Organic Self-organizing Bus-Based Communication Systems (OrganicBus)

Tobias Ziermann, Stefan Wildermann, Prof. Dr. Jürgen Teich

Hardware-Software-Co-Design

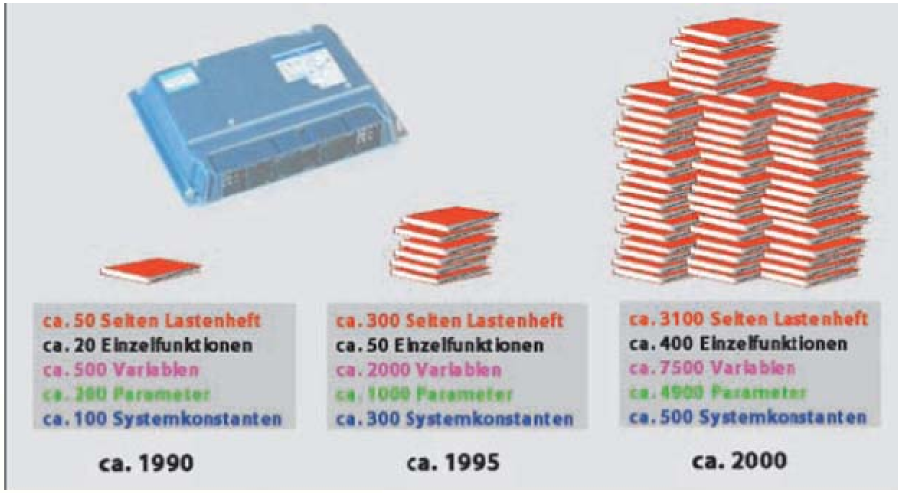
Universität Erlangen-Nürnberg

tobias.ziermann@informatik.uni-erlangen.de

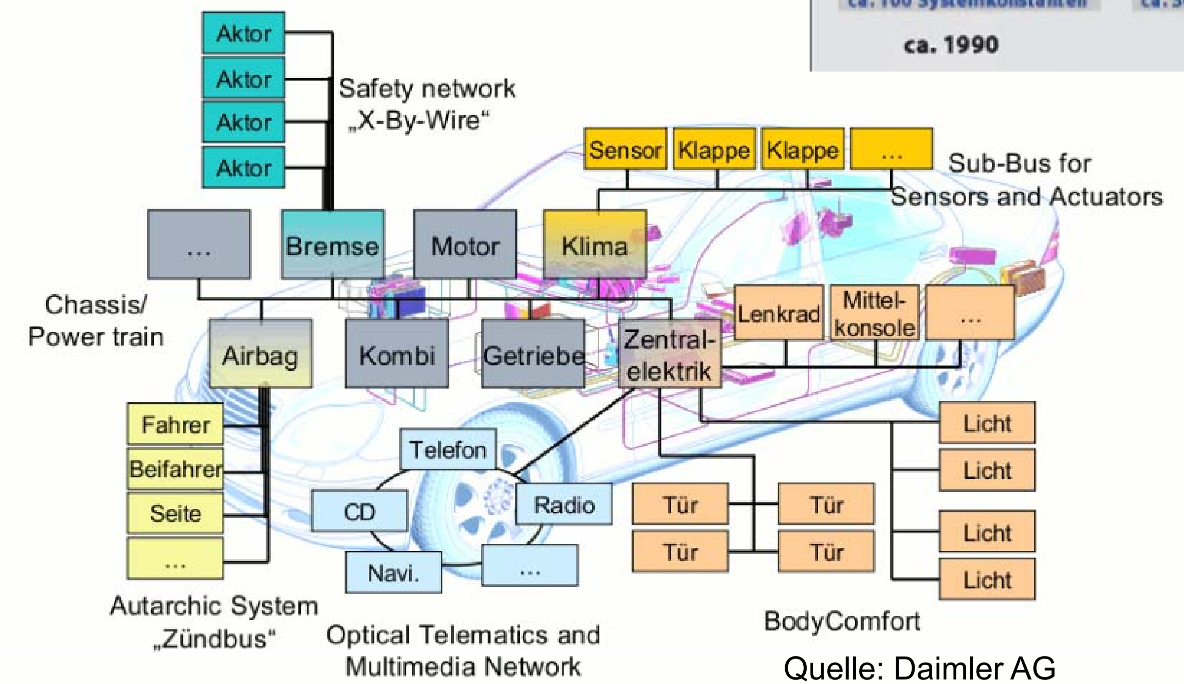
22.10.09

Motivation and Idea

- Increasing complexity in distributed embedded systems
- Static bus arbitration too restricted or suboptimal



