Analysis and Synthesis of Communication-Intensive Heterogeneous Real-Time Systems

Paul Pop
Computer and Information Science Dept.
Linköpings universitet

Outline

- ➔ Introduction
  - System-level design and modeling
    - Conditional process graph
  - The system platform
    - Time-driven vs. event-driven systems

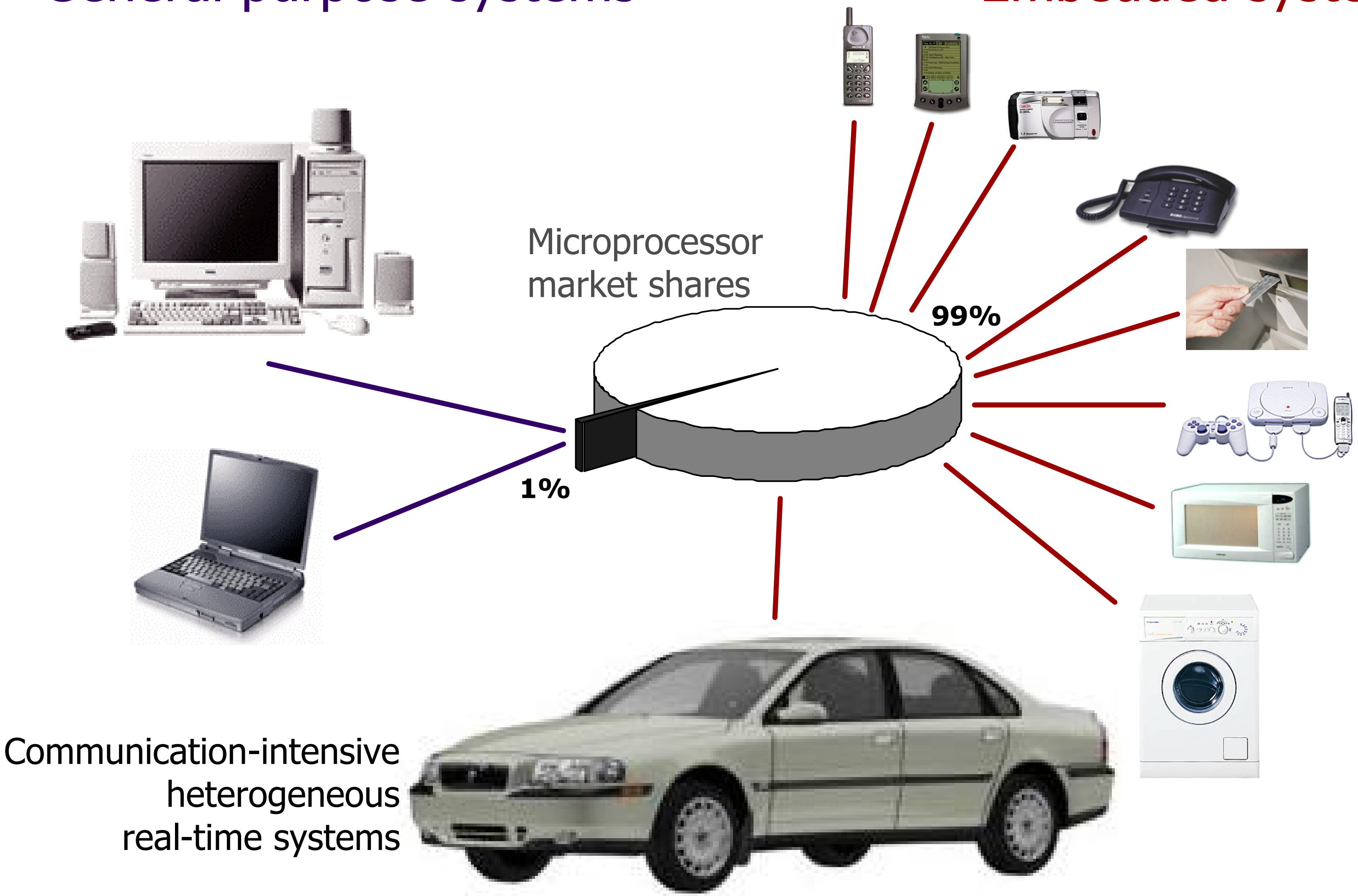
- Communication-intensive heterogeneous real-time systems
  - Time-driven systems
    - Scheduling and bus access optimization
    - Incremental mapping
  - Event-driven systems
    - Schedulability analysis and bus access optimization
    - Incremental mapping
  - Multi-Cluster Systems
    - Schedulability analysis and bus access optimization
    - Frame packing

