An Improved Scheduling Technique for Time-Triggered Embedded Systems

Paul Pop, Petru Eles, Zebo Peng

Dept. of Computer and Information Science
Linköping University
Sweden

Outline

- Motivation
- System Architecture
- Problem Formulation
- Scheduling Strategy
- Experimental Results
- Conclusions

Motivation

- **Embedded System Design.**
- **Scheduling, Communication, Bus Access.**

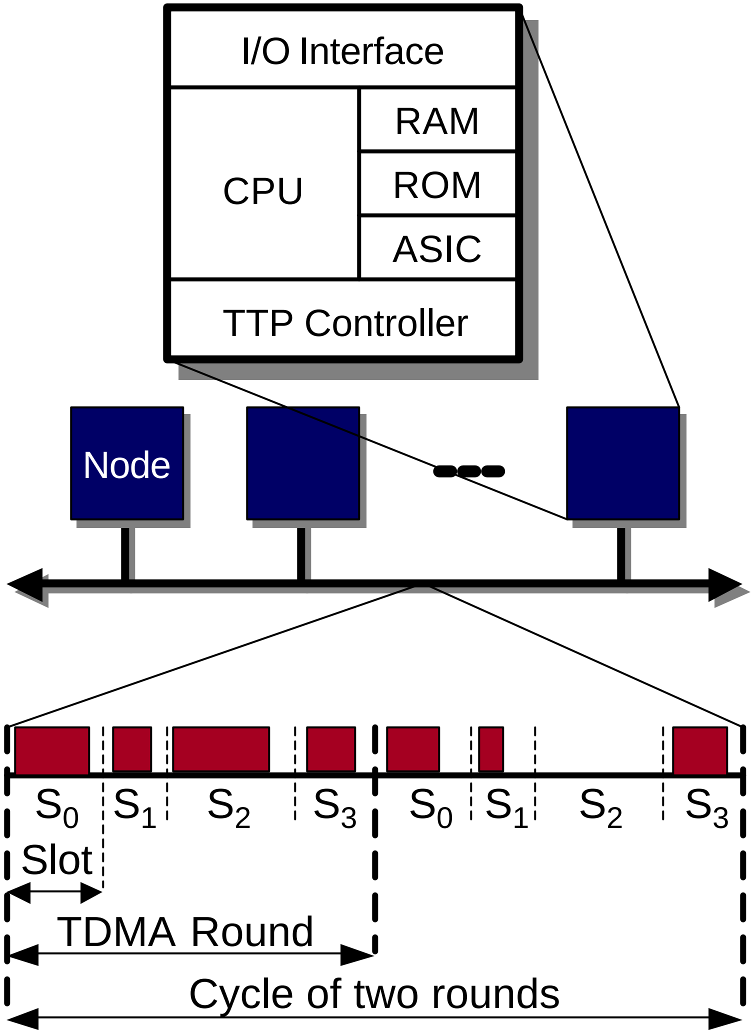
Characteristics:

- **Static nonpreemptive scheduling.**
- **System model captures both the flow of data and that of control.**
  Eles et al. Scheduling of Conditional Process Graphs for the Synthesis of Embedded Systems. DATE'98
- **Heterogeneous system architecture.**
- **Communications using the time-triggered protocol (TPP).**
  Kopetz, H., Grünsteidl, G. TTP-A Protocol for Fault-Tolerant Real-Time Systems. IEEE Computer '94

Message:

- **Improved schedule quality by considering the characteristics of the communication protocol.**

Hardware Architecture



- **Safety-critical distributed embedded systems.**
- **Nodes interconnected by a broadcast communication channel.**
- **Nodes consisting of: TTP controller, CPU, RAM, ROM, I/O interface, (maybe) ASIC.**
- **Communication between nodes is based on the time-triggered protocol.**

- **Bus access scheme: time-division multiple-access (TDMA).**
- **Schedule table located in each TTP controller: message descriptor list (MEDL).**

Software Architecture

- Real-Time Kernel running on the CPU in each node.
- There is a local schedule table in each kernel that contains all the information needed to take decisions on activation of processes and transmission of messages.
- Time-Triggered System: no interrupts except the timer interrupt.
- The worst case administrative overheads (WCAO) of the system calls are known:

U_t	WCAO of the timer interrupt routine
d_{PA}	process activation overhead
d_S	overhead for sending a message on the same node
d_{KS}	overhead for sending a message between nodes
d_{KR}	overhead for receiving a message from another node

Problem Formulation

Input

- Safety-critical application with several operating modes.
- Each operating mode is modelled by a conditional process graph.
- The system architecture and mapping of processes to nodes are given.
- The worst case delay of a process is known:

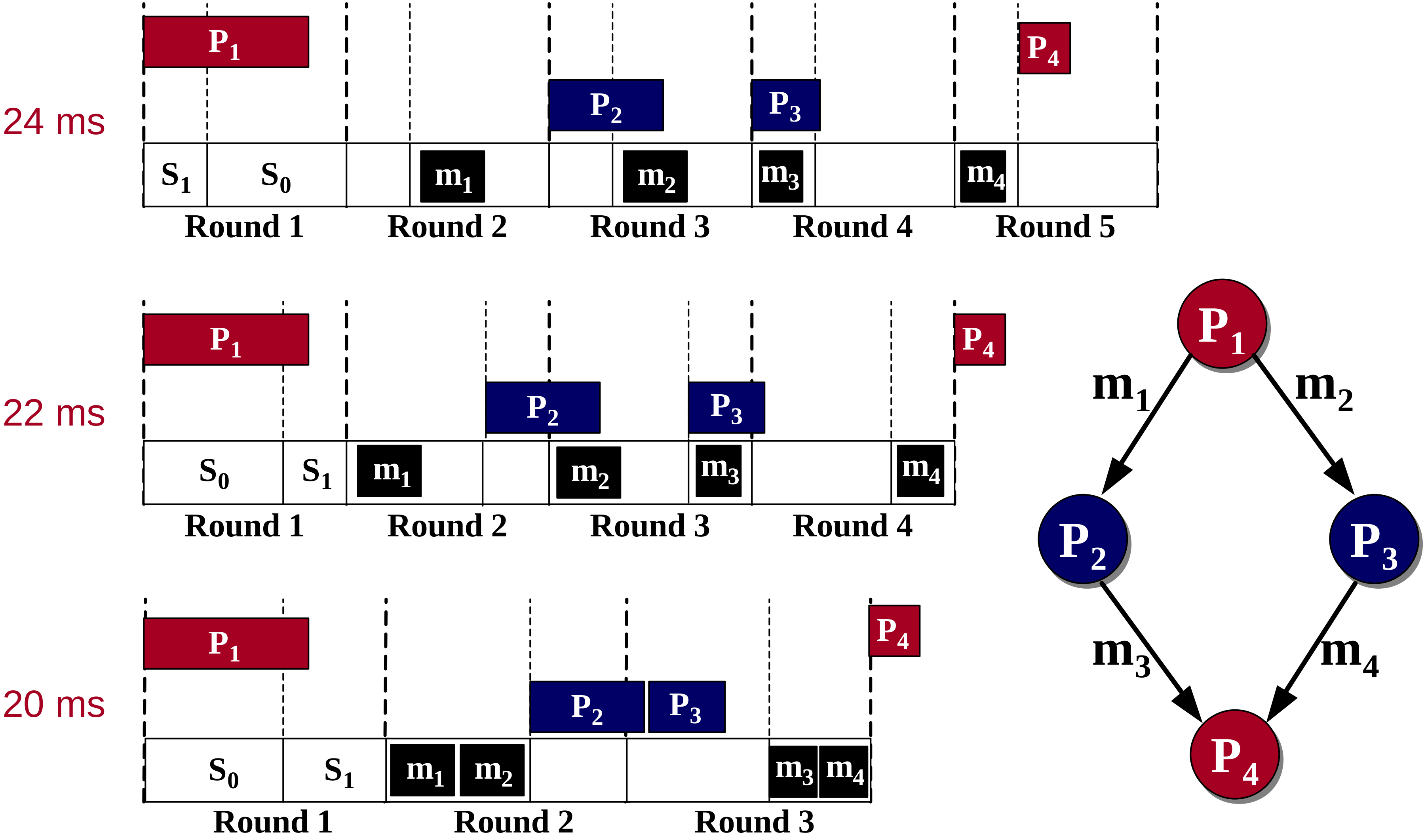
$$T_{P_i} = (d_{PA} + t_{P_i} + q_{C_1} + q_{C_2})$$

$$q_{C_1} = \sum_{i=1}^{N_{out}^{local}(P_i)} d_{S_i} \qquad q_{C_2} = \sum_{i=1}^{N_{out}^{remote}(P_i)} d_{KS_i} + \sum_{i=1}^{N_{in}^{remote}(P_i)} d_{KR_i}$$

Output

- Local schedule tables for each node and the MEDL for the TTP controllers.
- Delay on the system execution time for each operating mode,
  so that this delay is as small as possible.

