Scheduling and Communication Synthesis for Distributed Real-Time Systems

Paul Pop

Department of Computer and Information Science
Linköpings universitet

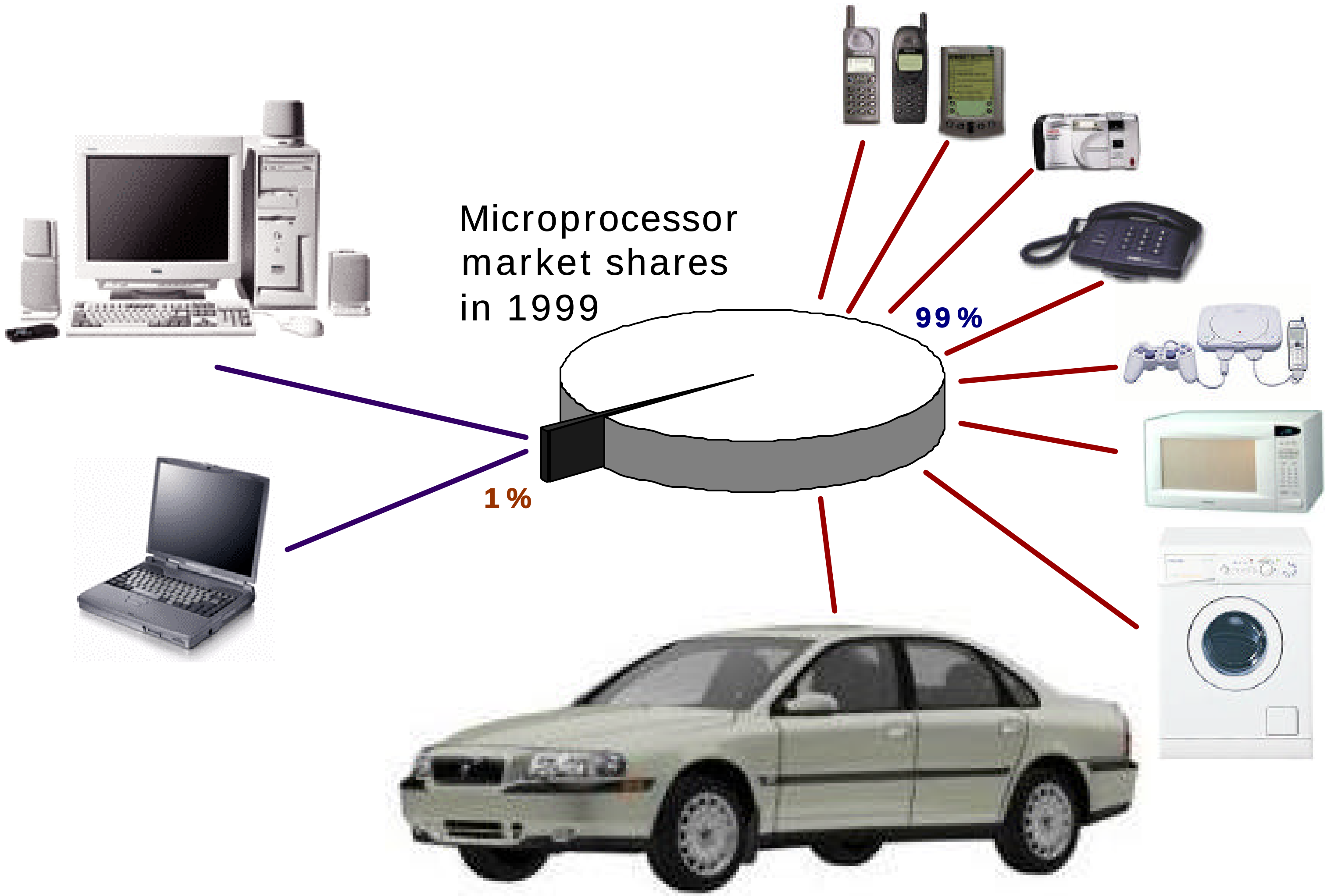
Outline

- Motivation
- System Model and Architecture
- Scheduling and Communication Synthesis
  - Time Driven Systems
  - Event Driven Systems
- Real Life Example
- Conclusions

