Modelling the Evolution of Cooperative Behavior in Ad Hoc Networks using a Game Based Model

Conference Paper · May 2007

DOI: 10.1109/CIG.2007.368084 · Source: DBLP

3 authors, including:

Pascal Bouvry
University of Luxembourg
399 PUBLICATIONS **2,184** CITATIONS

Modelling the Evolution of Cooperative Behavior in Ad Hoc Networks using a Game Based Model

Marcin Seredynski
University of Luxembourg
Faculty of Sciences,
Technology and Communication
6, rue Coudenhove Kalergi
L-1359, Luxembourg, Luxembourg
Email: marcin.seredynski@uni.lu

Pascal Bouvry
University of Luxembourg
Faculty of Sciences,
Technology and Communication
6, rue Coudenhove Kalergi
L-1359, Luxembourg, Luxembourg
Email: pascal.bouvry@uni.lu

Mieczyslaw A. Klopotek
Polish Academy of Sciences
Institute of Computer Science
ul. J.K.Ordona 21
01-237 Warsaw, Poland
Email: klopotek@ipipan.waw.pl

***Abstract*— In this paper we address the problem of cooperation and selfish behavior in ad hoc networks. We present a new game theory based model to study cooperation between nodes. This model has some similarities with the Iterated Prisoner's Dilemma under the Random Pairing game. In such game randomly chosen players receive payoffs that depend on the way they behave. The network gaming model includes a simple reputation collection and trust evaluation mechanisms. In our proposition a decision whether to forward or discard a packet is determined by a strategy based on the trust level in the source node of the packet and some general information about behavior of the network. A genetic algorithm (GA) is applied to evolve strategies for the participating nodes. These strategies are targeted to maximize the throughput of the network by enforcing cooperation. Experimental results show that proposed strategy based approach successfully enforces cooperation maximizing the network throughput.**

Keywords: Ad hoc networks, cooperation, selfish behavior, game theory

I. Introduction

A mobile ad hoc network is a network composed of two or more devices (nodes) equipped with wireless communications and network capability [1] [2]. Such network does not rely on any fixed architecture like base stations in traditional cellular networks or access points in wireless LANs. Routing functionality is incorporated into mobile nodes. Devices can directly communicate with each other only when they are located in their radio range. Otherwise, intermediate nodes should be used to forward packets. As a result, nodes beside sending their own packets are also expected to forward packets on behalf of others. An Ad hoc network is likely to be formed with small devices like laptops, PDAs, or smartphones that relay on batteries. Topology of such network may change quickly in an unpredictable way. Potential applications of wireless ad hoc networks include the tactical battlefield, emergency and rescue missions, as well as civilian ad hoc situations, such as conferences [2].

Since most of the devices participating in the network run on the batteries, the temptation to save energy might be very high. As shown in the literature [3] selfishness of the network participants can be a serious threat to the network. The solution to the selfish behavior problem could be so-called *self-policing mobile ad hoc networks* [4] [5] [6] [7] [8]. In such network nodes are equipped with a *reputation management system* combined with a *response mechanism*. Each node keeps its own rating of other network participants based on own experience and reputation data coming from other nodes. The idea of the cooperation enforcement mechanism based on the reputation is as follows. First, intermediate nodes should verify the reputation of the source of the packet that they are suppose to forward. If such packet comes from a node with a bad reputation then it is likely that it is going to be discarded by one of the intermediate nodes. This approach enforces cooperation because selfish nodes will not be able to use the network for their own purposes unless they contribute to the packet forwarding. Moreover, reputation management system can be helpful in finding the most reliable path from the source to the destination by avoiding untrusted nodes. Another possible approach to enforce cooperation is to introduce economic relations between the ability of sending own packets and forwarding packets for others [3].

Traditional tools to model ad hoc networks are not very good at modelling a high level property like cooperation [7]. This is why it is interesting to look at disciplines like economy and social science. In all of these disciplines game theory was used to study problems of conflict and cooperation among independent decision makers [9]. A game consists of a number of players taking actions (decisions) according to some strategies, with precise rules for the order in which players choose strategies, the information they have when they choose, and how they rate the desirability of resulting outcomes [9]. In game-theoretic terms cooperation in mobile network can be interpreted as a dilemma [6]. The node is tempted to get benefit (ability of sending packets) without cost (contribution to packet forwarding). However, if such behavior is noticed by other nodes then selfish node may end up at being excluded from the network. Selfish behavior would be risk free if a cooperation enforcement mechanism did not exist.

In this paper we address the problem of the selfish behavior in self-policing ad hoc networks. We propose a new game theory based model of the network and GA to evolve the behavior of its participants. The concept of the evolution of behavior using a game model and GA in a random

encounter scheme was already explored in the iterated Prisoner's Dilemma under Random Pairing game (IPDRP) [10]. Similarly to our problem it models the dilemma in an environment in which interaction sequences are short. But there are some important differences between such environment and ad hoc networks. Firstly, it assumes that only two players participate in each game and secondly it models a specific dilemma situation described by the Prisoner Dilemma payoff table.

