PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 49, NO. 3–4, PP. 223–237 (2005)

# NON-COOPERATIVE GAMES FOR SELF-ADAPTIVE TELECOMMUNICATION PROTOCOLS

Tibor DULAI

University of Veszprém
e-mail: tibor.dulai@irt.vein.hu

Received: Sept. 14, 2005

## Abstract

An important requirement of the new generation of telecommunication protocols is, that they should be able to adapt themselves to the changes of the environment without human interaction. The main problem of the system, which is built up by self-adaptive protocols is the question of stability. Results in the field of game theory, especially which are related to non-cooperative games could offer a good solution for this problem. In this article I attempt to introduce some kinds of application of game theory in case of self-adaptive telecommunication protocols.

*Keywords:* self-adaptive protocols, game theory, non-cooperative games, Nash-equilibrium.

## 1. Introduction

In our days' almost saturated telecom market the main point in increasing one service provider's role is the offer of more and more advanced services. That is the reason why service developers attempt to create services, which make the best of the available resources of the telecommunication network. It is not an easy task: beside the services which often have to deal with delay sensitive multimedia content, the attendant environment is largely heterogeneous because of the large scale of network element and end-user device types.

The basics of a telecommunication software are protocols. Protocols are the semantic, syntactic and timing rules of the communication and they are represented as finite state machines. These rules has to be created to be flexible, so that the protocols fit the best to the actual state of the network. It means that the protocols – and hereby the services also – have to be adapted to their environment. Section 2 deals with self-adaptive protocols and examines their possible role and their requirements in the future of telecommunication networks.

However, the software elements, which change their properties to adapt themselves to the environment raise an other problem: will the final system be stable or it becomes unstable? I attempt to apply the results of game theory for answering this question. In Section 3 I show why I chose non-cooperative games for this task.

Section 4 introduces some examples for situations in the field of telecommunication where self-adaptive protocols can reach the stable points of the system by the use of game theory. Finally in Section 5 I present two methods for finding the equilibrium points of some of the previously introduced examples.

## 2. The Role of Self-adaptive Protocols

The continuous development of technology creates more and more challenges for telecommunication software developers. The quick spreading of mobility increases the importance of ad-hoc networks, where mobile devices can join and secede from in dynamical manner. Moreover mobile terminals have to be able to communicate with the wired ones, and this fact also increases the heterogeneity what telecommunication has to face.

There is also a new feature: different types of contents have to be transmitted through the network even in case of mobile telecom, because also mobile devices are capable of handling multimedia contents. We can see that the actual challenge of telecommunication is to be ready for the discrepancies not only in the hardware but in the software field too.

The advantages of the technological development, what enrich services also cause some problems. If we want to use a new feature, we often have to make compromise in opposite virtues of the transmission. The developers' aim is to ensure the optimum for the user. (1) During the communication using mobile and portable devices long enough battery lifetime has to be ensured. The question is: which part of the service can be sacrificed for sparing energy. (2) Ad-hoc networks should be reliable, stable, self-organizing networks. Question could be: how do the communication entities co-operate in an ad-hoc network with each other, if they have different goals? (3) In the heterogeneous environment the telecommunication software has to be portable. What should be included in the software for making it portable reserving simplicity because of the restricted resources of mobile devices? (4) In our days mobile networks should deliver delay sensitive enhanced content with appropriate QoS through an unreliable, non-deterministic, fluctuating medium, what is electromagnetic wave. One of the most important questions is how to share the unreliable radio channels among more entities to ensure the desired Quality of Service for each of them.

Traditional static software is not the best solution, usually there is not any ideal constant parameter which would give the most optimal results for the whole communication time in a changing environment. Out and away the protocol has to adapt itself to the changes in execution time.

For the adaptation the protocol needs information about the environmental changes. In the followings we will see, that traditional layered protocol model is perhaps not the best solution for that.

### *2.1. Information for Adaptation*

Nowadays network architecture is created by layered protocols, they have their strictly separated functionality and they offer services to the upper layers. In this model communication takes place only between the neighbouring layers (*Fig. 1a*). The modular structure was applied because of the simplicity during its development.

It makes it possible to design complex systems easier than without the layered structure of the protocols. Although in case of the protocols, which should be capable to adapt themselves automatically to the changes of the environment, this model suffers of the lack of information sharing across the protocol layers.