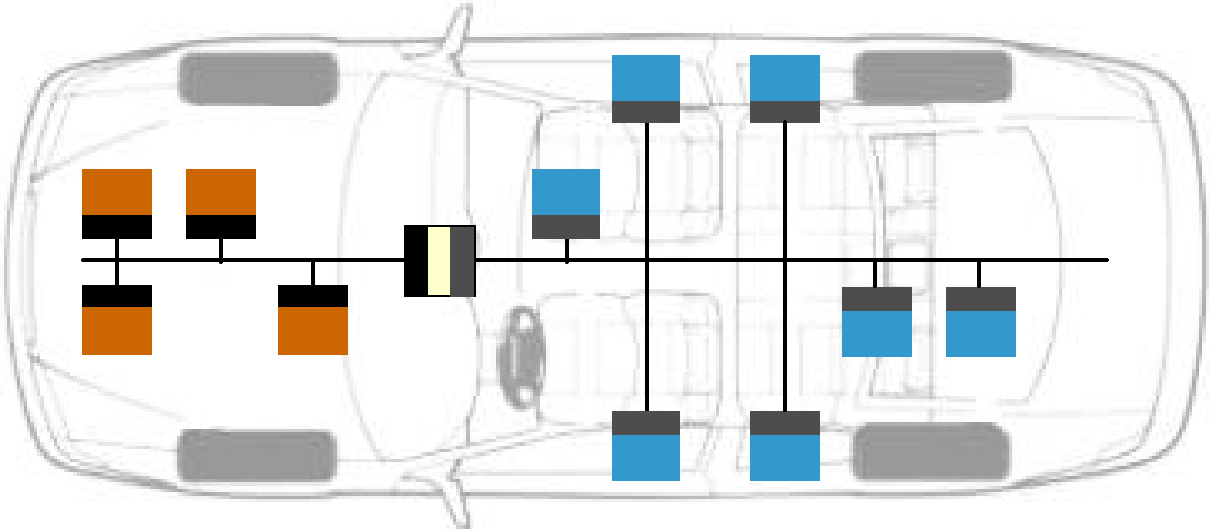
Embedded Systems

General purpose systems

Embedded systems

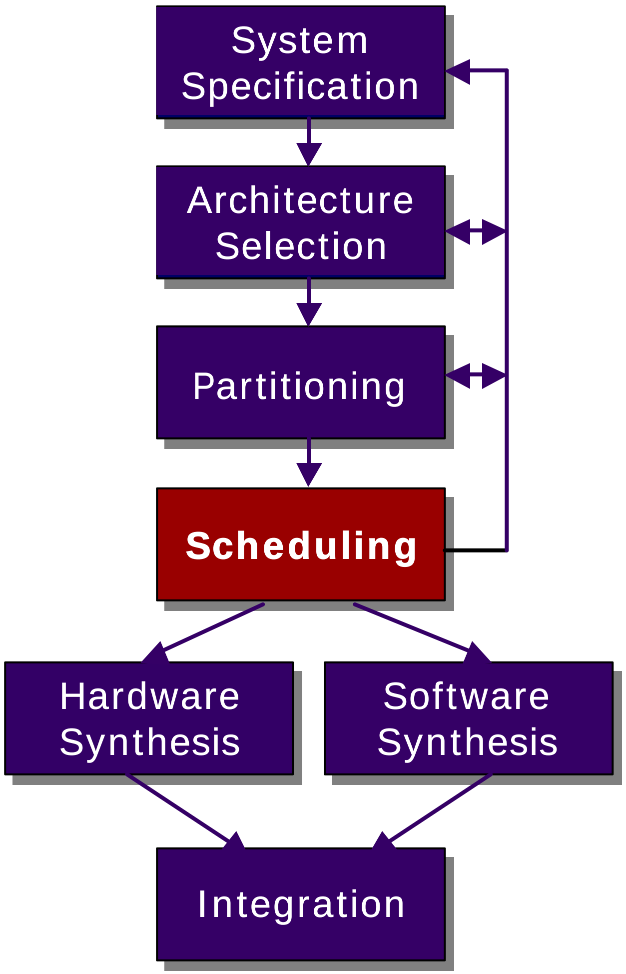


Distributed Real-Time Systems



- Safety critical applications (e.g. Drive-by-Wire):
  - timing constraints,
  - data and control dependencies.
- Communication protocols: Time Triggered Protocol (TTP), Controller Area Network (CAN).
- **Scheduling of processes and communication of messages: guaranteeing timing constraints.**

Hardware / Software Codesign



Goals of the thesis:

- **Scheduling...**

  Scheduling of processes and messages for distributed hard real-time applications with control and data dependencies in the context of a given communication protocol.

- **Communication synthesis...**

  Optimization of the parameters of the communication protocol so that the overall system performance is increased and the imposed timing constraints are satisfied.

Related Work

- Scheduling:
  - Static cyclic non-preemptive scheduling:

    P. Eles, G. Fohler, D. D. Gajski, H. Kasahara,

    H. Kopetz, K. Kuchcinski, J. Madsen, J. Xu.
  - Fixed priority preemptive scheduling:

    J. Axelsson, S. Baruah, A. Burns, J. W. Layland, C. L. Liu,

    K. Tindell, J. A. Stankovic, , W. Wolf , T. Y. Yen.
- Communication synthesis:

  G. Borriello, R. Ernst, H. Hansson, J. Madsen, R. B. Ortega, K. Tindell.

Characteristics and Message

- Distributed hard real-time applications.
- Heterogeneous system architectures.
- Systems with data and control dependencies.
- Scheduling of processes:

  Time triggered: Static cyclic non-preemptive scheduling,

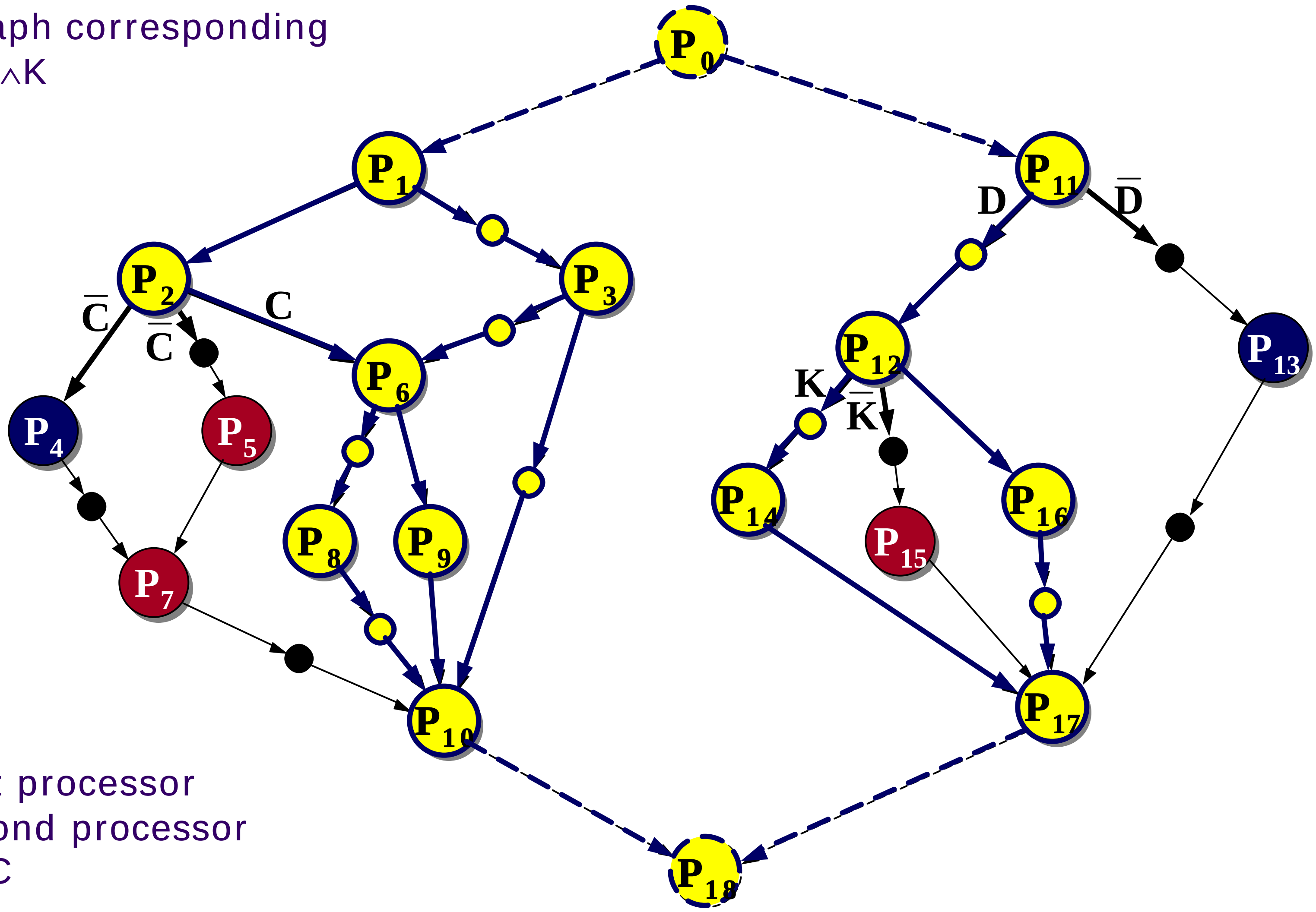
  Event triggered: Fixed priority preemptive scheduling.
- Communications using the time-triggered protocol (TPP).

- The performance of the system can be significantly improved by considering the **communication protocol** and the **control dependencies** during scheduling.

