Spontaneous Cooperation in Multi-domain Sensor Networks

Levente Buttyán, Tamás Holczer, and Péter Schaffer

Laboratory of Cryptography and System Security (CrySyS),
Department of Telecommunications,
Budapest University of Technology and Economics, Hungary
{buttyan, holczer, schaffer}@crysys.hu

Abstract. Sensor networks are large scale networks consisting of several nodes and some base stations. The nodes are monitoring the environment and send their measurement data towards the base stations possibly via multiple hops. Since the nodes are often battery powered, an important design criterion for sensor networks is the maximization of their lifetime. In this paper, we consider multi-domain sensor networks, by which we mean a set of sensor networks that co-exist at the same physical location but run by different authorities. In this setting, the lifetime of all networks can be increased if the nodes cooperate and also forward packets originating from foreign domains. There is a risk, however, that a selfish network takes advantage of the cooperativeness of the other networks and exploits them. We study this problem in a game theoretic setting, and show that, in most cases, there is a Nash equilibrium in the system, in which at least one of the strategies is cooperative, even without introducing any external incentives (e.g., payments).

1 Introduction

Multi-hop wireless sensor networks will be the near future's most powerful monitoring applications. These networks contain a large number of sensor nodes and some base stations which are collecting the information that the sensors measure. Sensor networks can be used for environmental monitoring (e.g., forest fire or earthquake detection), tracking of cars or material (e.g., freight transport, traffic monitoring), or monitoring the state of buildings [1].

An important design criterion for sensor networks is the minimization of the sensors' energy consumption. The reason is that sensors are often battery powered, and it is impractical, or in some cases, even impossible to change or recharge their batteries once they have been deployed. It is known that the energy consumption of transmitting a data packet is a super-linear function of the distance of the transmission. Practically, this means that, as far as energy consumption is concerned, it is more advantageous to transmit a packet in several small hops than to transmit it in a single large hop. Hence, if there are numerous sensors near to each other then they could transmit the packets together and by doing so, they can increase the lifetime of their batteries radically.

R. Molva, G. Tsudik, and D. Westhoff (Eds.): ESAS 2005, LNCS 3813, pp. 42–53, 2005.

In today's research of sensor networks it is generally assumed that all the sensors and base stations belong to one authority that can control the whole network. In this paper, we depart from this common assumption, and consider sensor networks that are deployed at the same physical area, but controlled by different authorities. In such a situation, the sensors that belong to one authority may reduce their transmission energy even further if their packets are forwarded by sensors that belong to another authority; an act that we call *cooperation*. There is a risk, however, that the sensors belonging to the other authority are not willing to help and they drop the foreign packets.

We study this problem in a game theoretic setting. The main question we are interested in is the following: Can cooperation emerge spontaneously in multi-domain sensor networks based solely on the self-interest of the nodes (or more precisely the authorities to which the nodes belong)? To put it in another way: Is the objective of increasing the lifetime of the network enough to foster cooperation between co-located sensor networks? Our analytical and simulation studies presented in this paper show that in most cases, the answer to these questions is affirmative.

The rest of the paper is organized as follows. In Section 2, we show some cheering analytical results on a simplified model. In Section 3, we extend the simple model to a more realistic one, and we present our simulation results. In Section 4, we report on some related work. Finally, we conclude in Section 5.

2 Simplified Model

We start to study the problem of spontaneous cooperation in a simplified model. We assume that there are only two sensor networks that co-exist at the same physical location, and each of them consists of a single base station and a single sensor. The placement of the base stations and the sensors is illustrated in Figure 1.

