Greening Ad Hoc Networks through Detection and Isolation of Defecting Nodes

Maurizio D'Arienzo
Dipartimento di Studi Europei e Mediterranei
Seconda Università di Napoli
maudarie@unina.it

Francesco Oliviero, Simon Pietro Romano
Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli "Federico II"
folivier@unina.it, spromano@unina.it

***Abstract*—Current ad hoc networks rely on a silent mutual agreement among nodes to relay packets towards the destinations. The effort made by each single node to serve the others is usually repaid with the chance to successfully set up its own traffic sessions. However, limited power, together with security concerns, can push certain nodes to refrain from cooperating. Such nodes will thus act as parasites, while the others will unawarely keep on trusting them for what concerns the mutual service agreement. In this paper we show how energy consumption in Ad Hoc Networks can be dramatically reduced if we stimulate cooperation by providing mechanisms for the detection and isolation of selfish nodes. We present a novel routing protocol exploiting a behavior-tracking algorithm based on game theory and allowing traffic to be forwarded only towards cooperative nodes. Through extensive simulations, we show how we can significantly reduce power wastage at the same time maximizing the delivery rate. Under this perspective, cooperation can definitely be seen as an incentive for all nodes, since it allows to optimize one of the most crucial parameters impacting the performance of ad hoc networks.**

***Index Terms*—Energy Efficiency, Fairness, Game Theory, Cooperation, Ad Hoc Routing Protocols.**

I. Introduction

Ad hoc networks are composed of several nodes with wireless connection capability. Differently from wired networks, in an ad hoc environment each node is an end system and a router at the same time. A transmission between a sender and a receiver happens with the help of one or more intermediate nodes that are requested to relay packets according to routing protocols designed for this kind of networks. A blind trust agreement among nodes makes it possible the right message forwarding. However, wireless nodes have often limited power resources, and some of them are asked to relay packets more frequently than they do with respect to their own relay requests. Thus, a good percentage of power is wasted to serve other nodes. Besides, the open nature of the current ad hoc network protocols raises some security concerns. In fact, although in the recent past there have been proposals of protocol modifications to enhance security, at present the aggregation of new nodes is usually uncontrolled and open to potential malicious users. In such a situation, a generic node of the network has to decide whether to trust or not to trust the other nodes. This obviously calls for a capability of each single node to somehow interpret (or, even better, predict) the behavior of the other nodes, since they represent fundamental *allies* in the data transmission process [1].

The situations in which a decision of a part depends on the predicted behavior of another part have been elegantly studied in game theory. Game theory has been already applied [2] [3] [4] to ad hoc networks with interesting results. The basic assumption is that all the players follow a rational behavior and try to maximize their payoff. The simplest games see the involvement of only two players who have to decide whether to cooperate or defect with the others. The best solution may not maximize the payoff, but can reach an *equilibrium* as proposed by Nash. One of the versions of this game is known as *prisoner's dilemma* and has an equilibrium in case both users decide to defect. This is true for the game played only one time while in its iterated version the situation is more complex and even cooperation can be convenient. In case of ad an hoc network, the player is a node that needs to cooperate with the others to send its traffic. However some nodes can decide to defect for a number of unspecified reasons and, as a first need, the other nodes should be informed of their behavior in order to react in the most appropriate way.

In this paper, we show how cooperation can be perceived by nodes as an incentive, thanks to the fact that it helps save the overall amount of energy needed for data transmissions. Differently from recent works proposed in the literature [5] [6], which aim at making the routing process become *natively* aware of the energy-related parameters, we herein propose a different approach, by leveraging cooperation in order to improve the overall energy efficiency of an ad hoc network without modifying the existing routing protocol. Our work is indeed complementary to the above mentioned proposals, in that it can co-exist with any routing protocol, be it legacy or energy-aware. We try and exploit a different perspective on energy efficiency, which is much more related to the behavioral patterns of the nodes rather than to the specific mechanisms and protocols adopted in the network.

Delving into some of the details of how we deal with the behavioral aspects of the problem at hand, we present in the paper an algorithm to identify and isolate defecting nodes. The algorithm takes inspiration from the results of game theory and keeps a local trace of the behavior of the other nodes. At the beginning the behavior of all the other nodes is unknown, but as soon as the first flows of traffic are exchanged among them, each node becomes gradually aware of the past behavior of the others, which can be either cooperative or defecting. Once the defecting nodes are identified, different countermeasures can be adopted. The current version of the algorithm makes the decision of not relaying packets coming from defecting nodes as long as they do not cooperate, but other, less disruptive policies can be considered and included. The algorithm is implemented in an existing ad hoc routing protocol and is validated in the ns-2 simulator. The current experimental results highlight the induced reduction of throughput of defecting nodes.

The paper is organized in six Sections. Section II deals with both background information and related work. Section III presents the algorithm we designed to infer behavioral information about the network nodes, whose implementation is described in Section IV. Results of the experimental simulations we carried out are presented in Section V, while Section VI provides concluding remarks and proposes some directions of future work.

II. Background and Related Work

In this section we try to shade light on the context of our contribution, by properly defining the scope of our research, as well as its application to the wide set of *green networking* proposals that have recently come to the fore in the international research community. We start by proposing a bird's eye view on the most recent works that have focused on energy-aware routing in ad hoc networks. Then, we move the focus to the most important aspect of our contribution, namely cooperation. Indeed, as we already pointed out, cooperation is a fundamental subject of our recent research and is herein studied under one of its most challenging facets, i.e. its use as an incentive for all the nodes of the network, thanks to the significant performance improvements that it entails in terms of energy savings associated with data transmissions.