The paper is organized as follows. In the next Section, related work is discussed. Next, in the Section III we show our trust and activity evaluation mechanisms. This is followed by the Section IV, where we explain our model game based model of ad hoc network. Then in Section V, where our strategy driven behavior is explained. Next, in Section VI we describe the evolution of strategies and network behavior using GA. Simulation results are presented in Section VII. Last Section concludes the paper.

II. Related work

A good survey of cooperation models with a game theoretical analysis can be found in [7].

In [11] authors present two techniques, *watchdog* and *pathrater* that aim at improving throughput of the network in the presence of selfish nodes. First, watchdog mechanism identifies selfish nodes and next, pathrater helps routing protocol to avoid this nodes. Such mechanisms do not discourage nodes from selfish behavior because selfish nodes are not excluded from the network. Authors show that in the network composed of 50 nodes with presence of 20 selfish nodes proposed mechanisms can increase the throughput by 17%.

In [8] authors propose a generic cooperation enforcement mechanism based on the reputation, which they call *CORE*. The solution is addressed to networks with low node density in which nodes are being part of a zone. The reputation is calculated using various types of data gathered by nodes. Three kinds of reputations are defined: subjective reputation, indirect reputation and functional reputation. More relevance is given to the past observations. Only positive values are exchanged between the nodes. This way a malicious broadcast of negative rankings for legitimate nodes is avoided. In such network selfish nodes are forced to contribute to the network operation. All service requests received from a misbehaving node will be ignored.

In [4] authors propose a mechanism called *CONFIDANT* whose goal is to make selfish behavior unattractive. It is based on selective altruism and utilitarianism. Both, the first and the second-hand observations are used. Similarly to CORE, packets coming from selfish nodes will not be forwarded by normally behaving nodes. Additionally, if a selfish node starts to behave correctly for a certain amount of time it might re-integrate with the network. CONFIDANT can be useful even when half of the nodes behave maliciously. In [5] authors further investigate the use of second-hand information. Bayesian approach to reputation systems is introduced: opinions that deviate from the first-hand observation and the majority opinion are excluded. As a result the reputation system is much more robust.

In [3] authors present an economic approach to the problem. Network is modelled as a market in which a virtual currency called *nuglet* is used. In such network nodes have to pay for the packets they want to send and are paid when they forward packets coming from other nodes. In order to protect against the fraud, nodes should be equipped with a tamper resistant security module made by a trusted manufacturer. Security issues of that model are further discussed in [12].

In [10] authors examine the evolution of cooperative behavior in the IPDRP. In opposite to iterated Prisoner's Dilemma in this game each player plays against a different randomly chosen opponent at every round. Each player has a single round memory strategy represented by a binary string of the length five. Each player is memorizing the result of its previous round encounter. The first bit of the strategy determines the first move of the player, while bits 2-4 define the moves for all possible scenarios in the previous round. Using GA authors analyze the evolution of both cooperation and strategies used by the players. In every generation each player plays 100 PD games (with randomly chosen opponent at every game).

III. Evaluation of trust

We assume that each node uses an omni-directional antenna with the same radio range. A source routing protocol is used, which means that a list of intermediate nodes is included in the packet's header. In our model the reputation information is gathered only by the nodes participating in the packet forwarding. Similarly to watchdog mechanism proposed in [11] each node monitors the behavior of the next forwarding node.

Reputation data is collected in the following way. Let's assume that node A wants to send a packet to node E using intermediate nodes B, C, and D. If the communication is successful then node E receives the packet and all nodes participating in that forwarding process update reputation information about each other. If communication fails (for example node D decides to discard the packet) this event is recorded by the watchdog mechanism of the node C. In such case node C forwards alert about selfish node D to the node B and then node B forwards it to the source node A (Fig. 1).

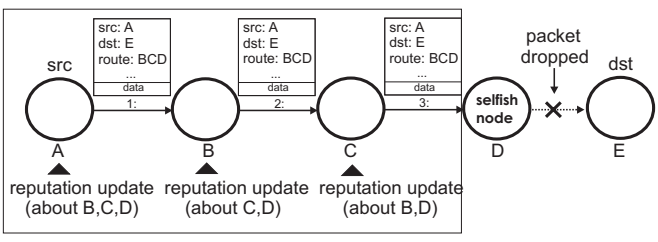

Fig. 1. Trust update mechanism example: communication failed because packet was discarded by the node D.

Lets suppose that node B wants to verify how trustworthy is node A (using available reputation data concerning node

B). In order to do this, first the fraction of correctly forwarded packets by node B is calculated (*forwarding rate*) and then the *trust lookup table* is used (Fig. 2).