- Motivation
- **System Model and Architecture**
- Scheduling and Communication Synthesis
  - Time Driven Systems
  - Event Driven Systems
- Real Life Example
- Conclusions

Conditional Process Graph (CPG)

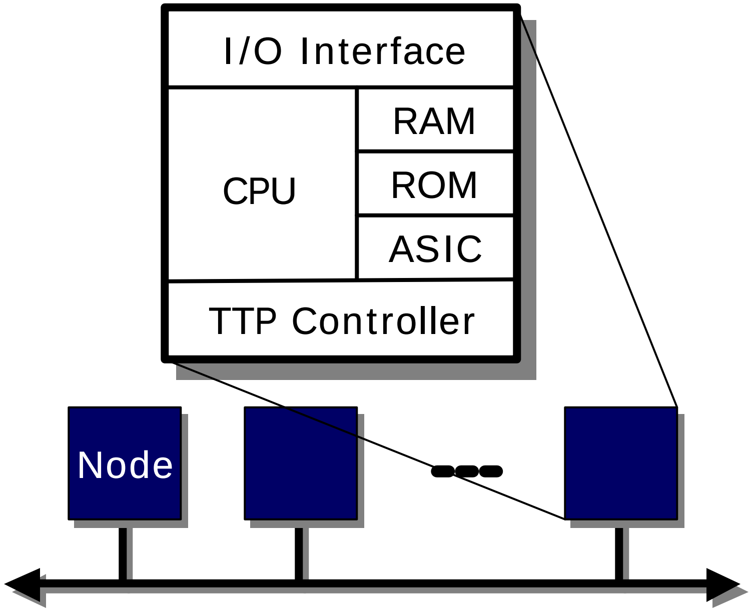
Subgraph corresponding to D∧C∧K



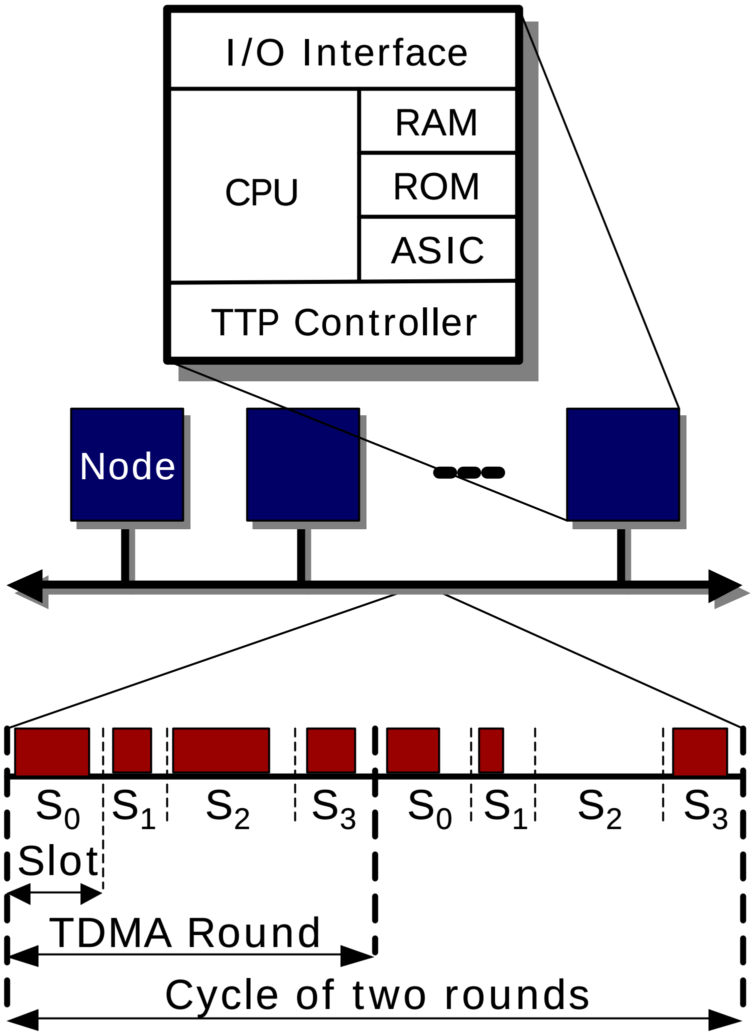
- First processor
- Second processor
- ASIC

Hardware Architecture

- Hard real-time distributed systems.
- Nodes interconnected by a broadcast communication channel.
- Nodes consisting of: TTP controller, CPU, RAM, ROM, I/O interface.
- Communication between nodes is based on the time-triggered protocol.



Time Triggered Protocol



- H. Kopetz, Technical University of Vienna.
- Intended for distributed real-time control applications that require high degree of dependability and predictability.
- Recommended by the X-by-Wire Consortium for use in safety critical applications in vehicles.
- Integrates all the services required in the design of fault-tolerant distributed real-time systems.

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

Outline

- Motivation
- System Model and Architecture
- **Scheduling and Communication Synthesis**
  - **Time Driven Systems**
  - Event Driven Systems
- Real Life Example
- Conclusions