A. Energy-aware routing in ad hoc networks

Routing table computation performed by a routing protocol based on energy measures can improve efficiency in ad hoc networks. In this regard, a great amount of energy-aware routing protocols for ad hoc networks have been proposed in the last years ([7], [8], [9], [10], [11]). They can be roughly classified based on their specific goals. They can in fact try to: (i) minimize the total power needed to transmit packets; (ii) maximize the lifetime of every single node; (iii) minimize the total power needed to transmit packets at the same time maximizing the lifetime of every single node. Some interesting energy-efficient route selection schemes, falling in one of the previous categories, are presented in [7] and briefly described in the following.

Minimum Total Transmission Power Routing (MTPR) is a routing protocol aimed at minimizing overall power consumption in ad hoc networks. Given a source s and a destination d, we denote with P_r the total transmission power for a generic route r from s to d. P_r is the sum of the power consumed for the transmission between each pair of adjacent nodes belonging to r. MTPR selects the route r^* such that $r^* = min_{r \in R} P_r$, where R is the set containing all possible routes from s to d. A simple shortest path algorithm can be used to find this route.

Minimum Battery Cost Routing (MBCR) associates each node n_i in the network with a weight $f_i(c_i(t)) = 1/c_i(t)$, where $c_i(t)$ is the battery capacity level of n_i at time t. Given a source s and a destination d, if we say E_r the sum of the nodes weights of a generic route r from s to d, MBCR selects the route r^* such that $r^* = min_{r \in R} E_r$, where R is the set containing all possible routes from s to d. Such a scheme will always choose routes with maximum total residual energy.

With *Min-Max Battery Cost Routing* (MMBCR), starting from the above definition of $f_i(c_i(t))$, for each route r from a source s to a destination d, a cost is defined as $C_r(t) = max_{i \in r} f_i(c_i(t))$. The chosen route r^* verifies the relation $C_{r*}(t) = min_{r \in R} C_r(t)$. MMBCR safeguards nodes with low energy level because it selects the route in which the node with minimum energy has more energy, compared to the nodes with minimum energies of the other routes.

Conditional Max-Min Battery Capacity Routing (CMMBCR) proposes an approach based on both MTPR and MMBCR. Let us consider the node of a generic route r from a source s to a destination d, with lowest energy. Let also $m_r(t)$ be its energy, and R the set of all the routes from s to d. If some paths with $m_r(t)$ over a specific threshold exist in R, one of these will be chosen using the MTPR scheme. Otherwise, the route r^* satisfying the relation $m_{r*}(t) = max_{r \in R} m_r(t)$ will be selected. This scheme suffers from an unfair increment of the forwarding traffic towards nodes with more energy [10].

Minimum Drain Rate (MDR [9]) proposes a mechanism that takes into account node energy dissipation rate, thus avoiding the above problem. MDR defines for each node n_i a weight $C_i = RBP_i/DR_i$, where RBP_i is the residual battery power and DR_i the drain rate of n_i. Intuitively, DR_i represents the consumed energy per second in a specified time interval. Now, let C_r be the minimum weight of a generic route r from a source s to a destination d. MDR selects the route r^* such that $C_{r*} = max_{r \in R} C_r$. In this way, residual energy level, as well as the energy consumption rate due to the incoming traffic to be forwarded, are jointly taken into account.

As we already stated, in this paper we do not embrace an approach aimed at modifying routing in order to let it become energy-aware. We rather propose to induce network nodes to cooperate, by demonstrating that a cooperative behavior turns out to have a significant effect on performance, in terms of reduction of the energy needed for data transmissions. In the next subsection we then focus on the behavioral aspects related to cooperation of the nodes of an ad hoc network.

B. Cooperation in ad hoc networks

Cooperation of nodes involved in an ad hoc network is usually induced because the efforts related to the offered services are compensated with the possibility to request a service from the other nodes. However current ad hoc network protocols do not provide users with guarantees about the correct behavior of other nodes that can eventually decide to act as parasites. Several works have identified the problem of stimulating cooperation and motivating nodes towards a common benefit. The main solutions rely on a virtual currency or on a reputation system, and more recently on game theory.

Virtual currency systems [12] [13] give well behaving users a reward every time they regularly relay a packet. They can then reuse the reward for their transmissions as long as they have a credit. The first issue of such systems is related to the need of a centralized server to store all the transactions among the users. Besides, the system is not completely fair with all nodes. Nodes placed at the boundary of the area are usually less involved in relay operations and then excluded from rewards, even if they are ready to be involved. Also, the messages regarding the transactions need to be secured in order to avoid spoofing of malicious nodes.

Reputation systems repeatedly monitor and build a map of trustworthy nodes on the basis of their behavior [4] [5] [6] [14] [15]. These systems distinguish between the *reputation*, which rates how well a node behaved, and *trust*, which represents how honest a node is. Most of these systems consider the reputation value as a metric of trust . A node is refrained to relay a packet coming from untrusted nodes, which are then excluded from the network operations. Several issues are related to the use of these systems. First, each node needs to maintain a global view of the reputation values with considerable caching. Some proposals keep local information, others disseminate reputations to other nodes, with an increased overhead due to the exchange of such messages. Reputation values can be modified, forged or lost during operations, and they can differ from node to node, which can bring to inconsistency, i.e. node X considers node Y trustworthy while node Z considers the same node untrustworthy.

To overcome some of these issues, it has been proposed to model the nodes taking part to an ad hoc network with game theory. There are different models studied in game theory and some of them have been already applied to ad hoc networks. In next subsections we review the most important models of game theory and some applications to ad hoc networks.