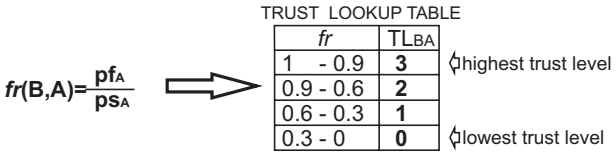

Fig. 2. Trust level evaluation. Node B verifies how trustworthy is node A. All data available to node B concerning behavior of the node A are used.

As a result one of the four possible trust levels is assigned. For example, forwarding rate of 0.95 results in the trust level 3.

If a node that wants to send a packet has more than one path available to the destination it will choose the one with the best reputation. A path rating (reputation) leading from node S to node D is calculated as a multiplication of all known forwarding rates of the nodes belonging to the route. An unknown node has a forwarding rate set to 0.5.

IV. An ad hoc gaming model

A. Description of the Ad Hoc Network Game

We define an *Ad Hoc Network Game* as a game in which one node (player) is originating the packet and some other nodes have to decide whether to forward or to discard it.

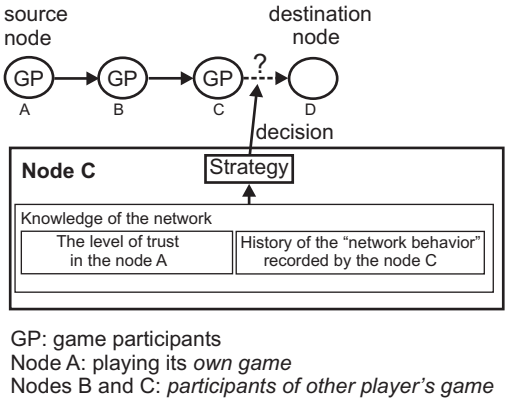

Fig. 3. A single ad hoc network game.

The number of game participants (GP) depends on the length of the path leading from the source to the destination node. Game participants are composed of the source node and all intermediate nodes. The destination node is not a part of the game. Each player is said to play *his own game* when being a source of a packet and is said to be a *participant of other players' game* when being an intermediate node. All intermediate nodes are chosen randomly. This simulates a network with a high mobility level, in which topology changes very fast. In the example shown in Fig. 3 the game is composed of 3 nodes: node A, B and C. Node A is the source of the packet while nodes B and C are intermediate nodes asked to forward the packet.

After the reception of the packet node B has to decide whether to forward or to discard the packet received from the node A. If node B decides to discard the packet then the game ends. Otherwise, it is the turn of node C to decide what to do with the packet. If all intermediate nodes decide to forward the packet, the communication is successful.

Each player uses a strategy that defines its reaction to forwarding request (see Section V for details). Such strategy depends on the history of the results of the two previous own games and the level of trust in the initiator (source node) of the packet.

After the game is finished all its participants receive payoffs according to the decisions they made. There are two payoff tables. One is applied for the source node and the other one for the intermediate nodes.

B. Payoff table and fitness function

The goal of payoffs is to capture essential relations between alternative decisions and their consequences. Payoff tables for a source node and intermediate nodes are shown in Fig. 4a. For the forwarding node the exact payoff depends only on the status of the transmission. If the packet reaches the destination then transmission status is denoted as S (success). Otherwise, if the packet is discarded by one of the intermediate node's then transmission status is denoted as F (failure). Payoffs received by the intermediate nodes depend on their decisions (packet discarded or forwarded) and on their trust level in the source node. Generally, the higher the trust level is the higher payoff is received by the node forwarding the packet. High trust level in the source node means that in the past this node already forwarded some packets for the currently forwarding node. So it is more likely that such node will be used in the future (when sending its own packets, routes with best reputation are chosen). This means that forwarding for such node might be considered as an investment of trust for the future situations. When a node decides to discard a packet it is rewarded for saving its battery live. On the other hand, such node will lose reputation among some of the network participants. Discarding packets originating from less trusted nodes should be better paid that discarding packets coming from untrusted nodes. Reason for this is that nodes with lower trust level will rather be avoided in the future communication so there is no real interest in building good trust relationship with such nodes.

The payoff table for intermediate nodes reflects the use of the reputation based cooperation enforcement system by network participants. If such system was not used, the payoff for selfish behavior (discarding packets) would always be higher than for forwarding. The reason for this is that selfish behavior would not be noticed in the network, so it would be always better to save energy by not participating to the packet forwarding.

An example of the game is shown in Fig. 4b.: node A wants to send a packet to node D. The path goes through

nodes B and C.