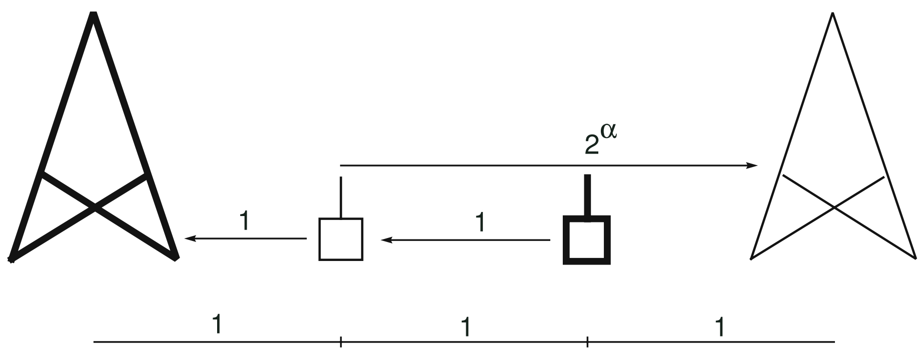

Fig. 1. Simple network

Now, we describe the operation of this simple system. We assume that time is divided into discrete time slots. In each time slot, each sensor wants to send a single data packet to its own base station, which contains its measurement data. We also assume that data packets are equal in size.

The packet can be sent to the base station directly in a single hop, or via the other sensor in two hops. Thus, at the beginning of each time slot every sensor has to decide the following:

– whether to request the other sensor to help in forwarding its own packet, or not, and
– whether to help in forwarding the other sensor's packet if it requests help, or not.

The decision made by the sensor defines its *move* in the time slot. Hence, we have four possible moves, each of which is denoted by a pair of letters as follows:

CC means that the sensor tries to get help from the other sensor and helps if the other sensor requests it;
CD means that the sensor tries to get help from the other sensor but it refuses to help if the other network requests it
DC means that the sensor does not ask for help, but it sends its packet directly to its base station, however, it helps if the other sensor requests forwarding;
DD means that the sensor does not ask for help from the other sensor, and does not help if the other sensor request forwarding.

In fact, C stands for cooperation and D stands for defection, and the first letter of the move defines how the sensor behaves concerning its own packet, and the second letter defines how it behaves when the other sensor's packet is concerned. For instance, making the move CD means that the node tries to cooperate when sending its own packet, but defects when the other sensor asks it to forward its packet.

Each pair of moves has a cost for both sensors, which are shown in Table 1. The costs are related to the energy consumption of the sensors and they are determined as follows:

– asking the other sensor to forward the packet has a unit cost, because this only requires to send the packet to a unit distance.
– forwarding the other sensor's packet also has a unit cost for similar reasons;
– sending the packet directly to the base station has a cost of 2^α, where α is the path loss exponent with usual values between 2 and 5, because this requires to send the packet to a distance of two units;
– dropping a packet has no cost.

We note that we are aware of the fact that in reality the cost of communication does not only depend on the distance, but there are also fix costs associated with the reception and the transmission of packets. In this simplified model, we set aside these fix costs.

The cells of Table 1 contain not only the costs for the two sensors, but also indicators of success, where 1 means that the packet reached the base station

Table 1. Costs and successes in the simple network (cost of row player, cost of column player; success of row player, success of column player)

	CC	CD	DC	DD
CC	$2,2$;$1,1$	$2,1$;$0,1$	$1,1+2^{\alpha}$;$1,1$	$1,2^{\alpha}$;$0,1$
CD	$1,2$;$1,0$	$1,1$;$0,0$	$1,1+2^{\alpha}$;$1,1$	$1,2^{\alpha}$;$0,1$
DC	$1+2^{\alpha},1$;$1,1$	$1+2^{\alpha},1$;$1,1$	$2^{\alpha},2^{\alpha}$;$1,1$	$2^{\alpha},2^{\alpha}$;$1,1$
DD	$2^{\alpha},1$;$1,0$	$2^{\alpha},1$;$1,0$	$2^{\alpha},2^{\alpha}$;$1,1$	$2^{\alpha},2^{\alpha}$;$1,1$

(success) and 0 means that it did not (failure). As an example let us consider the pair of moves $CC-CD$. In this case, the first sensor tries to send its packet via the other sensor, but the other sensor will drop it. On the other hand, the other sensor's packet will be sent via the first sensor to the base station successfully. Hence, the cost of the first sensor is 2 (1 for asking for forwarding its own packet and 1 for forwarding the other's packet), and the cost of the other sensor is 1 (the cost for asking for forwarding). Moreover, the first sensor records a failure, while the other one records a success.