- Summary of contributions

Embedded Systems

General purpose systems

Embedded systems



Microprocessor
market shares

99%

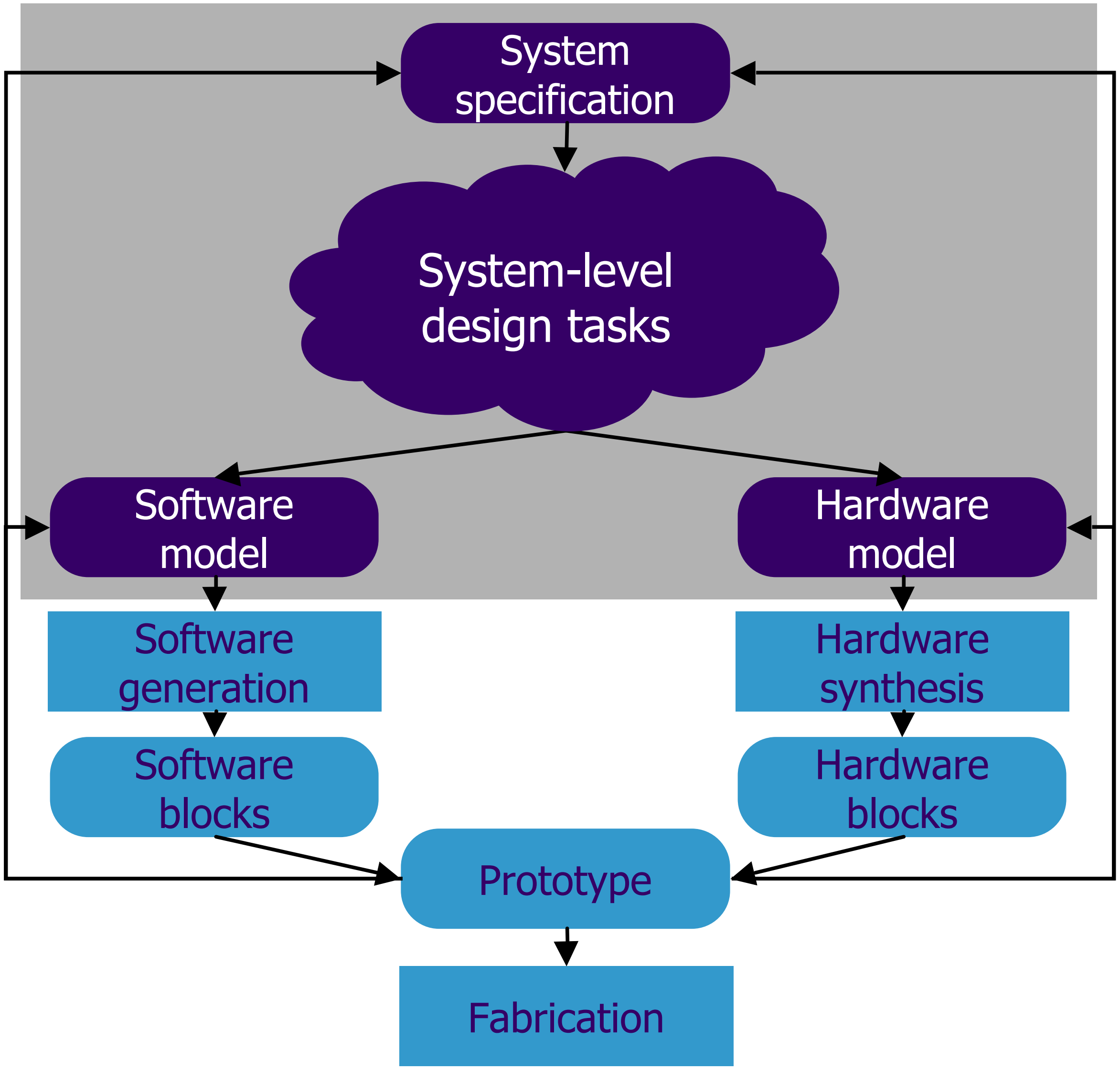
1%

Communication-intensive
heterogeneous
real-time systems

Embedded Systems Characteristics

- Dedicated functionality (not general purpose computers)
  - Embedded into a host system
  - Complex architectures
- Embedded systems design constraints
  - Correct functionality
  - Performance, timing constraints: **real-time systems**
  - Development cost, unit cost, size, power, flexibility, time-to-prototype, time-to-market, maintainability, correctness, safety, etc.
- Difficult to design, analyze and implement
  - System-level design
  - Reuse and flexibility