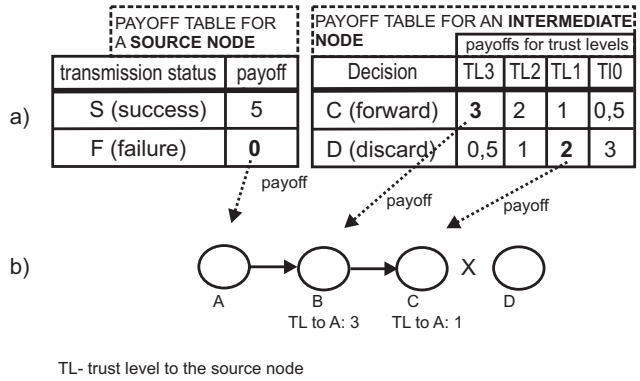

Fig. 4. Payoff tables for source and intermediate nodes (a), an example of a game: node D did not receive packet sent by node A (packed discarded by node C) (b).

After the reception of the packet node B decides to forward it and as a result it receives a payoff according to the payoff table for the intermediate node (Fig. 4a). The next node on the way to the destination (node C) decides to discard the packet and receives its appropriate payoff afterwards. Finally, the source node receives a payoff according to the status of the transmission (failure in the example shown).

The fitness value of each player is calculated as follows:

$$fitness = \frac{tps + tpf + tpd}{ne}, \tag{1}$$

where tps, tpf, tpd are total payoffs received respectively for sending own packets, forwarding packets on behalf of others and discarding them. The ne is a number of all events (number of own packets send, number of packets forwarded and number of packets discarded).

C. Types of players

Two types of players (nodes) are used in our game: *normal nodes* (NN) and *constantly selfish nodes* (CSN). A Normal node plays according to some strategy (which evolves in the evolutionary process). Its goal is to send maximum number of packets and save battery live at the same time. The CSN never cooperates (always drops packets). Such player is not included in the selection and reproduction. In each generation the number of CSN remains the same.

D. Tournament scheme

Strategy of each player is evaluated in a *tournament*. We define different tournaments varying in some parameters that represent specific network conditions. We call them *tournament environments* (explained in Section IV-C). In every tournament a number of ad hoc games is repeatedly played (as described in Section IV-A). Each tournament is composed of R rounds. In every round each player is a source of a packet exactly once (plays its own game) and participates in the packet forwarding several times (as a participant of other player's games). A destination node and intermediate nodes are chosen randomly depending on the *path environment* being used (see Section VII-C). Both maximum number of paths and maximum number of intermediate nodes are parameters. The tournament itself can be described as follows:

Tournament scheme

Step 1: Specify i (source node) as $i := 1$, K as a number of players participating in the tournament and R as a number of rounds.
Step 2: Randomly select player j (destination of the packet) and the intermediate nodes as described in Section VII-C.
Step 3: For each available path calculate its rating (as described in Section III) and select the path with the best reputation.
Step 4: Play the game (as described in Section IV-A).
Step 5: Update the memory of the source node (transmission status) according to the result of the game.
Step 6: Update payoffs of the source node i and all intermediate nodes (game participants) that received the packet.
Step 7: Update the reputation data among all game participants (as described in Section III).
Step 8: If $i < K$, then choose the next player $i := i + 1$ and go to the step 2. Else go to the step 9.
Step 9: If $r < R$, then $r := r + 1$ and go to the step 1 (next round). Else stop the game.

E. Evaluation of strategies in series of tournament environments

Strategies are evaluated in a series of *tournaments*. The evaluation scheme of all the players is shown in Fig 5.

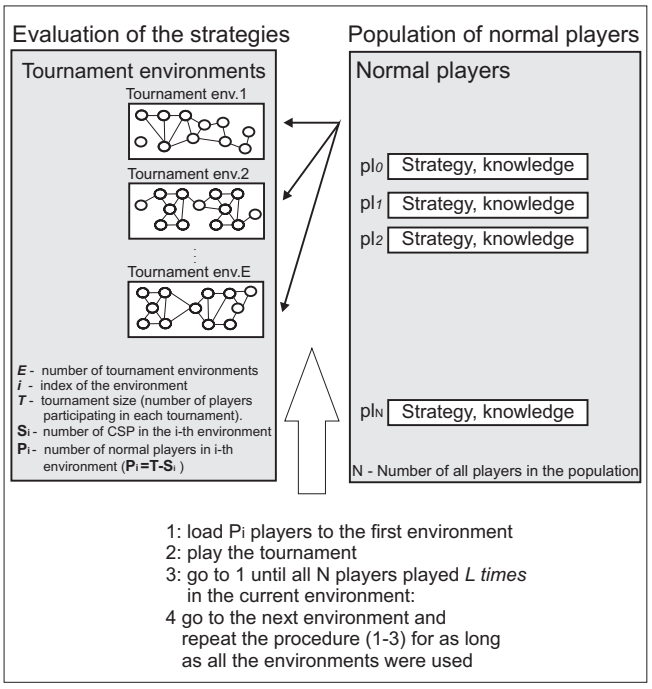

Fig. 5. Evaluation of strategies. Several tournament environments are used.

The total number of players participating in the tournament (*tournament size*) is the same in each environment.