Let's consider the case in which an application's information request depends on the status of the network. If the underlying network is unreliable, the requested content is only a textual one, otherwise the application wants to get multimedia content from a web page. This solution would improve the response time and decrease network congestion. If we refer to the OSI reference model, application layer needs information from the network layer in this example (*Fig. 1b*). It requires direct communication between the third and the seventh layer.

In an other case, the method which way the information source and the channel is coded for compressed image transmission depends also on the quality of the wireless channel. Compressed image is very sensible for transmission failures, and wireless systems can be characterized by limited bandwidth and high bit error rate. It is the reason for adapting the applied source and channel coding technics to the actual reliability level of the channel.

As a third example, the adaptive multipath routing in a network needs information about the state of the physical channel. Based on these data the resources of a network could be used in a more effective way and the offered services could have higher quality.

In the previous three examples protocols of the higher level layers have to be informed by the lower level layers. In the cross layer design methodology [1] it is called upward information sharing, and it is shown in *Fig. 1b*. Naturally there are situations, which fit to the opposite direction.

Better performance could be reached if the network layer knew about the expected communication load. If the application layer shares this information with the protocol layer, which is responsible for resource allocation, the network can apply more effective dynamical allocation strategies. In this case the information sharing happens in downward direction (*Fig. 1c*).

The more information an adaptive protocol has about the environment, the most effective way it can adapt itself to the actual situation. Cross layer design offers greater amount of information for self-adaptive protocols than the usual design to choose the values of their parameters. In the followings we will examine, what the essence of self-adaptive protocols is, how they handle the information they get to reach their optimal performances.

### *2.2. Self-adaptive Protocols*

As we have seen, self-adaptive (SA) protocols are protocols which attempt to do their best dynamically based on the information they get from their environments. The definition of self-adaptive software is: 'The self-adaptive software evaluates its own behaviour and changes behaviour when the evaluation indicates that it is

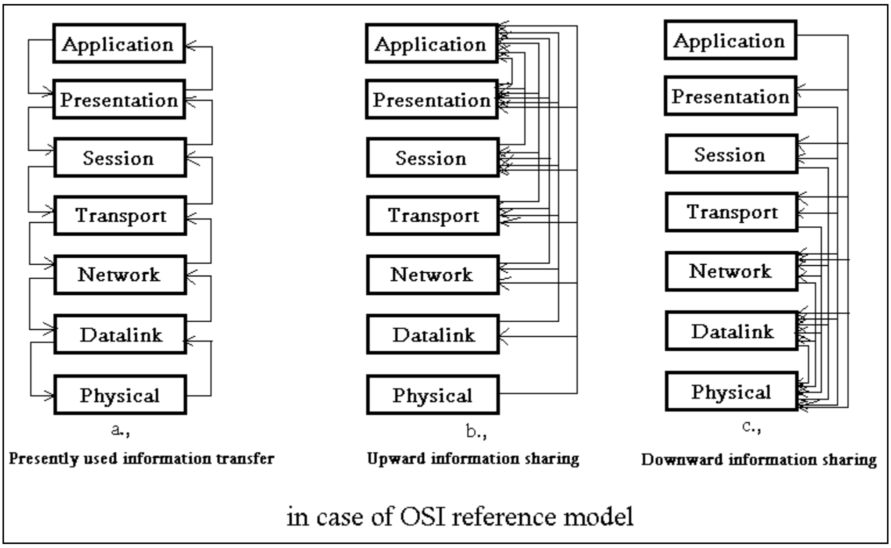


*Fig. 1.* Cross layer network architecture design

not accomplishing what the software intended to do or when better functionality or performance is possible.' [2] The evaluation is mostly based on the feedback, which is got from the environment. The structure of a self-adaptive protocol is presented in *Fig. 2*, based on the work of Katalin Tarnay [3].

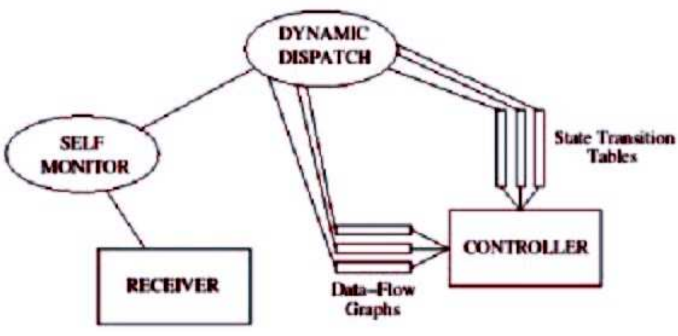


*Fig. 2.* Structure of a self-adaptive protocol

The protocol gets the feedback via the Receiver. The response of the environment, which is based on the behaviour of the system, is analysed by Self Monitor and the result is passed to Dynamic Dispatch. This unit is responsible for choosing the desired State Transition Table or Data Flow Graph for the work of the protocol as a response on the results of the monitoring process. Finally the Controller supervises the protocol functions based on these parameters. This description shows that both the protocol's static properties and the dynamical ones are adjusted according to the environment.

