Incentive scheme for voluntary and autonomous cooperation in distributed networks

LÁSZLÓ HARRI NÉMETH, RÓBERT SZABÓ

Budapest University of Technology and Economics,
Department of Telecommunication and Media Informatics
{nemethl, szabo}@tmit.bme.hu

Reviewed

Keywords: ambient networks, voluntary cooperation, game theory, peer-to-peer, distributed networks, promise theory

Today's communication networks are becoming increasingly dynamic in the sense that they do not have fixed infrastructure, or the configuration of infrastructure-based networks continuously changes. Examples include distributed access networks using WLAN technology, ad-hoc networks, ambient intelligence networks [1,2] or sensor networks. These networks have considerable independence and autonomy and they might frequently act in a selfish manner. Autonomy means that such networks have no central administrative or management principles that would determine their operation.

1. Introduction

In this kind of environment a distributed architecture becomes necessary for the voluntary cooperation of autonomous networks, which controls the cooperations [3]. No central confidence of infrastructure is to be assumed.

Promise theory is a graph theoretical framework, which simplifies the understanding of complex relationships in a network environment that requires compliance with diverse restrictions [3], [4]. According to the basic idea, fully autonomous nodes connect with each other through promises. The cooperative nodes organize groups. Every single promise implies a restriction on the behavior of the promising node.

In large scale distributed networks the components of the network share their services and network-management functions with each other. However, it is not a good choice for the nodes to share all their services with the others.

Each network node needs services from other nodes. If a node only requires services, but does not serve the requests of the other nodes, that means that this node behaves in a selfish way. In order to terminate such behavior in the network and motivate the nodes to cooperate, one may use several kinds of techniques. The principle of these solutions is that one rewards the generous nodes and punishes the selfish ones. If a node receives a reward, it is more likely that its requests will be served by other nodes. If a node receives a punishment, it will be less likely that such node is served. The game theory approach is the most suitable way to model the above described method. The most fitting game for this model is the general prisoner's dilemma. In order to make a decision whether or not to serve a certain service request, the nodes must store some kind of information about the behavior of the other nodes to make the system work.

Behavioral information and history can be stored basically in two ways: by shared history or by private histories [5]. The two storage methods have different drawbacks, in case of storage in a commonly used area a node may send false recommendation related to another node, that is to say it lies about another node and this can ruin the cooperation. To store information in a common field a distributed data-storage method is also required, e.g., by way of distributed hash tables. In case of a large number of nodes individually stored history results in infeasible memory requirements, so the above mentioned method can be used only to a limited extent.

Description of resource sharing by game theory models is a widely researched field, especially since the P2P file-distributed networks have become popular. Several approaches have been developed to motivate the participants of the network to share their resources. In these reputation-based incentive systems the nodes have a utility value, which they want to increase and maximize during their operation. The calculation of the utility value is based on the resource sharing level of the node and the extent of the utilization of other nodes. One of the most comprehensive studies in this field was conducted by Ion Stoica and his team [5], but many other valuable publications were made on this topic. These researches differ in several ways, e.g., the type of the game theory used to analyze the system. Ion Stoica and his team used an asymmetric model with two participants, while for example Philippe Golle conducted his analysis with a multi-agent reinforcement learning model [6].

Existing game theoretic descriptions are based on P2P principle, i.e., any participant may contact any other participant to request or to perform a service. The solution, described in this paper, differs from these approaches in the fact that a topological network is used to deliver service interactions as the chain of physical, node-to-node interactions. As an example, in ambient networks [1] the nodes have only a limited coverage area, so they can communicate directly only with their neighbours. Consequently, routing is required in the network, and a service request goes through several nodes. Therefore, upon a service request three different kinds of nodes participating in the process can be distinguished: an initiator node, which requests the service, a target node, from which the service is requested and optionally some transport nodes, which transmit the requests and the answers. Naturally, a node may request service from its direct neighbor. In this case the transport nodes are left out.

2. System Model

Game theory is a branch of mathematics trying to answer the question: which behavior is reasonable in a situation when the results and effects of a participant's decisions are also affected by other participants' decisions. The description of a game basically requires the specification of three elements: the players, the strategies and the payments, or in other word, payoffs.

Players are the participants of the game, who want to maximize their payoffs. By strategy we mean the behavior of the players, namely, the kind of decisions the players may come to. By payoff we mean the player's utility diagram, the value, which may be recorded to the player's credit at the end of the game. This value depends on the strategy the player has chosen and the strategies of other players. Since the player is rational, he wishes this utility value to be as high as possible. To reach this, the player has to consider the other players' decisions or decision options, as well as his own payoffs in relation to the above. There are several kinds and classifications of the games, e.g., normal form or extensive form games, symmetrical or asymmetrical, zero sum or non-zero sum games. The easiest way to specify a normal form game is the payoff matrix. This matrix shows the players, the strategies and the payoffs.

