The work was published in *Proceedings of Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2009), pp. 144-153, 2009.*

Self-organizing Bandwidth Sharing in Priority-based Medium Access

Stefan Wildermann, Tobias Ziermann, Jürgen Teich
Hardware/Software Co-Design
Department of Computer Science
University of Erlangen-Nuremberg, Germany
{stefan.wildermann, tobias.ziermann, juergen.teich}@cs.fau.de

Abstract

In this paper, we present an analysis of self-organizing bandwidth sharing in priority-based medium access. For this purpose, the priority-based Access Game is introduced. Analysis shows that a fair distribution of bandwidth cannot be achieved in this game. Therefore, we enhance this game by introducing a constraint that demands a small amount of the overall bandwidth being free. Fair bandwidth sharing is one Nash Equilibrium of this enhanced game, but not a unique one.

Based upon this theoretical analysis, a multi-agent reinforcement learning algorithm is proposed, where each agent tries to maximize its success rate for accessing the medium, while avoiding to violate the bandwidth constraint. We experimentally evaluate this mechanism for a system comprised of selfish agents. Experimental results show that the system is able to self-organize itself towards a fair distribution of bandwidth in a totally decentralized way without the need of global information or coordination.

1. Introduction

In real-time systems, access to shared resources is mainly implemented by priority-based approaches. A priority is assigned to each task, and, when requesting the resource, the task with highest priority is granted exclusive access. Due to its predictability in case of collisions, priority-based arbitration mechanisms are commonly used in real-time systems with hard deadlines, and applied for scheduling [1] as well as communication in distributed real-time networks [2]. The design of such systems is mainly done using analysis of the worst-case execution times [3]. However, when the average-case execution times of tasks significantly vary from their worst-case execution times, this approach can cause a waste of resources [1]. For example in multimedia applications using compressed frames, time spent for video encoding/decoding may change between each frame. Generally speaking, applications may not only require hard deadlines, but also other Quality of Service (QoS) requirements with soft deadlines as well as bandwidth requirements. For tasks with such requirements, packet delay or, for some applications, even loss can be tolerated and may even be unrecognized by an user. Another drawback of offline designed systems is that they are not flexible and thus hard to adapt online to system changes, such as removal or insertion of communicating nodes.

The purpose of our paper is to take a closer look onto priority-based medium access from a theoretical point of view, especially on systems in which Quality of Service depends on bandwidth. Here, an important issue is how to share the medium among communicating nodes. The drawback of priority-based access schemes is that selfish nodes with high priorities can block nodes with lower priorities leading to unfair bandwidth sharing, or, in the worst-case, to exclusive access of the node with highest priority. We use game theory to prove that selfish, i.e., greedy behavior leads to the blocking of tasks with low priorities. The problem is that nodes with low priorities are not able to influence the decisions of nodes with higher priorities. We present an mechanism to overcome this drawback where a small amount of the overall bandwidth has to be free. If this is not the case, all nodes receive a penalty (in our case, a payoff of 0). This gives nodes with low priorities the option to indirectly influence the decision of nodes with higher priorities. It is furthermore shown, that this enhanced game has a Nash Equilibrium where bandwidth is shared fairly – although it is not a unique one. We will therefore provide a multi-agent reinforcement learning (MARL) algorithm that enforces fair strategies. This algorithm works in a decentralized fashion where each agent observes its local environment,i.e., the shared medium in our case. Experimental results show that this algorithm converges to a fair bandwidth sharing, providing self-organizing properties.

The further outline of this paper is as follows. Section 2 gives a brief overview of game theoretic approaches to medium access, presents some use cases, and introduces a state-of-the-art priority-based communication protocol. Section 3 formulates medium access by means of game theory and illustrates the problem of priority-based access schemes. We then formulate and analyze the priority-based Access Game in Section 4. Section 5 presents an enhancement of the priority-based Access Game in which a fair bandwidth distribution is one Nash Equilibrium. A distributed MARL algorithm is provided in Section 6 which is experimentally evaluated in Section 7. Section 8 concludes this paper.

2. Overview

2.1. Related Work

Game theory has been often applied to study behavior of computer networks since it offers a framework to model and analyze situations in which players have to make decisions that have mutual, possibly conflicting consequences. In recent years, several game theoretical approaches to model and understand the dynamics of medium access and communication over an exclusively shared medium have been made [4], [5], [6]. The main focus of this research lies on contention-based medium access of Carrier Sense Multiple Access with Collision Detection (CSMA/CD). Such approaches use collision detection mechanisms: whenever a collision is detected, all transmissions are aborted and the senders wait for some random time before trying to transmit data again. Especially, selfish (i.e., greedy) behavior is analyzed. The problem is that the more often nodes try to send data, the more collisions may occur and less amount of data can be transmitted in the end. The purpose of the game theoretic analysis is to find protocols for medium access to overcome this problem. Further goals are to distribute bandwidth fairly [7] or to reduce latency and meet deadlines [8]. The Equilibria of these protocols are analyzed which are commonly reached when all nodes try to maximize their profit simultaneously (known as Nash Equilibria [9]). Results show that when designing protocols properly, the aforementioned goals can be achieved.

2.2. CAN as Priority-based Medium Access

Our mechanism is tailored for the use in the Controller Area Network (CAN) communication protocol. Nevertheless, the analysis and results are also valid for any priority-based medium access mechanism. In the following, a short introduction of CAN [2] is given to illustrate the basic principles of such protocols. The protocol can be divided into two ISO/OSI layers: physical layer and data link layer.

2.2.1. Physical Layer. The purpose of the physical layer is to find a suitable physical and electrical specification. In the CAN protocol, bits are represented by the Non-Return-To-Zero method. This means that a logical "zero" is represented by a voltage difference and a logical "one" by no voltage difference. The concept of dominant and recessive bits is used, leading to the behavior that, if two or more nodes simultaneously try to send, the dominant bit, i.e., a logical "zero", is always approved. This is important for the arbitration as explained in Section 2.2.2. The wiring is done by twisted pair cable (Bus High and Bus Low) and set up as shown in Fig.2.2.1.