## 3. Stability of Systems Built up by Self-adaptive Protocols

A self-adaptive protocol could work in a more efficient way than a static one. After all systems which are built up by self-adaptive protocols may suffer some problems. Let's consider a communication system, which is created by self-adaptive agents. Usually each agent of the system changes her SA protocols' parameters in a selfish way to maximize the user's utility. In this dynamical system stable points have to be found avoiding from the instable behaviour, that can be resulted because of the continuous changes of protocol parameters. The main question is: how to find the stable points of the communication network?

I attempt to apply the results of game theory [4] as a possible answer on this question. There is rich literature on finding the equilibrium point of a game. In a game there are players who do have their well defined individual goals which can be characterized by utility functions. Each player chooses and carries out actions at her decision points towards maximization of her utility. The main point in the game is that a player's payoff depends not only on his decision but also on the decision of the other players of the game. In the 'communication game' the game is the communication process itself, the players are the self-adaptive communication agents, and their strategies are represented by the settings of protocol parameters. Payoffs can be characterized by the resulted communication features (inverse of bit error rate, inverse of transmission time, etc.), and the goal of a player is to maximize his payoff. This goal causes the fact, that the players compete with each other, and they do not co-operate. It means that the communication game is a non-cooperative game.

The theoretical background of non-cooperative games is elaborated, especially by the work of John Nash: the Nash-equilibrium means the solution of non-cooperative games. A game reaches it's Nash-equilibrium, if no player has other strategy profile which results him in a better payoff in case of no change in the other players' strategy profile. In our case our task can be seen: to find the Nash-equilibrium of the communication game. It is sure that this game has got at least one Nash-equilibrium, because Nash theorem says: in case of mixed strategies every non-cooperative game has (at least one) Nash-equilibrium.

Now it can be seen in which direction to start. There are two main questions remaining: (1) What kinds of communication situations need game theory? (2) How to determine the stable points of these situations? In the next two sections I attempt to give some examples as part of the answers on these questions.

## 4. Non-cooperative Games in Telecommunication

In the previous sections I have already presented situations where some features of a telecommunication service had to compete with other ones. In this section I give four concrete examples for applying game theory to find the stable states of some dynamically changeable situations especially in the field of mobile telecommunication.

### *4.1. Random Access in Ad-hoc Wireless Networks*

One challenging field of mobile telecommunication is ad-hoc networking. These self-organizing mobile networks are created by devices which may enter and leave the network continuously. In these circumstances a stable, well functioning distributed network has to be created, which is able to adapt itself to the permanent change of its environment. The self-configuration of the network is based on the individual actions of the attendant mobile devices. While they attempt to carry out their own tasks the whole network should stay in a stable state. To transmit some data, the device has to get access to the channel at first. If there are more than one active devices which intend to transmit data at the same time in the network, there will be collision on the channel. The question is when should a device attempt to start the transmission. It is a good situation for involving the results of non-cooperative game theory [5]. Here the different devices are the players and they have their own retransmission strategies. In this game we can measure the payoff of a player as the inverse of his complete transmission time. Using game theory in case of random access seems to be a useful idea. MacKenzie and Wicker developed a game-theoretic version of slotted Aloha [6], the application of their result in case of more complex random access methods is a promising research area.

### *4.2. Energy Conservation of Mobile Devices in Fading Environment*

Another important field of mobile telecommunication research is the field of energy conservation. Because of mobility the mobile devices have to be small and easy units, even so they have to ensure long service time without recharging the battery. This means that they have to take the advantage of their transmission policy's dynamism as optimally as possible. The fluctuating and fading properties of the wireless channel and the usually bursty data source make it more difficult. The mobile devices have to change their channel sensitivities in the way which is powerful enough to do their transmission tasks correctly, but requires as small pieces of energy as possible. This situation can also be represented by a game, where the strategies of the players are their transmission policies and their outcomes can be measured by the amount of successfully transmitted bits during battery lifetime.

