Organic Self-organizing Bus-based Communication Systems

Tobias Ziermann, Stefan Wildermann, Jürgen Teich

Hardware-Software-Co-Design

Universität Erlangen-Nürnberg

tobias.ziermann@informatik.uni-erlangen.de

15.09.2011
Motivation

- Increasing complexity in distributed embedded systems
- Increasing demand on the communication
- Wired buses are used today

Source: Volkswagen

Source: Daimler AG

Source: Heidelberger Druckmaschinen AG
Goals of OrganicBus

- Planning of the communication is very difficult
  - Hand-based procedures are not practical
  - Design tools are pessimistic

- Solution: Organic Computing approach for priority-based bus communication:
  - Decentralized
  - Self-organizing
  - Self-optimizing
  - ...

- Idea: Decentralized run-time communication scheduling using **simple local rules**
Properties of Distributed Systems

- **Constraints of messages:**
  - Hard deadline
  - Soft deadline
  - Bandwidth

- **Occurrence of messages:**
  - Periodic
  - Sporadic
  - Bandwidth

- **Increase overall quality:**
  - Satisfaction of safety-critical requirements
  - Increase of number of fulfilled constraints
  - Improvement of bus utilization
  - Guarantee of fairness
Outline

- Motivation and Goals

- Bandwidth sharing
  - Penalty Learning Algorithm (PLA)
  - Results

- Response time reduction
  - Dynamic Offset Adaptation Algorithm (DynOAA)
  - Results

- Summary and Outlook
Problem Description

- Several nodes try to stream with maximum bandwidth
- Goal: Every node should get equal bandwidth
- Priority-based access unsuitable

Description as a Game:
- Set of Players
- Set of Strategies
- Payoff for each combination of played strategies

<table>
<thead>
<tr>
<th></th>
<th>Player 1</th>
<th></th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>wait</td>
<td></td>
<td>wait</td>
<td></td>
</tr>
<tr>
<td>0,0</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>send</td>
<td>1,0</td>
<td>send</td>
<td>1,0</td>
</tr>
<tr>
<td></td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Friedrich-Alexander-Universität Erlangen-Nürnberg
Tobias Ziermann
Solution

- Extension of the Game:
  - Sending probability is strategy
  - Demand that a small amount $\varepsilon$ of the available bandwidth always stays free.

- Payoff:
  - If sum of sending probabilities is less then $1 - \varepsilon$, then return percentage of successfully sent messages
  - Else return 0

- Fair bandwidth distribution is Nash equilibrium (Proof)

- But not the only one

- Development of multi-agent reinforcement learning algorithm: Penalty Learning Algorithm (PLA)
Results

The diagram illustrates the bandwidth used over a series of rounds. The x-axis represents the number of rounds, while the y-axis shows the bandwidth used. Two players are compared: player 1 is represented by a red line, and player 2 by a green dotted line. The bandwidth usage stabilizes after a certain number of rounds, with player 1 consistently using a higher bandwidth than player 2.
Results (20 Player)
Probabilistic/Periodic Access Method

- Probabilistic: Time independent representation
- Periodic: Deterministic behavior

(a) PLA with $\eta = 0.02$

Probabilistic: Time independent representation
Outline

- Motivation and Goals

- Bandwidth sharing
  - Penalty Learning Algorithm (PLA)
  - Results

- Response time reduction
  - Dynamic Offset Adaptation Algorithm (DynOAA)
  - Results

- Summary and Outlook
Problem Description

- Properties of control oriented communication:
  - Periodic messages with soft deadline
  - But short response times
  - Limited data rate

- Controller Area Network (CAN) widely used
  - Priority-based event-triggered access method

- Problem: Response times increase with workload

- Reason: On concurrent access messages with low priority get delayed
Solution

- Scheduling of messages to avoid concurrent access

- Example:
Given a set of streams that periodically send messages

Worst case response time (WCRT) is largest observed message delay during a given interval of time
Goal

- Find offsets to reduce WCRT
- Online algorithm because streams are asynchronous
Rating Approach

- Single-processor task scheduling:
  - Binary schedulability criterion for hard real-time tasks not applicable
- Diagram of the WCRTs of all streams
- Our approach: Rating function

\[ r(t) = \frac{\sum_{i=1}^{k} \frac{WCRT_i(t-P,t)}{T_i}}{k} \]
Dynamic Offset Adaptation Algorithm (DynOAA)

- Run on each node independently and forever:
  1. Monitor current bus communication
  2. Decide whether to adapt
  3. Adapt according to monitoring information
Simulation

- Evaluation by simulation

- Bit-accurate CAN simulator
  - Error free case
  - Worst-case bit stuffing
  - Synchronous simulation
  - Integrated online adaptation

- Test scenarios from Netcarbench (http://www.netcarbench.org/)
  - Typical automotive scenarios
  - 125 kbit/s data rate
  - Workload ranging from 50% to 90%
Results

- Rating over time with 10 random initial offsets for different scenarios

Graph showing the rating function over time for different load scenarios:
- 90% load; 269 streams
- 80% load; 254 streams
- 70% load; 202 streams
- 60% load; 173 streams
- 50% load; 145 streams
Adaptation to Changing System

- Simulation shows robustness to changing system during run-time
Multi-segment System Model

- Stream model extended by a source bus and a set of destination buses

- Central gateway:
  - Delays neglected
  - Priority-based access
  - Immediate start of retransmission after full reception
Multi-segment

- Difference: Handling of routed streams as non-adapting streams

- Modified algorithm to allow partial adaptation

- Scenarios are generated from single-segment scenarios:
  - Assigning source streams uniformly
  - Routing

- Preliminary results show the performance of DynOAA in multi-segment systems
Results

- Rating over time for different number of segments where all streams are routed to all other segments

![Graph showing rating over time for different number of segments](image-url)
Integration of All Approaches

- **Hard deadline**
  - Highest priorities
  - Analytical approach, e.g. EPOC

- **Soft deadline**
  - Periodic: DynOAA
  - Sporadic: Priority access

- **Bandwidth**
  - Lowest priority
  - PLA
Outlook

- Implement the algorithms on real hardware
  - Analyze overhead of organic bus protocol
  - Consideration of asynchronous communication with Controller Area Network (CAN)
  - Provide prototype and demonstrator

- Considered Platforms:
  - Standard PC
  - Prototype on FPGA
    - Softcore processor
    - Pure hardware
Preliminary Results

- Blue line: 50% load; 178 Streams; worst-case offsets
- Black line: 90% load; 271 Streams; random offsets
Summary

- Modeling and analysis of decentralized bus bandwidth allocation algorithms using game theory

- Development and simulation of two algorithms:
  - Penalty Learning Algorithm for bandwidth constraints
  - Dynamic Offset Adaptation Algorithm for soft real-time constraints

- Decentralized approach avoids single point of failure

- Online adaptation allows adjustment to current traffic
  - Allows higher utilization of bus

- Prototype will provide proof of concept
Thanks for your attention

- **Project page:**
  - [www12.informatik.uni-erlangen.de/research/organicbus/](http://www12.informatik.uni-erlangen.de/research/organicbus/)

- **Contact:**
  - Tobias Ziermann
  - tobias.ziermann@informatik.uni-erlangen.de
  - [www12.informatik.uni-erlangen.de/people/ziermann](http://www12.informatik.uni-erlangen.de/people/ziermann)