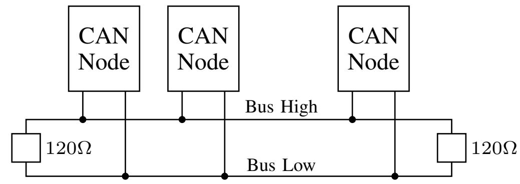


Figure 1. Setup of a Controller Area Network

2.2.2. Data Link Layer. CAN is a message-oriented protocol where nodes start to transmit as soon as data is available to send. Therefore, a bit-wise arbitration is used to avoid collisions. It can be described as follows: Each message starts with an unique identifier sequence which represents the message's priority. The sender transmits this identifier bit-wise and samples the actual bit on the bus in parallel. As soon as a sender is sampling a bit different from the one sent by it, it stops transmission. Since zero is the dominant bit, it is assured that the message with the lowest identifier remains at the end of the arbitration phase. The data frame of the remaining message can now be transmitted exclusively. This implements a priority-based access scheme where messages with lower identifiers have higher priorities for accessing the bus.

2.3. Use Cases

In this section, the importance of self-organizing bandwidth sharing in future real-time systems is illustrated. For example, [10] presents an automotive architecture where 29 electronic control units (ECUs) are connected in one priority-based network. One of the functions of the distributed ECUs is to transfer information from 360° sensors to several actuators. These sensors have equal importance to the quality of the control mechanisms. This means that additional information from one of the sensors is useless, if the other sensors cannot provide their data as well. Especially, when increasing the sampling rate of the sensors or adding additional components, the complete system has to be re-designed. However, the priority-based access does not allow to assign equal priority to all sensors. Today, the problem of finding proper priorities for the senders is solved by designing such systems based on worst-case assumptions to fulfill a static quality. This, of course, leads to unused communication resources what could decrease the overall quality of the mechanism, or even limit the number of sensor that can be used in the resulting system.

A more general use case is illustrated in Fig. 2. The system is divided into two classes of services. One class uses the upper priorities to ensure that the hard deadlines are met. The second class contains those services that require Quality of Services with no hard real-time constraints and are not critical for the function of the system. In this paper, we address the problem of providing equal bandwidth to the second class. This is, for example, the case for video

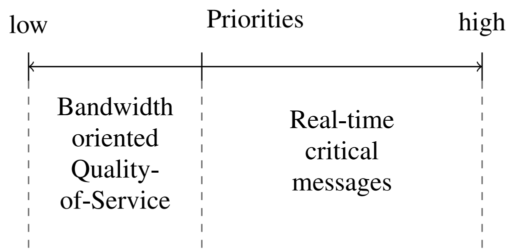


Figure 2. Partitioning of priorities. High priorities are used for messages with (hard) real-time constraints. Low priorities may be used for messages with bandwidth as Quality-of-Service.

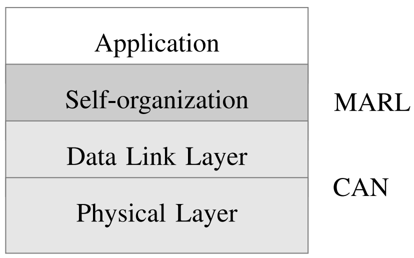


Figure 3. Example of a protocol stack used in a self-organizing Controller Area Network

and audio streams with flexible compression rates that are adjustable within a certain range by the user at run-time. A single stream would send with the lowest possible priority. In this case, the stream could use the bandwidth efficiently that is not used by real-time critical messages. But if a second stream or more are added, a method is needed to allow equal access for all streams to the remaining bandwidth. Else, the stream with the highest priority would displace the others.

2.4. Self-organization in Priority-based Medium Access

Following the principle as illustrated in Fig. 2, priority-based protocols are perfectly suited for a self-organizing behavior while still providing real-time guarantees for critical messages. The event-triggered message-oriented approach allows to add and remove nodes without interruption or reorganization of the communication network. In contrast to a contention-based approach, there is no waste of resources due to the collision avoidance of the access scheme. This paper proposes a protocol stack as depicted in Fig. 3. While real-time critical messages are directly passed to the Data Link layer of the CAN protocol, the remaining messages are first processed by a MARL-based self-organization layer. This layer determines when to send a message, in our case with the goal of achieving fair bandwidth sharing among all bandwidth-oriented messages. In this way, a designer does not have to care about priority assignment for the messages. Furthermore, new functionality can be added at run-time. The only prerequisite for this is that each message is assigned with an unique identifier. Elsewise, more than one node could be granted access, and errors would occur as a consequence.

3. Game Theoretical Analysis of Medium Access

Game theory is a branch of mathematics that helps to analyze learning in multi-agent systems [11]. It enables the formal definition and evaluation of multi-agent decision problems, i.e., conflict situations where the success of an individual agent depends on the decisions made by others. In the following, we will give some background definitions and then describe the medium access games used in this paper.

3.1. Preliminaries

A *game* consists of a set of *players* $\mathcal{N} = \{1, ..., n\}$, a set of *strategies* S_i available to player i, and a specification of *payoffs* for each combination of player strategies. Let $s_i \in S_i$ be the strategy of player i. Then the payoff of an individual player is given by the utility function $u_i(\mathbf{s})$ with $\mathbf{s} = (s_1, ..., s_n)$ being the strategies taken by all players.

An *equilibrium* is reached if each player chooses the strategy that is a *best response* to the strategies chosen by the other players. Although an equilibrium might not produce the best outcome of the game, each player receives the highest achievable payoff given the other players' strategies, and thus has no incentive to deviate from its strategy.

So-called *selfish players* try to maximize their own utility. When all players try to maximize their utilities simultaneously, a *Nash Equilibrium* is achieved [9]. For further analysis, we denote the strategy vector containing the strategies of all players but player i by $\mathbf{s}_{-i} = (s_1, s_2, ..., s_{i-1}, s_{i-1}, ..., s_n)$.