The outcome depends both on the fluctuation of the channel in time, in frequency and in space, and also on the co-channel interference besides the individual strategy.

### *4.3. Fair Capacity Sharing Ensuring Appropriate QoS*

The 3rd generation mobile systems make multimedia data transmission possible. This means that mobile users are able to have resort to services with very different QoS requirements. It results that the network has to multiply data streams with different bit rates to transmit them through the wireless channel. For the sake of decreasing the amount of control information the medium access is usually not controlled by the base station, but it is decentralized. That way a situation is created where the communicating units fight for getting such a wide bandwidth, which is enough to ensure the required quality during the transmission. In addition they want to get it as cheap as possible. (Usually the cost of a very high data rate transmission can be extremely big.) The strategies of the players are medium access and data rate policies in this communication game, while the utility function can be determined as a metric of the achieved QoS reduced by the price of that.

### *4.4. Aspect-oriented Telecommunication Software with Time Constraints*

In contrast to object-oriented paradigm aspect-oriented programming (AOP) gives the advantage of dynamic software architecture creation. Besides the core software we may develop different aspects with different functions and depending on the needs or on some conditions we can include them into the core software. If we think of an OO software, the clearing of the screen procedure may crosscut several objects. Screen clearing can be implemented in one aspect. AOP can be extremely important in telecommunication industry where the offered additional services vary day by day. The aspects can be developed in different parts of the world, AOP makes distributed development possible. Moreover the use of aspects offers the possibility to change the architecture of a software in runtime.

Crosscutting concerns can be modularized as aspects and they can be added to the software at join points if we want. The decision can be adapted to the actual state of the environment. When we consider error handling aspects and the critical parameter is the running time of the software, the situation can be described by a game as follows: The strategy of players is their aspect waving policies at join points and the set of the game's outcomes is consisted of the reciprocal of the sum of the aspects' execution times at all join points.

In the next section I present a method for determining the equilibrium point(s) of this game, and next to that I give an other possible solution for finding Nash-equilibrium in case of continuous strategies, e.g. in case of the game presented in 4.3.

## 5. Determination of Nash-Equilibrium

In this section I present two different methods for determining the Nash-equilibrium of a game. The first one can be applied for games with discrete strategies, the example I use is the aspect-oriented game presented in 4.4, and the second one will give the solution of games with continuous strategies. In this case it is applied for the fair capacity sharing game, which was introduced in 4.3.

### *5.1. Solution of the 'Aspect-Oriented' Game*

An AO telecommunication software with time constraints can be considered as a game. Let's have a look at the problem, which was introduced in 4.4. There is a software which has more components: the core software and some aspects. In the core software there are joint points, where aspects can be connected. We use an aspect at a joint point, if we want to use the function of the aspect at that point. Naturally, the execution of the aspect takes time. Each time the software runs, the core part will be executed. In contrast to this, aspects are used only if the conditions are fulfilled, which guard the joint points, where the aspects are connected.

Let us suppose that the aspects are used for error handling in this case, and the quick execution has the same importance than the application of the aspects. It means that we have to balance between the fastness and the correctness of the software. The game which models this situation is the following: the players are the error handling aspects. Each player has two possible actions: the aspect is either applied at a joint point or not. There are aspects, which are related to each other: if one of them is executed, the other one has to be executed too or it is forbidden to execute the second one. There are some joint points in the core software and the goal is to minimize the maximal possible time interval which is needed for the execution of the aspects at the joint points. It means that the payoff is the sum of the reciprocal of execution times at all joint points.

The players of this game are driven by their own interests, it is a non-cooperative game. To find the game's equilibrium, I attempted to use mixed strategies. In case of clear strategies the players can choose between their strategies only to apply one of them or not (in our case it means two strategies: execution or no execution), however in case of mixed strategies a probability distribution is assumed over the clear actions. Nash theorem says that any finite game has mixed-strategy Nash-equilibrium. It means, that if we assign probabilities to the aspects' actions at the joint points, the game has to have Nash-equilibrium at least in case of one possible probability distribution. *Fig. 3* shows, what likelihood it has, that an aspect is executed at a certain joint point.