Quelle: VW



Quelle: Daimler AG

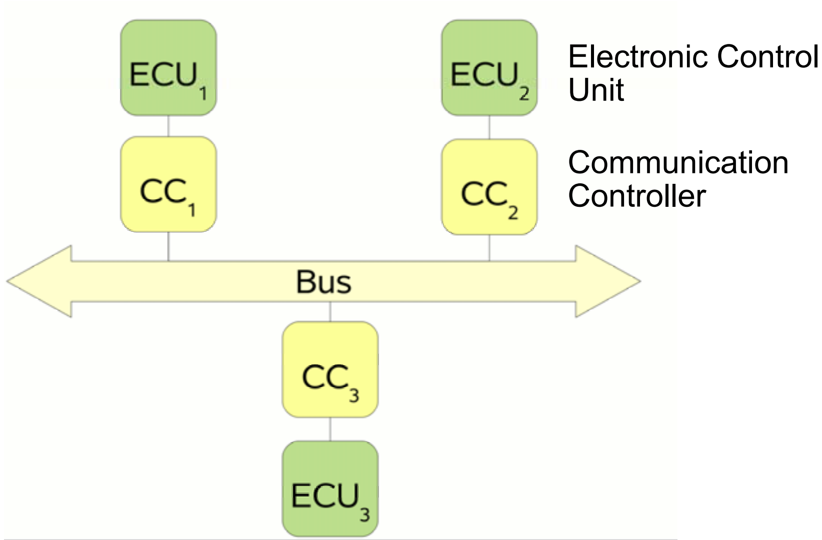
- Potential for optimization through *self-organizing bus systems (OrganicBus)*

Agenda

- Goals
- State of the art
- Roadmap
- First Milestones
- Conclusion

Goals

- Organic approach for bus-based communication systems
- Desired properties:
  - Decentralized arbitration
  - Self-organization
  - Self-healing
  - Self-optimization of bus access and traffic



- Separation of function and communication
- Reach the above properties using **simple local rules**
- Overcome the drawbacks of pure offline optimization

Major Goal: Increase overall Quality

- Typical types of messages to distinguish:
  - Hard deadline
  - Soft deadline
  - Bandwidth

- Definition of Quality (Objectives - in order of relevance):
  - Satisfaction of safety critical requirements
  - Increase of number of fulfilled constraints
  - Improvement of bus utilization
  - Guarantee of fairness

Features of our Decentralized Approach

- Apply multi-agent reinforcement learning techniques
- Apply game theoretic analysis for validation of protocols
- Use observer/controller architecture to preserve real-time properties
- Simulation and real hardware implementation
- Advantages:
  - Approach flexible to system changes
  - Robust to single node failures
  - Scalable