Problem Formulation

Input

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

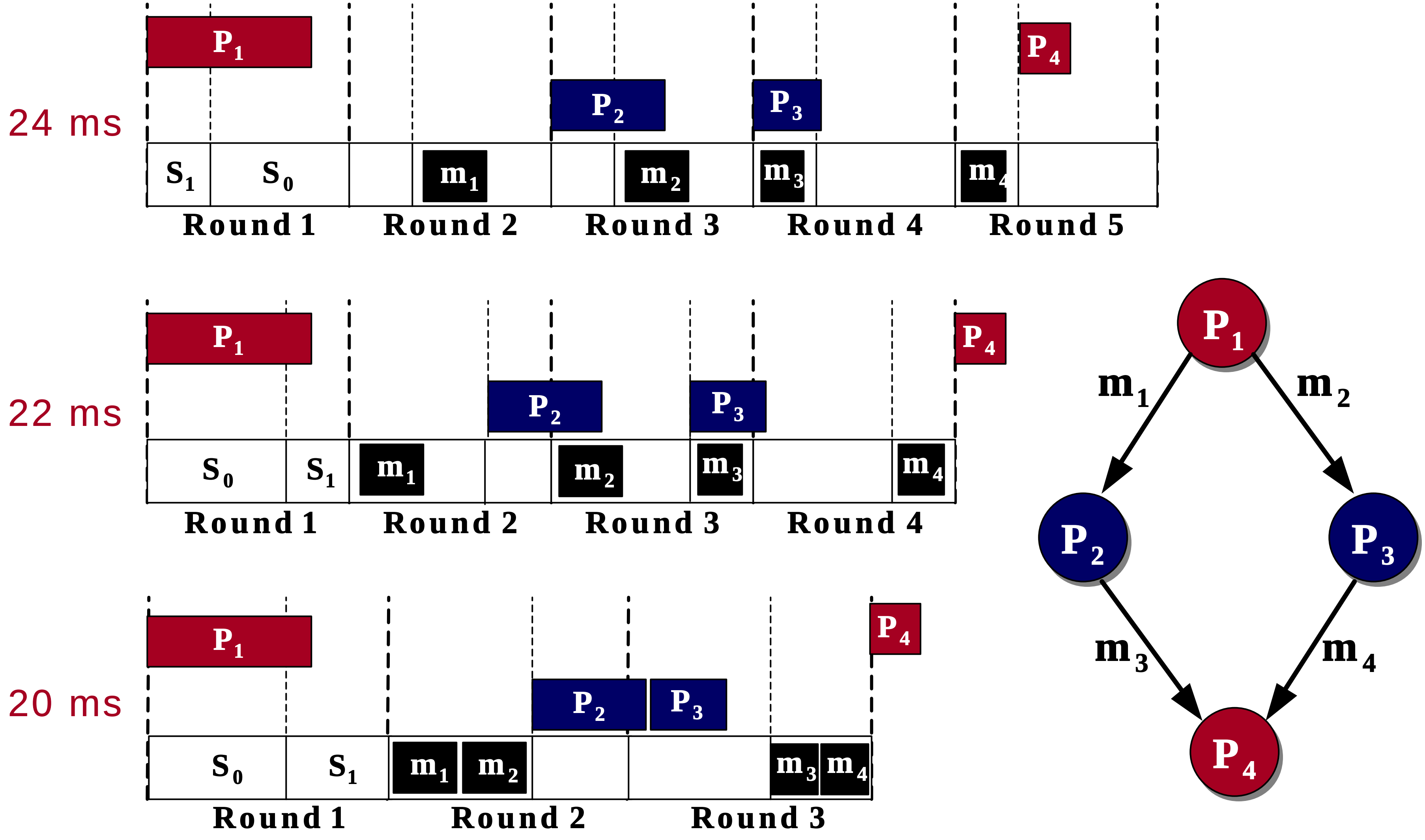
Output

- 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.

Note

- Processes scheduled with **static cyclic non-preemptive scheduling**, and messages according to the TTP.

Scheduling Example



Scheduling Strategy

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

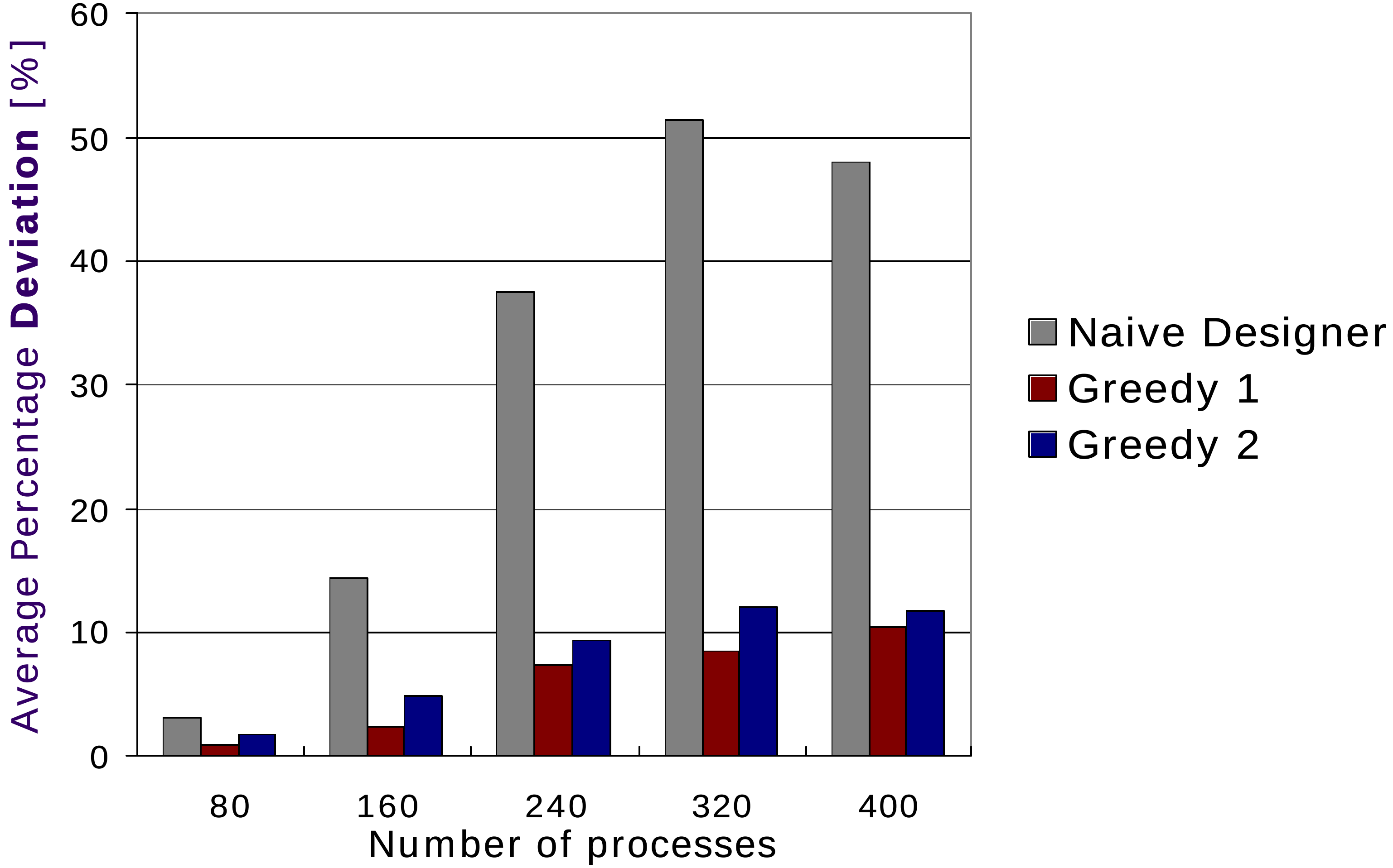
- Priority function for the list scheduling.

2. The optimization 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.

Experimental Results

Deviations from the near-optimal schedule lengths obtained by SA:



Outline

- Motivation
- System Model and Architecture
- **Scheduling and Communication Synthesis**
  - Time Driven Systems
  - ➔ **Event Driven Systems**
- Real Life Example
- Conclusions

Event Triggered Processes

Problem Formulation

Input

- An application modelled using conditional process graphs.
- Each process has an execution time, a period, a deadline, and a priority.
- The system architecture and mapping of processes are given.

