Evolution of Strategy Driven Behavior in Ad Hoc Networks Using a Genetic Algorithm

Marcin Seredynski[1], Pascal Bouvry[1], Mieczyslaw A. Klopotek[2]

[1]University of Luxembourg
Faculty of Sciences, Technology and Communication
6, rue Coudenhove Kalergi, L-1359, Luxembourg, Luxembourg
{marcin.seredynski, pascal.bouvry}@uni.lu

[2]Polish Academy of Sciences
Institute of Computer Science
Ordona 21, 01-237 Warsaw, Poland
klopotek@ipipan.waw.pl

Abstract

In this paper we address the problem of selfish behavior in ad hoc networks. We propose a strategy driven approach which aims at enforcing cooperation between network participants. Each node (player) is using a strategy that defines conditions under which packets are being forwarded. Such strategy is based on the notion of trust and activity of the source node of the packet. This way network participants are enforced to forward packets and to reduce the amount of time of being in a sleep mode. To evaluate strategies we use a new game theory based model of an ad hoc network. This model has some similarities with the Iterated Prisoner's Dilemma under the Random Pairing game where randomly chosen players receive payoffs that depend on the way they behave. Our model of the network also includes a simple reputation collection and trust evaluation mechanisms. A genetic algorithm (GA) is applied to find good strategies. Experimental results show that approach can successfully enforce cooperation among ad hoc networks participants.

1. Introduction

A mobile ad hoc network is a network composed of two or more devices (nodes) equipped with wireless communications and network capability [6] [13]. 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. Topology of such network may change quickly in an unpredictable way.

Since most of the devices participating in the network run on batteries, the temptation to save energy might be very high. As shown in the literature [7] [9] [11], 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* [1] [2] [3] [5] [10]. In such networks 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. Firstly, intermediate nodes should verify the reputation of the source of the packet that they are requested 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. Another possible approach to enforce cooperation is to introduce economic relations between the ability of sending own packets and forwarding packets for others [7].

In game-theoretic terms cooperation in mobile network can be interpreted as a dilemma [3]. Nodes are 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.

1-4244-0910-1/07/$20.00 ©2007 IEEE.

Energy saving can be done by discarding packets or switching into a *sleep mode*. However the the greatest saving is done when wireless network interface is operating in a sleep mode [4]. The power consumption is about 98% lower comparing to the one in the *idle mode*. The significantly higher idle power consumption reflects the cost of listening to the wireless channel. If a node wants actively participate to the network then its network interface should be in the idle mode, ready to receive traffic from its neighbors. Being in a sleep node will not decrease nodes' reputation because such behavior will be unnoticed by other network participants (since it is not possible to distinguish between node being in a sleep mode and node temporally leaving the network). For these reasons nodes should be rewarded based on their activity in the network.

In this paper we address the problem of the selfish behavior in self-policing ad hoc networks. Our approach aims at enforcing cooperation using the notions of trust and activity. We propose a strategy driven behavior of network participants. The decision whether to forward or discard packets depends in the trust level to the source node and its activity. We propose a new game theoretic-based model of the ad hoc network whose goal is to evaluate strategies. This model has some similarities with the Iterated Prisoner's Dilemma under the Random Pairing (IPDRP) game in which randomly chosen players receive payoffs that depend on the way they behave [12]. A GA is used to search for good strategies.

The paper is organized as follows. In the next Section, related work is discussed. Then in Section 3 we show our trust and activity evaluation mechanisms and we explain proposed strategy driven behavior. This is followed by Section 4, where we present our game based model of ad hoc network. Next, in Section 5 we describe how strategies are evolved using GA. Simulation results are presented in Section 6. Last Section concludes the paper.