Players participate in the tournament environments in the following way:

Evaluation of strategies in several tournament environments

Step 1: Let E be a number of tournament environments, T - a tournament size (a number of players participating in each tournament), N - population size (a number of normal nodes) Si - a number of selfish players in the $i-th$ environment, Pi - a number of normal nodes in each environment ($Pi = T - Si$) and L - number of times every player plays in each of the tournaments. Clear the memory (reputation/transmission status history) of all N players and specify i as $i := 1$.

Step 2: Randomly choose Pi players among all the players that played less then L times in the current environment.

Step 3: Play the tournament in the $i-th$ environment (as described in Section IV-D).

Step 4: If all players already played the $i-th$ environment L times, then go to the Step 5. Otherwise, go to the step 2.

Step 5: If $i < E$, then $i := i + 1$ and go to the step 2. Else stop the evaluation.

V. Coding the strategy

The decision whether to forward or to discard the packet is determined by the strategy represented by a binary string of length 18. An example of a strategy is shown in Fig. 6a.

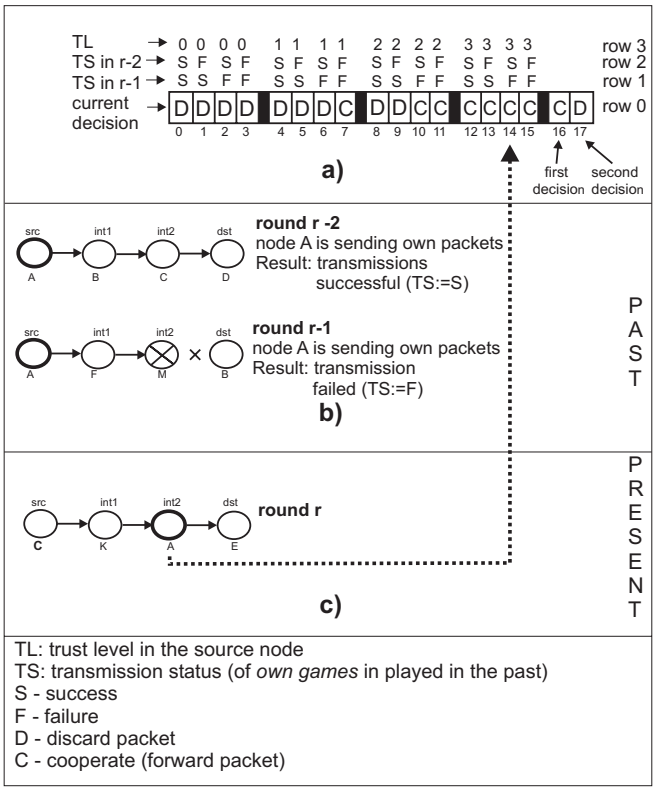

Fig. 6. Coding and using a strategy: coding the strategy (a), collecting past experiences (when sending its own packets in two successive games) (b), using the strategy when being asked to forward a packet (c).

The exact decision (row 0) is based on a finite history of a *transmission status* of the packets originated by the node (row 1 and row 2) and *trust level* in the source node of the packet (row 3). The history is limited to the last two own games (at rounds $r-1$ and $r-2$). If last packet sent by the node was successfully received by the destination then transmission status for a given round is denoted as S (success). If the packet was discarded by one of the intermediate nodes then the status is denoted as F (failure). There are 16 possible combinations of the transmission status history and trust level in the source node. Decisions for each case are represented by bits no. 0-15. Bits no. 16 and 17 represent the first two decisions (no transmission status available at that time). Decision C stands for cooperation (forward the packet) and D stands for defect (discard the packet). For example, lets suppose that node A receives a packet originally coming from node C (Fig. 6c). Assuming that node A has a trust 3 in node C and its game in the round $r-2$ finished with success and game in round $r-1$ finished with failure (Fig. 6b) then according to the strategy shown in Fig. 6a the decision would be to forward the packet (decision C, bit no. 14).

VI. Evolution of the behavior using GA

In order to analyze behavior of the network under particular conditions and to search an optimal strategy we use similar evolutionary technique as in [10]. There are N players participating in all defined tournaments. At the beginning of the evolution randomly generated strategies are assigned to each of N players. Then, the series of tournaments are executed according to the scheme described in Section IV-E. Next, selection and reproduction operators are applied on the current population of strategies: fitness value of each player's strategy is calculated as the average payoff obtained in all the tournaments. Then N pairs of strategies are selected using roulette wheel selection with a linear scaling. The new strategies are obtained by applying crossover and mutation operators to each of N selected pairs. Standard one-point crossover is used. One of the two strategies created after crossover is randomly selected to the next generation. Finally, the standard uniform bit flip mutation is applied. As a result a new population of strategies for each player is created. The process is repeated for a predefined number of times.

VII. Experiments

A. A number of players

The total number of normal nodes (population size) is 100. Number of players (both NP and CSN) participating in each tournament environment is 50. The exact proportion of particular type of players depends on the tournament environment.