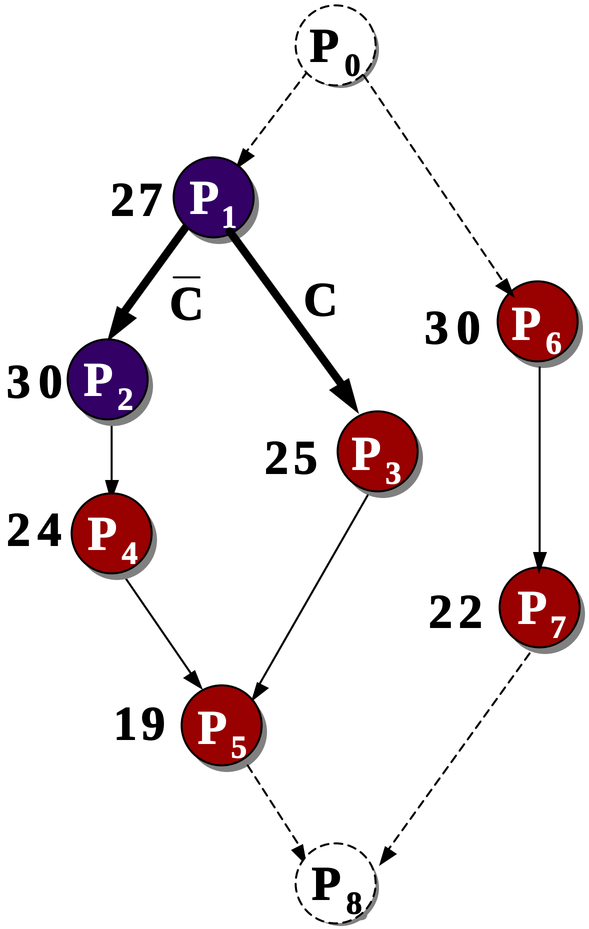
Output

- **Schedulability analysis** for systems modelled using CPGs.
- Schedulability analysis for the time-triggered protocol.
- The **MEDL** for the TTP controllers so that the process set is schedulable on an as cheap (slow) as possible processor set.

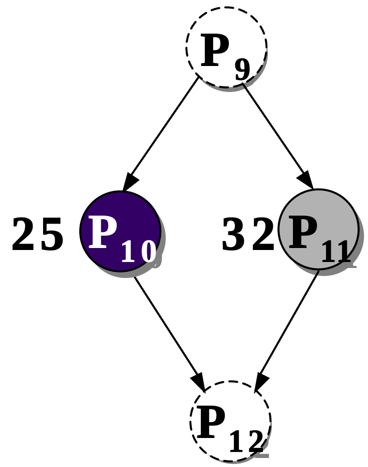
Note

- Processes scheduled with **fixed priority preemptive scheduling**, and messages according to the TTP.