2 Related work

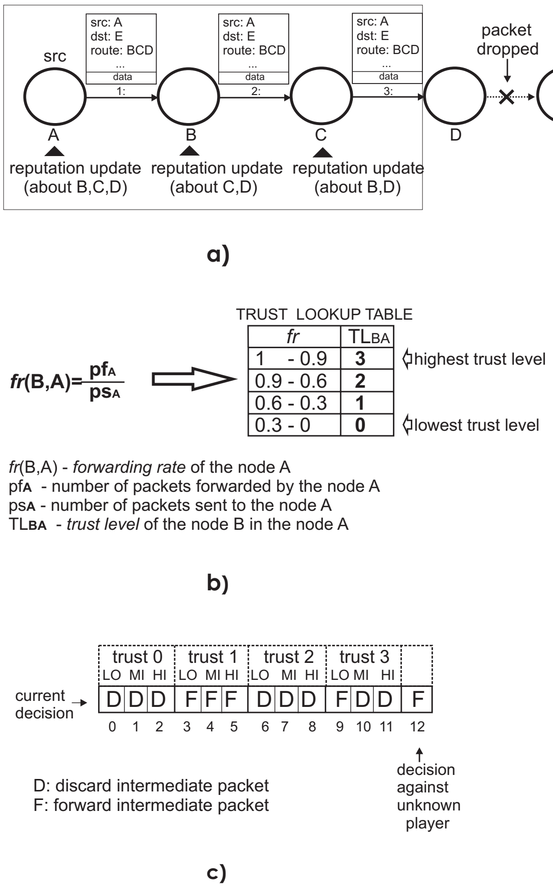
A good survey of cooperation models with a game theoretical analysis can be found in [5]. In [9] authors present two techniques, *watchdog* and *pathrater* that aim at improving throughput of the network in the presence of selfish nodes. First, watchdog mechanism identifies selfish nodes and next, pathrater helps routing protocol to avoid these 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 [10] 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. 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 [2] authors propose a mechanism called *CONFIDANT* whose goal is to make selfish behavior unattractive. 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. In [1] use of second-hand information is further investigated. In [7] 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. Security issues of that model are further discussed in [8]. In [12] 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 memorizes 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.

3 Evaluation of trust, activity levels and coding of the strategy

3.1 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 nodes directly participating in the packet forwarding. Similarly to watchdog mechanism proposed in [9] each node monitors the behavior of the next forwarding node.

Reputation data is collected in the following way. Let's suppose 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


src
src: A
dst: E
route: BCD
...
data
1:
2:
3:
packet dropped
dst
A
B
C
D
E
reputation update (about B,C,D)
reputation update (about C,D)
reputation update (about B,D)
a)
$fr(B,A)=\frac{pf_A}{ps_A}$
highest trust level
lowest trust level
b)
decision against unknown player
c)

TRUST LOOKUP TABLE

fr	TL_{BA}
1 - 0.9	**3**
0.9 - 0.6	**2**
0.6 - 0.3	**1**
0.3 - 0	**0**

fr(B,A) - *forwarding rate* of the node A
pfA - number of packets forwarded by the node A
psA - number of packets sent to the node A
TLBA - *trust level* of the node B in the node A

	trust 0			trust 1			trust 2			trust 3			
	LO	MI	HI	LO	MI	HI	LO	MI	HI	LO	MI	HI	
current decision	D	D	D	F	F	F	D	D	D	F	D	D	F
	0	1	2	3	4	5	6	7	8	9	10	11	12

D: discard intermediate packet
F: forward intermediate packet

Figure 1. Trust update mechanism example: communication failed because packet was dropped by node D (a), trust level evaluation (b), coding of the strategy (c).

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 a).

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. 1b). 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 source node has more than one path available to the destination it will choose the one with the best reputation. A path rating is calculated as a multiplication of all known forwarding rates of all nodes belonging to the route. An unknown node has a forwarding rate set to 0.5.

3.2 Evaluation of the activity level

We define three activity levels: low (LO), medium (MI) and high (HI). Those levels are calculated using the same reputation data as used for trust evaluation. In order to verify the activity level of a source node an intermediate node calculates the average value of all packets forwarded by all known nodes (denoted as av). This value is next compared with the number of packets forwarded by the source node. If this number belongs to a range $< av - 0.2 * av...av + av * 0.2 >$ then the medium activity level is assigned. Low activity level is assigned in the case when this number is smaller than that range while high level in the case when this number is above that range.