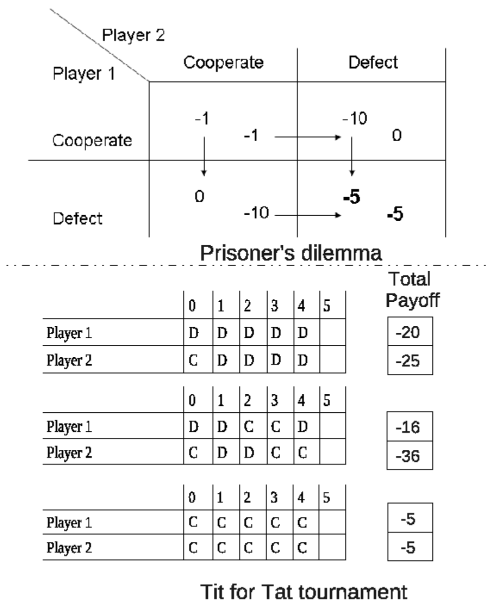
C. *Game theory Basics*

Game theory is a branch of applied mathematics that witnessed a great success thanks to the application of its results to a wide selection of fields, including social sciences, biology, engineering and economics. Game theory covers different situations of conflicts regarding, in a first attempt, two agents (or *players*), and in the generalized version, a population of players. Each of these players expects to receive a reward, usually named *payoff*, at the end of the game. The basic assumption is that all the players are self interested and rational: given a utility function with the complete vector of payoffs associated with all possible combinations, a rational player is always able to place these values in order of preference even in case they are not numerically comparable (e.g. an amount of money and an air ticket). This not necessarily means that the best value will be selected, since the final reward of each player is strongly dependent on the decision of the other players. Each player is then pushed to plan a *strategy*, that is a set of actions aiming at a total *payoff* maximization, provided that he is aware that the other players will try to do the same.

Games are now classified in several categories according to various properties. If the players tend to be selfish in the achievement of the best payoff, the game is classified as non cooperative rather than cooperative. When there is a common knowledge of the utility function for all players, the game is with complete information, otherwise it is considered incomplete. There are several real world examples that fall in one of these categories. Here we are mainly interested in the difference between strategic and extensive games. In strategic (also known as *static*) games the players make their decision simultaneously, without any knowledge of the others' intention. Even if the game is repeated, the players are still unaware of others' plans and do not have the chance to react to a previous action. This last opportunity is instead available in case of extensive games. Such games are played more than once and the players can evaluate what the others did at least during the last tournament, so that they can potentially decide to modify their strategy for the next move. Also, the payoff is cumulated at the end of each round rather than accounted for only once.

One of the fundamental problems of game theory is known as *prisoner's dilemma*, which can be represented in the matrix format of Fig. 1: two suspects of a crime are arrested and jailed in different cells with no chance to communicate between each other. They are questioned by the police and receive the same deal: if one confesses (*defect*) and the other stays silent (*cooperate*), the first is released, the second is convicted and goes to prison with a sentence of 10 years, the worst; if both stay silent (*cooperate*), they go to prison for only 1 year; if both testify against the other (*defect*) they go to prison with a sentence of 5 years. The situation in which they both stay silent (*cooperate*) is the more convenient to both of them; however, it was demonstrated that a rational behavior is to confess (*defect*) and receive the sentence of 5 years, and this situation represents the only equilibrium, as first introduced by Nash [16] [17]. Hence, the prisoner's dilemma falls in the field of strategic non-cooperative games.

In its basic form the prisoner's dilemma is played only once and has been applied to many real life situations of conflict, even comprising thorny issues of state diplomacy. Another version of the prisoner's dilemma is played repeatedly rather than a single time and is known as iterated prisoner's dilemma (ITD), which turned out to be a cooperative game under certain circumstances [18][19]. The goal of both players still is the maximization of their payoff, as the cumulated payoff earned at each stage. If the number of rounds is finite and known in advance, the strategy of always defecting is still the only situation of equilibrium and the game is still non-cooperative. However, in case the number of repetitions is infinite, it was demonstrated that the choice to always defect is not the only equilibrium as even the choice of cooperating may be an equilibrium. In this case, one of the strategies that let players maximize their payoff is the so-called *Tit for Tat* game, in which each player repeats the past behavior of the other player: a player is keen to cooperate if the other node behaved correctly the last time, otherwise it defects. If we



	0	1	2	3	4	5	Total Payoff
Player 1	D	D	D	D	D		-20
Player 2	C	D	D	D	D		-25

	0	1	2	3	4	5	
Player 1	D	D	C	C	D		-16
Player 2	C	D	D	C	C		-36

	0	1	2	3	4	5	
Player 1	C	C	C	C	C		-5
Player 2	C	C	C	C	C		-5

Fig. 1: The tit-for-tat strategy in action

consider the first five tournaments of a two players game, a player who defects (D) against a cooperative (C) player adopting the tit for tat strategy would play (D,D,D,D,D) and earn $(0,-5,-5,-5,-5) = -20$. If the first player decides to cooperate two times out of five (D,D,C,C,D), he would earn $(0,-5,-10,-1,0) = -16$. In case he always cooperates, his payoff would be $(-1,-1,-1,-1,-1) = -5$, which is the best he can achieve. So, continued cooperation for the iterated prisoner's dilemma also yields the best payoff. Despite this benefit, the main result of the tit for tat strategy is that it stimulates the cooperation. We base our algorithm to mitigate the node selfishness on the results of this version of the game.

D. *Game Theory applied to Ad Hoc Networks*

One of the first proofs of the improvements produced by cooperation in ad hoc networks is presented in [2]. The authors provide a mathematical framework for studying the effects of cooperation in ad hoc networks. They first introduce a normalized acceptance rate (NAR) as the ratio between the successful relays provided to the others and the relay requests made by the node. Then they propose two models, namely GTFT (Generous Tit for Tat) and m-GTFT for the case of multiple players, to give the (rational) nodes the chance to make a decision concerning the possibility to cooperate or defect with other nodes, and they analytically demonstrate that these models represent a Nash equilibrium. In such a situation, a node does not improve its NAR to the detriment of the others. Also, at the opposite of reputation schemes, each node can maintain per session rather than per packet information, thus leading to a scalable solution.