Example



G_1: 200

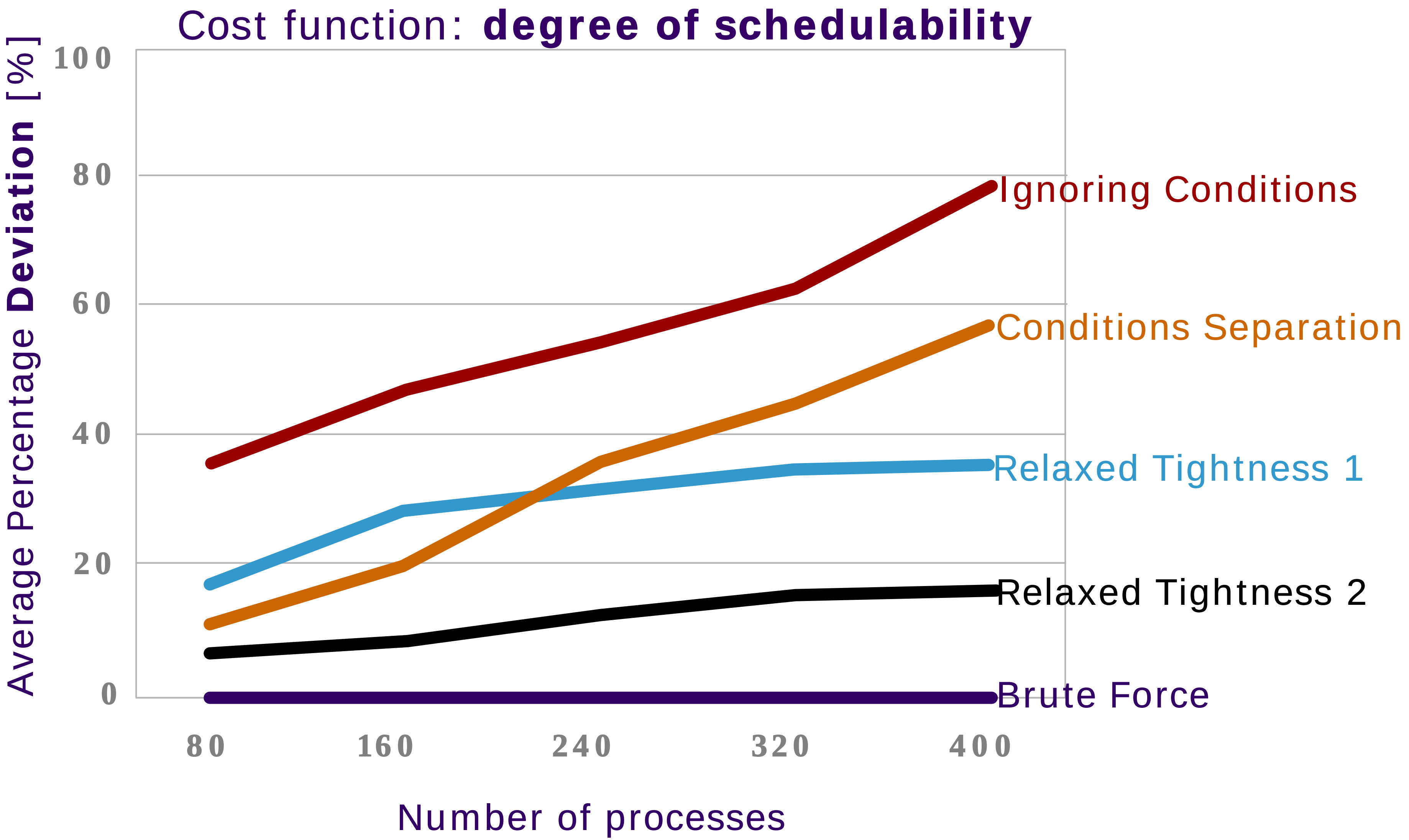


G_2: 150

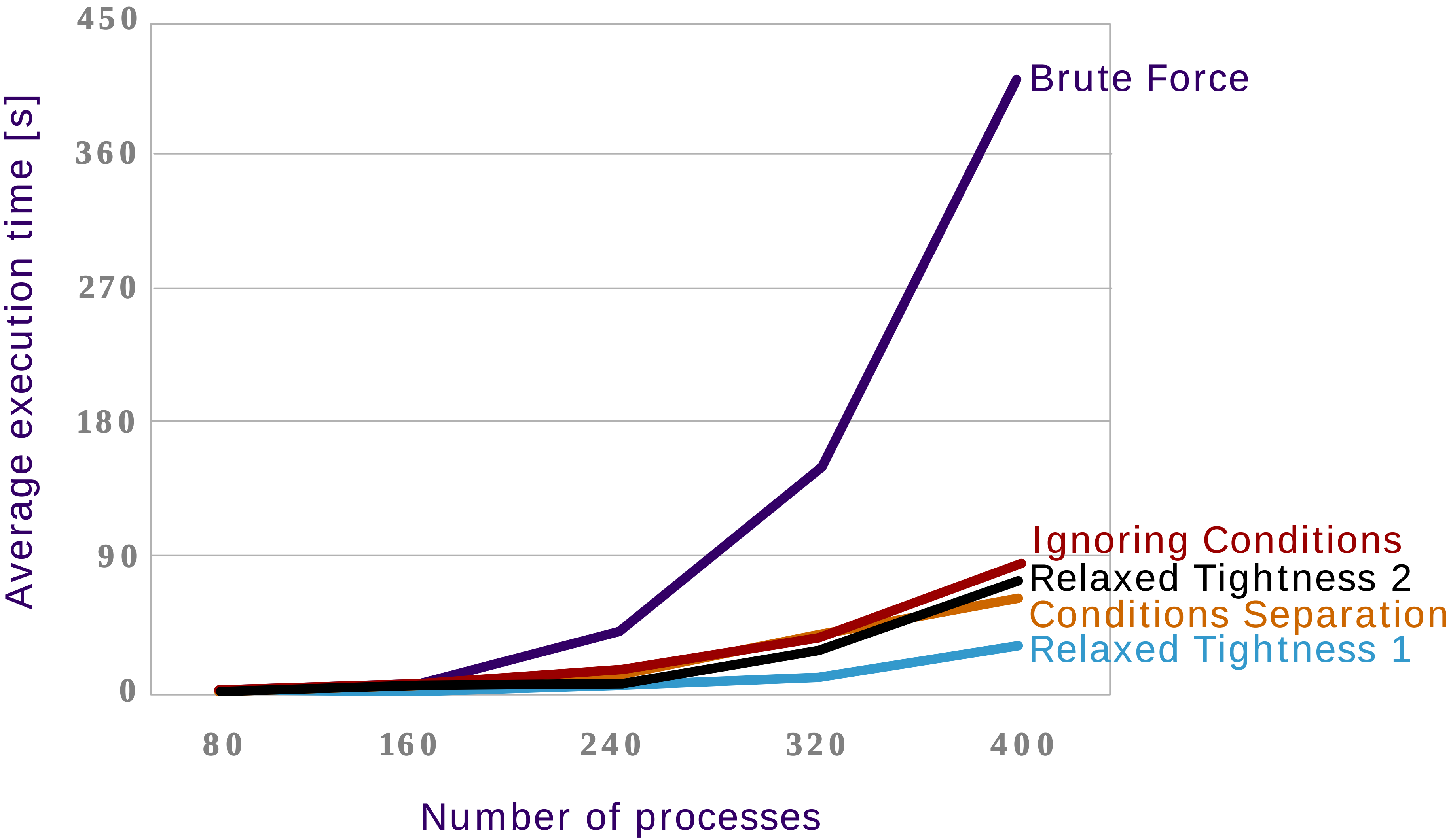
CPG	Worst Case Delays	
	No conditions	Conditions
G_1	120	100
G_2	82	82

Deadline: 110

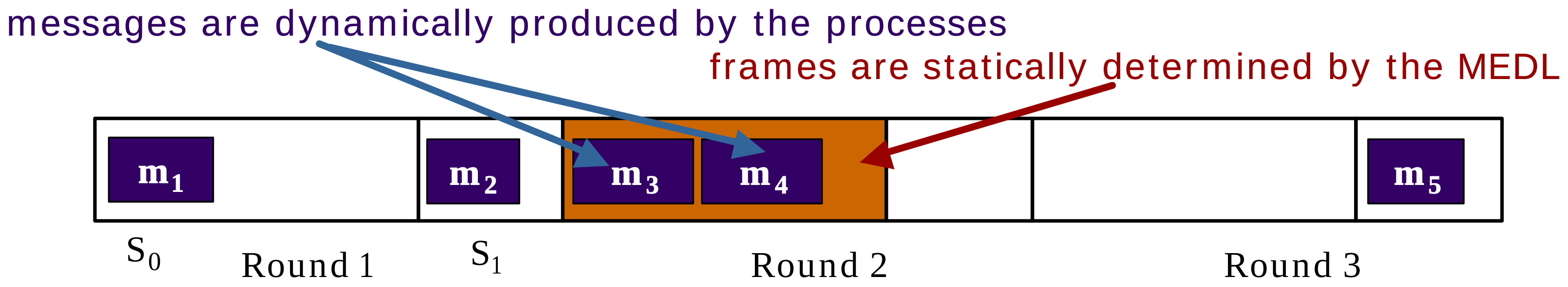
Experimental Results



Experimental Results (Cont.)



Scheduling of Messages over TTP



1. Single message per frame, allocated statically:
   Static Single Message Allocation (**SM**)

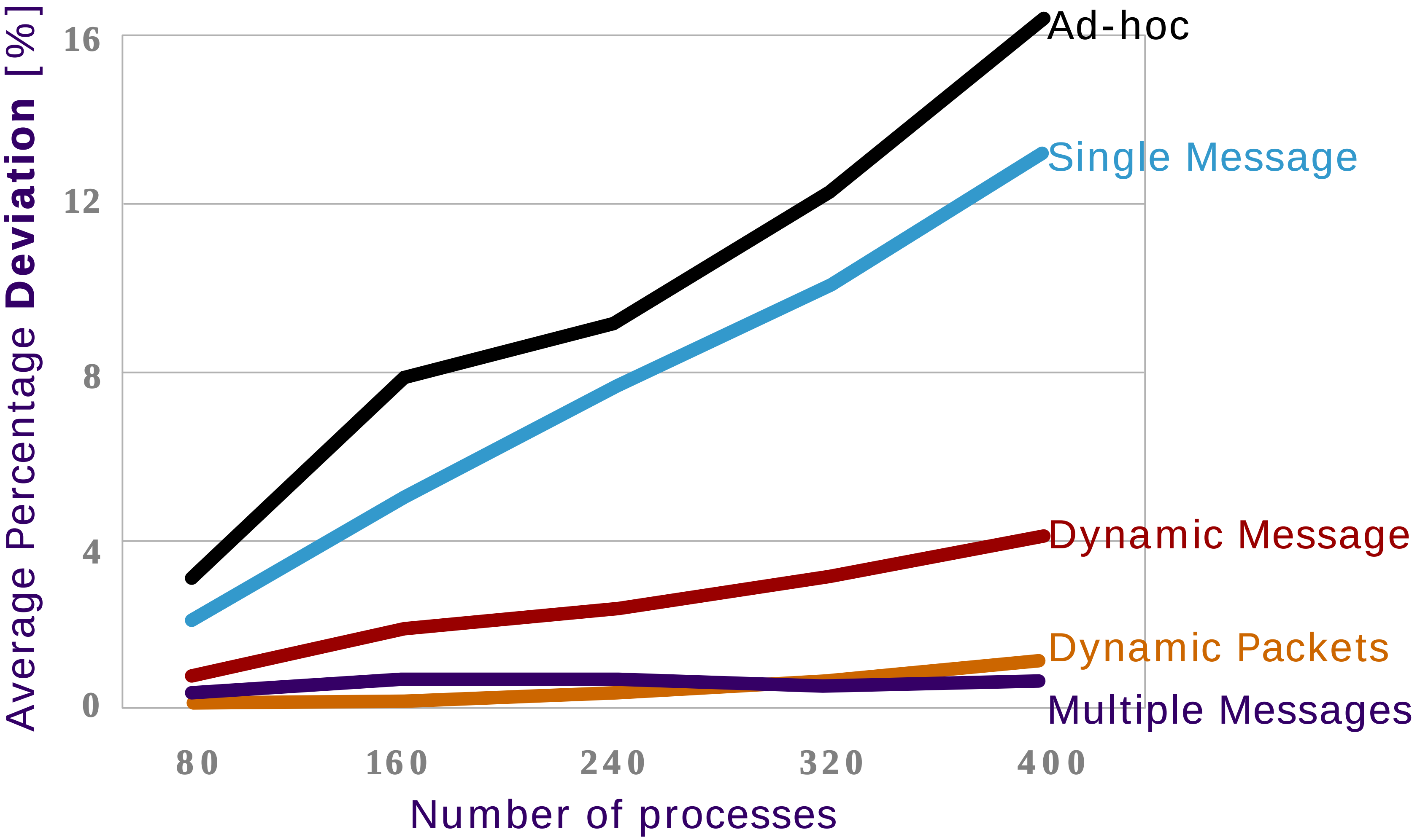
2. Several messages per frame, allocated statically:
   Static Multiple Message Allocation (**MM**)

