooperates in every time slot. In this case, since all the other nodes play TFT, every node will always fully cooperate, and hence, every node will experience a normalized throughput equal to 1 in each time slot. This can easily be derived from the i/o formulae describing the behavior of the nodes, which take a simple form due to the simplicity of

TABLE 2
Table of Symbols Used in the Paper

| <i>Symbol</i> | <i>Definition</i> | <i>Section where it is defined</i> |
|----------------------|---|------------------------------------|
| TFT | Tit-For-Tat strategy | Section 2 |
| $p_i(t)$ | cooperation level for node i | Section 3.2 |
| r | a given route | Section 3 |
| s | the source of a given route | Section 3 |
| l | number of forwarder nodes on a given route | Section 3 |
| f_k | the k -th forwarder on a given route | Section 3.2 |
| $T_s(r)$ | the traffic sent by source s on route r | Section 3.2 |
| $\tau(r, t)$ | throughput enjoyed by the source on route r in time slot t | Section 3.2 |
| $\hat{\tau}(r, t)$ | normalized throughput enjoyed by the source on route r in time slot t | Section 3.2 |
| $\hat{\tau}_j(r, t)$ | normalized throughput route r in time slot t leaving node j | Section 3.2 |
| $u_s(\tau(r, t))$ | a utility function of the node s | Section 3.2 |
| $\xi_s(r, t)$ | benefit enjoyed by the source on route r in time slot t (determined by $\tau(r, t)$) | Section 3.2 |
| $\eta_{f_j}(r, t)$ | forwarding cost of node f_j on route r in time slot t | Section 3.2 |
| c | cost of forwarding one unit of traffic | Section 3.2 |
| $\pi_i(t)$ | total payoff of node i in time slot t | Section 3.2 |
| $S_i(t)$ | set of routes where node i is a source | Section 3.2 |
| $F_i(t)$ | set of routes where node i is a forwarder | Section 3.2 |
| σ_i | strategy function of node i | Section 3.3 |
| Θ | automaton representing the network in the meta-model | Section 4 |
| Θ_{-i} | automaton representing the network minus node i in the meta-model | Section 4 |
| $x_i(t)$ | output of the machine of node i in time slot t | Section 4 |
| $y_i(t)$ | input of the machine of node i in time slot t | Section 4 |
| L | a dependency loop | Section 4 |
| ω | discounting factor | Section 5 |
| $\bar{\pi}_i$ | total discounted payoff of node i | Section 5 |
| $\Phi(r, i)$ | set of all forwarders on route r upstream from node i including node i | Section 5 |
| $\Phi(r)$ | set of all forwarders on route r | Section 5 |
| Φ | set of all forwarders in the network | Section 5 |
| $src(r)$ | the source of route r | Section 5 |
| $\delta_{i,src(r)}$ | the length of the dependency loop of node i with the source of route r minus one | Section 5 |
| V | function that defines how nodes aggregate the behavior of others | Section 6 |

the strategy function of the TFT strategy. As a consequence, we have that $y_i(t) = 1$ for every t , and $x_k(t) = 1$ for every k and for every t . In this case, (14) becomes:

$$\begin{aligned}\bar{\pi}_i &= \sum_{t=0}^{\infty} \left[u_i(T_i \cdot y_i(t)) - \sum_{r \in F_i} T_{src(r)} \cdot c \cdot \prod_{k \in \Phi(r,i)} x_k(t) \right] \omega^t \\ &= \frac{1}{1-\omega} \cdot \left(u_i(T_i) - c \cdot \sum_{r \in F_i} T_{src(r)} \right).\end{aligned}$$

This means that by fully cooperating, the payoff of node i reaches the upper bound expressed in (25); in other words, there is no better strategy for node i than full cooperation.

APPENDIX B

GLOSSARY

See Table 2.

ACKNOWLEDGMENTS

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322 (<http://www.terminodes.org>). The authors are thankful to John Dorsey and Paolo Santi for their comments. The authors would also like to thank the anonymous reviewers for their helpful and detailed comments that provided appropriate guidelines to improve the paper.

REFERENCES

- [1] T. Alpcan, T. Basar, R. Srikant, and E. Altman, "CDMA Uplink Power Control as a Noncooperative Game," *Wireless Networks, The J. Mobile Comm., Computation and Information*, Nov. 2002.
- [2] R. Axelrod, *The Evolution of Cooperation*. New York: Basic Books, 1984.
- [3] S. Buchegger and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation of Nodes—Fairness in Dynamic Ad-Hoc NeTworks)," *Proc. Third ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 80-91, June 2002.
- [4] L. Buttyán and J.-P. Hubaux, "Enforcing Service Availability in Mobile Ad Hoc WAnS," *Proc. First ACM/IEEE Int'l Workshop Mobile Ad Hoc Networking and Computing (MobiHoc '00)*, Aug. 2000.
- [5] L. Buttyán and J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Mobile Ad Hoc Networks*, vol. 8, no. 5, Oct. 2003.
- [6] J.R. Douceur, "The Sybil Attack," *Proc. Int'l Workshop Peer-to-Peer Systems*, Mar. 2002.
- [7] M. Félegyházi, L. Buttyán, and J.-P. Hubaux, "Equilibrium Analysis of Packet Forwarding Strategies in Wireless Ad Hoc Networks—the Static Case," *Proc. Conf. Personal Wireless Comm. (PWC '03)*, Sept. 2003.
- [8] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.
- [9] D. Goodman and N. Mandayam, "Network Assisted Power Control for Wireless Data," *Mobile Networks and Applications (MONET)*, vol. 6, pp. 409-415, 2001.
- [10] J.P. Hubaux, T. Gross, J.Y. Le Boudec, and M. Vetterli, "Towards Self-Organized Mobile Ad Hoc Networks: The Terminodes Project," *IEEE Comm. Magazine*, Jan. 2001.
- [11] Y. Jin and G. Kesidis, "Nash Equilibria of a Generic Networking Game with Applications to Circuit-Switched Networks," *Proc. IEEE INFOCOM '03 Conf.*, Mar.-Apr. 2003.