In [20] the authors prove the selfishness property of the nodes in a MANET by using the Nash equilibrium theorem [16]. They define a generic model for node behavior that takes into account also energy consumption due to the transmission process. By adopting a punishment based technique they prove that it is possible to escape from the theoretically unique equilibrium point of non-cooperation and to enforce a cooperation strategy under specific conditions.

In [21] the authors also fucus on forwarding mechanisms. They provide a model for node behavior based on game theory in order to determine under which conditions cooperation with no incentives exists. They prove that network topology and communication patterns might significantly help enforce cooperation among nodes.

Game theory has been also used to improve routing algorithms in wireless networks. An actual implementation of a game theory model in the AODV routing protocol with two distinct approaches has been proposed in [3]. The first plays a deterministic tit for tat game and the second a randomized version of the same game deployed with a genetic algorithm. In both cases, they achieve better performance in terms of experienced delay and packet delivery ratio in case of cooperation of nodes. The models are tested in a simulated environment and rely on static distribution of nodes' behavior profiles while not supporting a mechanism for a dynamic adaptation to changed situations.

III. ALGORITHM DESCRIPTION

In an ad hoc network, the number of nodes and links can change during time, so we consider the number of nodes $N(t)$ as a function of time t. We also define a dynamic array $C(t)$ of $N(t)$ elements for each node of the network. The generic element $c_i(t)$ of $C(t)$ assumes the values (UNKNOWN, COOPERATE, DEFECT) meaning that the behavior of node i at time t is respectively unknown, cooperative or non cooperative. At time $t = 0$ all the values are set to UNKNOWN, since at the beginning each node is not aware of the behavior of the other nodes.

Suppose the generic node s of the network needs to send some traffic to the destination d. The first task is to discover an available path, if it exists, to reach the destination. To this purpose, we consider a source based routing protocol capable of discovering a list $A(t)_{(s,d)i}$ $\forall i : 0 < i < P$ of P multiple paths. All the nodes in the list $A(t)_{(s,d)i}$ are considered under observation and marked as probably defecting in the array $C(t)$ unless a positive feedback is received before a timeout expires. The sender s starts sending his traffic along all the discovered paths. If the destination node generates D acknowledgement messages containing the list of all the nodes $L_{(s,d)i}$ $0 < i < D$ traversed, as it happens in some source based routing protocols, the sender s is informed about the behavior of intermediate nodes. For each acknowledgement message received, the sender s can make a final update of the array $C(t)$ by setting the matching elements $c_i(t)$ contained in the list $L_{(s,d)i}$ as cooperative. Notice that the last update overwrites the previous stored values and represents the most recent information concerning the behavior of a node. An example of the evolution of the described algorithm is presented in Fig. 2.

Given this algorithm, each node is aware of the behavior of other nodes and can react in the most appropriate way. For

$A(t)|_{A,F}=\{A,C,E,F\}$
$A(t)|_{A,F}=\{A,B,C,E,F\}$
$A(t)|_{A,F}=\{A,C,E,F\}$
$A(t)|_{A,F}=\{A,B,C,D,F\}$

Node	A	B	C	D	E	F
expected	-	D	D	D	D	-
final	-	U	U	U	U	-

Timeout expired

Node	A	B	C	D	E	F
expected	-	D	D	D	D	-
final	-	C	C	C	D	-

U	-	Unknown
C	-	Cooperative
D	-	Defecting

Fig. 2: Algorithm description

example, a node can refuse to relay packets of defecting nodes, or operate a selective operation like queuing their packets and serving them only if idle and not busy with the service requested by cooperative nodes. In this first proposal, we rely on the harsh policy of packet discarding, and this brings to the isolation of defecting nodes. However, a defecting node can even gain trust of other nodes if it starts to cooperate. The array $C(t)$ is not static over time and its values are continuously updated. In fact, due to the dynamic situation of ad hoc networks, the search of available paths is frequently repeated, and the list $A_{(s,d)}$ consequently updated. Hence, if a defecting node decides to cooperate, its identification address will be included in one of the acknowledgement messages $L_{(s,d)i}$ sent to the sender s and its aim to cooperate will be stored in the array $C(t)$.

The situation described here for the pair (s, d) is replicated for all the possible pairs of nodes that try to interact, but each node stores only one array $C(t)$ that is updated upon reception of any acknowledgement message, wherever it comes from. Furthermore, not all the packets relayed are checked in order to verify the nodes' behaviors, but only a sample of them, thus keeping the total overhead under control.

IV. An ad hoc network routing protocol for discovery of defecting nodes

The algorithm introduced in the previous section has been implemented in an existing source based routing protocol for ad hoc networks. We first modified this protocol to support the search of multiple paths, and then included the new algorithm for the identification of non cooperative nodes. In the next subsections we present the existing protocol, with respect to both its basic and newly added features.