Definition 1: A strategy vector $\mathbf{s}^*$ is a *Nash Equilibrium* iff

$$u_i(s_i, \mathbf{s}^*_{-i}) \leq u_i(s^*_i, \mathbf{s}^*_{-i}) \quad \forall s_i \in S_i, \ \forall i.$$

This means that no player achieves a higher payoff when deviating from its strategy s^*_i, given that the other players keep their currently played strategies s^*_j, $j \neq i$.

3.2. Medium Access Games in Normal Form

Our theoretical analysis of medium access is based on the contention-based Access Game introduced in [7]. It models the contention phase where multiple users content for medium access. In this section, the difference between contention- and priority-based Medium Access is illustrated by using games in normal form where, for each strategy combination, the resulting payoff of each user is given. If the game has only two players, it is convenient to define

		Player 2	
		wait	send
Player 1	wait	0, 0	0, 1
	send	1, 0	0, 0

Figure 4. 2-player contention-based Access Game in normal form.

		Player 2	
		wait	send
Player 1	wait	0, 0	0, 1
	send	1, 0	0, 1

Figure 5. 2-player priority-based Access Game in normal form. Player 2 has higher priority than Player 1.

the game as a matrix as shown in figures 4 and 5. Each cell contains the payoffs $(u_1(s_1, s_2), u_2(s_1, s_2))$ received by the row player (Player 1) and the column player (Player 2), respectively, if the row player chooses strategy s_1 and the column player strategy s_2.

The game given in Fig. 4 represents the contention-based Access Game. Both players have the options to "send" or "wait". A player receives a payoff of 1 if it is granted medium access, and 0 else. Contention-based medium access uses *collision detection*. Here, successful medium access is granted in case only a single player chooses to send and all others choose to wait. Otherwise, collision occurs and no player is allowed to access the medium.

The primary goal of any rational player would be to maximize its expected payoff. However, as can be seen from the figure, there exists a certain conflict: If both players are greedy, choosing "send", they will never successfully access the medium. If, on the other hand, one player chooses strategy "wait", it also has no option to access the medium. The equilibrium of this game is given in [7]. It can also be shown that bandwidth is shared fairly in this equilibrium. A brief overview of this analysis will be given in Section 3.3.

To illustrate the problem of priority-based medium access, the corresponding 2-player game is represented in Fig. 5. Priority-based medium access uses *collision avoidance*. This means, in case more than one user tries to send, the user with highest priority is granted medium access. For the two player game, Player 2 has higher priority than Player 1. As can be seen, strategy s_2="send" *dominates* strategy s_2="wait" for Player 2. This means that, no matter how Player 1 may play, Player 2 always successfully accesses the medium when playing "send"

3.3. The Contention-based Medium Access Game

Rakshit et al. have analyzed the contention-based Medium Access Game using mixed strategies in [7]. This means that, instead of playing with discrete strategies "wait" and "send", the strategies are given as a probability distribution on strategy space S_i. More specifically, player i sends with probability p_i and waits with probability $1-p_i$. The Access Game with n players is formally defined as follows.

Definition 2: The *Access Game* $\mathcal{G}$ is defined as tuple $\mathcal{G} := (\mathcal{N}, (S_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}})$, where $\mathcal{N}$ is the set of players, S_i is the set of mixed strategies for player i, $S_i := \{p_i | p_i \in [0,1]\}$, and $u_i(\mathbf{p})$ is the utility function of player i depending on the strategies chosen by all players $\mathbf{p} = (p_1, p_2, ..., p_n)$.

The payoff is given as an utility function $u_i(\mathbf{p})$ that corresponds to the probability that user i successfully accesses the medium:

$$u_i(\mathbf{p}) = Pr_i(\text{Successful}) = p_i \cdot \prod_{j\neq i}(1-p_j).$$

This means, the utility is the probability that player i chooses "send" and all other players chose "wait".

It was shown that this game has a *Constrained Nash Equilibrium* leading to a fair bandwidth sharing.

4. The Priority-based Access Game

In this section, we adapt the contention-based Access Game to priority-based medium access, and show formally that there exists no equilibrium leading to fair bandwidth sharing in the standard game.

4.1. Game Formulation

Priority-based access schemes are characterized by the fact that collisions are resolved in an *arbitration phase*. In the arbitration phase, the active node with highest priority is determined. In this context, the term *active node* denotes a node that has data to send. In the following, we will use the terms *node*, *user*, *player*, and *message* interchangeably: A node specifies a controller connected to the bus (see Fig. 2.2.1). But for our further analysis, we state that each node just has one message to send and, thus, the same priority as this message. Nevertheless, the framework may easily be mapped to the case where a node has more message types with different priorities to send.

To adapt the Access Game defined in Def. 2 to priority-based medium access, following assumptions are made – as in [7] – and used throughout this paper:

A1: Each user always has a message to send.
A2: Each user has messages of equal length.
A3: The system is stable, so no technical failures occur.

The users $i \in \mathcal{N}$ are ordered according to their priorities $prior(i)$ with

$$prior(1) < prior(2) < ... < prior(n). \quad (1)$$

At the beginning of the arbitration phase, the player with highest priority is chosen from all players that want to send and granted access to the medium. The utility function has to be adopted accordingly.

Definition 3: The *priority-based Access Game* is an Access Game defined by tuple $\mathcal{G} = (\mathcal{N}, (S_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}})$ and with utility function

$$u_i(\mathbf{p}) = p_i \cdot \prod_{j>i}(1 - p_j). \quad (2)$$

Like for the classic Access Game, the utility function $u_i(\mathbf{p})$ corresponds to the probability that user i successfully transmits a data packet. In this case, this is the probability that user i chooses to "send" and all users with higher priorities chose to "wait". The strategies of users with lower priorities do not influence this payoff.