In the figure we can see that there are M joint points in the core software and we've got N error handling aspects which can be waved into the core software. The likelihood of applying aspect $j$ at joint point $k$ is $p_{j,k}$. As it was mentioned earlier, the likelihood of some aspect's use can depend on other aspects' use. It means that join points

| aspects | 1 | 2 | 3 | 4 | ... | M |
|---|---|---|---|---|---|---|
| 1 | $p_{1,1}$ | $p_{1,2}$ | $p_{1,3}$ | $p_{1,4}$ | ... | $p_{1,M}$ |
| 2 | $p_{2,1}$ | $p_{2,2}$ | $p_{2,3}$ | $p_{2,4}$ | ... | $p_{2,M}$ |
| 3 | $p_{3,1}$ | $p_{3,2}$ | $p_{3,3}$ | $p_{3,4}$ | ... | $p_{3,M}$ |
| ... | ... | ... | ... | ... | ... | ... |
| N | $p_{N,1}$ | $p_{N,2}$ | $p_{N,3}$ | $p_{N,4}$ | ... | $p_{N,M}$ |

*Fig. 3.* Mixed strategies in case of the 'aspect-oriented' game

there may be relations between the likelihood values in the table. It depends on the concrete situation, on the concrete software. The required time for executing all the aspects, which are applied at joint point $k$ is

$$T_k = \sum p_{j,k} * t_{j,k} \tag{1}$$

if $t_{j,k}$ denotes the time needs at joint point $k$ to execute aspect $j$. Based on this, we are able to determine the desired likelihood values in a given situation to get the optimal result. In our example we want to reach the case, where the overall execution time is minimal. It means, that we have to find the worst case, where the execution of aspects at a joint point takes the most time and we have to find the best case of the worst cases. That is the minimization of the maximal time-interval used for aspect-execution at joint points. Formally:

$$opt = \min \sum_k \max T_k = \min \sum_k \max \sum p_{j,k} * t_{j,k} \tag{2}$$

It is a minimax exercise. Solving it, we reach the Nash-equilibrium of the game. Nash-equilibrium is the state of a game, where nobody is interested in changing his own strategy unless anybody else does it. The solution has this property, because if any player changes his strategy (e.g. executes further aspects at a joint point), it may result in longer execution time, so the outcome of the game may decrease.

### 5.2. *Solution of Games with Continuous Strategies*

There are games, where we can not apply the method which was presented in 5.1 for finding the equilibrium point, because the number of the actions, which the players can carry out is infinite, so we should assign likelihood to infinite number of actions. Usually players have to choose their actions from an interval - which contains infinite number of possible values for the player (e.g. 0 - 10 Mbit/s) - in case of this kind of games. These are games with continuous strategies [7]. Fair capacity sharing game, which was presented in 4.3 belongs to this class of games, because the transmission quality of a communication agent can vary its level continually within an interval.

In fair capacity game the players want to transmit data through the fluctuating wireless medium. Their income is in relation with the bandwidth they get and the time it takes to push the data through the channel. In fact, we may think of the quality as the bit rate of the transmission. However, the higher quality costs more than the lower. An agent's income decreased by this cost results the agent's profit. Each agent's goal is to maximize its profit.

For the sake of simplicity let us suppose that there are only two communication agents in the system. The given solution is well applicable also for systems, which have more than two agents, we only have to work in more dimensions. Let denote $Q_1$ the quality level of the first agent's transmission and $Q_2$the second one's. The quality of service is influenced by both the result of medium access and the data rate policy of the agents. To visualize the way of the solution, let us suppose that the function which expresses the communication income of a quality unit can be described as follows:

$$I = 10 - Q_t \quad \text{if } Q_t < 10, \tag{3}$$
$$I = 0 \quad \text{otherwise,}$$

where

$$Q_t = Q_1 + Q_2. \tag{4}$$

It means, that if the sum of the desired bit rates of the two agents is over a limit (e.g. the capacity of the channel, which is 10 units in this example), then none of the two agents has profit. If the sum of the agents' desired bit rates is small, the income can be a great value (e.g. the unused capacity of the channel can be sold to other firms). Naturally, the transmission needs resources, which have their prices. For the sake of simplicity let's consider the case where the cost is linearly proportional to the transmitted bits. Let it be expressed by

$$C(Q_i) = 3 * Q_i \tag{5}$$

for both agents. It means, that the cost of a quality unit is 3 units. Based on the previously defined functions the communication profit ($\Pi_i$) of agent $i$ can be derived as follows:

$$\Pi_i(Q_1, Q_2) = I * Q_i - C(Q_i) = (10 - (Q_1 + Q_2)) * Q_i - 3 * Q_i \tag{6}$$

It means that the profit – a metric of QoS – of one agent depends not only on its own desired transmission quality but also on its desired quality of the other agent's transmission.