We assume that the sensors record the results (success or failure) of the last few time slots in a buffer that we call *history*. One can think of the history as a binary vector of a fixed length. We assume that each sensor's next move is a function of its history. We call this function the *strategy* of the sensor.

Here we make an important restriction on the strategy space (set of possible strategies). We assume that each sensor wants to keep the weight of its history (i.e., the number of successful slots in the recent past) above a threshold, which we call the *weight threshold*. Intuitively, this means that we do not want to allow too many unsuccessful slots in the history, because that would mean that the base station does not receive measurement data with high enough rate (characteristic to the application). Therefore, when the weight of the history approaches the weight threshold, the sensor is not allowed to make risky $C*$ moves (i.e., CC and CD), but it is required to send its data directly to the base station (i.e., to make a $D*$ move). This situation is called the *constraint state*. Using strategies that suggest $D*$ moves in the constraint state guarantees that the weight threshold is never violated.

Note that a longer history with a lower weight threshold results in a system with more freedom. On the other hand, a shorter history with a higher threshold results in a much stricter system.

Each sensor has some initial battery B. In each time slot, the battery levels of the sensors decrease. The amount of this decrease depends on the pair of moves made in the time slot and their associated costs. When a sensor runs out of its battery, it dies. The other sensor can continue to send data to its base station if it still has some battery.

Note that the above mentioned concepts describe together an extensive game, where the players are the sensors, the possible moves (made simultaneously by both players) in each round (except for the constraint states) are CC, CD, DC, DD, the information sets are defined by the content of the histories, and the set of strategies are the functions that assign a move to every possible history with

Table 2. Best lifetimes with two-step strategies (lifetime for row player; lifetime for column player), B initial battery, ρ weight threshold, α path loss exponent, $\epsilon_{1,2}$ payoff from transient states

	CC/DD	CD/DD
CC/DD	$\frac{B}{2}$; $\frac{B}{2}$	$\frac{B}{\rho 2^\alpha+(1-\rho)}$; $\frac{B}{\rho 2^\alpha+(1-\rho)}+\epsilon_1$
CD/DD	$\frac{B}{\rho 2^\alpha+(1-\rho)}+\epsilon_1$; $\frac{B}{\rho 2^\alpha+(1-\rho)}$	$\frac{B}{\rho 2^\alpha+(1-\rho)}+\epsilon_2$; $\frac{B}{\rho 2^\alpha+(1-\rho)}+\epsilon_2$

the restriction that only $D*$ moves are assigned to a history that represents a constraint state. The game ends when both sensors run out of their batteries. The payoff of a player is its lifetime, which is represented by the number of rounds it survived. Lifetime is a good payoff function since the authorities want to run their network as long as possible with some constraints on their success.

Once we have a game, we can look for Nash equilibria with the highest possible lifetime. A Nash equilibrium is a strategy pair such that none of the players can increase its utility by unilaterally changing its strategy. It is quite reasonable to choose one of these Nash equilibria as an operating point in real systems. If there are more then one Nash equilibria, the equilibrium with the highest lifetimes is chosen.

In order to make the analysis feasible, we further restrict the strategy space. Let us consider first the *two-step strategies*. These strategies suggest a fix move if the player is not in a constrained state (independently of the actual weight of the history), and another fix move if the player is in a constrained state. A two-step strategy is denoted by m/m', where m is the move chosen in an unconstrained state and m' is the move chosen in a constrained state. For instance, the strategy CC/DD selects CC in an unconstrained state and DD in a constrained state. Therefore, we have eight two-step strategies, because in a constrained state only $D*$ moves are possible.

We performed an exhaustive search on this strategy space (there are $8\times8 = 64$ pairs of strategies to consider), and looked for Nash equilibria. We found that CC/DD and CD/DD dominate the other strategies. The CC/DD strategy is a cooperative strategy, while the CD/DD is an uncooperative one. By eliminating the dominated strategies, we get a reduced game. The lifetimes for the sensors in this reduced game are shown in Table 2, where ρ denotes the weight threshold, and B denotes the initial battery level. ϵ_1 and ϵ_2 comes from transient states like starting and ending the game. There are two Nash equilibria: $(CC/DD, CC/DD)$ and $(CD/DD, CD/DD)$. The first one results in full cooperation, while the second one results in full defection. However, if $\rho > \frac{1}{3}$ (and $\alpha \geq 2$ which is a fundamental condition in our model), then the cooperative equilibrium results in a higher lifetime for both players.