State of the Art

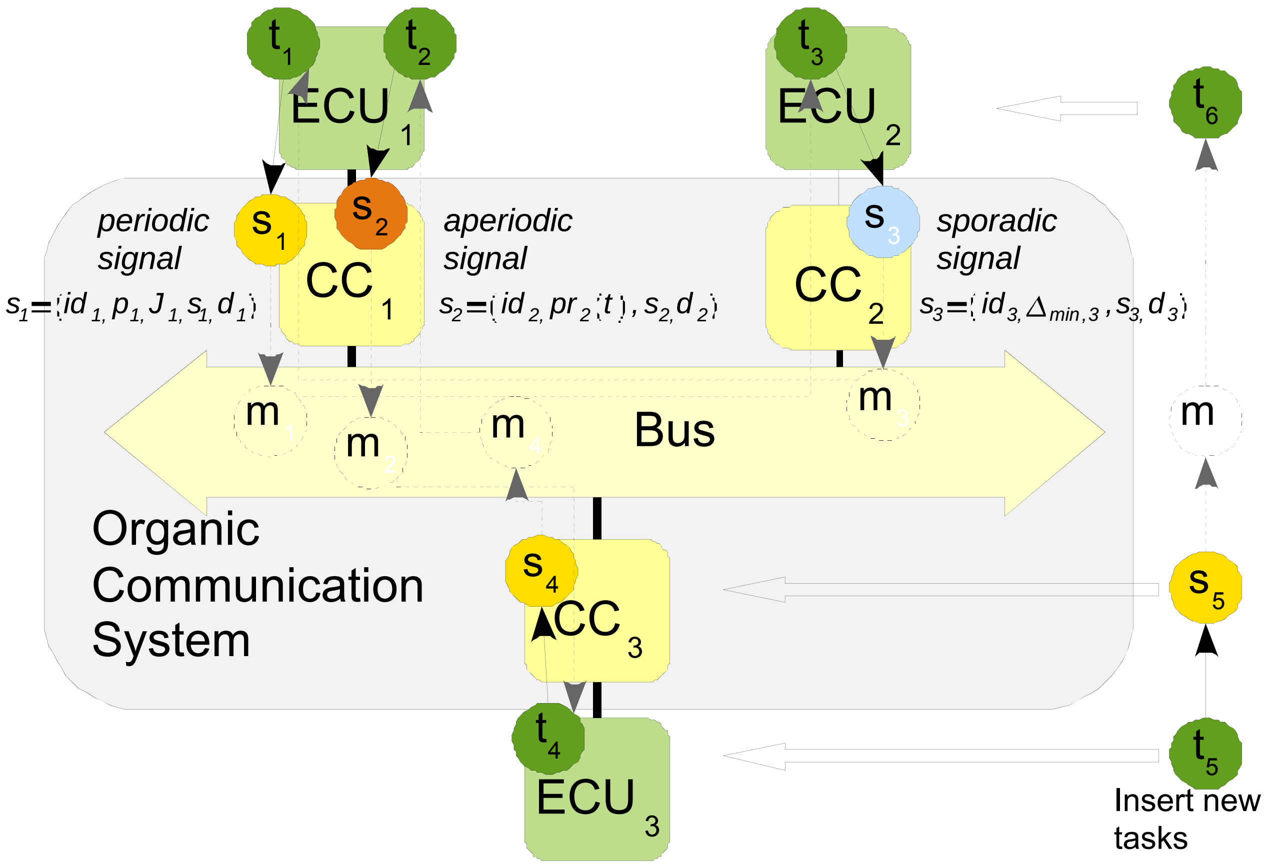
- Internet uses simple decentralized mechanisms, e.g. TCP
  - Fulfill requirements
  - Difficult to predict
- Embedded performance analysis for organic computing (EPOC) [Stein, Ernst]
  - Starting from offline solution
  - Hard real-time
- Feedback scheduling
  - Control loops optimize schedule
  - Centralized scheduling mechanism

Roadmap

1. Definition of a system model
2. Game-theoretic analysis
3. Development and rating of algorithms
4. Physical simulation
5. Implementation on real hardware

Definition of a System Model

- Separation between application and communication
- Describe messages by properties



Game-Theoretic Analysis

- Game Theory as a formal tool to define and analyze multi-agent decision problems

- A **game** consists of
  - a set of **players** $N = \{1,\ldots, n\}$
  - a set of **strategies** S_i available to each player *i*
  - a specification of **payoffs** for each combination of strategies

- Payoff of player i: $u_i(\mathbf{s})$
  - strategies taken by all *n* players: $\mathbf{s} = (s_1,\ldots,s_n)$
  - strategy of player *i*: $s_i \in S_i$

Equilibria

- *Nash Equilibrium* is an important characteristic to evaluate stable states in games

- When all players are selfish, a Nash Equilibrium is reached.

- *Selfish players* try to maximize their own payoff.

- No player achieves a higher payoff when choosing another strategy than his current strategy, when the other players keep their current strategies

Medium Access as 2-Player Game

- Models contention phase of medium access where users content for medium access.
- Players have the strategies {*send*, *wait*}
- The payoff is
  - 1, if granted access to the medium
  - 0, else

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

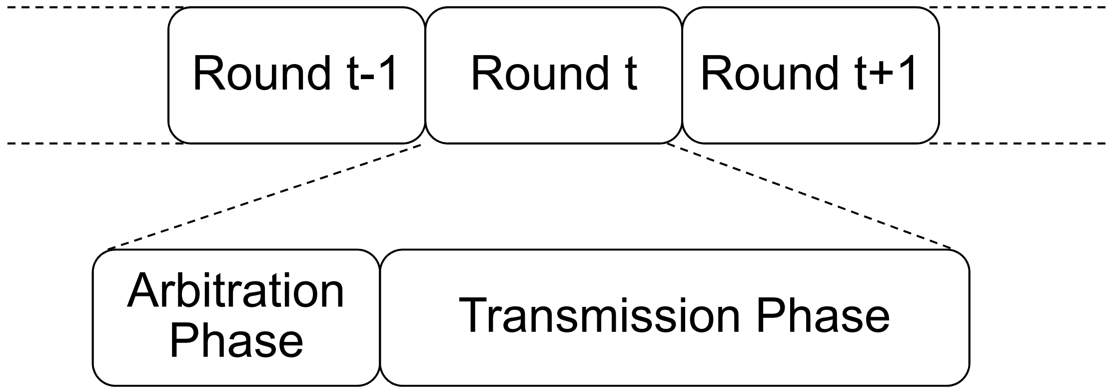
Contention-based access (e.g., WLAN), Analysis in [1]