4.2. Fairness in the Priority-based Access Games

The priority-based Access Game is now used to define and analyzed fair bandwidth sharing. Although there are potentially many different ways to define fairness, we propose the following formal definition:

Definition 4: A *fair strategy* $\tilde{\mathbf{p}}$ guarantees that each user has the same probability of getting access to the medium.

In that case, the following *fairness constraint* holds:

$$u_1(\tilde{\mathbf{p}}) = u_2(\tilde{\mathbf{p}}) = ... = u_n(\tilde{\mathbf{p}}) \quad (3)$$

We now want to consider a medium where $b \in [0, 1]$ determines the available bandwidth. Following Def. 4, a fair strategy might look as follows.

Theorem 1: A fair strategy $\tilde{\mathbf{p}}$ of the priority-based Access Game is achieved if user i sends with probability $\tilde{p}_i = \frac{b}{n-(n-i)\cdot b}$ for all $i \in \mathcal{N}$.

Proof: From Def. 4,we have that

$$\begin{aligned} u_1(\tilde{\mathbf{p}}) = u_2(\tilde{\mathbf{p}}) = ... = u_n(\tilde{\mathbf{p}}) \\ u_i(\tilde{\mathbf{p}}) = \tfrac{b}{n}. \end{aligned} \quad (4)$$

Assume that the priorities of users are given according to (1). Then for two users i and $i+1$, the following relation between their sending probabilities is given when playing a fair strategy:

$$\begin{aligned} \tilde{p}_i \cdot (1 - \tilde{p}_{i+1}) \cdot \prod_{j>i+1}(1 - \tilde{p}_j) &= \tilde{p}_{i+1} \cdot \prod_{j>i+1}(1 - \tilde{p}_j) \\ \tilde{p}_i &= \frac{\tilde{p}_{i+1}}{1 - \tilde{p}_{i+1}}. \end{aligned} \quad (5)$$

We now prove that strategies $\tilde{p}_i = \frac{b}{n-(n-i)\cdot b}$ fulfill (4) by complete induction:

- For $i = n$, we have $u_n(\tilde{\mathbf{p}}) = \frac{b}{n}$ according to (4) and thus

$$u_n(\tilde{\mathbf{p}}) = \tilde{p}_n = \frac{b}{n} = \frac{b}{n - (n - n) \cdot b}.$$

- Under assumption that $\tilde{p}_{i+1} = \frac{b}{n-(n-(i+1))\cdot b}$ holds, we obtain using (5):

$$\begin{aligned} \tilde{p}_i &= \frac{\tilde{p}_{i+1}}{1 - \tilde{p}_{i+1}} = \frac{\frac{b}{n-(n-i-1)\cdot b}}{1 - \frac{b}{n-(n-i-1)\cdot b}} = \\ &= \frac{b}{n - n \cdot b + i \cdot b} = \frac{b}{n - (n - i) \cdot b}. \end{aligned}$$

□

In the following section, we take a closer look on the fairness condition in the priority-based medium access when played by selfish users.

4.3. Unfairness of Selfish Users

From Theorem 1, we know the strategies leading to fair bandwidth sharing. The theorem shows that each player would have to know the number of players n and the overall placement of its priority, i.e., its index i. In this paper, however, we want to establish a self-organizing multi-agent framework, where each player needs as few global information as possible. That is why we concentrate on agents that just have a local perception of their environment. Such users want to maximize their own payoff. In the following, we denote such agents as *selfish users*.

The problem already indicated in Section 3.2 is that selfish behavior results in the blocking of other users. The proof is given in the following.

Theorem 2: Selfish players may block players with lower priorities forever leading to unfair bandwidth sharing.

Proof: Each user i wants to maximize its utility u_i. As can be seen from (2), the best response of player i depends on $\prod_{j>i}(1 - p_j)$:

1) If $\prod_{j>i}(1-p_j) > 0$, then u_i is maximized for $p_i = 1$. In this case, user i blocks all users with lower priorities.
2) If $\prod_{j>i}(1 - p_j) = 0$, then u_i is maximized for any $p_i \in [0, 1]$ and there exists at least one $p_j = 1$ with $j > i$.

□

Especially, the player with highest priority has utility function $u_n(\mathbf{p}) = p_n$ which is maximized for $p_n = 1$. Therefore,

$$u_i(\mathbf{p}_n, \mathbf{p}_{-n}) = p_i \cdot \prod_{i<j<n}(1 - p_j) \cdot (1 - p_n) = 0 \quad \forall i < n.$$

This means that all other users are blocked by user n which has the highest priority.

5. Enhancement of the Priority-based Access Game

The main problem leading to unfair sharing lies in the fact that users with low priorities have no option to influence the decisions of users with higher priorities. On the other hand, users with higher priorities do not realize when overwriting users with lower priorities in the arbitration phase. The idea is to demand from the system that a small amount of the bandwidth stays free; this amount is denoted by ϵ. If this is not the case, the utilities of all users are set to 0. This gives nodes with low priorities the option to indirectly influence the decision of nodes with higher priorities, which may eventually lead to a fair solution.

The probability of the medium being free corresponds to the probability that all users decide not to send which is given by

$$Pr(\text{Medium is free}) = \prod_{j \in \mathcal{N}} (1 - p_j). \tag{6}$$

The claim that amount ϵ of the bandwidth stays free can be expressed as $Pr(\text{Medium is free}) \geq \epsilon$ for our theoretical analysis. Consequently, the utility function has to be redefined:

Definition 5: The *enhanced priority-based Access Game* is a priority-based Access Game with utility function

$$u_i = \begin{cases} p_i \cdot \prod_{j>i} (1 - p_j), & \text{if } \prod_{j \in \mathcal{N}} (1 - p_j) \geq \epsilon \\ 0, & \text{else} \end{cases} \tag{7}$$

We will now analyze the Nash Equilibria of this game and furthermore show that following a fair strategy leads to a Nash Equilibrium.