3.3 Coding the strategy

The decision whether to forward or discard the packet is determined by the strategy represented by a binary string of the length 13. An example of a strategy is shown in Fig. 1c. The exact decision is based on two elements: trust level in the source node and its activity level. There are 12 possible combinations of trust and activity levels. Decisions for each case are represented by bits no. 0-11. Bit no. 12 defines behavior against an unknown node. Decision F means ”forward packet” while D stands for the opposite (drop the packet). For example, lets suppose that node B receives a packet originally coming from node A. Assuming that node B has a trust level 3 in node A, and nodes’ A activity is ”LO” then according to the strategy shown in Fig. 1c the decision would be to forward the packet (F, bit no. 9).

4 An ad hoc gaming model

4.1 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.

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. 2 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.

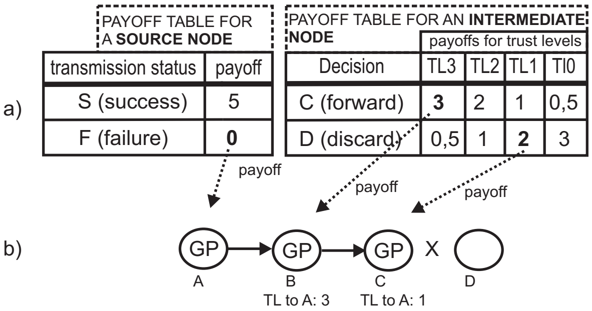
After the reception of the packet node B has to decide whether to forward or to discard the packet received from the node A. If node B decides to discard the packet then the game ends. Otherwise, it is the turn of node C to decide what to do with the packet. If all intermediate nodes decide to forward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

4.2 Payoff table and fitness function

The goal of payoffs is to capture essential relations between alternative decisions and their consequences. There are two payoff tables. One is applied for the source node and the other one for the intermediate nodes. Payoff tables for a source node and intermediate nodes are shown in Fig. 2a. For the source 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 (packet discarded by one of the intermediate node's), the transmission status is denoted as F (failure). Payoffs received by 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. 2b.: node A wants to send a packet to node D. The path goes through nodes B and C. 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. 2a). 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).



PAYOFF TABLE FOR A **SOURCE NODE**

transmission status	payoff
S (success)	5
F (failure)	**0**

PAYOFF TABLE FOR AN **INTERMEDIATE NODE**

	payoffs for trust levels			
Decision	TL3	TL2	TL1	Tl0
C (forward)	**3**	2	1	0,5
D (discard)	0,5	1	**2**	3

TL- trust level to the source nodeGP: game participants
Node A: playing its *own game*
Nodes B and C: *members of other player' game.* Node B: destination

Figure 2. 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).

The fitness value of each player is calculated as follows:

$$fitness = \frac{tps + tpf + tpd}{ne}, \qquad (1)$$

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).

4.3 Types of nodes

Two types of nodes (players) are used in our game: *normal nodes* (NN) and *constantly selfish nodes* (CSN). A Normal node plays according to some strategy (that 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.

4.4 Tournament scheme and evaluation of strategies

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*. In every tournament a number of ad hoc games is repeatedly played (as described in Section 4.1). 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 mode* being used (see Section 6.1). 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.

Step 3: For each available path calculate its rating (as described in Section 3.1) and select the path with the best reputation.

Step 4: Play the game (as described in Section 4.1).

Step 5: Update payoffs of the source node i and all intermediate nodes (game participants) that received the packet.

Step 6: Update the reputation data among all game participants (as described in Section 3.1).

Step 7: If $i < K$, then choose the next player $i := i + 1$ and go to the step 2. Else go to the step 8.

Step 8: If $r < R$, then $r := r + 1$ and go to the step 1 (next round). Else stop the tournament.

Strategies are evaluated in a series of *tournament environments*. The evaluation scheme is shown in Fig 3.