A more interesting class of strategies are the *weight aware strategies*. These strategies choose the next move as a function of the weight of the history. Thus, a weight aware strategy can be represented as $m_1/m_2/\ldots/m_k$, where m_1 is the move that is chosen when the weight of the history is maximal, and m_k is chosen when the weight of the history is just above the weight threshold. k is a parameter whose value depends on the history size and the value of the weight threshold. This class contains more complex and more reactive strategies.

After running 20 different exhaustive simulations, with different parameter sets, we found that the strategy that achieves the best Nash equilibrium is always the same: $(CD/CD/\ldots/CD/CC/DD)$. We call it the smart strategy. The smart strategy tries to ask for help in the first steps (the CD moves, which are cheap moves), but provides help (the CC move before the DD move) only in a state when the weight threshold is nearly violated in the hope that its nice behavior will be reciprocated. In other words, the smart strategy first tries to exploit the other. If this is successful, then it will never cooperate. However, if the other strategy is not exploitable, then it will change to a cooperative behavior. In the long run, the strategy keeps the actual weight of the history near to the weight threshold, which means that it cooperates only as much as necessary. This turns out to be a very effective behavior to save battery and leads to a rational cooperation.

In summary, we can see that in the simplified model, which contains two base stations and two sensor nodes, cooperative Nash equilibria exist based on smart strategies that try to optimize the amount of cooperation. In the next section we will investigate if the same is true in a more general model.

3 Generalized Model

After the cheering results of the simplified model in Section 2 we have examined much bigger and more complex systems. We have developed a simulator that corresponds to the model described in the first part of Section 2 with some extensions.

The generalized model uses many sensors per authority randomly placed on the playground with uniform distribution. The possible moves are the same as those in the simplified model, but in the generalized model each pair of moves has a cost that depends not only on the distance of the transmissions and the path loss exponent α, but also on some fix costs associated with the sending and receiving of packets. The fix cost of sending and the fix cost of receiving are constant values, which represent the energy consumption for connecting to the communication channel and to process the packets.

The principle of routing in the model is finding the minimum energy path towards the base station [9]. This means that every node has to forward on the path which has the minimum energy cost among all the possible paths. Every node maintains three paths: one in its own network (for the defective moves) and two in the global network (i.e., where all the nodes are possible forwarders). The global network paths are maintained for being able to make cooperative moves. The two distinct cooperative paths are towards the two base stations. These three paths can be the same depending on the placement.

Both networks have a threshold value (success threshold) which defines the minimum number of packets that the base station has to receive in each time slot, and the time slot is considered successful only if at least that number of packets reach the base station. The lifetime of a network is the total number

Table 3. Parameters for the simulations (the parameters are motivated in the *example* and in [7])

Parameter	*Value*
Number of sensors per domain	10-20-40 (20)
Distribution of the sensors	uniformly random
Area size	100x100 m
Position of the base (common base)	[50,50]
Position of the bases (separate bases)	[45,50] and [55,50]
Initial battery	10 million units
Reception fix cost	3000 units
Sending fix cost	2000 units
Success threshold	0.7-0.8-0.9 (0.8)
Weight threshold	0.6
History length	5
Energy drop-off (α)	2-3-4 (3)

of time slots that elapsed until the weight of the history becomes zero. The objective of the game is to reach the best possible lifetime under the constraint that the weight threshold of the history has to be respected.