B. Parameters of tournament environments

In order to test strategies in various networking conditions we defined four tournament environments, called TE1, TE2, TE3 and TE4. The only difference between them is the

number of CSN players. The numbers of CSN associated with each environment are shown in Tab. I.

TABLE I
PARAMETERS OF TOURNAMENT ENVIRONMENTS (TE).

	TE1	TE2	TE3	TE4
number of CSN	0	10	25	30
number of normal nodes	50	40	25	20

Number of CSN varies from 0 to 30, depending on the tournament environment.

C. Path length

When a node wants to send a packet (when playing its own game) first a path length (number of hops) is chosen and then the number of available paths of previously selected length is randomly generated. Path length is chosen according to predefined probabilities. A number of hops from the source node to the destination varies from 2 to 8. Path length is chosen according to predefined probabilities as shown in Tab. II.

TABLE II
PROBABILITY OF SELECTING A PARTICULAR NUMBER OF HOPS TO THE DESTINATION (PATH LENGTH).

	Path length probability
2 hops	0.4
3 hops	0.3
4 hops	0.1
5-8 hops	0.05

Additionally, for each path length a number of available alternate paths to the destination is available according to the probabilities shown in Tab.III. In general, the longer the path is, more likely less routes to the destination are going to be available.

TABLE III
PROBABILITY OF THE NUMBER OF AVAILABLE PATHS FOR EACH PATH LENGTH.

	1 path	2 paths	3 paths
2-3 hops	0.5	0.3	0.2
4-6 hops	0.6	0.25	0.15
7-8 hops	0.8	0.15	0.05

D. Evaluation cases

We examine the evolution of behavior among network participants in two cases. In the first case (*case 1*) players are evaluated in only one selected environment (as described in Section IV-D). Four independent evolutions (one for each tournament environment) are obtained. Players use different strategies for each environment (evolutions of behavior are independent for each environment). In such approach the evolved strategies are suppose to perform best in one particular environment.

In the second case (*case 2*) in each generation players are evaluated in all defined test environments (as described in Section IV-E). It means that each player uses his own unchangeable strategy in all environments. The evolved strategies are suppose to be general enough to perform well in all environments. Each player plays the tournament twice in each environment.

E. Parameters of GA

The following parameters are used for the experiments: crossover probability: 0.9; mutation probability 0.001; number of rounds in the tournament: 300; number of generations: 500. The unknown nodes have a default trust value assigned to 1. All the experiments are repeated 60 times and the average value is calculated as a result.

F. Results: evolution of cooperation

We define *cooperation level* as a percentage of packets that originated by normal nodes and successfully reached the destination. The results for both evaluation cases (described in the Section VII-D) are shown in Fig. 7.

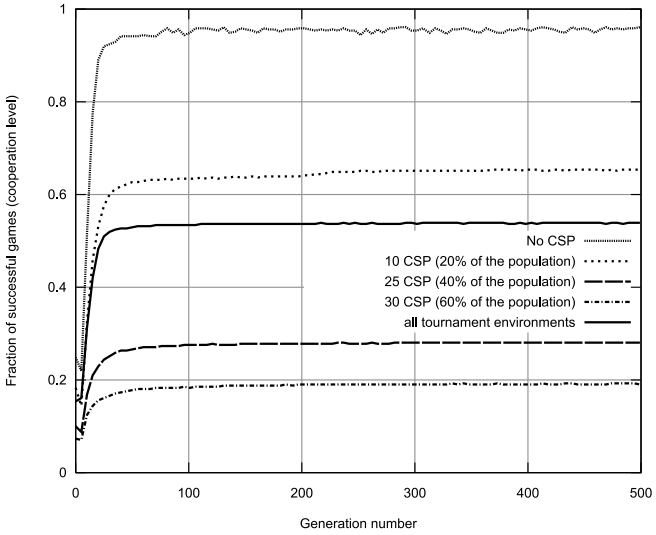

Fig. 7. The evolution of cooperation.

The following results are obtained for separate evolutions in each environments (first evaluation case): when players play in the CSN free tournament (TE1), the level of cooperation is about 96%. When 20% of the population of the tournament is composed of CSN (TE2, 10 CSN) the cooperation level drops to 65%. For the 25 CSN tournament (TE3) the level of cooperation is close to 28%. When most of the population (60%) is composed of CSN (TE4, 30 CSN) the cooperation level drops to about 19% (which means that only 19% of packets originated by non-CSN nodes reach the destination).

When players players are evaluated in all tournament environments (second evaluation case) the cooperation level is close to 54%. Additional results for the second evaluation

case are shown in Tab.IV. These results are taken from the last generations (average value of all experiments). In the second column one can see the cooperation level measured independently for each environment. In the third column the cooperation level for the paths that did not contain CSN is shown. Percentage of paths that did not contain CSN is shown in the last column (when sending packets, normal nodes try to avoid CSN by choosing paths with the best reputation).