3. Several messages per frame, allocated dynamically:
   Dynamic Message Allocation (**DM**)

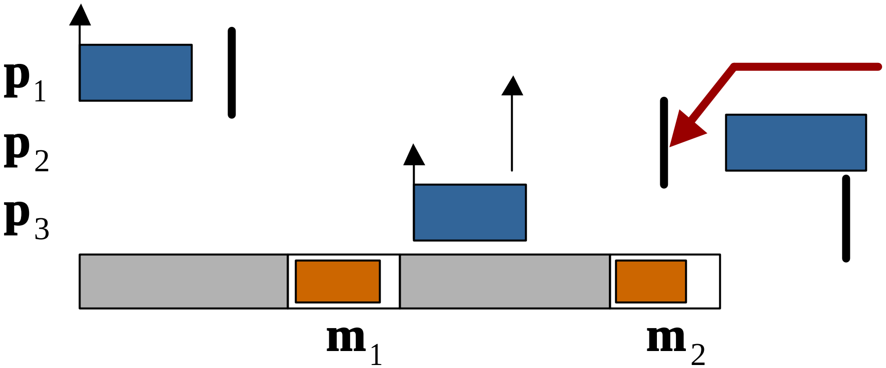
4. Several messages per frame, split into packets, allocated dynamically
   Dynamic Packets Allocation (**DP**)

Experimental Results

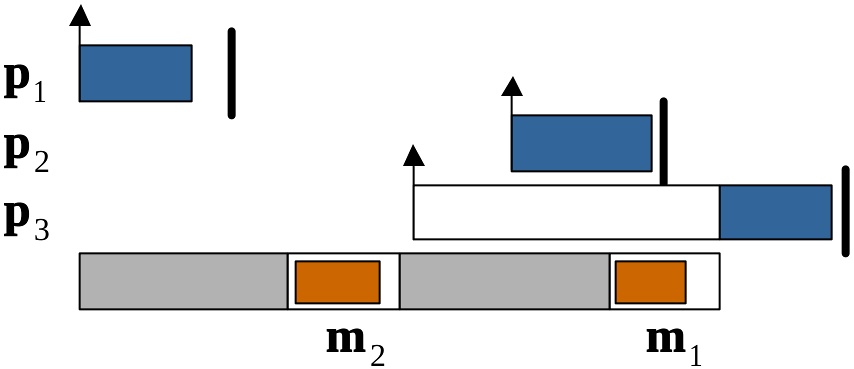
Cost function: **degree of schedulability**



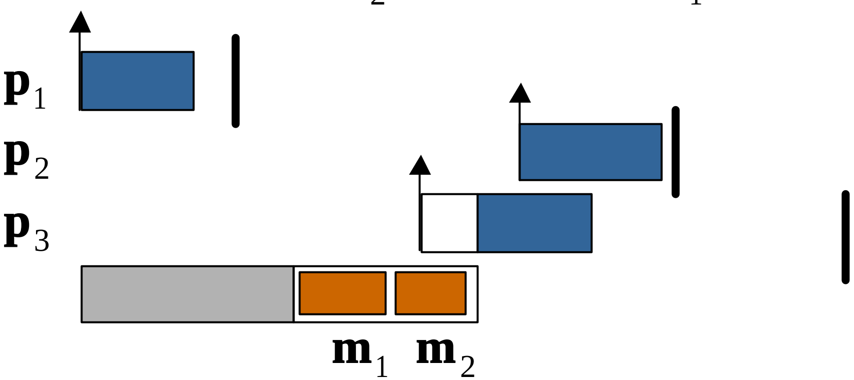
Optimizing Bus Access (SM and MM)



Process p_2 misses its deadline!



Swapping m_1 with m_2:
all processes meet their deadlines.



Putting m_1 and m_2 in the same slot:
all processes meet their deadline,
the communication delays are reduced.

OptimizeDM

OptimizeDM: Find the slot sizes that maximize the "degree of schedulability"

for each node N_i **do**

  $MinSize_{Si}$ = max(size of messages m_j sent by node N_i)