Example: In an office building it is usual to deploy temperature and movement sensors. The temperature sensors measure the actual temperature and forward it to the air conditioning system. The movement sensors gather information about which zone is visited or abandoned and forward it to the security system. The two systems ask for information regularly (once in every second) but it is not crucial to get the information in every time slot. The temperature can be controlled and the security can be guaranteed with enough accuracy if some of the measurements are successful (let us say three out of the last five). The systems can work properly if they get enough measurement data in a time slot. While the sensors are usually deployed redundantly a given proportion can execute the task (let us say 80 % of the sensors). If the given proportion of data is arrived to the control systems, then the missing information can be deduced.

We have investigated two main type of scenarios. In one of them (common base scenario), there is a single common base station that collects the information from all of the nodes (independently from the authority they belong). In the other (separate base scenario), both networks have their own base stations. In the common base model, the base station is placed in the middle of the playground, while in the separate bases model, the base stations had the same distance from the theoretical middle of the playground.

We performed 100 simulation runs for each parameter setting with different topology. The concrete values for the simulations are shown in Table 3. The values in parenthesis are the defaults. For each run we made an exhaustive search in the strategy space to find the best strategy pairs (i.e., those that form a Nash equilibrium and generate the highest lifetimes).

In the extended model, it is not so easy to determine which equilibrium is a cooperative equilibrium. Two strategies can act in a cooperative way in case

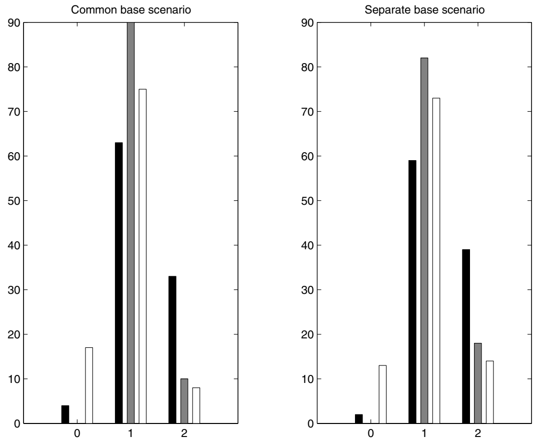

Fig. 2. Distribution of equilibrium classes (number of nodes per domain = 10 (black), 20 (gray), 40 (white))

of one topology and in an uncooperative way in case of another topology. In other words, the topology and the strategies both can influence the cooperation. Therefore, we classified the equilibria into the following three classes:

- *Class 0:* If the networks play strategies that form this type of equilibrium, then neither of them ever forwards a packet for the other. (no cooperation)
- *Class 1:* If the networks play strategies that form this type of equilibrium, then one of them forwards some packets for the other. (semi cooperation)
- *Class 2:* If the networks play strategies that form this type of equilibrium, then both of them forward some packets for the other. (full cooperation)

If the game had more than one best Nash equilibria, then we considered the most cooperative ones (i.e., those that have the highest class number).

The simulation results are shown in Figures 2, 3, and 4. In each figure, the left hand side chart shows the results of the common base scenario, and the right hand chart shows the results of the separate base scenario. On the x axis, we show the equilibrium classes (0, 1, 2), and on the y axis, the percentage of simulations where the best Nash equilibria fell in a given equilibrium class.

Figure 2 shows how the distribution of the different equilibrium classes depends on the number of nodes. One can see that in most cases the best Nash equilibria result in some kind of cooperation, although semi-cooperation has a higher probability than full cooperation.

Figure 3 shows how the distribution of the different equilibrium classes depends on the path loss exponent α. If α is high, then full cooperation is the best choice, because it costs a lot of battery energy to send to a far sensor. If full cooperation occurs, then the average sending distance is smaller, which is very advantageous when the path loss exponent is large.