TABLE IV
COOPERATION LEVEL (CL) FOR EACH ENVIRONMENT MEASURED SEPARATELY (SECOND COLUMN), CL IN CASE WHEN NO CSN WERE INCLUDED IN THE PATH (THIRD COLUMN), PERCENTAGE OF PATHS CHOSEN WITH NO CSN (FORTH COLUMN). RESULTS TAKEN FROM THE LAST GENERATIONS OF THE SECOND EVALUATION CASE.

	Cl	Cl no when no CSN	CSN-free paths
TE1	0.997	0.997	100%
TE2	0.656	0.996	65.91%
TE3	0.281	0.986	28.55%
TE4	0.193	0.972	19.89%

Cooperation levels measured for each environment separately were almost the same as in the first case when the evolution was performed for each environment separately (players were using strategies that evolved for the particular environment). The difference between each of the environments was in the number of CSN which resulted in the number of CSN-free paths available. When the cooperation was measured excluding paths containing CSN, its level was quite similar in all environments (97%-99%, third column).

In Tab.V one can see how *forwarding requests* coming from normal nodes and CSN were treated in the network. We define a forwarding request as a situation in which a node is asked to forward a packet. Requests coming from normal and CSN nodes are shown.

TABLE V
RESPONSE TO PACKET FORWARDING REQUESTS COMING FROM NORMAL NODES AND CSN. RESULTS TAKEN FROM THE LAST GENERATIONS OF THE SECOND EVALUATION CASE.

	Normal players	CSN
Number of requests	515994	173456
Req. accepted	77.46%	4.3%
Req. rejected by NP	0.28%	52.78%
Req. rejected by CSN	22.26%	42.9%

There were 515994 forwarding requests coming from normal players. Around 77% of them were accepted (packet forwarded). Most of the rejections came from CSN (22%). The acceptance percentage of requests coming from CSN was only 4.3%. All unknown nodes have a trust level 1 by default. All forwarded packets coming from CSN were forwarded at the beginning of the tournament, at the time when CSN were seen as unknown nodes. As the reputation of CSN decreased with time, such nodes did not manage to send any more packets.

The case in which there are no CSN simulates a situation in which all nodes try to minimize the use of battery but at the same time they want to send the maximum possible number of packets. So, if the selfish behavior does not allow sending the desired number of packets then the node is modifying its strategy to the more cooperative one. In the CSN-free environment on can see that nodes decide to cooperate (and as a result gain trust) for most of the times because it is the only way to use the network for its own purposes. The CSN nodes are not interested in sending its own packets so the cooperation enforcement system will not convince them to participate in packet forwarding. With the presence of CSN, nodes become more restrictive to the less trusted nodes. This is probably because they "learn" that nodes with low trust will not change its behavior (which is only true for CSN).

G. Payoffs received by normal players

During each generation players receive payoffs for sending own packets and discarding or forwarding packets received from other nodes. The results for the first evaluation case are shown in Fig. 8.

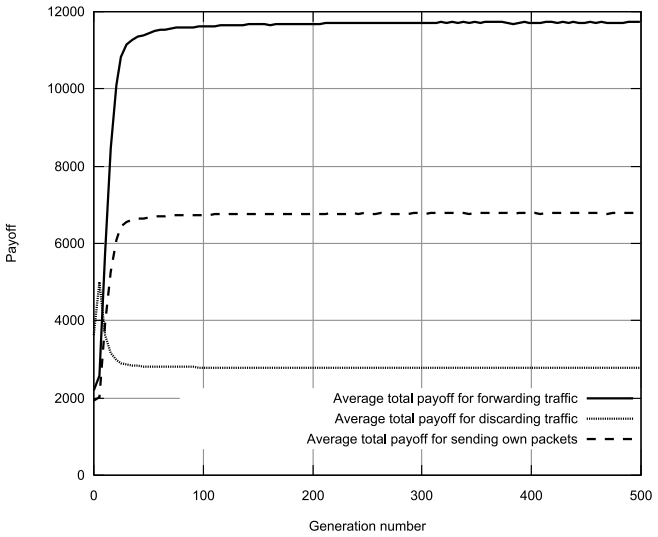

Fig. 8. Payoffs for sending, discarding and forwarding packets

The fitness value (not shown in the figure) starts from 1.23 (first generation) and raises up to 2.82 in the last generation. The highest payoff in the final generation is received for forwarding packets. Payoff for discarding traffic is by far the lowest.

H. Winning strategies

During the evolutionary process the initial randomly generated strategies evolve and as a result the cooperation level in the network decreases. We analyze strategies for each trust level separately (sub-strategies). Firstly, strategies that evolved when being evaluated in all environments (evaluation case 2) are shown. The sub-strategies that were present in

more than 3% of populations in the last generations are presented in Tab.VI.