5.1. Analysis of the Nash Equilibrium

A Nash Equilibrium is achieved when all users try to maximize their payoffs. As already defined in Def. 1, for users being in an equilibrium no user will profit from deviating from its strategy. We now look at the impact of the enhanced game on the remaining users.

Theorem 3: Users $i \in \mathcal{N}$ are in a Nash Equilibrium when choosing strategies $\bar{\mathbf{p}}$ with

$$\prod_{i \in \mathcal{N}} (1 - \bar{p}_i) = \epsilon. \tag{8}$$

Proof: To prove that $\bar{\mathbf{p}}$ is a Nash Equilibrium, we show that users maximize their payoffs when choosing strategies $\bar{\mathbf{p}}$ by looking at the following two cases where the users deviate from this strategy:

- If $\prod_{j \in \mathcal{N}} (1 - p_j) < \epsilon$, there is to much communication on the bus and, as defined in (7), the utilities are set to

$$u_i = 0 \quad \forall i \in \mathcal{N}. \tag{9}$$

- If $\prod_{j \in \mathcal{N}} (1 - p_j) > \epsilon$, then we have following relation for strategy p_i of user i:

$$\begin{aligned} (1 - p_i) \cdot \prod_{j \neq i} (1 - p_j) &> \epsilon \\ p_i &< 1 - \epsilon \cdot \frac{1}{\prod_{j \neq i} (1 - p_j)}. \end{aligned}$$

What can be seen is that if user i would choose strategy p_i' defined as

$$p_i' = 1 - \epsilon \cdot \frac{1}{\prod_{j \neq i} (1 - p_j)} \tag{10}$$

with

$$(1 - p_i') \cdot \prod_{j \neq i} (1 - p_j) = \epsilon. \tag{11}$$

its utility would be maximized [1], since

$$u_i(p_i, \mathbf{p_{-i}}) < u_i(p_i', \mathbf{p_{-i}}) \tag{12}$$

Consequently, we have that $u_i(p_i, \bar{\mathbf{p}}_{-\mathbf{i}}) \leq u_i(\bar{p}_i, \bar{\mathbf{p}}_{-\mathbf{i}})$, $\forall p_i \in S_i, \ \forall i$. □

From Theorem 3 we know that the Nash Equilibrium has to fulfill $\prod_{j \in \mathcal{N}} (1 - \bar{p}_j) = \epsilon$. We now show that there exists a Nash Equilibrium that is a fair strategy as stated in Theorem 1.

Theorem 4: The fair strategy $p^* = (p_1^*, ..., p_n^*)$ with $p_i^* = \frac{b}{n-(n-i) \cdot b}$ is a Nash Equilibrium.

Proof: The proof of this theorem is done by simply evaluating $\prod_{1 \leq i \leq n} (1 - p_i^*)$. Due to the nature of the extended priority-based Access Game, the available bandwidth is set to $b = 1 - \epsilon$.

$$\begin{aligned} \prod_{1 \leq i \leq n} (1 - p_i^*) &= \prod_{1 \leq i \leq n} \left(1 - \frac{1-\epsilon}{n-(n-i)\cdot(1-\epsilon)}\right) = \\ = \prod_{1 \leq i \leq n} \frac{i+n\cdot\epsilon-i\cdot\epsilon-1+\epsilon}{i+n\cdot\epsilon-i\cdot\epsilon} &= \prod_{1 \leq i \leq n} \frac{n\cdot\epsilon+(1-\epsilon)\cdot(i-1)}{n\cdot\epsilon+(1-\epsilon)\cdot i} = \\ = \frac{(n\cdot\epsilon+(1-\epsilon)\cdot(1-1))\cdot \prod_{1<i\leq n} (n\cdot\epsilon+(1-\epsilon)\cdot(i-1))}{(n\cdot\epsilon+(1-\epsilon)\cdot n)\cdot \prod_{1\leq i<n} (n\cdot\epsilon+(1-\epsilon)\cdot i)} &= \\ = \frac{n\cdot\epsilon\cdot \prod_{1\leq i<n} (n\cdot\epsilon+(1-\epsilon)\cdot i)}{n\cdot \prod_{1\leq i<n} (n\cdot\epsilon+(1-\epsilon)\cdot i)} &= \epsilon \end{aligned} \tag{13}$$

So, strategy p^* fulfills (8) and is a Nash Equilibrium. □

The proof shows that, as soon as the players reach the fair strategy, no user has an incentive to deviate from this strategy. In the next sections, we propose and experimentally evaluate a MARL algorithm for achieving such a behavior.

1. Note that node i may not choose $p_i' > 1 - \epsilon \cdot \frac{1}{\prod_{j \neq i} (1-p_j)}$, since else the first case would hold resulting in utilities of 0.

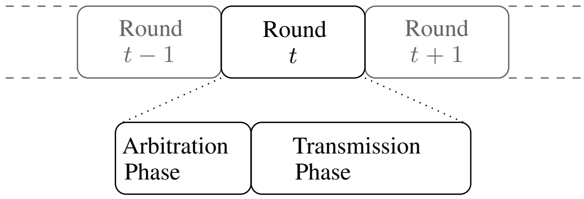


Figure 6. The two phases, *arbitration phase* and *transmission phase*, that define the rounds of the priority-based Access Game.

6. Self-organizing Bandwidth Sharing

The goal of this paper is to establish a self-organizing framework for fair bandwidth sharing in priority-based medium access. Section 4.2 has shown that each user would have to know the overall number of users n as well as the overall placement of its priority, so far denoted by its index i.

Although a system could be designed using this global information, this would result in a very inflexible system. Especially, when adding or removing nodes, the whole setup would have to be updated. The appointed goal of self-organization in distributed systems is to reach a global behavior from some simple local rules. In the context of game theory, this means that each participant tries to maximize its local payoff. Section 4.3, however, has shown that following this strategy would lead to an unfair distribution of bandwidth. Especially, nodes with low priorities could starve out. The theoretical analysis in Section 5 extended the standard game by a bandwidth constraint. When demanding a minimum of free bandwidth, even a system comprised of selfish users could reach an equilibrium state where bandwidth is shared fairly. Still the problem exists that, although the fair strategy results is a Nash Equilibrium, it is a not a unique one.