Figure 4 shows how the distribution of the different equilibrium classes depends on the success threshold. One can see that the success threshold does not

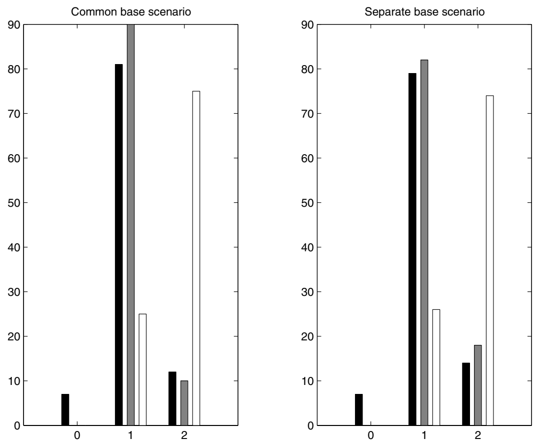

Fig. 3. Distribution of equilibrium classes ($\alpha = 2$ (black), 3 (gray), 4 (white))

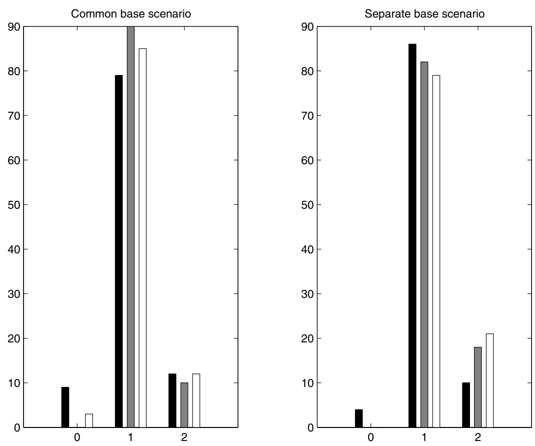

Fig. 4. Distribution of equilibrium classes (success threshold = 0.7 (black), 0.8 (gray), 0.9 (white))

have much influence on the distribution. If the success threshold is higher, than a little more fully cooperative Nash equilibria occur, but the success threshold seems to be a not as important parameter as the path loss exponent or the number of nodes.

As we have seen above, when co-located sensor networks are allowed to collaborate in the packet forwarding effort, some form of cooperation can emerge spontaneously, by which we mean that in the best Nash equilibria, at least one of the networks forwards some packets on behalf of the other network. It is clear that this cooperative behavior is more advantageous (meaning results in a longer lifetime) for the cooperating network than a defective behavior, given the

Table 4. Average gain in lifetime in the common base scenario and in the separate base scenario

Non-default parameter	Separate base scenario	Common base scenario
-	6.5%	6.1%
$n = 10$	15.5%	15.6%
$n = 40$	1.5%	0.6%
$\rho = 0.7$	4.4%	3.2%
$\rho = 0.9$	8.7%	7.8%
$\alpha = 2$	1.9%	2.2%
$\alpha = 4$	34.7%	31.0%

strategy of the other network, since a Nash equilibrium consists of best response strategies. In order to quantify this advantage, we performed the following experience. For each simulation run[1], we determined (i) the networks' lifetimes when both networks ignore each other and use only their own nodes for forwarding, and (ii) the networks' lifetimes in the best Nash equilibrium when the networks are allowed to collaborate. In both (i) and (ii), we took the smaller lifetime value (i.e., the lifetime of the network that lives shorter), and we computed the ratio of the values obtained. Finally, we averaged the ratio values over the 100 simulation runs (for each parameter setting). One can interpret the result of this computation as the average gain in lifetime when the networks are allowed to collaborate compared to the case when they operate independently from each other.

The results are shown in Table 4. Each row of the table belongs to a particular parameter setting, where all but one of the parameters have the default values shown in Table 3, and the first cell of the row shows the non-default parameter value. The second and the third columns of the table contain the average gain in lifetime in the common base and in the separate base scenarios, respectively. As we can see, the average gain in lifetime can be as high as 34% in the common base scenario and 31% in the separate base scenario when $\alpha = 4$.

4 Related Work

There are several articles that address the problem of cooperation in ad hoc networks (see e.g., [2, 3, 6]). However, these papers deal with the question of how cooperation can be encouraged by the introduction of some incentives (e.g., payments or reputations). Thus, indirectly, all these papers assume that cooperation cannot emerge by itself, but it must be stimulated. In contrast to this, we study spontaneous cooperation in this paper.