TABLE VI
EVOLVED SUB-STRATEGIES WHEN BEING EVALUATED IN ALL ENVIRONMENTS (EVALUATION CASE 2)

Trust 0	Trust 1	Trust 2	Trust 3
0000 (99%)	1111 (98%)	1111 (90%)	1111 (99%)
-	-	1011 (4%)	-

One can see that for the trust level 0 the winning sub-strategy is "always discard incoming traffic", while for other trust levels the opposite strategy "always forward" wins. Next, the dominating sub-strategies for case when being evaluated in one environment only (first evaluation case) are shown. We demonstrate the results for the environment TE1 (Tab.VII) and TE4 (Tab.VIII).

TABLE VII
EVOLVED SUB-STRATEGIES WHEN BEING EVALUATED IN THE ENVIRONMENT TE1 (CSN-FREE)

Trust 0	Trust 1	Trust 2	Trust 3
0100 (18%)	0111 (74%)	1011 (32%)	1111 (98%)
0000 (16%)	0011 (10%)	1101 (19%)	-
0010 (15%)	0101 (8%)	1111 (17%)	-
1000 (9%)	1101 (3%)	1110 (13%)	-

For the TE1 environment only one dominating sub-strategy evolved. It was "always forward" when the source node has a trust level 4. For other trust levels several sub-strategies evolved. In general, when comparing to the evaluation case 2 (VI), strategies for trust levels 1 and 2 are less cooperative while for lowest trust level slightly more cooperative.

TABLE VIII
EVOLVED SUB-STRATEGIES WHEN BEING EVALUATED IN THE ENVIRONMENT TE4 (30 CSN)

Trust 0	Trust 1	Trust 2	Trust 3
0000 (99%)	1111 (79%)	1111(51%)	1111 (98%)
-	0111 (16%)	0111(42%)	-

Sub-strategies evolved in TE4 environment (Tab.VIII) are quite similar to the evaluation case 2, that is allowing cooperation in trust levels 1-3 and rejecting forwarding request coming from nodes with trust 0 level.

A default trust level for unknown nodes is set to 1. In all evaluation cases, the evolved strategies for the trust level 1 were cooperative (i.e., strategies with many 1s). As a result, new nodes can easily join the network and start sending own packets.

When network conditions are unknown the most reasonable approach would be to use strategies that were evaluated when being evaluated in many environments. If those conditions are somehow known it would be best to use strategies that evaluated in specific network conditions.

VIII. CONCLUSIONS

Traditional tools to model ad hoc network are not very good at modelling a high level property like cooperation and at testing cooperation enforcement systems. In this paper, we have proposed a new game based model to examine the evolution of cooperative behavior in ad hoc network. It is composed of three elements: a game based model of an ad hoc network, a reputation system and GA. Using GA appropriate strategies for the network participants were evolved. Experimental results showed that the proposed cooperation enforcement mechanism based on strategies was good enough to enforce high level of cooperation among the nodes that were interested in sending their own packets. Fair contribution to the packet forwarding was the only way to be able to send its own packets. Evolved rules allow new nodes to join the network. However, if the population was composed of a high number of selfish nodes which were not interested in sending their own packets, the cooperation level in such network strongly decreased. Our future work will address such issues like false accusations, temporary failures, noise, etc.

ACKNOWLEDGMENTS

This research benefits from FNR SECOM-SIM funding.

REFERENCES

[1] M. Ilyas and I. Mahgoub, Eds., *Mobile Computing Handbook*. Auerbach Publications, 2005.
[2] C. Perkins, Ed., *Ad Hoc Networking*. Addison-Wesley, 2001.
[3] L.Buttyan and J.-P.Hubaux, "Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks," Swiss Federal Institute of Technology, Tech. Rep. DSC/2001/001, 2001.
[4] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the confidant protocol," in *Proc. ACM 3rd International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, 2002, pp. 226–236.
[5] ——, "The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-Hoc Networks," in *Proc. Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03)*, 2000, pp. 131–140.
[6] ——, "Self-policing mobile ad-hoc networks by reputation systems," *IEEE Communications Magazine, Special Topic on Advances in Self-Organizing Networks*, vol. 43, no. 7, July 2005.
[7] S. Giordano and A. Urpi, *Mobile Ad Hoc Networking*. Wiley-IEEE Press, 2004, ch. 13.
[8] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proc. IFIP 6th Conference on Security Communications, and Multimedia (CMS'02)*, 2002, pp. 107–121.
[9] C. Camerer, *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003.
[10] N. Namikawa and H. Ishibuchi, "Evolution of cooperative behavior in the iterated prisoner's dilemma under random pairing in game playing," in *Proc. IEEE Press Congress on Evolutionary Computation (CEC'05)*, 2005, pp. 2637–2644.
[11] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. ACM/IEEE 6th International Conference on Mobile Computing and Networking (MobiCom'00)*, 2000, pp. 255–265.
[12] L.Buttyan and J.-P.Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 8, no. 5, Oct. 2003.