A. Multipath source based routing in ad hoc networks

The dynamic configuration of an ad hoc network topology makes the routing protocols used in wired networks unsuitable for this kind of networks. Hence, several new protocols have been designed and made available to manage this collection of wireless nodes. To the purpose of identifying defecting nodes, an acknowledgment, or missed acknowledgement technique is needed. Among the many ad hoc routing protocols, AH-CPN (Ad Hoc Cognitive Packet Network) [22] is designed to support QoS and make an intense use of acknowledgement messages independently from the transport protocol in use. AH-CPN is the wireless version of CPN (Cognitive Packet Network) [23], a proposal for a self aware network architecture

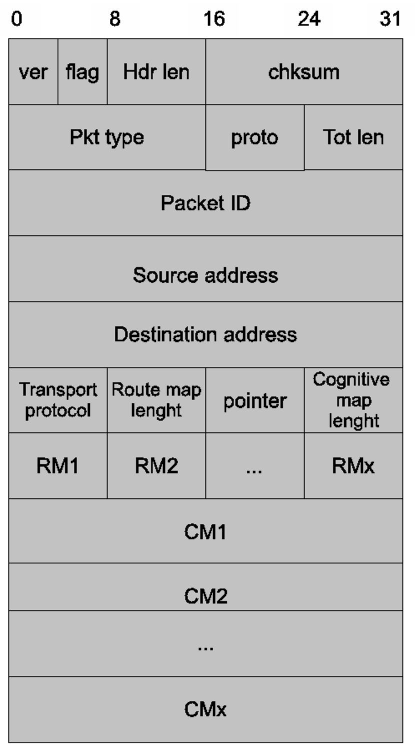


Fig. 3: The CPN header

with native support for QoS. Both in AH-CPN and CPN, the presence of a neural network engine enables to undertake dynamic and fast routing decisions as soon as a condition, like for example a congested link or a different user's requirement, has changed. An always active traffic of smart packets discovers new paths according to specific QoS goals, e.g. discovering paths that minimize the delay or maximize the throughput. This information is made available to the interested nodes that can send traffic along the defined path on a source based routing basis. The smart traffic keeps on looking for the specific goals, and in case a better path is found, the sender is informed and can update its routing path.

There are four different kinds of packets in AH-CPN, all sharing the same header (depicted in Fig. 3): Smart Packets (SP), Smart Acknowledgements (SA), Dumb Packets (DP), and Dumb Acknowledgements (DA).

Smart packets are those described at the beginning of this section. They are lightweight packets containing a QoS goal sent by a sender to a destination. These packets are routed with the Random Neural Network (RNN) [24] algorithm that runs on each node and which selects the next hop by taking into account the past behavior of the link. Every time a SP traverses a node the route map (RM) field is updated with the node's address. Once at the destination, a SA is generated and sent backwards along the RM received in the SP. Finally, the actual data can be sent across the network in a DP, which is prepared with the whole path copied in the RM field. Internal nodes relay DPs to the next hop excerpted from the RM field, and they add timestamp information useful to evaluate the round trip time (RTT) between each pair of nodes along the path. These RTT data are stored in special *mailboxes* present in each node and provide the RNN algorithm with precious information concerning the past behavior of a link. Once the DP reaches its destination, a DA is sent along the reverse path. Notice that differently from IP networks, in CPN the acknowledgements are generated upon reception of each single packet, whatever the transport protocol is. This feature is helpful in the deployment of our algorithm to identify defecting nodes, as we will soon explain.

The basic CPN version looks for one available path, the best in terms of the requested QoS goal. We modified this protocol to search for multiple paths. To this purpose, SPs are initially sent via flooding to collect a certain number of available paths (up to a well defined threshold, which can be properly configured at setup time). To prevent loops, SPs are marked with an identification number ID, and those with the same ID touching a node for the second time are discarded. SPs reaching the same destinations with different contents for what concerns the routing map RM are considered valid, and SAs are sent backward to inform the sender. The sender collects the different SAs and updates its routing table. DPs are sent on a round robin basis. Once the available paths are discovered, the transmission of SPs is not terminated; it is rather repeated periodically for path maintenance, to check if the topology has changed, and in our case also to verify if there is a different configuration concerning the behavior of nodes.

B. Identification and isolation of defecting nodes

We provide the multipath source based routing protocol with the support for identification and isolation of defecting nodes. The array $C(t)$ is computed and stored at each node. Its dimension can change according to the number of nodes active in the ad hoc area. When node a needs to send traffic to node b, SPs are immediately sent in flooding. We make the assumption that non cooperative nodes try to cheat by forwarding inexpensive SPs, that do not carry any payload, while they do not relay DPs containing the real data. In case the non cooperative nodes decide to block the SPs forwarding, they are immediately discovered as non cooperative and have no chance to cheat. In this scenario, every time a SP traverses a node, its cognitive map is extended with the label of the visited node. Once at the destination, the complete cognitive map is copied into the DA and sent back to the sender along the reverse path. Obviously, this is repeated for all the discovered paths, so at the end of this process node a has a complete knowledge of all the available paths, including those comprising cheating nodes, and these are all stored in $A(t)_{(a,b)}$. At the time of the first transmission, the real data are packed in multiple DPs and sent along all the available paths on a round robin basis, but the interested cheating nodes will not relay them. Since in CPN a destination b must send an acknowledgement message DA whatever the transport protocol is, node a will receive only the DAs containing the successful paths, i.e. those without cheating nodes. This information, as described before, helps finalize the array $C(t)$ with the list of cooperative and defecting nodes, and the traffic is sent only along the path or the paths composed of cooperative nodes rather than towards all the available paths. When one of the cheating nodes requests the relaying of a message to node a, it is aware of his past behavior and can decide to drop all its packets, while it can regularly relay packets coming from cooperative users.

The situation concerning the cooperation and the selection of paths is not static and can change during time, so isolated nodes are not banned forever from the network. Although the traffic from a node is delivered only along paths composed of cooperative nodes, sending nodes keep on checking periodically the paths containing the defecting nodes. Should a defecting node decide to change its behavior and begin to cooperate, the routing protocol soon detects this change and admits again the node to the transmission of flows. This way, a node reacts following a *Tit for Tat* strategy.