Cooperation without incentives has been studied in [8], but there are important differences between that paper and our work. First, the authors of [8] study cooperation in ad hoc networks, while we are considering cooperation in sensor

[1] Recall that for each parameter setting, we had 100 simulation runs with different topologies.

networks. Second, in [8], the nodes are collected into energy classes, which represent the heterogeneity of the nodes, whereas in our model, the nodes have equal resources. Finally, in [8], randomly chosen pairs of nodes communicate with each other, while in our case, every sensor communicates with the base stations.

In [4], the authors study the conditions under which cooperation (without incentives) is the best strategy in static ad hoc networks. Unlike the model in [8], their model takes into account the topology of the network. The main difference between [4] and our work is that energy consumption of the nodes is not considered in [4], whereas it has a central role in this paper. In addition, in [4], the nodes communicate with each other in a peer-to-peer manner, while we are considering sensors communicating with base stations.

The paper of Félegyházi et al. [5] stands most near to our work. In that paper, the authors investigated exactly the same problem as we do in this paper, nonetheless, their model and simulator is remarkably different from ours. First, in their model the payoff received after a successful round (in which enough sensors managed to send their data to the base station successfully) is a subjective value that represents the importance of a successful round for the given authority. In our case, there are no payoffs after the rounds, but instead the lifetime of the network is the payoff received at the end of the game. Second, in this paper, we introduce a constraint on the available moves after a certain number of unsuccessful rounds. This guarantees that a minimum level of quality of service is maintained in the network (i.e., base stations do receive data from sensors at least with a predefined rate), which indeed is a very important practical requirement. In [5], no lower bound on the success rate is guaranteed. Finally, we define the notion of lifetime differently: for us a network is dead when a certain percentage of its nodes die, while in [5], the death of the first node means the death of the whole network.

5 Conclusion and Future Work

In this paper we examined if cooperation is possible without the usage of incentive mechanisms in multi-domain sensor networks. First, we analyzed a simple network consisting of two sensors and two base stations, and found that in this simple setting, the best Nash equilibria (where the lifetime of the sensors is the highest) consist of cooperative strategies. Then we generalized our model from two nodes to many nodes, and used a two dimensional layout. We classified equilibria into non-cooperative, semi-cooperative, and fully cooperative. We found that in most cases, the best Nash equilibria belong to the cooperative classes. Especially, in the case when the path loss exponent is large, full cooperation is the best strategy.

In terms of future work, we intend to study more in detail how the distribution of the different equilibrium classes depend on the parameters the density and the topology. If this dependence can be characterized precisely, then it becomes possible to *engineer cooperation* by fine-tuning the parameters and adjusting the topology (if the application permits that) appropriately.

Acknowledgements

The authors are thankful to István Vajda for the helpful comments and discussions. The first author is also grateful to Jean-Pierre Hubaux and Márk Félegyházi for initial discussions on the simplified model presented in this paper.

This work has partially been supported by the Hungarian Scientific Research Fund (T046664). The first author has been further supported by IKMA and by the Hungarian Ministry of Education (BÖ2003/70).

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, Vol. 38, No. 4, pp. 393-422, March 2002.
2. S. Buchegger and J-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation Of NodesFairness In Dynamic Ad-hoc NeTworks). In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 80-91, June, 2002.
3. L. Buttyán and J.-P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8(5), October 2003.
4. M. Félegyházi, J.-P. Hubaux, and L. Buttyán. Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks. *to appear in IEEE Transactions on Mobile Computing*
5. M. Félegyházi, J.-P. Hubaux, and L. Buttyán. Cooperative Packet Forwarding in Multi-Domain Sensor Networks. In *Proceedings of the First International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS)*, March 2005.
6. P. Michiardi, R. Molva. CORE: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. In *Communication and Multimedia Security 2002*, September 2002.
7. Rahul C. Shah, Jan M. Rabaey: Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2002.
8. V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Cooperation in Wireless Ad Hoc Networks. In *Proceedings of IEEE INFOCOM'03*, San Francisco, Mar 30 - Apr 3, 2003.
9. F. Ye, A. Chen, S. Lu, and L. Zhang. A scalable solution to minimum cost forwarding in large sensor networks. In *Proceedings of the Tenth International Conference on Computer Communications and Networks*, pp. 304-309, 2001.