In order to understand the operation of the system first we should discuss the prisoner's dilemma. There are many versions of this game. The basic idea is that two prisoners, suspected of a crime are imprisoned in separate cells. They have the same options: if a prisoner testifies against the other he will be released and the other is punished to 10 years' imprisonment. If neither of them testifies, they receive 6 months each, if both of them testify, they get 6 years each. The prisoners must not communicate with each other hence they are unable to cooperate (non-cooperative game). Thus the duration of the punishment may be considered as a kind of negative utility we wish to minimize. The payoff matrix of the above described game is illustrated in *Table 1* (in a cell the first number is the payoff of the Player 1 /utility/ and the second number belongs to the Player 2).

The difference between the original and the generalized game is that several restrictions and rules were defined for the payoff values. Based on the above various prisoner's dilemma games may be described which fulfill these rules. We do not discuss these in details.

For asymmetric games, like a client-server interaction, the classical prisoner-dilemma game can be extended as shown in *Table 2*. The numbers in Table 2 indicate the utility and payoffs of certain players. This game is played many times by the participants of the network and the scores are cumulated. In the very case of *Table 2*, when a node requests a service from another node, two events can occur: the node either serves the client node's request, in which case the server node receives -1 point and the client receives 7 points, or the server rejects the request, so each of them receive 0 points.

The players may have 3 different strategies: always cooperate, always defeat (never cooperative) and to be reciprocative. The first strategy means that the node fulfils every inbound request unconditionally. The second strategy is the opposite of the first: the node never fulfils any request.

The information about the behavior of the requesting nodes stored by the nodes becomes relevant in the reciprocative strategy. Using this strategy, the decision of a node whether to serve the requesting node or not, is based on some stored information. During the pro-

Table 1.
Payoff matrix - Classical prisoner's dilemma game

[years]		Player 2: Do not testify	Player 2: Testify
Player 1	Do not testify	-0.5 / -0.5	-10 / 0
	Testify	0 / -10	-6 / -6

Table 2.
Payoff matrix for the game played by nodes

		Server player: Perform service	Server player: Do not make attention
Client player	Request service	7 / -1	0 / 0
	Do not request service	0 / 0	0 / 0

cedure the nodes collect their scores (or loose them) game by game. Each node compiles statistics about which strategy has been the most profitable for them. If a node considers that another strategy would be more profitable than the one it currently uses, it changes strategy. In this case the identifier of the node also changes, so the information about this node stored by the others loses its relevance. (A traitorous node is an exception to this rule, since it keeps its identifier even if it changes strategy. This issue will be discussed later.)

A node may increase its utility not only by serving, but also transferring requests. The value of transferring requests is identical to the value of serving a request. For the requesting node, it is practically transparent who provides the service. The transport of the services is implemented in a way of a routing mechanism. The nodes are aware of the routes through which they can reach other nodes, thus they know which of their neighbors they have to turn to first if they request service from a specific node.

The following question may arise: why would a server node perform services upon a client's request if this results in a negative score for such a node? The answer lies behind the previously described private history stored by the nodes. If a node does not perform services to the other nodes, then sooner or later its requests will be declined as well, so it would be unable to collect scores. This means that in the long term it would not profit from such operation. Performing or not performing services also depends on the relationship of the serving node with other nodes, since as it will be subsequently shown, in certain cases a node may prefer the non-cooperative strategy to the other strategies.

Additionally, a traitorous type of node has also been introduced into the system with the following operation: When this kind of node changes strategy its identifier remains unchanged and the information stored by the other nodes about it also remains valid. Theoretically, a node like this may cooperate with every other node in the first part of the operation, while it refuses to serve any requests in the second part, since due to the high score collected in the first part its requests will most likely be served by the other nodes, which conduct reciprocative strategy. We have examined the operation of the system also in the presence of such of nodes.

During the operation of the system the nodes also store information on the nodes they had previous connections with. The nodes "remember" the clients which had requested services from them. They use this memory when they act as client nodes and they are more likely to request services from those nodes which had already requested services from them. Thus, a node can return a service by performing a request for the other node.

Due to this principle, during the simulation the behavior of the network converges to a relatively stable condition, and although some strategy changes may occur at the last stages of the simulation, no drastic u-turns take place, thus the system becomes stable.