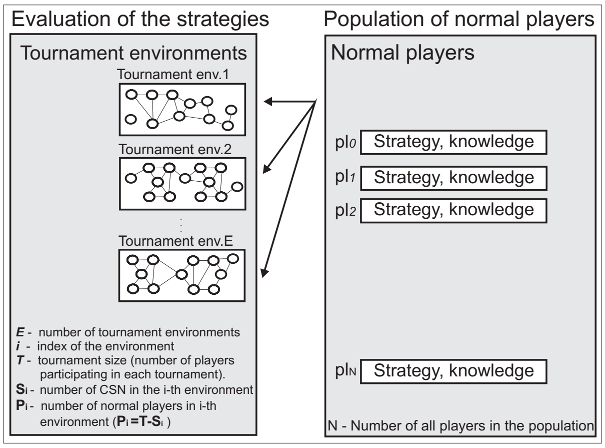

Evaluation of the strategies
Population of normal players
Tournament environments
Tournament env.1
Tournament env.2
Tournament env.E
Normal players
pl0 Strategy, knowledge
pl1 Strategy, knowledge
pl2 Strategy, knowledge
plN Strategy, knowledge
E - number of tournament environments
i - index of the environment
T - tournament size (number of players participating in each tournament).
Si - number of CSN in the i-th environment
Pi - number of normal players in i-th environment (Pi =T-Si)
N - Number of all players in the population

Figure 3. 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 series of 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/activity data) of all N players and specify i as $i := 1$.

Step 2: Randomly choose Pi players among all 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 4.4).

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.

5 Evolution of the behavior using GA

In order to analyze behavior of the network under particular conditions and to search optimal strategies we apply similar evolutionary technique as in IPDRP problem [12] except that we use a tournament selection instead of a roulette one. 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 4.4. 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 a tournament selection. 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.

6 Experiments

6.1 Conditions of experiments

Tournament environments: in order to test strategies in various networking conditions we defined four tournament environments, called TE1, TE2, TE3 and TE4. The

Table 1. Parameters of tournament environments (TE).

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

Table 2. Probability of selecting a particular number of hops to the destination (path length).

	Shorter paths	longer paths
2 hops	0.2	0.1
3-4 hops	0.3	0.1
5-8 hops	0.05	0.1
9-10 hops	0.00	0.15

only difference between them is the number of CSN players. Numbers of CSN associated with each environment are shown in Tab. 1.

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.

Selecting paths: path modes. When a node wants to send a packet (when playing its own game) first, a path length (number of hops) is chosen and next the number of available paths of previously selected length is randomly generated. Path length is chosen according to predefined probabilities. The number of hops from the source node to the destination varies from 2 to 10. We use two path modes (referred later as *shorter (SP)* and *longer path modes (LP)*) differing in the probability of selecting particular number of hops leading from the source node to the destination. The first mode has a higher probability of selecting shorter paths while the second one has a higher probability of selecting longer paths. These probabilities are shown in Tab. 2. Additionally, for each path length a number of available alternate paths to the destination is available according to the probabilities shown in Tab.3. In general, the longer the path is, more likely less routes to the destination are going to be available.

Parameters of GA. The following parameters of GA are used: 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 taken as a result.

Evaluation cases. We examine the evolution of behavior among network participants in four cases. In the first two cases players are evaluated in only one environment while in cases 3 and 4 players are evaluated in all environments. For cases 1-3 paths are chosen according to the shorter path mode while for case 4 a longer path mode is used (Tab. 4).

Table 3. 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

Table 4. Parameters of evaluation cases.

	tournament environment	path mode
case 1	1 (CSN)	shorter paths (SP)
case 2	3 (30 CSN)	shorter paths (SP)
case 3	1-4	shorter paths (SP)
case 4	1-4	longer paths (LP)

6.2 Results: evolution of cooperation

We define *cooperation level* as a percentage of packets that originated by normal nodes and than successfully reached the destination. The results for all evaluation cases are shown in Fig. 4. When players play in the CSN free tournament (case 1), the level of cooperation reaches about 97%. On the contrary, when most of the population (60%) is composed of CSN (case 2, 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 are evaluated in all tournament environments (third and fourth evaluation case) the cooperation level reaches 38% and 54% respectively.