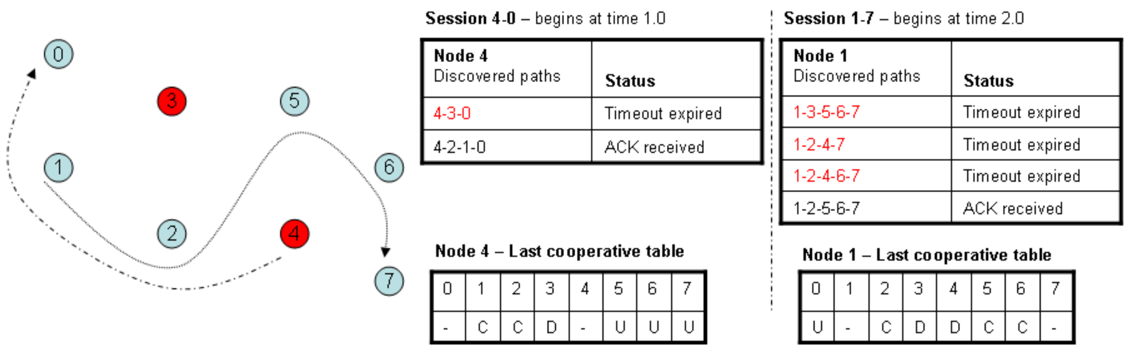
V. Experimental results

In a wireless scenario, normal operation can often lead to a high level of iniquity. The introduction of a system able to detect defecting nodes can instead increase the fairness of a wireless ad hoc network in terms of both delivery ratio and energy consumption. To show these two aspects, we repeatedly ran two type of experiments in the ns-2 simulator.

In the first series of experiments we designed a scenario associated with several working conditions on a simple wireless testbed composed of 8 nodes (see Fig.4), labeled from 0 to 7. In such network we set up the following conditions: (i) node 3 defects all the time; (ii) the behavior of node 4 dynamically changes over time; (iii) all the other nodes are cooperative. The duration of the experiments is set to 10 minutes. The defection of a node means that the relay of traffic to serve other nodes is totally stopped, so the percentage of node 3's cooperation is always 0% (of the total time). As far as node 4 is concerned, five situations are considered, most of them offering the other nodes the chance to reply with a *tit for tat* strategy:

1) Node 4 never cooperates. Requests of relay are never forwarded, so the percentage of cooperation is 0%;
2) Node 4 follows a switching behavior: assuming that the time is divided in 4 equal slots of 150 seconds each, node 4 cooperates during the first 75 seconds of the second and fourth slot interval, then it defects all the time; the total percentage of cooperation is hence 25%;
3) Node 4 still switches its behavior: it defects during two slots and cooperates in the other two; in this case the total percentage of cooperation is thus 50%;
4) Node 4 switches its behavior in a way that is opposite to the one described in the second item of this list: node 4 defects during the first 75 seconds of the first and third slot interval, then it cooperates all the time; the total percentage of cooperation is hence 75%;
5) Node 4 always cooperates; all relay requests are served, for a final percentage of cooperation of 100%.

Two equal sessions of constant bit rate traffic are activated between node 4 and node 0 and node 1 and node 7, respectively at time 1.0 and at time 2.0. In the ideal situation of all



Session 4-0 – begins at time 1.0

Node 4 Discovered paths	**Status**
4-3-0	Timeout expired
4-2-1-0	ACK received

Node 4 – Last cooperative table

0	1	2	3	4	5	6	7
-	C	C	D	-	U	U	U

Session 1-7 – begins at time 2.0

Node 1 Discovered paths	**Status**
1-3-5-6-7	Timeout expired
1-2-4-7	Timeout expired
1-2-4-6-7	Timeout expired
1-2-5-6-7	ACK received

Node 1 – Last cooperative table

0	1	2	3	4	5	6	7
U	-	C	D	D	C	C	-

Fig. 4: The simulated testbed

cooperating nodes, the shortest paths would be $(4, 3, 0)$ and $(1, 2, 4, 7)$. However, node 3 is always defecting, so the path $(4, 3, 0)$ turns out to be unavailable and the traffic coming from node 4 is forced along the other available path $(4, 2, 1, 0)$. As long as node 1 does not generate traffic, it does not have the chance to track the behavior of node 4, so the relay requests coming from node 4 are regularly served. At time 2.0 node 1 begins the discovery of paths to reach node 7. Besides the other choices, the best path $(1, 2, 4, 7)$ is soon discovered and selected to immediately generate traffic. If node 4 follows a switching behavior, then node 1 has the chance to react in compliance with the *tit for tat* strategy. Notice that in case node 4 is in a defecting state, node 1 can still send traffic to the destination along the path $(1, 2, 5, 6, 7)$.

In Fig. 5 we report the delivery ratio of node i as the ratio $G_i(t) = r_i(t)/s_i(t)$ at the end of the experiment ($t = 10min$) between the number of bytes correctly received at destination $r_i(t)$ and the total number of bytes sent $s_i(t)$. The x axis represents the percentage of node 4's cooperation, the y axis is the final delivery ratio $dr_i(t)$. Initially (left part of the x axis in Fig. 5), node 4 is fully defecting; the same applies to node 3. Traffic from node 4 towards node 0 is regularly sent between time 1.0 and time 2.0 because node 1 did not generate any request and did not yet test the behavior of the other nodes. At time 2.0, however, node 1 tries to send traffic to node 7 and hence has the chance to verify the behavior of the other nodes. Among the other discovered paths, it realizes that paths comprising nodes 4 and 3 are not working, so as soon as the timeout expires it marks nodes 3 and 4 as defecting and immediately stops relaying traffic coming from node 4. The final delivery ratio dr_1 of node 1 is closer to the ideal value because the alternative path $(1, 2, 5, 6, 7)$ is soon discovered and used for the entire duration of the experiment. The delivery ratio dr_4 is instead severely reduced.