Scheduling Example



Scheduling Strategy

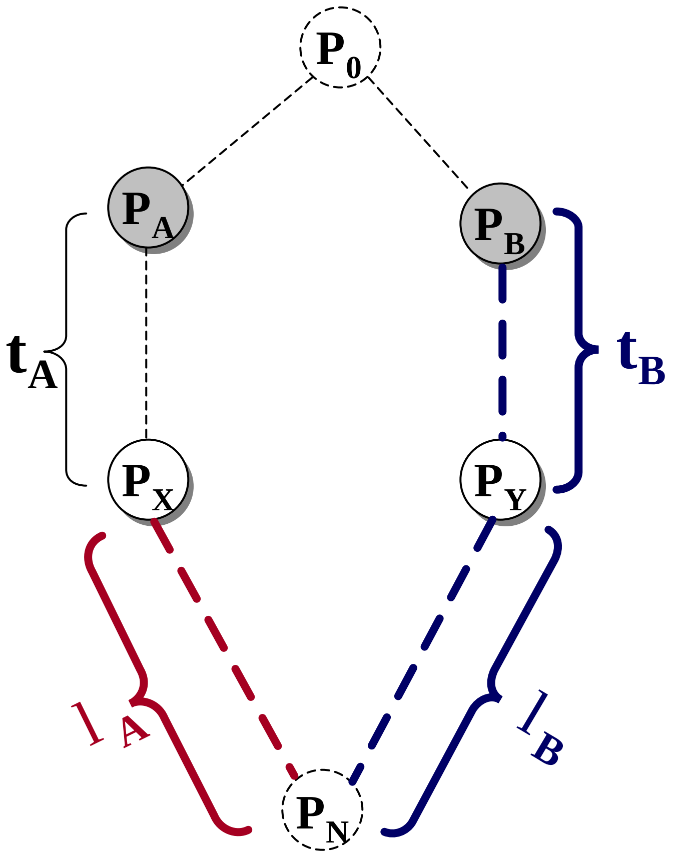
1. The scheduling algorithm has to take into consideration the TTP.

    - priority function for the list scheduling

2. The optimisation of the TTP parameters is driven by the scheduling.

    - sequence and lengths of the slots in a TDMA round are determined to reduce the delay
    - two approaches: Greedy heuristic, Simulated Annealing (SA).
    - two variants: Greedy 1 tries all possible slot lengths, Greedy 2 uses feedback from the scheduling algorithm.
    - SA parameters are set to guarantee near-optimal solutions in a reasonable time.

Partial Critical Path Scheduling



L_{PA}= max(T_curr+t_A+l_A, T_curr+t_A+t_B+l_B)

L_{PB}= max(T_curr+t_B+l_B, T_curr+t_B+t_A+l_A)

Select the alternative with the smaller delay:

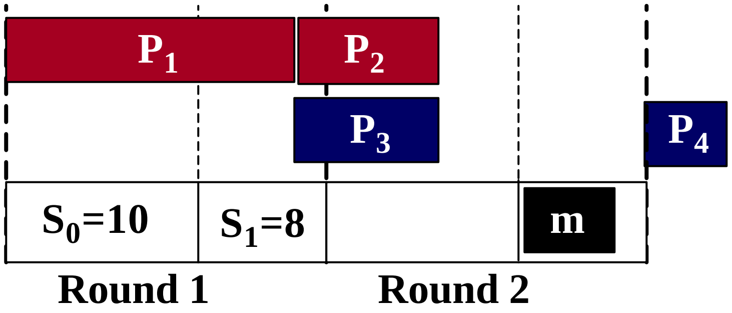
L = max(L_{PA}, L_{PB})