3. Numerical Results

The examination of the above described system was conducted by way of simulation. The simulation was divided into cycles and every node requested service from another node in each cycle, that is, they played the above described game. The game goes through the entire service path, that is, the path on which the performance of services takes place between the client and server nodes. Each simulation contains 1,000 cycles. The examination of the operation of the system was conducted with respect to several cases.

The storing method of histories stored about the nodes was examined both from short-term and long-term respect. If we store such information only for a short-term, this means, that a node may quickly "whitewash" itself, so the system is forgiving, however this behavior might be disadvantageous for the other nodes subsequently. However the storage of long-term history requires extra memory and for satisfactory operation an efficient search must be implemented as well. These two cases we examined in relation to private and shared history.

During the simulation we examined the operation of a network containing 100 nodes. The nodes were randomly positioned, so the topology developed in this way is also random. We examined which strategy is the most profitable for a certain node. The use of a certain strategy depends on several circumstances, e.g., on the position of the node in the network (whether it has a few or a lot of neighbours) or the strategies its neighbours use. At the beginning of the simulation the strategies were randomly distributed between the nodes in the same proportion, thus 1/3 of the nodes were cooperative, 1/3 were defective (non-cooperative) and 1/3 played the reciprocative strategy. In general it can be established that in most cases the cooperative and the reciprocative strategies were the most profitable ones. However, in certain cases, in some parts of the network the non-cooperative behavior became more popular. The system acted differently if the presence of traitorous nodes were also allowed, the proportion of which was set to 25%.

During the simulation the network approached to a stable state. This means that the majority of the nodes were not interested in strategy change and the frequency of strategy changes decreased in the entire network. The diagrams show the number of nodes that use a certain strategy in a certain simulation cycle, but it does not indicate which specific nodes use such strategy, so we cannot find out if the strategy was used by the same nodes or some others. To demonstrate the aforementioned characteristics, we prepared a network topology in each simulation cycle, which indicates each strategy by different color.

By examining these topologies, we came to the conclusion that the bulk strategy changes took place at the beginning of the simulation and at further stages no substantial changes happened. The examination of this process provides an opportunity to focus on the distribution of the strategies depending on position within the topology.

Fig. 1. shows the topology reached by the end of the simulation in case of various simulation scenarios. It is obvious that in the case of the presence of traitorous nodes, the number of the non-cooperative nodes is larger than the number of defective nodes if only normal operating nodes are present in the network. It is worth noticing when the short-term history is used and some traitorous nodes are present every node behaved in a non-cooperative way towards the others shown at extension of the right-hand side portion of the graph. Thus, this effect spread over in that part of the network and such behavior could be observed at the presence of the traitorous nodes. In those parts of the network where the nodes are relatively densely positioned the behavior of the nodes is more or less the same, however, there are some areas, where, because of the presence of the traitorous nodes, the nodes become less cooperative.