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 (case 1) 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. 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). 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.

Table 5. Cooperation levels measured independently for each environment for evaluation cases 3 and 4.

	Coop. lev in case 3	Coop. lev in case 4	CSN-free paths in case3	CSN-free paths in case4
TE1	99%	99%	100%	100%
TE2	66%	41%	66%	41%
TE3	28%	7%	29%	12%
TE4	19%	5%	20%	8%

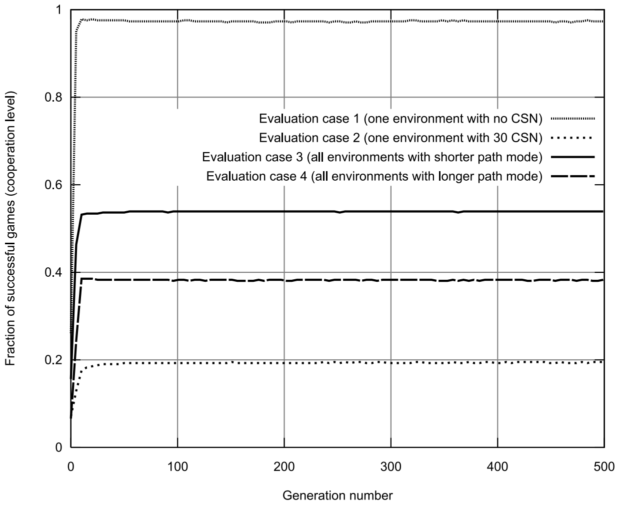


Figure 4. The evolution of cooperation.

Additional results for third and fourth evaluation cases are shown in Tab.5. These results are taken from the last generations (average value of all experiments). In the second and third columns one can see the cooperation levels measured independently for each tournament environment. The percentage of paths that did not contain CSN is shown in the last two columns. When sending packets, normal nodes try to avoid CSN by choosing paths with the best reputation. Its not always possible and depends mainly on two factors: number of CSN and a path length. The difference in cooperation level between evaluation cases 3 and 4 was related to the fact that in case 4 it was much more difficult to avoid CSN (longer paths mode used). For example, in TE4 only 8% paths did not contain CSN, while in case 3 it was 20%. This resulted in cooperation levels of 5% and 19% respectively (in TE4).

In Tab.6 one can see how *forwarding requests* coming from normal nodes and CSN were treated in the network in evaluation cases 3 and 4. We define a forwarding request as a situation in which a node is asked to forward a packet.

Table 6. Response to packet forwarding requests coming from normal nodes and CSN. Results for evaluation case 3 (EC3) and case 4 (EC4) are shown.

	Requests from normal players EC3, (EC4)	Requests from CSN
Req. accepted	77, (78)%	4, (3)%
Req. rejected by NP	0.23, (3.5)%	53, (49)%
Req. rejected by CSN	22, (18)%	43, (47)%

Table 7. Most popular strategies for the evaluation case 3 and 4.

Shorter paths (eval. case 3)	Longer paths (eval. case 4)
010 101 101 111 1	010 000 111 111 1
000 111 111 111 1	000 000 111 111 1
000 111 101 111 1	000 010 111 111 1
000 011 111 111 1	000 000 101 111 1
010 011 101 111 1	010 000 101 111 1

Around 77% of requests coming from normal nodes were accepted. Most rejections came from CSN (18-22%, depending on the evaluation case). The main difference was in rejection of packets originated by normal nodes. In the evaluation case 3 only 0.23% of such requests were rejected by normal nodes, while in the evaluation case 4 more than 3% packets were dropped. The acceptance percentage of requests coming from CSN was only around 4% in both cases.

6.3 Winning strategies

During the evolutionary process the initial randomly generated strategies evolved and as a result the cooperation level in the network decreased. Several strategies emerged in the last generations of our experiments. In Tab.7 five most popular strategies for both evaluation cases are shown. One can see that a decision against an unknown player (last bit) is to forward. As a result, new nodes can easily join the network and start sending own packets.