Each agent wants to make his profit maximum. This information is part of the common knowledge. It means, that agent1 knows, that if its moves first and determines a quality level for itself, agent2 chooses the point of its reduced state space, which makes its profit maximum. Before explaining it graphically, let's get acquainted with isoprofit curves!

The isoprofit curve means the collection of points, which belong to the same profit value of an agent. Based on the profit function (6) we can determine the isoprofit curves of the agents as it is illustrated in *Figs. 4* and *5*. *Fig. 4* shows some isoprofit curves of agent2.

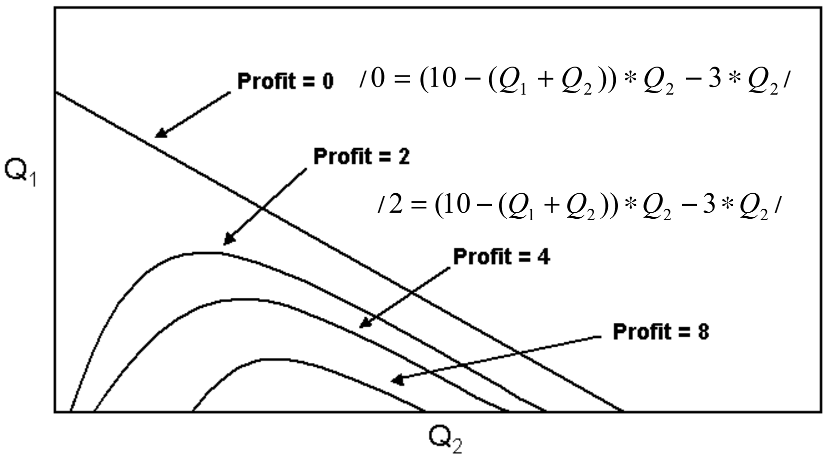


*Fig. 4.* Isoprofit curves of communication agent2

If the profit value of agent2 is equal to zero, it means that

$$\Pi_2(Q_1, Q_2) = 0 \tag{7}$$

$$(10 - (Q_1 + Q_2)) * Q_2 - 3 * Q_2 = 0 \tag{8}$$

Let us suppose that $Q_2 \neq 0$:

$$10 - Q_1 - Q_2 - 3 = 0 \tag{9}$$

If the profit value of agent2 is equal to 2:

$$(10 - (Q_1 + Q_2)) * Q_2 - 3 * Q_2 = 2 \tag{10}$$

$$7 * Q_2 - Q_1 * Q_2 - Q_2^2 = 2 \tag{11}$$

$$Q_1 = \frac{7 * Q_2 - Q_2^2 - 2}{Q_2} \tag{12}$$

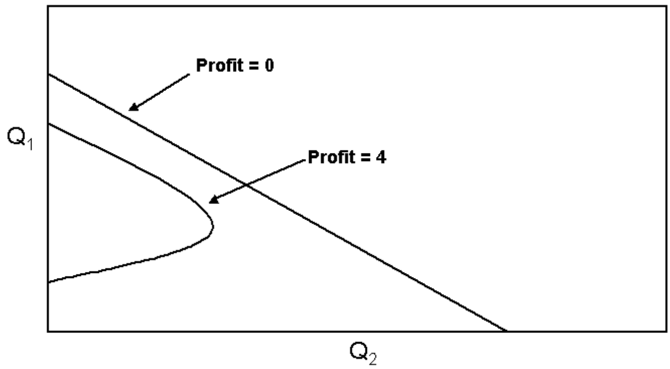


*Fig. 5.* Isoprofit curves of communication agent1

Similarly, *Fig. 5* shows some isoprofit curves of agent1.

With the help of isoprofit curves we are able to determine the optimal level of the desired quality of transmission. Let us suppose that agent1 sets 3 as the level of its transmission's quality. It can be seen in *Fig. 6* as a line:

$$Q_1 = 3. \tag{13}$$

Now agent2 can choose only points as its transmission quality, which fit onto the line. Among the points of the line there is only one point, which results the highest profit value for agent2 in these circumstances. This point is the point of agent2's isoprofit curve, which curve is tangential to $Q_1 = 3$ line. If agent2 chose other point, the point would fit onto agent2's isoprofit curve, which curve represents less profit value for agent2. By solving the maximalization problem of agent2's profit function, we can determine the set of all optimal reaction of agent2 to all level of agent1's quality selection. This set of points is the curve which is called the reaction curve of agent2. It can be seen in *Fig. 6*.