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

Priority-based Medium Access with Player 1 having higher priority

Priority-based Access Game

- Arbitration and transmission phase define the rounds of the priority-based Access Game



- Mixed Strategies:
  - Not playing with discrete strategies "*wait*" and "*send*".
  - Strategies are given as a probability distribution.
- Player *i sends* with probability p_i and *waits* with $1 - p_i$
- Payoff $u_i(\mathbf{s})$: Percentage of successfully sent messages

Unfairness of Selfish Users

- Goal is to achieve a *fair bandwidth sharing*, i.e.,

$$u_1(\mathbf{p}) = u_2(\mathbf{p}) = \ldots = u_n(\mathbf{p})$$

- However, selfish behavior leads to unfair bandwidth sharing and blocking of users with low priorities.
- **Example:** 2-Player case

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

Nash Equilibria

Strategy "send" dominates strategy "wait" for Player 1

→ Player 1 always chooses "send"

→ Player 2 never sends

Enhancement to reach Fairness

- Main problems:
  - Users with low priorities cannot influence users with higher priorities.
  - Users with high priorities do not realize when blocking an user with lower priority.

- Idea: Demand that a small amount ε of the available bandwidth stays free.

- Payoff $u_i(\mathbf{s})$:
  - If sum of sending probabilities is less then 1- ε then percentage of successfully sent messages
  - Else 0

Analysis of the Nash Equilibrium

- It can be shown that a fair bandwidth sharing is one Nash Equilibrium of the enhanced game:

$$u_1(\mathbf{p}) = u_2(\mathbf{p}) = \ldots = u_n(\mathbf{p}) = \frac{1-\varepsilon}{n}$$

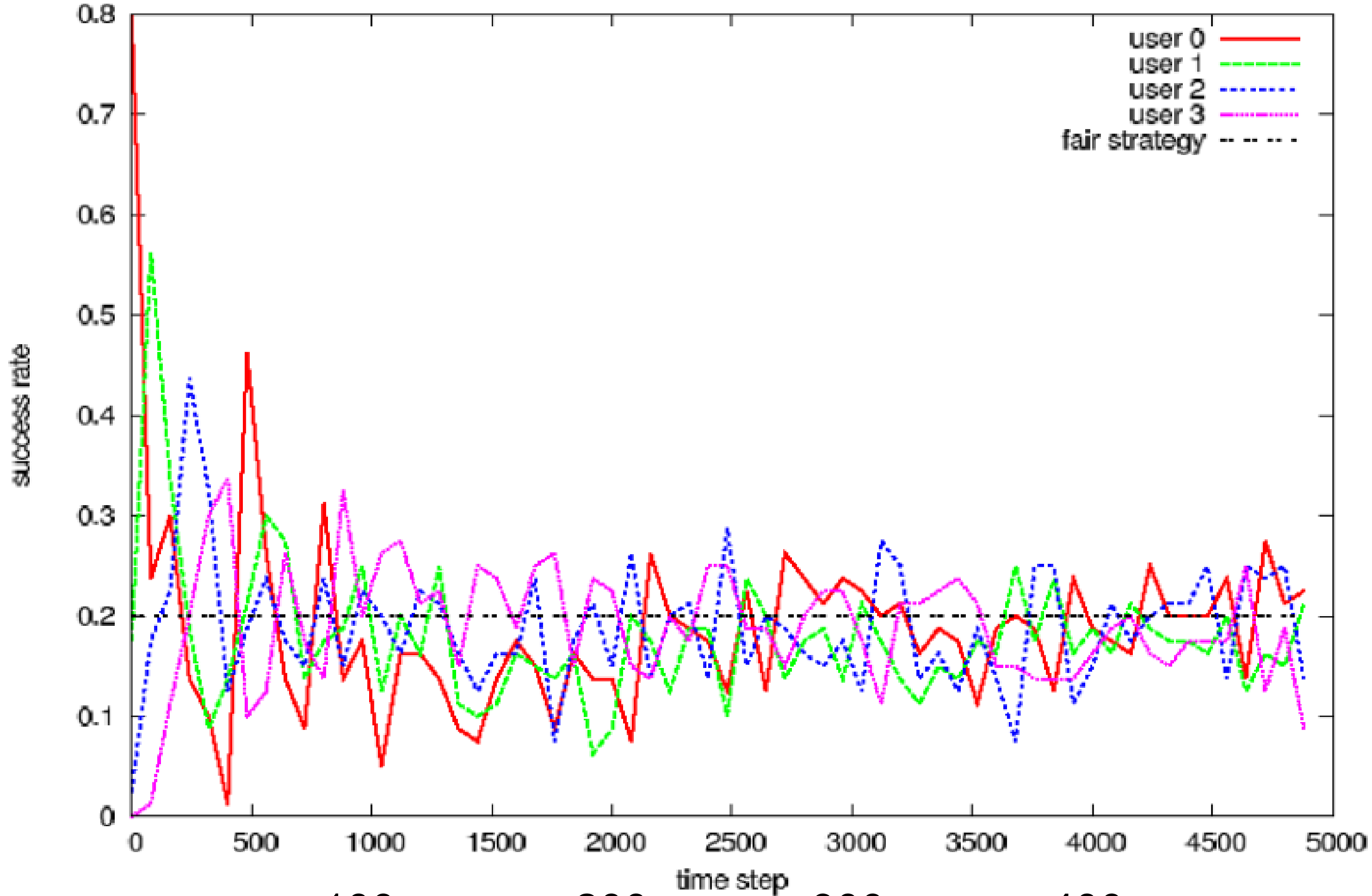
- This means, it is possible to design a self-organizing system for fair bandwidth sharing.

