Scheduling with Optimized Communication for Time-Triggered Embedded Systems

Pop, Paul; Eles, Petru; Peng, Zebo

Published in:
Hardware/Software Codesign, 1999. (CODES '99) Proceedings of the Seventh International Workshop on

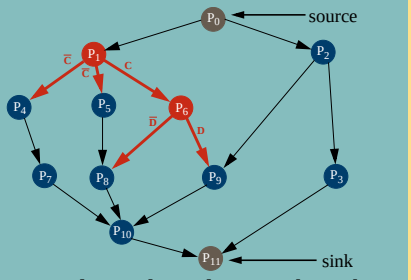
Link to article, DOI:
10.1109/HSC.1999.777428

Publication date:
1999

Document Version
Publisher's PDF, also known as Version of record

Scheduling with Optimized Communication for Time-Triggered Embedded Systems

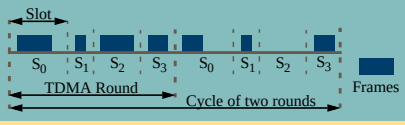
Conditional Process Graph



- Directed, acyclic polar graph with conditional edges used for system representation.
- Each node represents a process assigned to a processing element.
- Conditional edges (in red) have an associated condition. Transmission will take place only if the associated condition is satisfied.
- A process is activated when all its inputs have arrived and issues its outputs when it terminates.
- A process can not be preempted.

Time-Triggered Protocol

- TTP was designed for distributed real-time applications that require predictability and reliability.
- The bus access scheme is time-division multiple-access (TDMA).
- Each node can transmit only during a predetermined time interval, its TDMA slot. A sequence of slots for all nodes is a TDM round. Several rounds can form a cycle that is repeated periodically.
- Every node has a TTP controller.
- The TDMA access scheme is imposed by a message descriptor list (MEDL) located in every controller.
- MEDL serves as a schedule table for the TTP controller which knows when to send/receive a message.



Summary

- Scheduling algorithms proposed can be used both for performance estimation and for system synthesis.
- System model capturing both the flow of data and that of control.
- Communication using time-triggered protocol (TTP) implementation.
- Improved schedule quality by considering the overheads of the real-time kernel and communication protocol.

Hardware Architecture

- Safety-critical distributed embedded systems.
- Nodes connected by a broadcast communication channel.
- Nodes consisting of a TTP controller, a CPU, a RAM, a ROM and an I/O interface to sensors and actuators.
- Communication is based on the TTP.

Software Architecture

- Real-time kernel on the CPU in each node.
- Local schedule table in each kernel.
- The worst case administrative overheads:

U_t	timer interrupt routine
δ_{PA}	process activation overhead
δ_S	sending a message on the same node
δ_{KS}	sending a message between nodes
δ_{KR}	receiving a message from another node

Problem Formulation

- Input: Safety-critical application modelled by a conditional process graph.
- Mapping of processes to nodes is given.
- Worst case execution delay of a process P_i is:

$$T_{P_i} = (\delta_{PA} + t_{P_i} + \theta_{C_1} + \theta_{C_2}) \cdot (1 + U_t) \text{ where:}$$

$$\theta_{C_1} = \sum_{i=1}^{N_{out}^{local}(P_i)} \delta_{S_i} = \sum_{i=1}^{N_{out}^{remote}(P_i)} \delta_{KS_i} + \sum_{i=1}^{N_{in}^{remote}(P_i)} \delta_{KR_i}$$

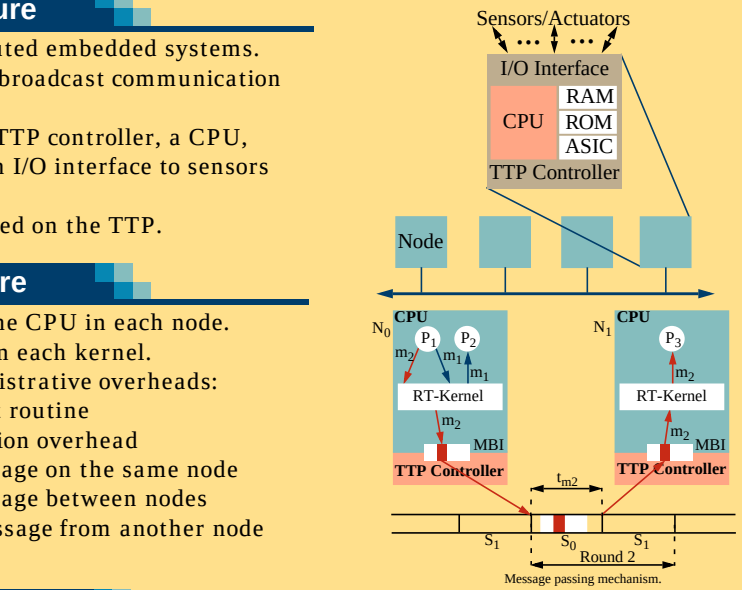
- Output: Local schedule tables and MEDL, delay (smallest) on the system execution.

Scheduling Strategy

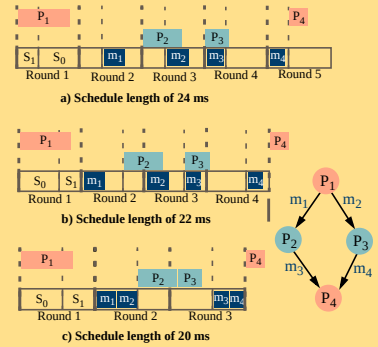
- Sequence and lengths of the slots in a TDMA round are determined to reduce the delay.
- Previous work extended to handle scheduling of messages within TTP for a given TDMA configuration.
- Two approaches: Greedy heuristic, Simulated Annealing.
- Two variants: Greedy 1 tries all slot lengths, while Greedy 2 uses feedback from the schedule_message.
- SA parameters are set to guarantee finding near-optimal solutions in a reasonable time.

No.of proc.	Naive Designer		Greedy 1			Greedy 2		
	aver.	max.	aver.	max.	time.	aver.	max.	time
80	3.16%	21%	0.02%	0.5%	0.25s	1.8%	19.7%	0.04s
160	14.4%	53.4%	2.5%	9.5%	2.07s	4.9%	26.3%	0.28s
240	37.6%	110%	7.4%	24.8%	0.46s	9.3%	31.4%	1.34s
320	51.5%	135%	8.5%	31.9%	34.69s	12.1%	37.1%	4.8s
400	48%	135%	10.5%	32.9%	56.04s	11.8%	31.6%	8.2s

Percentage deviations of the schedule lengths produced by Greedy 1 and Greedy 2 from the lengths of the near-optimal schedules obtained with the SA algorithm, and the average execution time expressed in seconds.



Message passing mechanism.



Scheduling example.

By modifying the TDMA parameters we can improve the schedule quality.

```
greedy
  for each slot
    for each node not yet allocated to a slot
      bind (node, slot, minimum possible length for this slot)
      for every slot length (or recommended slot lengths)
        schedule in the context of current TDMA round
        remember the best schedule for this slot
      end for
    end for
    bind (node, slot and length
      corresponding to the best schedule)
  end for
  return solution
end
```


Paul Pop, Petru Eles and Zebo Peng
Embedded Systems Laboratory (ESLAB)
Dept. of Computer and Information Science, Linköping University