The solution contains the partial derivation of agent2's profit function with respect to $Q_2$

$$\frac{\partial \Pi_2}{\partial Q_2} = 7 - Q_1 - 2 * Q_2^* \tag{14}$$

and the finding of the value of $Q_2$*, which makes the derived function equal to zero.

$$7 - Q_1 - 2 * Q_2^* = 0 \tag{15}$$

$$\begin{array}{l} Q_2^* = \frac{7-Q_1}{2}, \quad \text{if} \quad Q_1 \leq 7 \\ Q_2^* = 0, \quad \text{if} \quad Q_1 > 7. \end{array} \tag{16}$$

$Q_2$* is the reaction curve of agent2.

$$\frac{\partial \Pi_2}{\partial Q_2} = (10 - (Q_1 + Q_2^*)) - Q_2^* - 3 = 0$$

$$Q_2^* = \begin{cases} \frac{7 - Q_1}{2}, & \text{if} \quad Q_1 \leq 7 \\ 0, & \text{if} \quad Q_1 > 7 \end{cases}$$

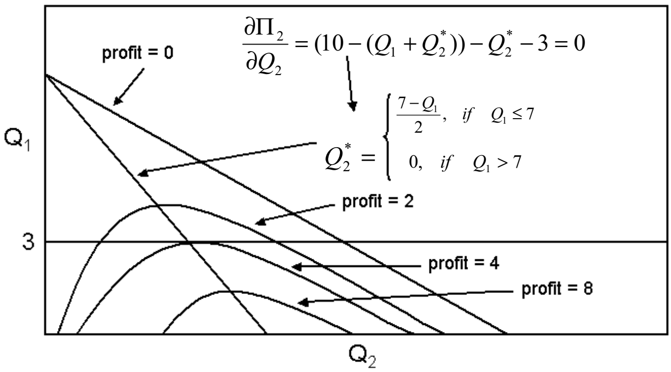


*Fig. 6.* Reaction curve of communication agent2

In game theory we suppose that the players are rational decision makers. It means that they choose the strategy, which maximizes their profits. That's why agent2 will choose a point, which fits onto her reaction curve, and agent1 knows this fact. Based on this information agent1 can determine his optimal level of transmission quality: he has to find his isoprofit curve, which is tangential to the reaction curve of agent2. If we have a look at agent2's reaction curve (*Fig. 7*), we can realize, that the point, in which agent1's isoprofit curve is tangential to agent2's reaction curve, is the best choice of agent1. The other points of agent2's reaction curve fit onto agent1's isoprofit curves, which mean less profit for agent1. To find the equilibrium point of the game, we have to find agent1's isoprofit curve, whose extreme fits onto agent2's reaction curve. It means that we have to replace $Q_2$ with the function of agent2's reaction curve ($Q_2$*) in agent1's profit function, and we have to derive the function after that with respect to $Q_1$, as it is illustrated in *Fig. 7*.

$$\Pi_1(Q_1, Q_2) = (10 - (Q_1 + Q_2)) * Q_1 - 3 * Q_1 \tag{17}$$

$$\Pi_1(Q_1, Q_2) = (10 - (Q_1 + \frac{7 - Q_1}{2})) * Q_1 - 3 * Q_1 \tag{18}$$

$$\Pi_1(Q_1, Q_2) = \frac{7}{2} * Q_1 - \frac{Q_1^2}{2} \tag{19}$$

$$\frac{\partial \Pi_1}{\partial Q_1} = 3,5 - Q_1 \tag{20}$$

If we make this result equal to zero, we find agent1's optimal choice as his desired transmission quality level.

$$3,5 - Q_1 = 0 \tag{21}$$

$$Q_1 = 3,5 \tag{22}$$

If we replace $Q_1$ with 3,5 in agent2's reaction curve, we get agent2's optimal choice.

$$Q_2 = \frac{7 - 3,5}{2} = 1,75 \tag{23}$$

$$\frac{\partial \Pi_1}{\partial Q_1} = 3,5 - Q_1 = 0$$

$$\Pi_1(Q_1, Q_2) = (10 - (Q_1 + Q_2)) * Q_1 - 3 * Q_1$$

$$Q_1 = 3,5$$

$$Q_2^* = \begin{cases} \frac{7 - Q_1}{2}, & \text{if} \quad Q_1 \leq 7 \\ 0, & \text{if} \quad Q_1 > 7 \end{cases}$$