In this section, we provide a MARL algorithm to lead the system to a fair strategy. Medium access is divided into rounds to adapt the game theoretical framework to the physical communication protocol. This is illustrated in Fig. 6. Each round consists of an *arbitration phase* and a *transmission phase*. In the arbitration phase, all nodes that want to send a message arbitrate the medium. The node with highest priority is granted access and transmits its message in the transmission phase.

Algorithm 1 provides an online protocol for the agents. The goal is to collect local information by observing the environment, i.e., the shared medium. In the arbitration phase, each agent sends or waits depending on its probability. If successful, the message is transmitted in the following transmission phase. Since bandwidth cannot be determined in a single round, each agent repeats the same strategy for an arbitrary interval of rounds. The number of successfully transmitted messages $nSuccess$ (Line 5) as well as the number of overall transmitted messages $nOverallMsgSent$ (Line 9) is counted. After the end of that interval, the agent's

Algorithm 1 Online protocol for adaptive agent

1: **while** (true) **do**
2: $time = time + 1$;
3: arbitrate bus with probability p_i;
4: **if** (access granted) **then**
5: $nSuccess = nSuccess + 1$;
6: transmit the message;
7: **end if**
8: **if** (bus was not free) **then**
9: $nOverallMsgSent = nOverallMsgSent + 1$;
10: **end if**
11: **if** ($time \geq interval$) **then**
12: doLearning();
13: $nOverallMsgSent = 0$;
14: $nSuccess = 0$;
15: $time = 0$;
16: **end if**
17: **end while**

sending probability is updated using a learning algorithm and the counters are reseted (lines 11-16).

Algorithm 2 doLearning()

1: $load = \frac{nOverallMsgSent}{interval}$;
2: $success = \frac{nSuccess}{interval}$;
3: **if** ($load > 1 - \epsilon$) **then**
4: $\Delta = -success$;
5: **else**
6: $\Delta = R^{(t)} - success$;
7: **if** ($\Delta > 0$) **then**
8: $\Delta = \Delta \cdot (1 - p_i)$;
9: **else**
10: $\Delta = \Delta \cdot p_i$;
11: **end if**
12: $R^{(t+1)} = (1 - \gamma) \cdot R^{(t)} + \gamma \cdot success$
13: **end if**
14: $p_i = p_i + \eta \cdot \Delta$;
15: limit(p_i);

The purpose of Algorithm 2 is to adapt the user's sending probability, so that its success rate for accessing the bus is maximized, while trying that all users do not use more than $1 - \epsilon$ of the available bandwidth. As already stated before, we want to use as few global information as possible. In the proposed algorithm, the only data each user has to know is ϵ. The algorithm is based on the *load* on the communication medium and the user's *success rate*. Both properties of the current interval can be calculated as shown in Lines 1 and 2, based on the data provided by Algorithm 1. The size of the interval[2] may be chosen arbitrarily by each user. However, it

2. Note that the interval has to be the same as in Algorithm 1.

has to be sufficiently big to allow a good approximation of the system behavior. The load corresponds to the probability of the medium not being free, i.e., $1 - \prod_{j \in \mathcal{N}} (1 - p_j)$. It is used to determine if there is more bandwidth used than the allowed $1 - \epsilon$. Depending on this, the algorithm follows two different behaviors: If the constraint was violated, a *coordination behavior* is applied which reduces the user's sending probability. Else, a *payoff maximization behavior* is adopted.

The coordination behavior determines Δ which is used to reduce the user's sending probability. The value $success$ corresponds to $p_i \cdot \prod_{j>i} (1 - p_j)$. Therefore, the change in Δ (and thus p_i) is largest when p_i is close to 1 and users with higher priorities did not use any bandwidth. This means that greedy acting users are penalized more than non-greedy ones.

The payoff maximization behavior tries to maximize the user's *expected reward* R which corresponds to its expected success rate. The goal is to maximized the reward for valid outcomes where the bandwidth constraint is not violated. Therefore, the expected reward is only updated when the bandwidth constraint was not violated. We set the initial expected reward to $R^{(0)} = 1 - \epsilon$. Reward maximization is inspired by the *weighted policy learning algorithm* introduced in [12] for 2-player games, and applied for an n-player problem in [13]. The idea of this algorithm is to slow down learning when moving away from a stable strategy, and to speedup learning when moving towards a stable strategy. The idea of this algorithm is to start learning fastest when the gradient changes its direction and then to gradually slow down learning. For this, it calculates the policy gradient Δ which is weighted depending on the user's sending probability.

After determining Δ, the user's sending probability is updated according to Line 14, where η is the *learning rate*. The purpose of limit(p_i) is to ensure that p_i stays within reasonable limits, in our case $[0, 1 - \epsilon]$.

7. Experimental Evaluation

Section 5 has shown that the fair strategy is only one equilibrium of (infinitely) others. This means that we cannot state that the fair strategy is the only outcome a distributed multi-agent system will choose. Therefore, goals of the experiments are to evaluate how the agents self-organize in a simulated environment, and in how far they really converge to a fair outcome. A further aim is to determine how the system scales with the number of users, and how it reacts to system changes, i.e., insertion of additional nodes.

The experimental setup was chosen as shown in Fig. 6. In each round, all users acted according to Algorithm 1. The learning interval was set to 80 time steps, the learning rate was set to $\eta = 0.5$, and the discount factor was set to $\gamma = 0.5$. This parameters have shown good experimental results.