$l_A > l_B$ Þ $L_{PA} < L_{PB}$
$l_B > l_A$ Þ $L_{PB} < L_{PA}$

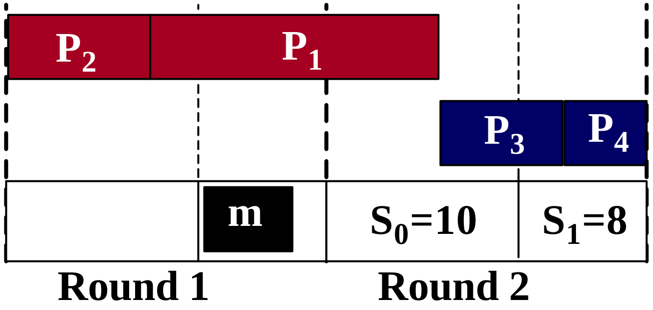
Use l as a priority criterion.

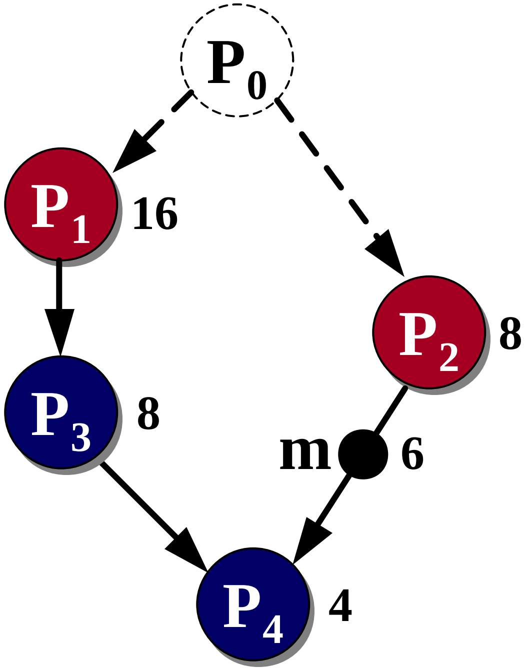
Priority Function Example

40 ms



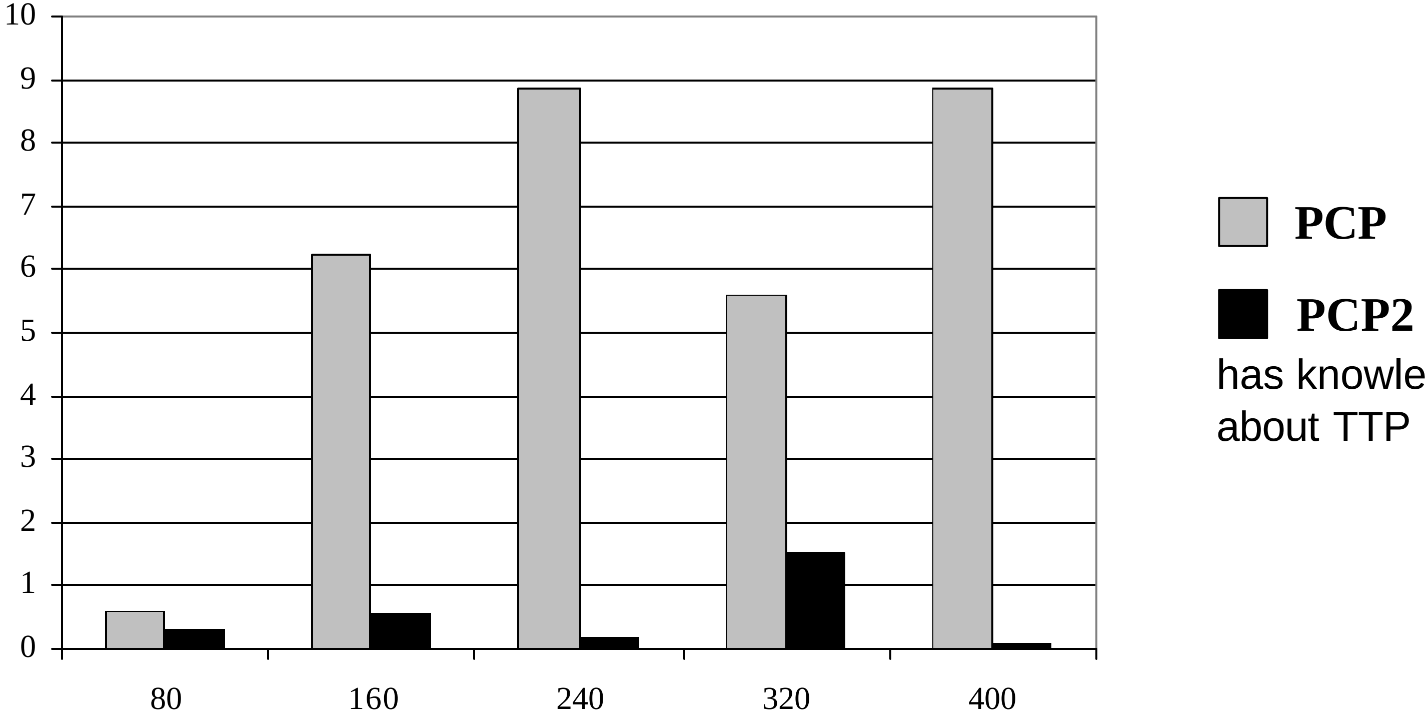
36 ms





Experimental Results

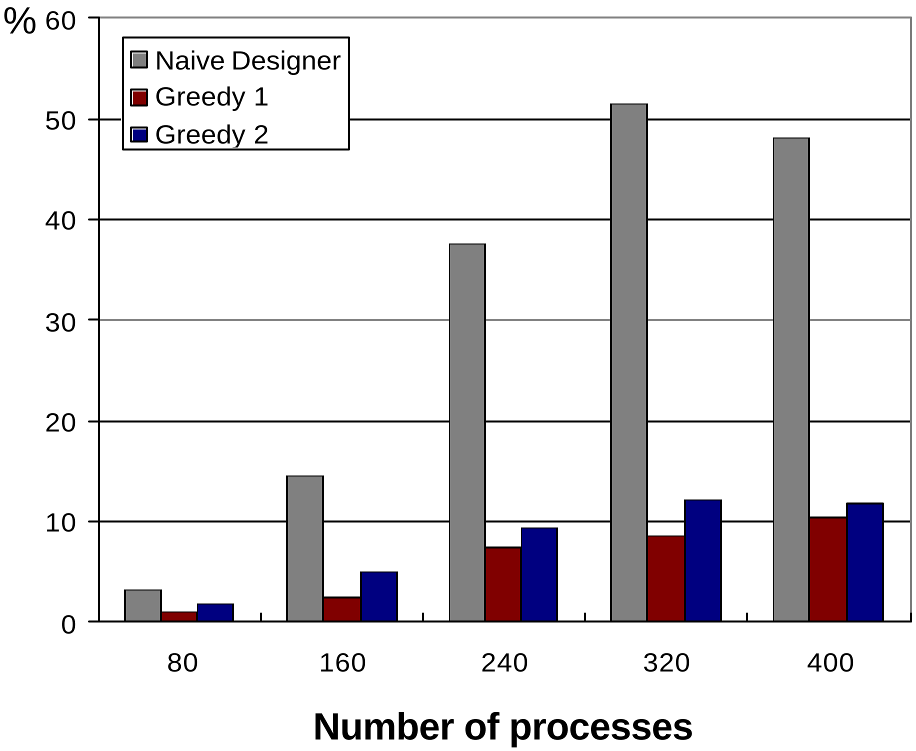
Average percentage deviations from the lengths of the best schedule between PCP and PCP2



has knowledge about TTP

Experimental Results (Cont'd)

Average percentage deviations from the lengths of near-optimal schedules



- **The Greedy approach is producing accurate results in a very short time (few seconds for graphs with 400 processes).**
- **Greedy 1 produces better results than Greedy 2 (but it is slightly slower).**
- **SA finds near-optimal results in a reasonable time.**
- **A real-life example implementing a vehicle cruise controller validated our approach.**

Conclusions

- An approach to process scheduling for the synthesis of safety-critical distributed embedded systems.
- Communication of data and conditions based on TTP.
- Scheduling algorithm tailored to the communication protocol.
- Communication has been optimised through packaging of messages into slots with a properly selected order and lengths.
- Improved schedule quality by considering the overheads of the real-time kernel and of the communication protocol.
- Evaluation based on experiments using a large number of graphs generated for experimental purpose as well as real-life examples.