- However, the Nash Equilibrium is not a unique one.

- Providing a Multi-Agent Reinforcement Learning algorithm to lead the system to a fair strategy

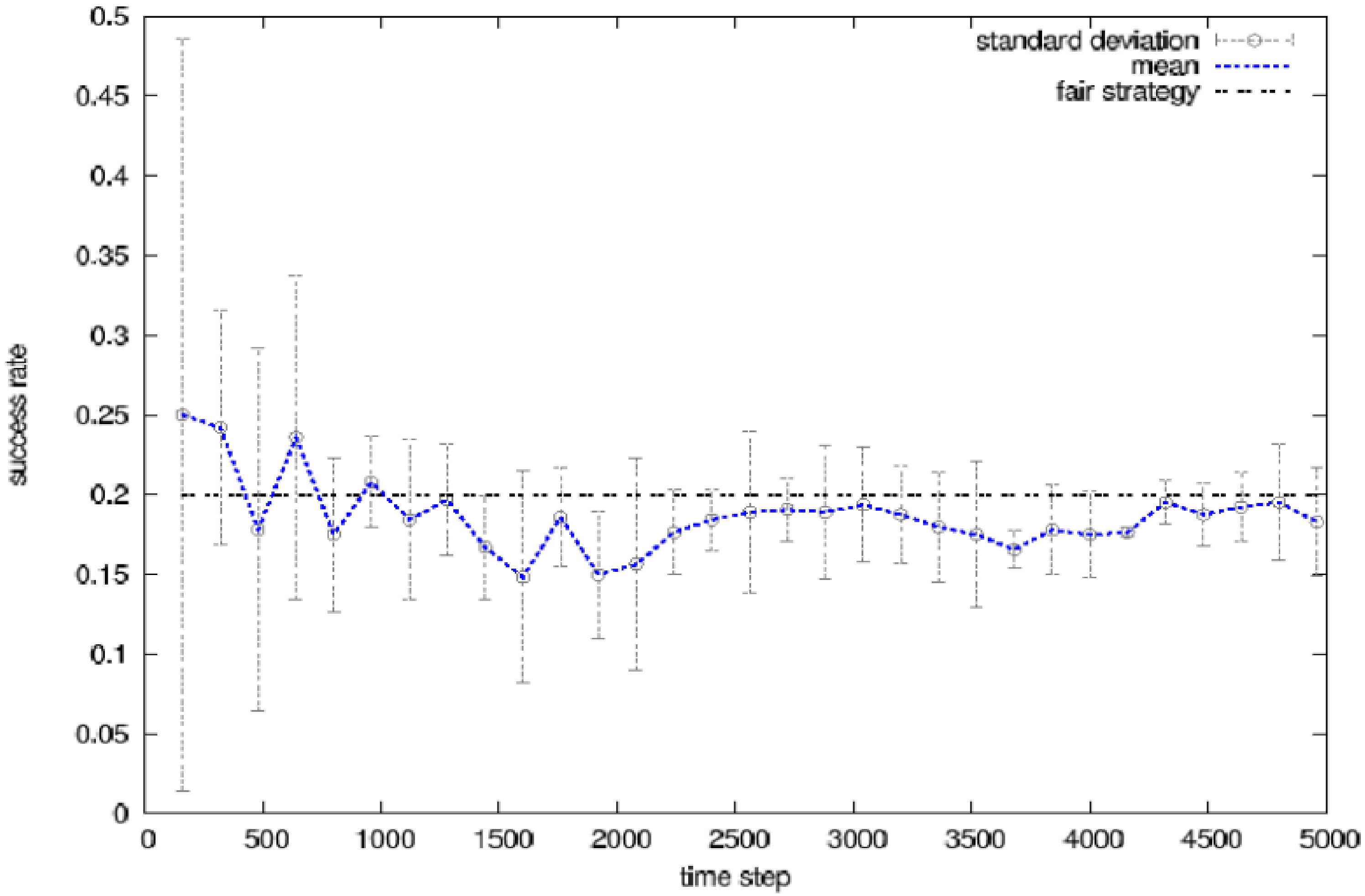
Experimental Results – 4 Users

Success rates (= bandwidth used) of 4 Players and ε = 0.2



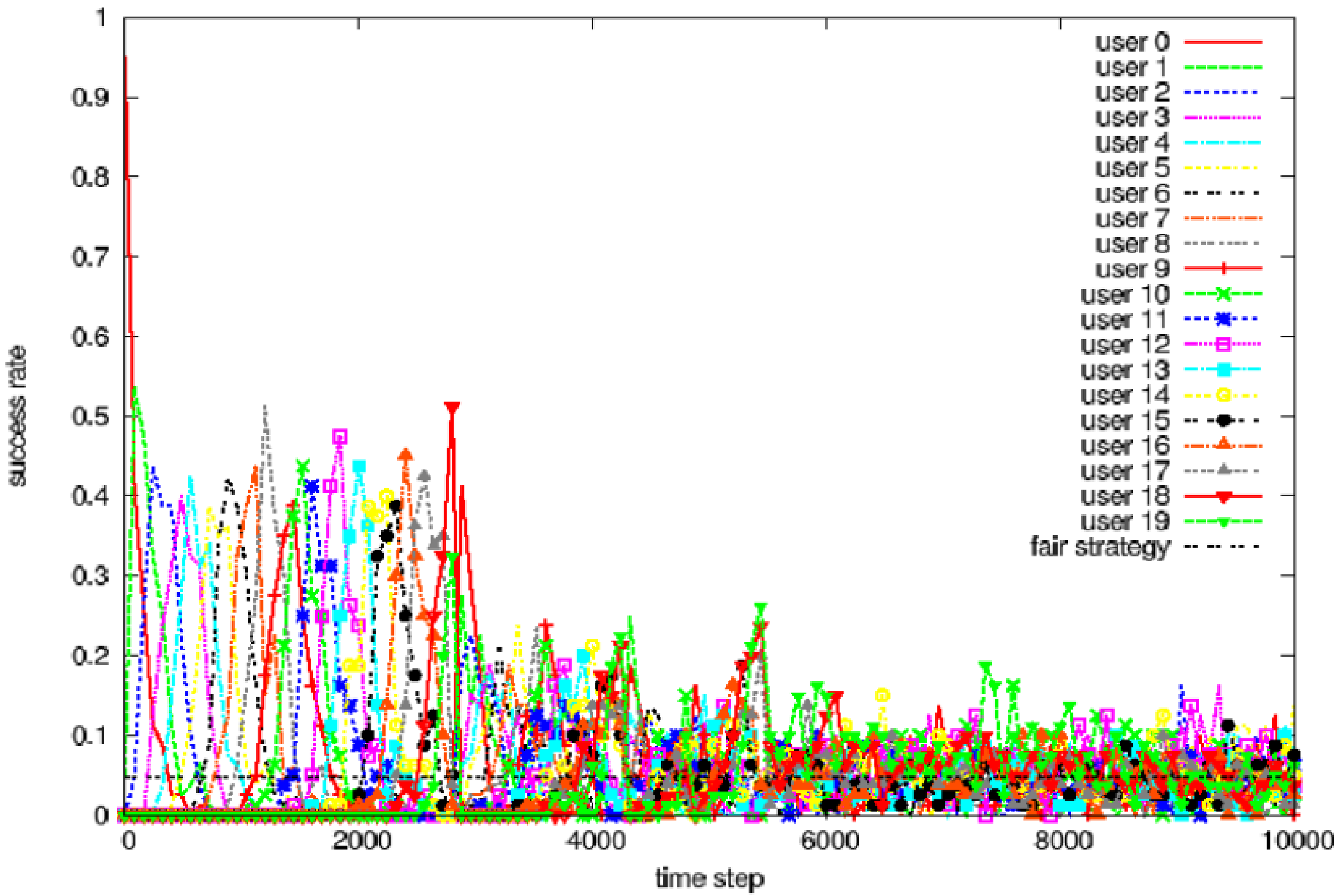