7.1. Convergence and Scalability

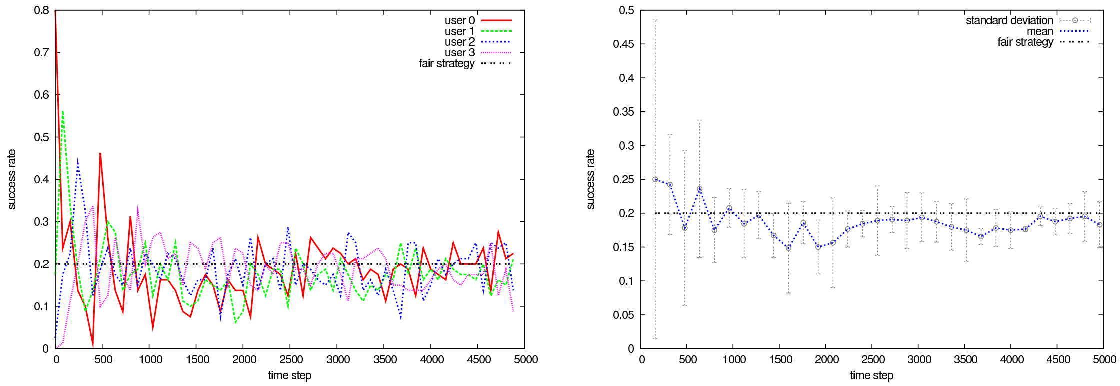
Fig. 7 shows the results of a run with four users, and ϵ set to 0.2. Fig. 7 (a) displays the success rate of the users in each interval. At the start, the users with high priorities, here User 0 and User 1, block the users with lower priorities. Since the bandwidth constraint is violated, they have to reduce their sending probability. At approximately time step 1000, all users oscillate around the fair strategy. Fig. 7 (b) shows the mean bandwidth used by each player as well as the standard deviation. The values were calculated over 160 time steps, i.e., two times the users' learning intervals. The mean bandwidth converges towards the bandwidth used by each user in a fair strategy. The standard deviation decreases over time, showing that the success rate of all users get approximately the same.

This experiment shows two major results. First, the users converge very fast. After three learning intervals, i.e., 240 time steps, no user is blocked by users with higher priorities. The second result is that the algorithm does not converge to a stable strategy. The reason for that is that the utility function is not continuous. When exceeding the bandwidth used, all payoffs drop to 0. Even a small increase of the sending probability may lead to the bandwidth constraint being violated. So, the strategies oscillate around the strategy. Further experiments have shown that the oscillation cannot be reduced by choosing smaller learning rates close to 0. In this case, convergence even takes longer.

Fig. 8 presents the results of a run with 20 users. Fig. 8 (a) shows the success rates of the users for each interval. Starting with the highest user, each user consecutively achieves a high success rate but has to reduce its sending probability for the next learning interval. Fig. 8 (b) shows the mean bandwidth used and the standard deviation. At approximately time step 5000, the users have converged. The algorithm scales nearly linearly compared to the 4-user experiment.

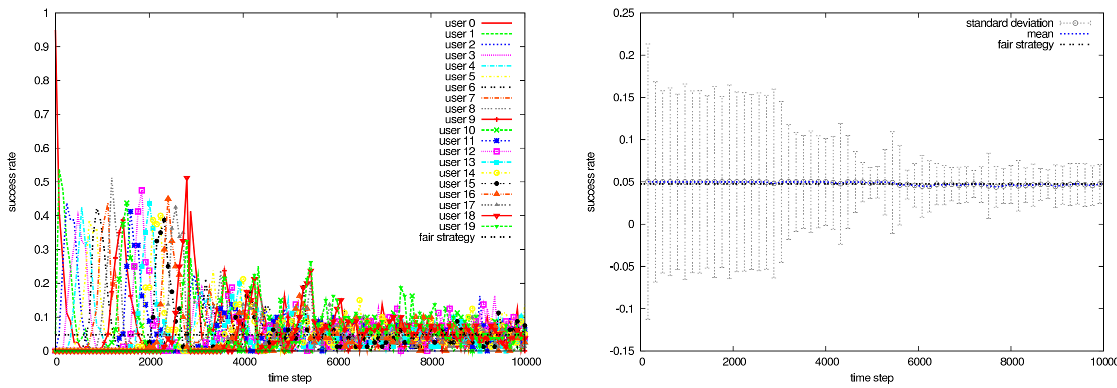
7.2. Flexibility

In further experiments, we have added new users into systems that have already self-organized themselves. Fig. 9 presents the results of an experiment, where the users have already run for 20000 time steps before adding a fifth user with higher priority than the other ones. In reaction to this insertion, all users reduce their sending probability and oscillate around the new fair strategy, where each user gets 16% of the bandwidth. Experiments have shown that it is harder for the system to self-organize when adding a user that has higher priority than the existing ones. This can be seen in this experiment. Although the new user reduces its sending probability very quickly, it takes nearly 100 learning



(a) (b)

Figure 7. Results of the learning algorithm played with 4 users and $\epsilon = 0.2$, (a) shows the success rate of each user in its interval, (b) shows the mean success rate and standard deviation.



(a) (b)

Figure 8. Results of the learning algorithm played with 20 users and $\epsilon = 1/21$, (a) shows the success rate of each user in its interval, (b) shows the mean success rate and standard deviation.

intervals (8000 time steps) before it also converges towards the fair strategy. Fig. 10 presents the results of the same experiment with 20 users. In this much more dynamic setup, the newly added user converges much faster to the fair strategy.

8. Conclusion

In this paper, we have proposed a self-organizing framework for bandwidth sharing in priority-based medium access. The overall goal is to simplify the design of distributed real-time systems and to make them more flexible to system changes. The game theoretic analysis, however, has shown that self-organization is not achievable without modifications. Therefore, the *enhanced priority-based Access Game* was introduced, where a small fraction of the overall bandwidth has to be free. We have shown that for this game a Nash Equilibrium exists that leads to a fair sharing, although it is no unique one. A decentralized MARL algorithm was presented. Here, each user observes the communication on the medium. The only additional information that has to be provided is the amount of bandwidth that has to be free. The experimental evaluation has shown that this algorithm provides self-organizing properties where each user converges towards a fair distribution of bandwidth without the need of further global information or coordination. In future work, we want to continue our analysis. We want to evaluate the behavior when relaxing the assumptions made in Section 4.1. We furthermore want to extend the system to also support the self-organization of real-time critical tasks.