It is easier to analyze the global tends by looking at sub-strategies (strategy for each trust level). The results for the evaluation case 3 and 4 are shown in Tab.8 and Tab.9. Only sub-strategies that appeared in more than 3% of final populations are shown. For trust level 3 the same strategy dominated: "111 - always forward". The activity level of a source node is not taken into account at this trust level. For trust 2 strategies evolved for case 4 were slightly more

Table 8. Evolved sub-strategies for the case 3 (short paths).

Trust 0	Trust 1	Trust 2	Trust 3
010 (40%)	101 (33%)	101 (35%)	111 (99%)
000 (33%)	111 (25%)	111 (27%)	-
001 (11%)	011 (20%)	001 (21%)	-
011 (16%)	001 (19%)	011 (17%)	-

Table 9. Evolved sub-strategies for the case 4 (long paths).

Trust 0	Trust 1	Trust 2	Trust 3
000 (54%)	000 (53%)	111 (51%)	111 (99%)
010 (45%)	010 (34%)	101 (37%)	-
-	100 (8%)	001 (7%)	-
-	-	011 (5%)	-

cooperative: 93% of strategies said to forward packets for at least two activity levels, while for case 3 about 79% were in favor of cooperation in such case. This situation changes for the trust level 1. Strategies that evolved for the case 4 were by far less cooperative. The dominating strategy was ”000 - newer cooperate, 53%” while strategies that allowed cooperation (in only one activity level - 010 and 100) were present in 42% of last populations. For the case 3 cooperative level was only slightly lower than for trust 2. For the trust level 0 the evolved approach for the case 3 was slightly more cooperative than in case 4. In general, one can say that sub-strategies evolved in the evaluation case 3 were more cooperative than those from case 4 (with the exception of trust level 2, where the opposite is true). This was related to the fact that in the case 4 longer paths were used and as a result the probability of successfully sending a packet was lower (since it was more likely that a CSN would appear in the path). In such case normal nodes were becoming less cooperative against requests coming from nodes with lower trust levels in order to force the cooperation. All forwarded packets coming from CSN were forwarded at the beginning of the tournament, at the time when CSN were seen as an unknown nodes. As the reputation of CSN decreased with time, such nodes did not manage to send any more packets.

7 Conclusions

In this paper, we have proposed a strategy driven approach to enforce cooperations in ad hoc networks. It uses the notion of trust and activity. We have found such strategies using a game based model of the network and GA. Our model of the network includes a simple reputation collection mechanism. Experimental results showed that the proposed cooperation enforcement approach 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. The exact evolution of strategies depends on the network conditions being used at their evaluation. To achieve best results one should know what kind of network are those strategies target.

References

[1] S. Buchegger and J.-Y. L. Boudec. 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)*, pages 131–140, 2000.

[2] 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)*, pages 226–236, 2002.

[3] S. Buchegger and J.-Y. L. Boudec. Self-policing mobile ad-hoc networks by reputation systems. *IEEE Communications Magazine, Special Topic on Advances in Self-Organizing Networks*, 43(7), July 2005.

[4] L. Feeney and M. Nilsson. Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. In *Proc. The IEEE Conference on Computer Communications (INFOCOM'01)*, pages 1548–1557, 2001.

[5] S. Giordano and A. Urpi. *Mobile Ad Hoc Networking*, chapter 13. Wiley-IEEE Press, 2004.

[6] M. Ilyas and I. Mahgoub, editors. *Mobile Computing Handbook*. Auerbach Publications, 2005.

[7] L.Buttyan and J.-P.Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. Technical Report DSC/2001/001, Swiss Federal Institute of Technology, 2001.

[8] L.Buttyan and J.-P.Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8(5), Oct. 2003.

[9] 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)*, pages 255–265, 2000.

[10] 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)*, pages 107–121, 2002.

[11] P. Michiardi and R. Molva. Simulation-based analysis of security exposures in mobile ad hoc networks. In *Proc. European Wireless Conference*, 2002.

[12] 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)*, pages 2637–2644, 2005.

[13] C. Perkins, editor. *Ad Hoc Networking*. Addison-Wesley, 2001.