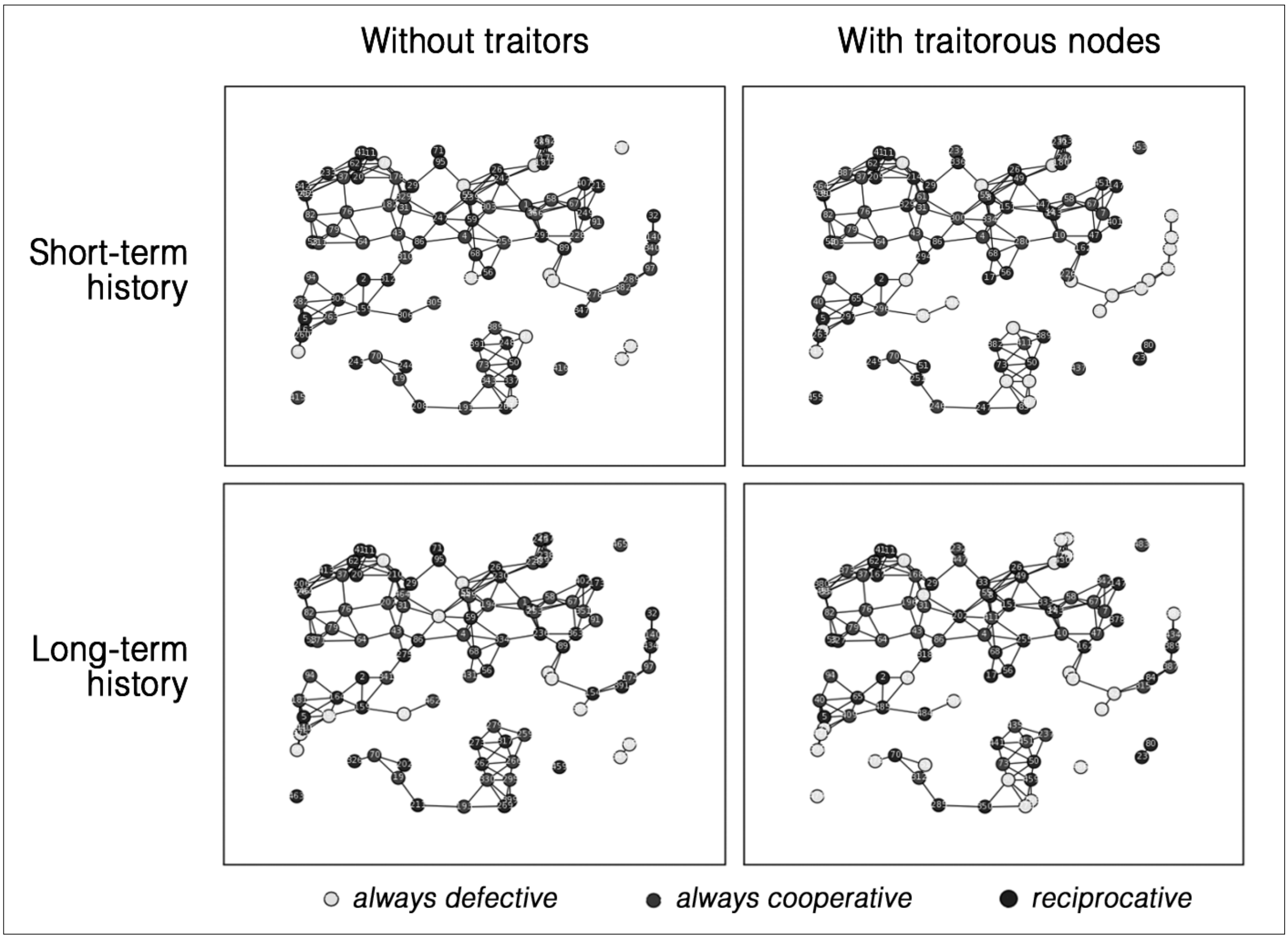
Fig. 2. shows the distribution of the nodes using specific strategies. It can be seen that, if traitorous nodes are present, the distribution of the nodes is more unsteady, the nodes more frequently change strategies. This effect can be clearly seen also when comparing the solutions using short-term and long-term history. In accordance with previous diagram it can be observed that how many nodes followed the various strategies by the end of the simulation. At the presence of the traitorous nodes the difference is clearly noticeable, by the end of the simulation more nodes used the strategy of never cooperating with the others.

4. Summary

In summary, we may establish that the proposed incentive system is able to motivate the nodes to voluntary cooperation. In some cases this cooperation is high-level and the number of the non-cooperative nodes is insignificant, while in other cases some parts of the network form non-cooperative groups.

The study of the system may be continued different ways, e.g., we might examine a specific situation when the nodes are not steady, but change their positions. In this case we certainly must provide effective routing for the nodes to be able to find each other in a quickly changing network. Several studies were made to this effect, however small but frequent changes the network topology had a significant effect on the nodes' strategy selection.