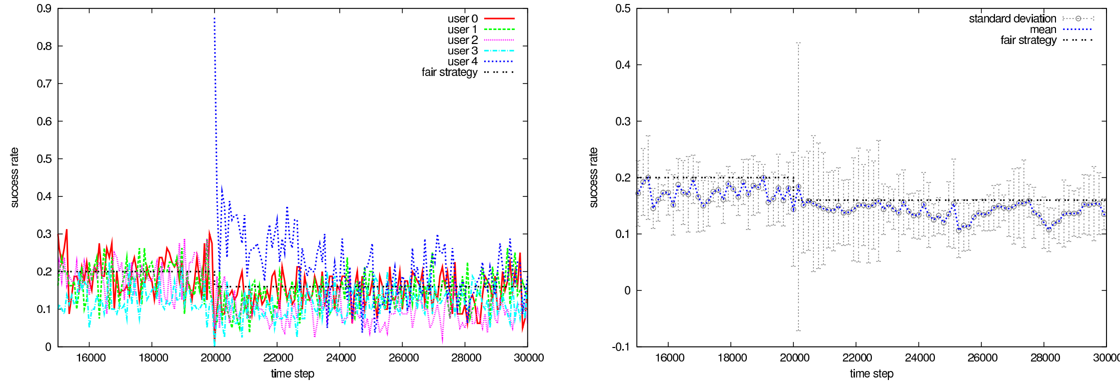
Acknowledgment

This work was supported in part by the German Science Foundation (DFG) in the project under contract TE 163/15-1.

References

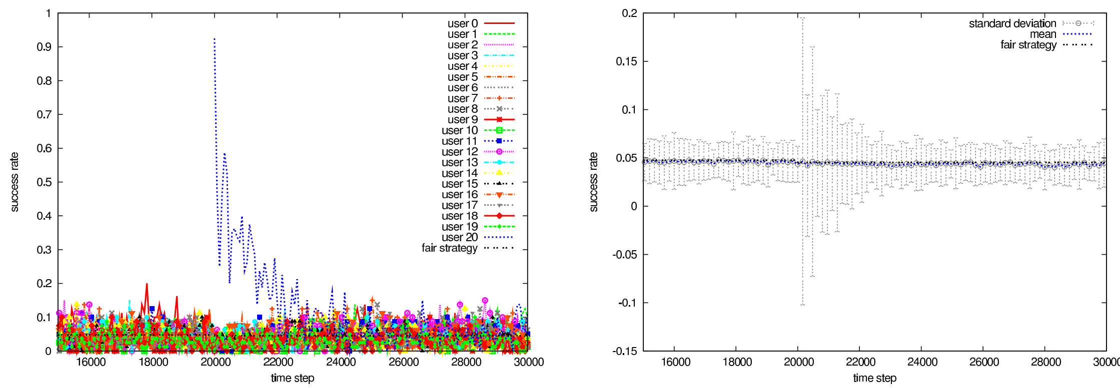
[1] L. Sha, T. Abdelzaher, K.-E. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A. K. Mok, "Real time scheduling theory: A historical perspective," *Real-Time Syst.*, vol. 28, no. 2-3, pp. 101–155, 2004.

[2] CAN, "Controller Area Network," http://www.can.bosch.com/.



(a) (b)

Figure 9. Results of 4 users experiment and a new player with higher priority being added at time step 20000.



(a) (b)

Figure 10. Results of 20 users experiment and a new player with higher priority being added at time step 20000.

[3] K. Tindell, H. Hansson, and A. Wellings, "Analysing Real-Time Communications: Controller Area Network (CAN)," in *Proc. 15th IEEE Real-Time Systems Symposium*. San Juan, Puerto Rico: IEEE Society Press, December 1995.

[4] L. Chen, T. Cui, S. H. Low, and J. C. Doyle, "A Game-Theoretic Model for Medium Access Control," in *Proceedings of International Wireless Internet Conference (WICON)*, Austin, TX, Oct. 2007, (invited). [Online]. Available: http://netlab.caltech.edu/publications/rag07.pdf

[5] Z. Fang and B. Bensaou, "Fair Bandwidth Sharing Algorithms based on Game Theory Frameworks for Wireless Ad-hoc Networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2004, pp. 1284–1295.

[6] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs," in *SIGCOMM*, 2005, pp. 121–132.

[7] S. Rakshit and R. K. Guha, "Fair Bandwidth Sharing in Distributed Systems: A Game-Theoretic Approach," *IEEE Transactions on Computers*, vol. 54, no. 11, pp. 1384–1393, 2005.

[8] A. Fiat, Y. Mansour, and U. Nadav, "Efficient contention resolution protocols for selfish agents," in *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 179–188.

[9] J. Nash, "Non-Cooperative Games," *The Annals of Mathematics, Second Series*, vol. 54, no. 2, pp. 286 – 295, 1951.

[10] A. Davare, Q. Zhu, M. Di Natale, C. Pinello, S. Kanajan, and A. Sangiovanni-Vincentelli, "Period optimization for hard real-time distributed automotive systems," June 2007, pp. 278–283.

[11] L. Busoniu, R. Babuska, and B. De Schutter, "A Comprehensive Survey of Multiagent Reinforcement Learning," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 2, pp. 156–172, March 2008.

[12] S. Abdallah and V. Lesser, "Learning the task allocation game," in *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2006, pp. 850–857.

[13] ——, "Multiagent reinforcement learning and self-organization in a network of agents," in *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2007, pp. 1–8.