As node 4's percentage of cooperation increases up to 100%, the delivery ratio dr_4 also increases until it reaches a value close to dr_1 when there is full cooperation. Although node 3's defection makes the path $(4, 3, 0)$ unavailable, the routing protocol discovers the alternative path $(4, 2, 1, 0)$ composed of cooperative nodes, while the shortest $(1, 2, 4, 7)$ is regularly available in this case. This is the only situation in which node 4 maximizes its goodput. In the intermediate cases the trend is linear and clearly demonstrates the correct

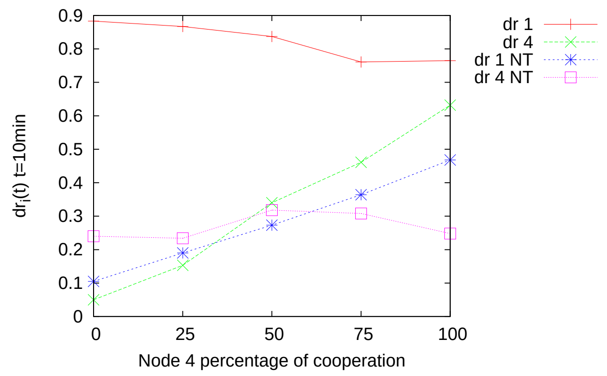


Fig. 5: Delivery ratio of Node 1 and Node 4

implementation of the *tit for tat* reaction mechanism, as node 1 cooperates only when node 4 does the same. Delivery ratio dr_1 remains more or less unaltered independently of node 4's behavior, thanks to the fact that node 1 has a chance to discover alternative cooperative paths.

We compared these results with the situation in which the nodes are unable to detect the defecting behavior. We mark these sessions with *NT* in the same Fig.5. The situation is now opposite to the previously analyzed case because delivery rate dr_4 outperforms dr_1 in the case of node 4's full defection. Node 1 is now unaware of node 4's defection; hence, while its traffic is not relayed, it regularly relays the incoming packets having node 4 as source. Anyway, both delivery ratios (dr_1 and dr_4) are lower than in the previous case. This time the lack of tracing of nodes defection affects even node 4's performance, because such node tries to forward traffic not only along the path $(4, 2, 1, 0)$ but also along the uncooperative path $(4, 3, 0)$, which explains the halved final delivery ratio.

We then evaluated the effective energy spent by node 1 and node 4 to successfully deliver their packets to the respective destinations. This energy is calculated as $E_{eff_i} = Ec_i * dr_i$, being Ec_i the energy consumed by node i and dr_i the delivery ratio computed as described above. Fig.6 illustrates the average effective energy consumed by node 1 and 4 in both situations of detection (active and inactive), as well as in all the aforementioned conditions of cooperation. Notice how the trace corresponding to the detection enabled is always lower compared to the case when detection is disabled. Besides, both traces decrement as the cooperation increases, and reach their

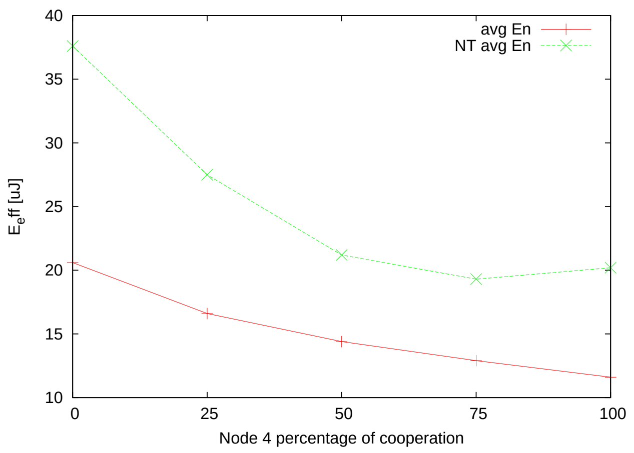


Fig. 6: Average effective energy

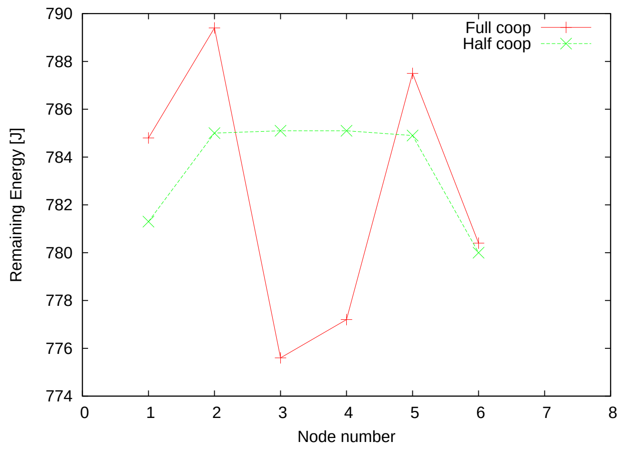


Fig. 7: Effect of defection on energy distribution

lowest values when cooperation is high.

In the second series of experiments we describe how an appropriate combination of defection and cooperation can yield a better distribution of the energy. We made use of the same 8 node testbed with two sessions of equal constant bit rate traffic between node 0 and 7 and vice versa, with a total duration of still $t = 10$ minutes. We evaluated the remaining energy at the end of the experiment for all the *inner* nodes labeled from 1 to 6 in the two different situations of (i) full nodes' cooperation and (ii) partial cooperation of node 3 and 4 for 50% of time. The final levels of energy are reported in Fig.7 showing a better balance when node 3 and 4 are semi-cooperating while keeping the same average consumption of all nodes, which is of 782.4 J with a variance of 31.6 in the former case, and of 783.5 J with a variance of 5.2 in the latter one.