Figure 1. Distribution of nodes by strategies in the topology graph



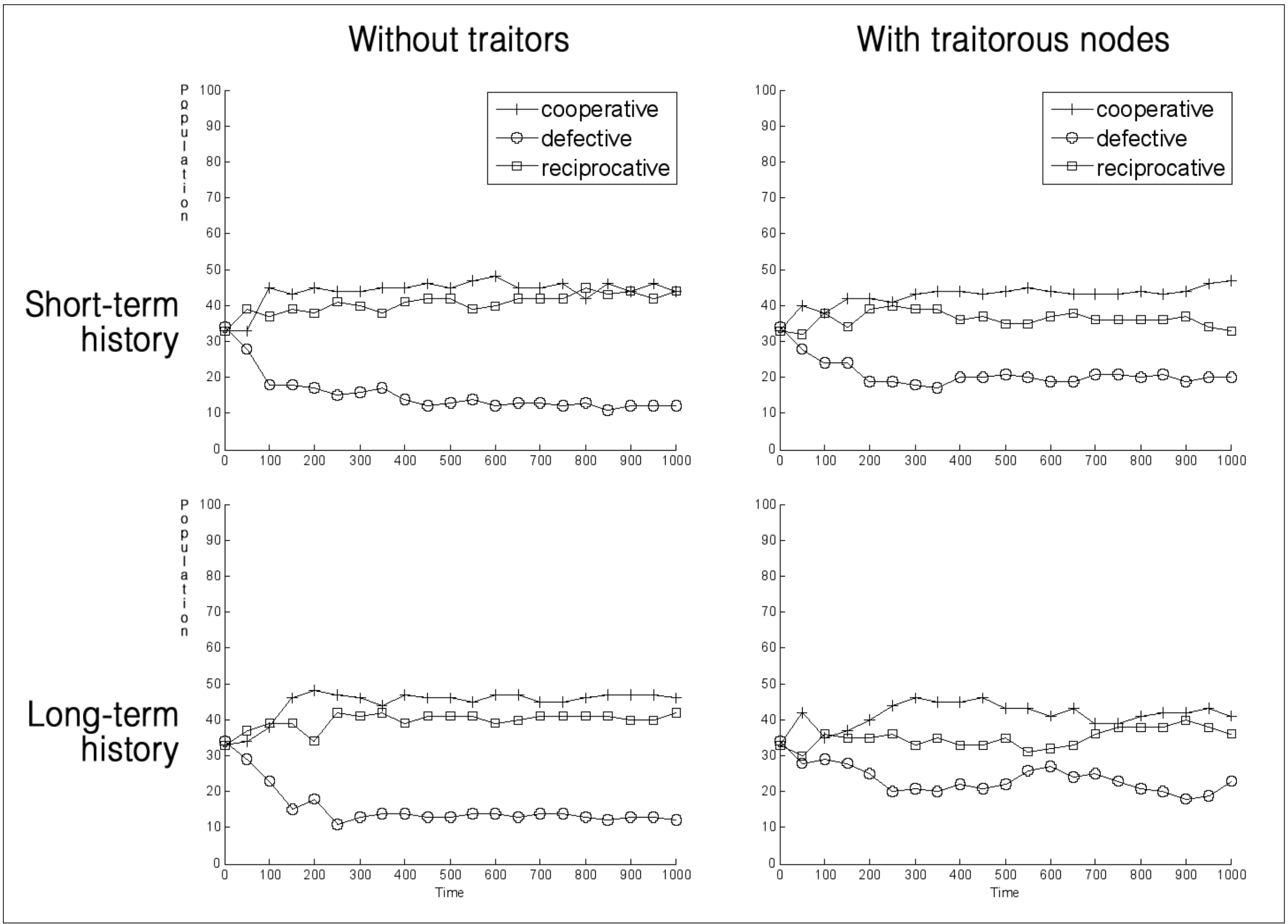


Figure 2. The numbers of the nodes using the different strategies during the simulation

This means, that we may not draw many conclusion from describing diagrams like the above ones. The study of such case constitutes the subject of further research.

References

[1] N. Niebert, H. Flinck, R. Hancock, H. Karl, C. Prehofer, Ambient Networks – Research for Communication Networks Beyond 3G, 2004.

[2] Kovács Balázs, Simon Csaba, "Ambient" hálózatok, 2005.

[3] Mark Burgess, An Approach to Understanding Policy Based on Autonomy and Voluntary Cooperation, Lecture Notes on Computer Science, 2005.

[4] Mark Burgess and Siri Fagernes, Pervasive Computer Management: A Model of Network Policy with Local Autonomy, IEEE Transactions on Networking, 1999.

[5] Michalel Feldman, Kevin Lai, Ion Stoica, John Chuang, Robust incentive techniques for peer-to-peer networks, ACM Conference on Electronic Commerce, June 2004.

[6] Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, Incentives for sharing in peer-to-peer networks, 3rd ACM conference on Electronic Commerce, Tampa, Florida, USA, 2001.

Authors

László Harri Németh obtained his M.Sc. in Computer Engineering, graduated in Budapest University of Technology and Economics in 2006. Currently he is a Ph.D. student of Budapest University of Technology and Economics, Department of Telecommunication and Mediainformatics. His research interests are peer-to-peer and ambient networks and Wi-Fi based positioning techniques for presence and location based services. He participated in the development of positioning algorithms for WLANpos, a Wi-Fi based indoor local positioning system.

Róbert Szabó is an associate professor at the Department of Telecommunication and Media Informatics, Budapest University of Technology (BME). He is the head of the High Speed Networks Laboratory (HSNLab) at BME; and is the President of the Telecommunications Section of the Scientific Association for Infocommunications, Hungary. His main research interests are architectures, protocols and performance of communication networks.