end for

for each slot S_i

  $BestSize_{Si} = MinSize_{Si}$

  for each *SlotSize* in [$MinSize_{Si}$...$MaxSize$] **do**

    calculate the *CostFunction*

    if the *CostFunction* is best so far **then**

      $BestSize_{Si} = SlotSize_{Si}$

    end if

  end for

  $size_{Si} = BestSize_{Si}$

end for

end OptimizeDM

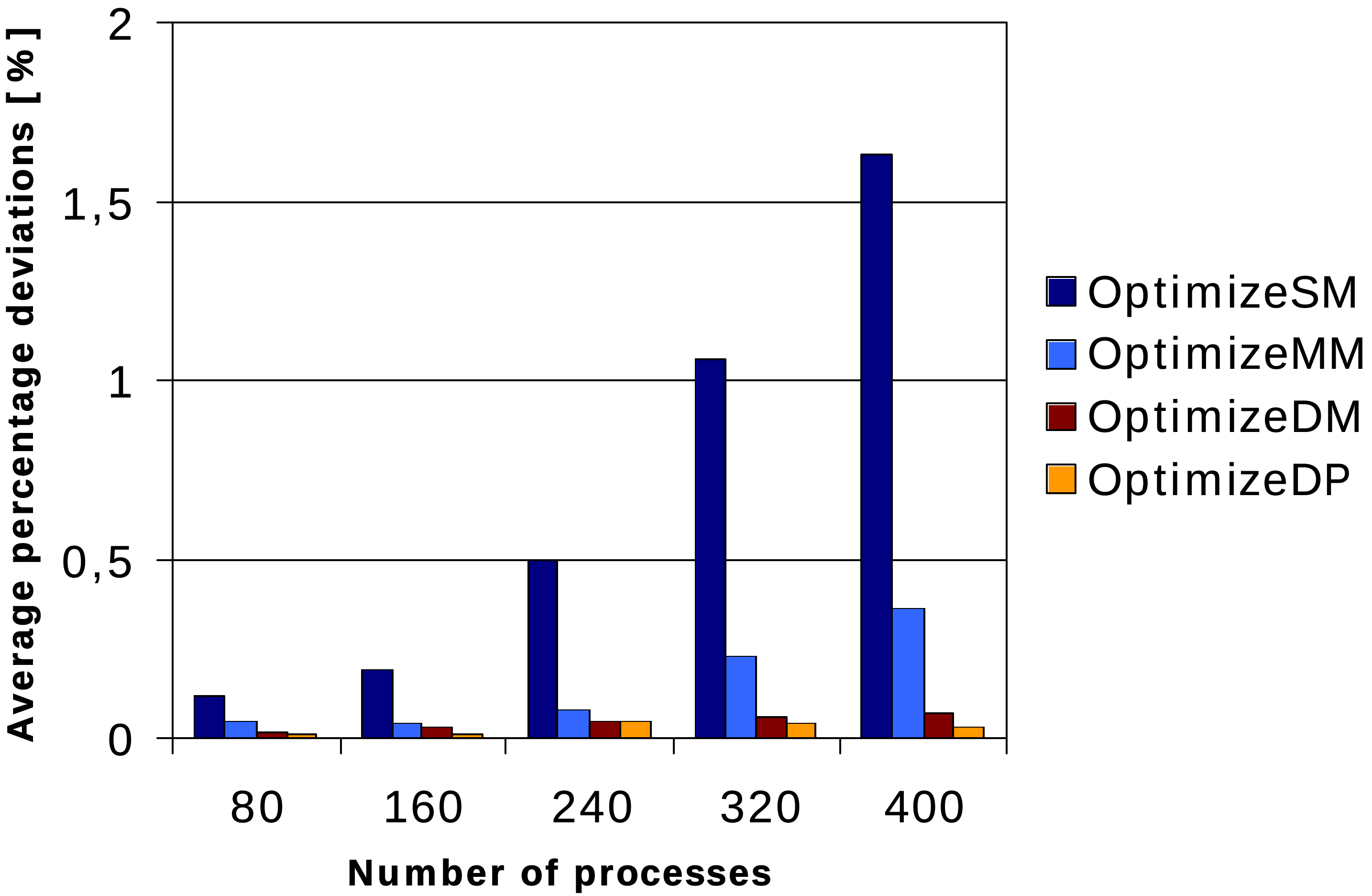
Initialization:
the size of a slot in a TDMA round has to accommodate the largest message sent by the corresponding node.

Greedy heuristic: finds the local optimum for each slot.

Local optimum:
find the best size for this slot, the size that leads to the "best so far" cost function.

Experimental Results

The quality of the greedy optimization heuristics:



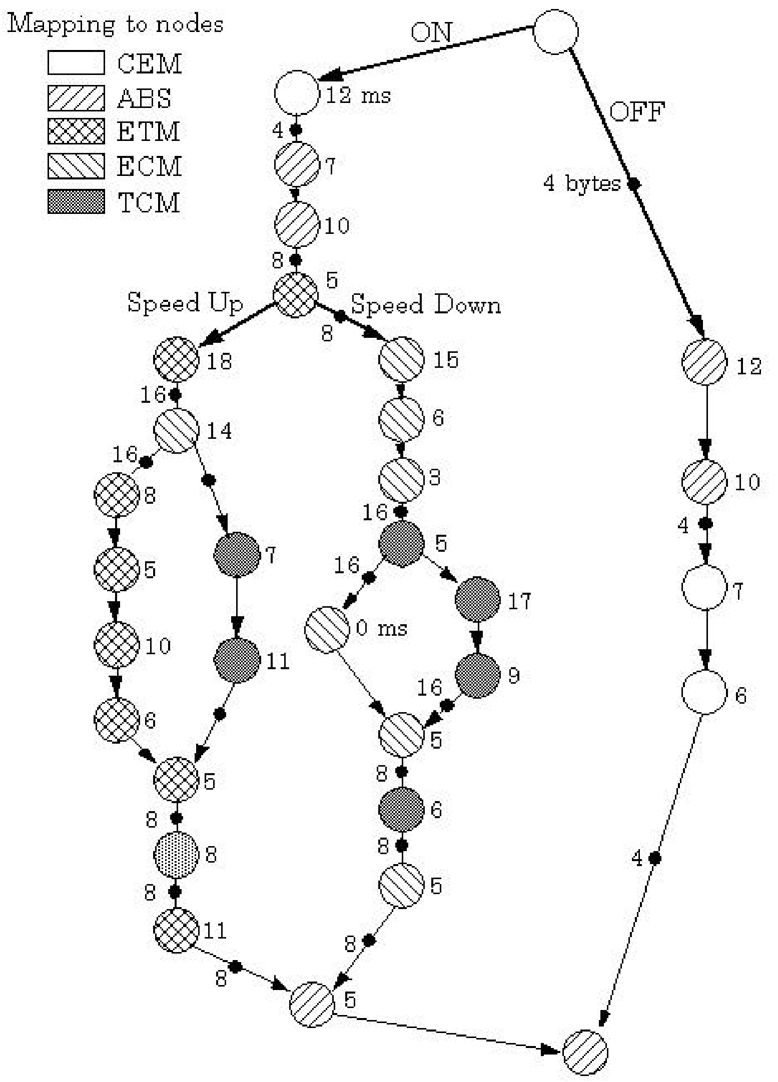
Outline

- Motivation
- System Model and Architecture
- Scheduling and Communication Synthesis
  - Time Driven Systems
  - Event Driven System
- **Real Life Example**
- Conclusions

Real Life Example

- **Vehicle cruise controller.**
- Modelled with a CPG of 32 processes and two conditions.
- Mapped on 5 nodes: CEM, ABS, ETM, ECM, TCM.

- Time triggered processes: (deadline 400 ms)
  - Ad-hoc: 429 ms
  - Greedy 1: 314 ms
  - Greedy 2: 323 ms
  - SA: **302** ms
- Event triggered processes: (no messages, deadline 130 ms)
  - Ignoring Conditions: 138 ms
  - Conditions Separation: 132 ms
  - Relaxed Tightness 1, 2: **124** ms
  - Brute Force: **124** ms



Conclusions and Future Work

Time triggered processes:

- Extension to static scheduling for CPGs to handle TTP.
- Improved schedule quality by using new priority function that considers the time triggered protocol.
- Significant performance improvements can be obtained by optimizing the access to the communication channel.

Event triggered processes:

- Schedulability analysis with the TTP: four message scheduling approaches compared based on the issue of schedulability.
- Significant improvements to the “degree of schedulability” through the optimization of the bus access scheme.
- The pessimism of the analysis can be drastically reduced by considering the conditions.

Mapping of processes and **architecture selection**.

Contributions

Time triggered processes:

- Static scheduling strategy for systems with both control and data dependencies.
- Optimization strategies for the synthesis of the bus access scheme.

Event triggered processes:

- Less pessimistic schedulability analysis for hard real-time systems with both control and data dependencies
- Schedulability analysis for the time-triggered protocol.
- Optimization strategies for the synthesis of the bus access scheme.