VI. Conclusions

In this paper we showed how cooperation positively affects the performance of an ad hoc network, by helping reduce the overall energy consumption associated with data transmissions. We demonstrated through simulations that cooperation actually acts as an incentive for nodes, since it allows for a lower average energy expenditure with respect to the packets successfully delivered. We also studied the positive impact of cooperation on nodes' delivery ratio, which is considered a key performance indicator for any networked environment. Finally, we gave a first proof that if a subset of core nodes deliberately opts for defecting, the energy consumption can be better distributed among the nodes.

We do believe that the behavior-based approach that we presented in this work can be effectively exploited in a number of alternative scenarios, since it actually works along a dimension, which turns out to be complementary to other potential approaches, like, for example, ad-hoc designed energy-efficient routing paradigms.

This work is clearly a first step towards the study of cooperation effects in ad hoc networks. Among the numerous improvements that we identified and that represent directions of our future work, we firstly mention a more detailed analysis of the dependence of the performance improvements deriving from cooperation on the specific network topology taken into account. Apart from this, we also intend to study how the specific location of a node in the ad hoc network topology affects its performance and consequently its willingness to cooperate. This requires that a thorough analysis of the tradeoff between relaying other nodes' packets and sending one's own data is conducted.

References

[1] M. D'Arienzo, F. Oliviero, and S.P. Romano. Smoothing selfishness by isolating non-cooperative nodes in ad hoc wireless networks. In *Advances in Future Internet*, AFIN '10, pages 11–16, Washington, DC, USA, 2010. IEEE Computer Society.

[2] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Cooperation in wireless ad hoc networks. In *INFOCOM 2003.*, vol. 2, pp. 808–817, April 2003.

[3] K. Komathy and P. Narayanasamy. Trust-based evolutionary game model assisting aodv routing against selfishness. *J. Netw. Comput. Appl.*, 31(4), pp. 446–471, 2008.

[4] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *ACM MobiCom '00*, pp. 255–265, New York, USA, 2000.

[5] F. Olivero and S. P. Romano. A reputation-based metric for secure routing in wireless mesh networks. In *IEEE GLOBECOM 2008.*, pp. 1–5, December 2008.

[6] K. Mandalas, D. Flitzanis, G.F. Marias, and P. Georgiadis. A survey of several cooperation enforcement schemes for MANETs In *IEEE Int. Symp. on DOI* pp. 466 - 471, 2005

[7] C. K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks", IEEE Communications Magazine, 2001.

[8] P. Sondi and D. Gantsou, "Voice Communication over Mobile Ad Hoc Networks: Evaluation of a QoS Extension of OLSR using OPNET", Proceedings of AINTEC'09, Bangkok.

[9] D. Kim, J. J. Garia Luna Aceves, K. Obraczka,J. Cano and P. Manzoni, "Power-aware routing based on the energy drain rate for mobile ad hoc networks", in Proceedings of IEEE $11th$ International Conference on Computer Communications and Networks, Pages $562-569$, 2002.

[10] Floriano De Rango, Marco Fortino, "Energy efficient OLSR performance evaluation under energy aware metrics", in Symposium on Performance Evaluation of Computer and Telecommunication Systems, Pages $193-198$, 2009.

[11] S. Mahfoudh, P. Minet, "Survey of Energy Efficient Strategies in Wireless Ad Hoc and Sensor Networks", IEEE International Conference on Networking, Cancun, Mexico, Pages $1-7$ (2008).

[12] S. Zhong, Y. Yang, and J. Chen. Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks. In *INFOCOM 2003.*, vol 3, pp. 1987 - 1997, 2003.

[13] L. Buttyán and J. P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mob. Netw. Appl.*, 8(5), pp. 579–592, October 2003.

[14] Y. Po-Wah and C.J. Mitchell. Reputation methods for routing security for mobile ad hoc networks. *Mobile Future and Symp. on Trends in Communications*, pp. 130-137, 2003.
[15] S. Buchegger and J.-Y. Le Boudec. Performance analysis of the confidant protocol. In *ACM MobiHoc '02*, pp. 226–236, New York, USA, 2002.
[16] J. Nash. Non-Cooperative Games. *The Annals of Mathematics.*, 54(2), pp. 286–295, 1951.
[17] J. F. Nash. Equilibrium Points in n-Person Games. In *Proceedings of the National Academy of Sciences of the United States fo America.*, 36(1), pp. 48–49, 1950.
[18] R. Axelrod. *The Evolution of Cooperation*. Basic Books, 1988.
[19] R. Axelrod and D. Dion. The further evolution of cooperation. *Science*, 242(4884), pp. 1385–1390, December 1988.
[20] A. Urpi, M. Bonuccelli, and S. Giordano. Modelling Cooperation in Mobile Ad Hoc Networks: A Formal Description of Selfishness. In *Proceedigns of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks.*, 2003.
[21] M. Félegyházi, J. P. Hubaux, and Levente Buttyán. Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks. *IEEE Transaction on Mobile Computing.*, 5(5), pp. 463–476, 2006.
[22] E. Gelenbe and R. Lent Power-aware ad hoc cognitive packet networks. *Ad Hoc Networks*, 2(3), pp. 205–216, July 2004.
[23] E. Gelenbe, R. Lent, and Z. Xu. Design and performance of cognitive packet networks. *Perform. Eval.*, 46(2-3), pp. 155–176, 2001.
[24] E. Gelenbe. Learning in the recurrent random neural network. *Neural Comput.*, 5(1), pp. 154–164, 1993.