100ms 200ms 300ms 400ms

Experimental Results – 4 Users



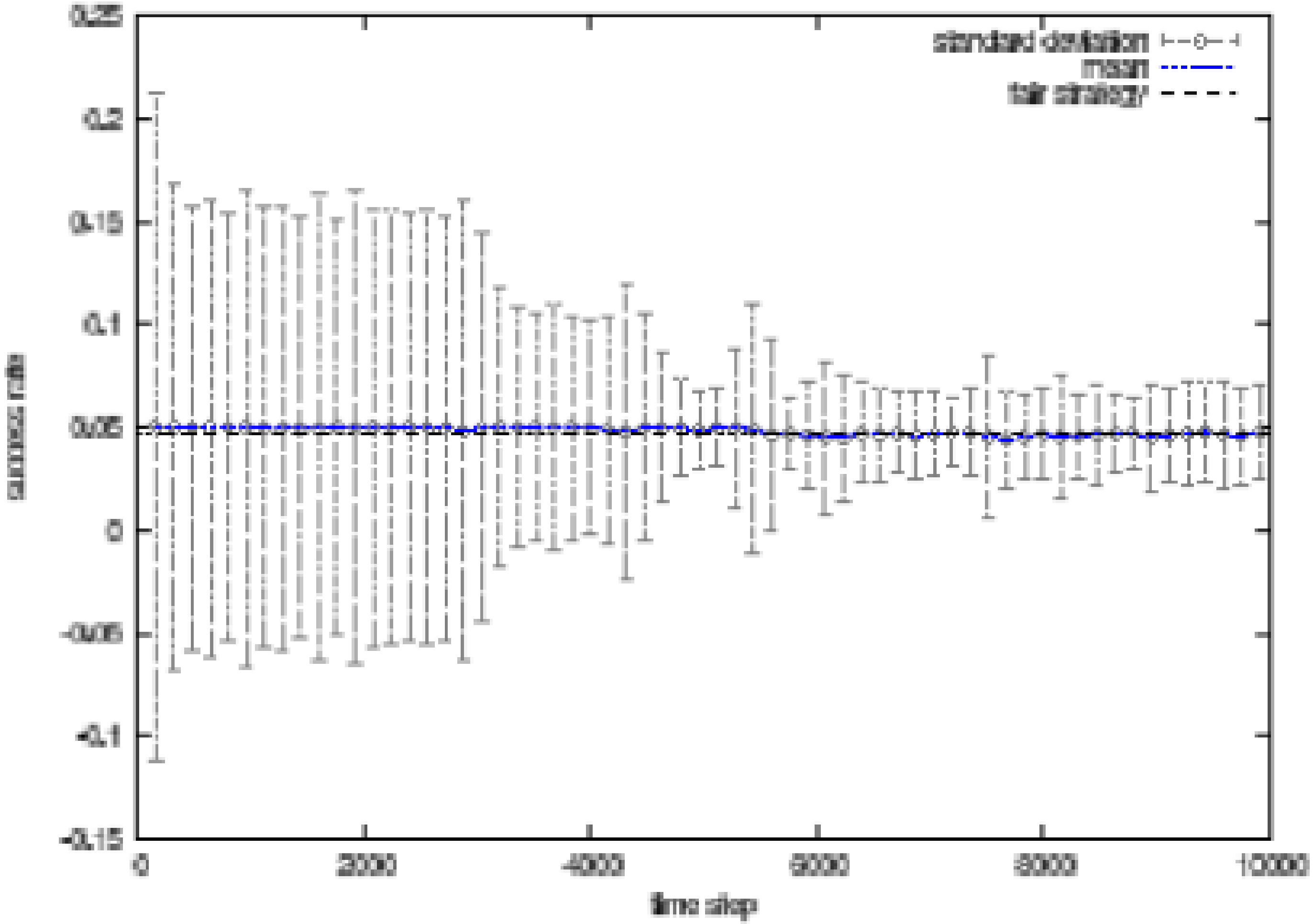
mean success rate with standard deviation