- [12] Y. Korilis, A. Lazar, and A. Orda, "Architecting Noncooperative Networks," *IEEE J. Selected Areas in Comm.*, vol. 13, no. 8, 1995.
- [13] Y. Korilis and A. Orda, "Incentive Compatible Pricing Strategies for QoS Routing," *Proc. IEEE INFOCOM '99 Conf.*, Mar. 1999.
- [14] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking (Mobicom '00)*, pp. 255-265, 2000.
- [15] P. Michiardi and R. Molva, "Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," *Proc. Conf. Comm. and Multimedia Security 2002*, Sept. 2002.
- [16] J. Nash, "Equilibrium Points in N -Person Games," *Proc. Nat'l Academy of Sciences*, vol. 36, pp. 48-49, 1950.
- [17] M.J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1994.
- [18] Y. Qiu and P. Marbach, "Bandwidth Allocation in Wireless Ad Hoc Networks: A Price-Based Approach," *Proc. IEEE INFOCOM '03 Conf.*, Mar.-Apr. 2003.
- [19] A. Rapaport and A.M. Chammah, *The Prisoner's Dilemma*. Ann Arbor, Mich.: Univ. of Michigan Press, 1965.
- [20] T. Roughgarden, "Selfish Routing," PhD thesis, Cornell Univ., May 2002.
- [21] V. Srinivasan, P. Nuggehalli, C.F. Chiasserini, and R.R. Rao, "Cooperation in Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM '03 Conf.*, Mar.-Apr. 2003.
- [22] R. Trivers, "The Evolution of Reciprocal Altruism," *Quarterly Rev. of Biology*, vol. 46, pp. 35-57, 1971.
- [23] A. Urpi, M. Bonuccelli, and S. Giordano, "Modeling Cooperation in Mobile Ad Hoc Networks: A Formal Description of Selfishness," *Proc. WiOpt '03 Conf.: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Mar. 2003.
- [24] L.M. Wahl and M.A. Nowak, "The Continuous Prisoner's Dilemma: I. Linear Reactive Strategies," *J. Theoretical Biology*, vol. 200, pp. 307-321, 1999.
- [25] H. Yaïche, R.R. Mazumdar, and C. Rosenberg, "A Game Theoretical Framework for Bandwidth Allocation and Pricing in Broadband Networks," *IEEE/ACM Trans. Networking*, vol. 8, no. 5, Oct. 2000.
- [26] M. Xiao, N.B. Schroff, and E.K.P. Chong, "Utility-Based Power Control in Cellular Systems," *Proc. IEEE INFOCOM '01 Conf.*, Apr. 2001.
- [27] S. Zhong, Y.R. Yang, and J. Chen, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM '03 Conf.*, Mar.-Apr. 2003.



Márk Félegyházi received the MSc degree in electrical engineering from the Budapest University of Technology and Economics (BUTE) in 2001. From 1999 to 2001, he was with Ericsson Research in Budapest, Hungary. During his work at Ericsson, he was involved in Mobile IP protocol testing and the design of ad hoc network protocols for the Bluetooth short-range wireless technology. Currently, he is a research and teaching assistant at EPFL, Switzerland. His current research interest covers cooperation issues in ad hoc and sensor networks. He is a student member of the IEEE and a member of the ACM and he has served as a reviewer for various conferences and journals including ACM MobiHoc, ACM Mobicom, IEEE Infocom, and the *IEEE Transactions on Mobile Computing*. For more information, please check <http://icawwww.epfl.ch/felegyhazi/>.



Jean-Pierre Hubaux joined the faculty of EPFL in 1990; he was promoted to full professor in 1996. His research activity is focused on mobile networking and computing, with a special interest in fully self-organized wireless ad hoc networks. In particular, he has performed research on cooperation aspects, security, power efficiency, and distributed algorithms for ad hoc and sensor networks. During the last few years, he has been strongly involved in the definition

and launching phases of a new National Competence Center in Research named "Mobile Information and Communication Systems" (NCCR/MICS), see <http://www.terminodes.org>. Within his first year at EPFL, he defined the first curriculum in communication systems. From October 1999 until September 2001, he was the first chairman of the Communication Systems Department. He served as the general chair for the Third ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002), held on the EPFL campus. He is an associate editor of the *IEEE Transactions on Mobile Computing* and of the *Elsevier Journal on Ad Hoc Networks*. He has held visiting positions at the IBM T.J. Watson Research Center and at the University of California at Berkeley. In a former life, he spent 10 years in France with Alcatel, where he was involved in R&D activities, mostly in the area of switching systems architecture and software. He is a senior member of the IEEE. For more information, please check <http://lcawww.epfl.ch/hubaux>.



Levente Buttyán received the MSc degree in computer science from the Budapest University of Technology and Economics (BUTE) in 1995, and the PhD degree from EPFL, Switzerland, in 2002. In 2003, he joined the Department of Telecommunications at BUTE, where he currently holds a position as an assistant professor and works in the Laboratory of Cryptography and Systems Security (CrySyS). His research interests are in the design and analysis of security protocols for wired and wireless networks, including wireless sensor networks and ad hoc networks. More information is available at <http://www.hit.bme.hu/~buttyan/>.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**