In this example the Nash-equilibrium of the game is the point, where the transmis-

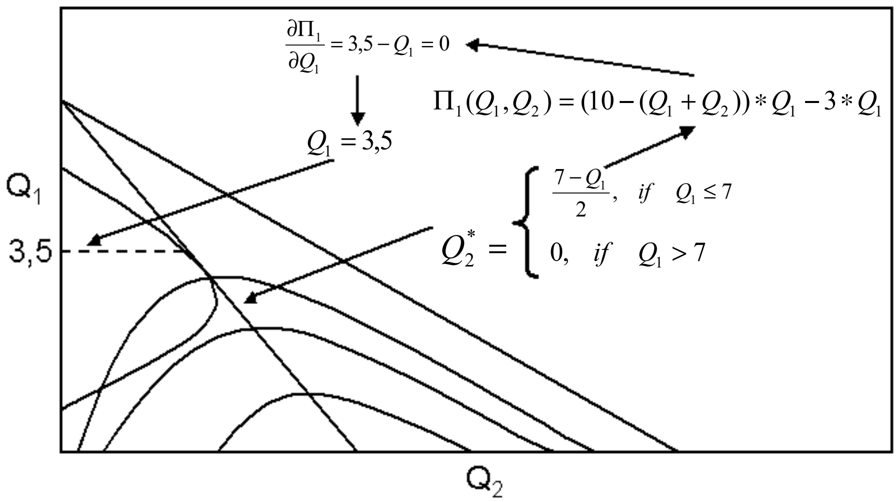


*Fig. 7.* Determination of the equilibrium point of game with continuous strategies

sion quality of agent1 is set to 3,5 units, while agent2's transmission quality is at 1,75 units.

The presented two examples give methods for finding a communication game's Nash-equilibrium both with discrete and with continuous strategies and they are useful methods for finding stable points of communication games, which are based on self-adaptive protocols.

## 6. Conclusion

In this article I attempt to highlight that in the future of telecommunication a big role of self-adaptive protocols is expected. The main question of them is how to determine the stable points of a system, which is built up by self-adaptive protocols, if these protocols change their behaviours continuously to adapt themselves to the actual state of the system's environment.

I attempt to answer this question by applying the results of the game theory, especially by using the research issues of the area of non-cooperative games. Agents of a telecommunication system usually compete with each other, that's why Nash-equilibrium can give the stable points of a communication game. Nash-equilibrium represents the stable point(s) of a non-cooperative game.

I introduced four game models in the field of telecommunication and showed examples for finding Nash-equilibrium of a system both in case of mixed discrete strategies and in case of continuous strategies.

Based on this work I'm on developing adaptive game-theoretic protocol for mobile telecommunication and I plan to simulate it in NS (Network Simulator) in the future.

## Acknowledgement

I would like to thank my supervisor, Prof. Katalin Tarnay for the tremendous help and encouragement she gave me during my work and studies.

## References

[1] HAAS, Z. J., Design Methodologies for Adaptive and Multimedia Networks, *IEEE Communication Interactive*, **39/11**, November 2001.
[2] ROBERTSON, P. – LADDAGA, R. – SHROBE, H., Introduction: The First International Workshop on Self-Adaptive Software, IWSAS 2000, Springer-Verlag, New York (2001).
[3] TARNAY, K., Self-adaptive protocol, In Self-adaptive Software: Applications, *Second International Workshop, IWSAS 2001, Balatonfüred*, Hungary, May 2001, Revised Papers, **2614**, pp. 106–112, Springer-Verlag, Berlin, Heidelberg, 2001.
[4] MYERSON, R. B., Game Theory - Analysis of Conflict, Harvard University Press, London, England, 1991.
[5] MACKENZIE, A. B. – WICKER, S.B., Game Theory and the Design of Self-Configuring, Adaptive Wireless Networks, *IEEE Communications Magazine*, November 2001, pp. 126-131.
[6] MACKENZIE, A. B. – WICKER, S.B., Selfish Users in Aloha: A Game Theoretic Approach, *Proceedings of the IEEE VTC*, Oct 2001, pp. 1354–57.
[7] BIERMAN, H. S. – FERNANDEZ, L., Game Theory with Economic Applications, Addisen - Wesley Publishing Company, Inc., Massachusetts, 1993.