Experimental Results – 20 Users



20 Players and ε = 1/21. Mean success rate and standard deviation.

Experimental Results – 20 Users



20 Players and ε = 1/21. Mean success rate and standard deviation.

Roadmap

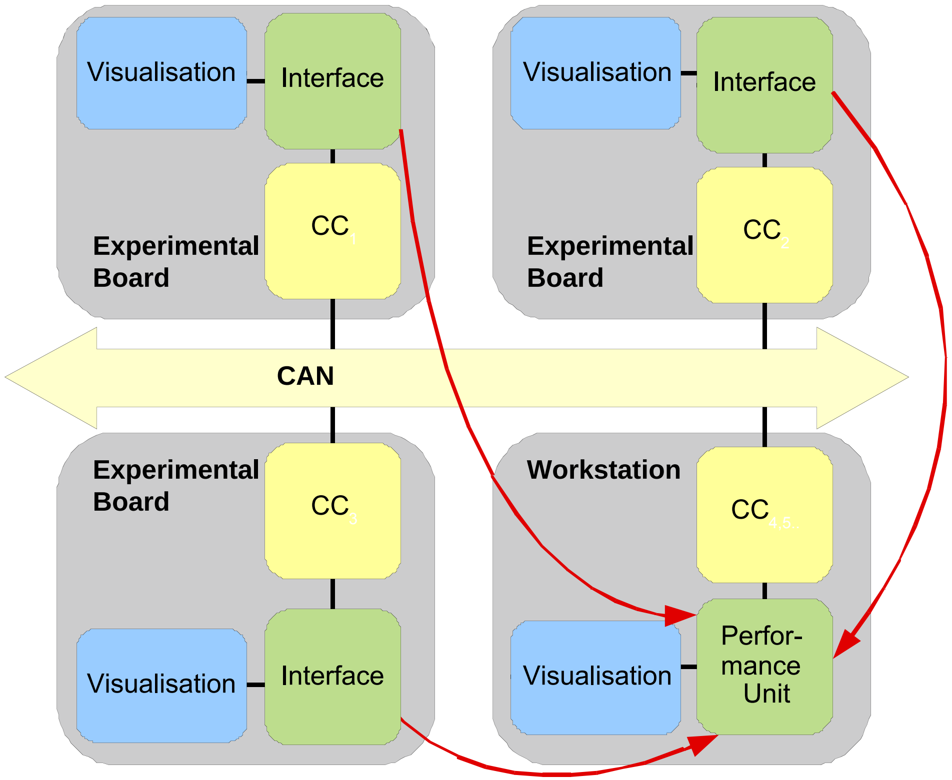
1. Definition of a system model
2. Game-theoretic analysis
3. Development and rating of algorithms
4. Physical simulation
5. Implementation on real hardware

Physical Simulation

- First step towards real implementation
- Analyze overhead of communication protocol
- Consideration of asynchronous communication with Controller Area Network (CAN)

Implementation on Real Hardware

- Implementation of organic bus arbitration algorithms on FPGA-boards
- Connecting boards and workstation via CAN
- Additional point-to-point connections for debugging purpose



Conclusions

- 3 Steps to achieve *Organic Self-organizing Bus-Based Communication Systems:*
  - Theoretical foundation (game theory)
  - Models and learning techniques
  - Real hardware implementation

Thanks for your Attention

- Project page:
  - www12.informatik.uni-erlangen.de/research/organicbus/
- Contact:
  - Tobias Ziermann
  - tobias.ziermann@informatik.uni-erlangen.de
  - www12.informatik.uni-erlangen.de/people/ziermann

Online Protocol for Adaptive Agents

- Collect local information by observing the environment, i.e., the shared medium, for some rounds.
  - *SuccessRate*: the rate of successfully arbitrating the medium
  - *ContentionRate*: the rate the medium was not free

- Update the sending probability using a learning algorithm:

  Let Δ be the change of the probability

  IF *ContentionRate* > 1 - ε **THEN**

  // Coordination behavior behavior

  Δ = -*SuccessRate*;

  ELSE

  // Payoff maximization behavior

  Apply weighted policy learning algorithm inspired by [2];

  END IF

References

[1] Sudipta Rakshit; Guha, R.K., **"Fair bandwidth sharing in distributed systems: a game-theoretic approach,"** *Computers, IEEE Transactions on* , vol.54, no.11, pp. 1384-1393, Nov. 2005

[2] Abdallah, S. and Lesser, V. 2006. **"Learning the task allocation game,"** *Proceedings of the Fifth international Joint Conference on Autonomous Agents and Multiagent Systems*. AAMAS '06. ACM, New York, NY, 850-857.