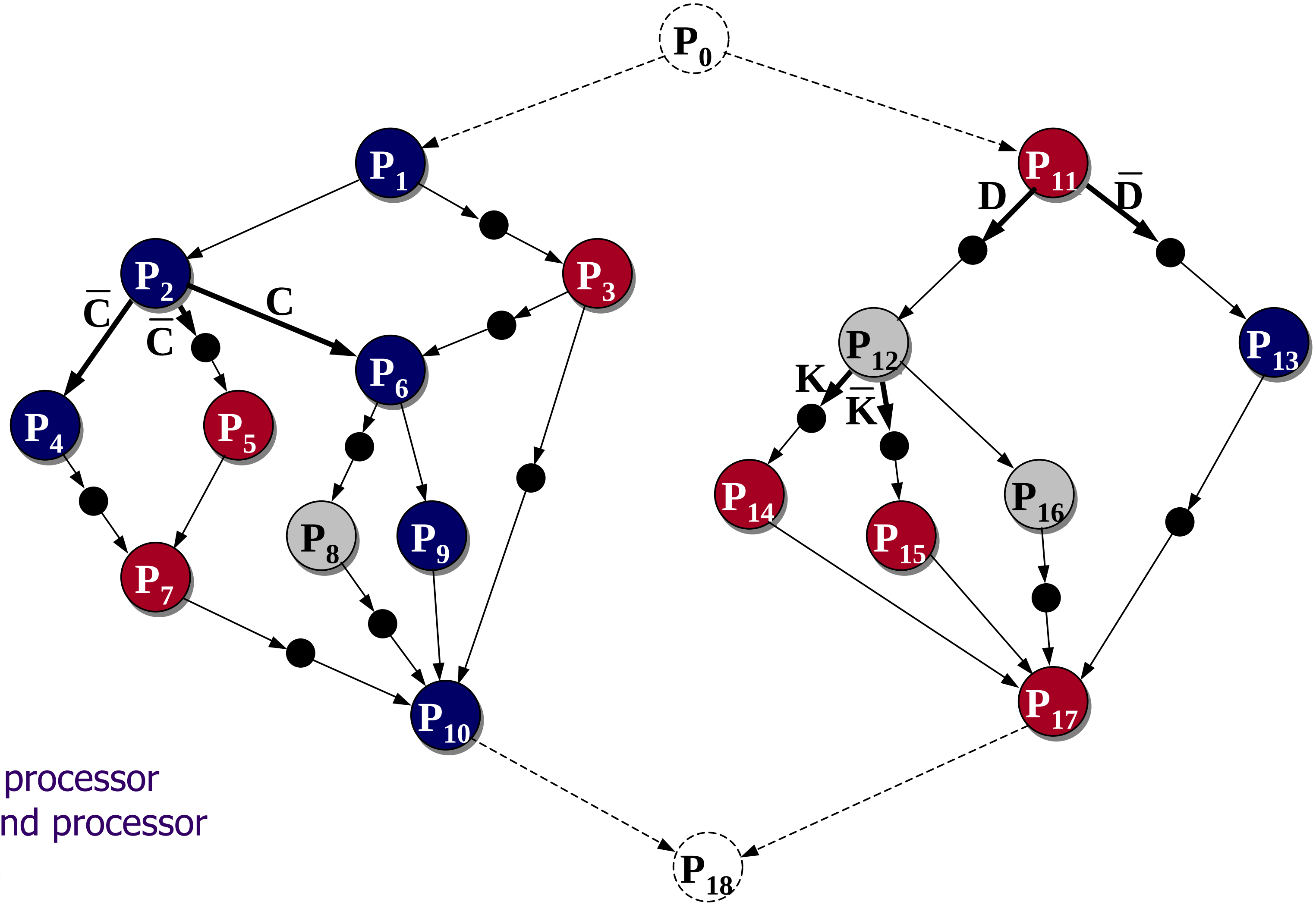
System-Level Design



In this thesis

- Scheduling
- Bus access optimization
- Mapping

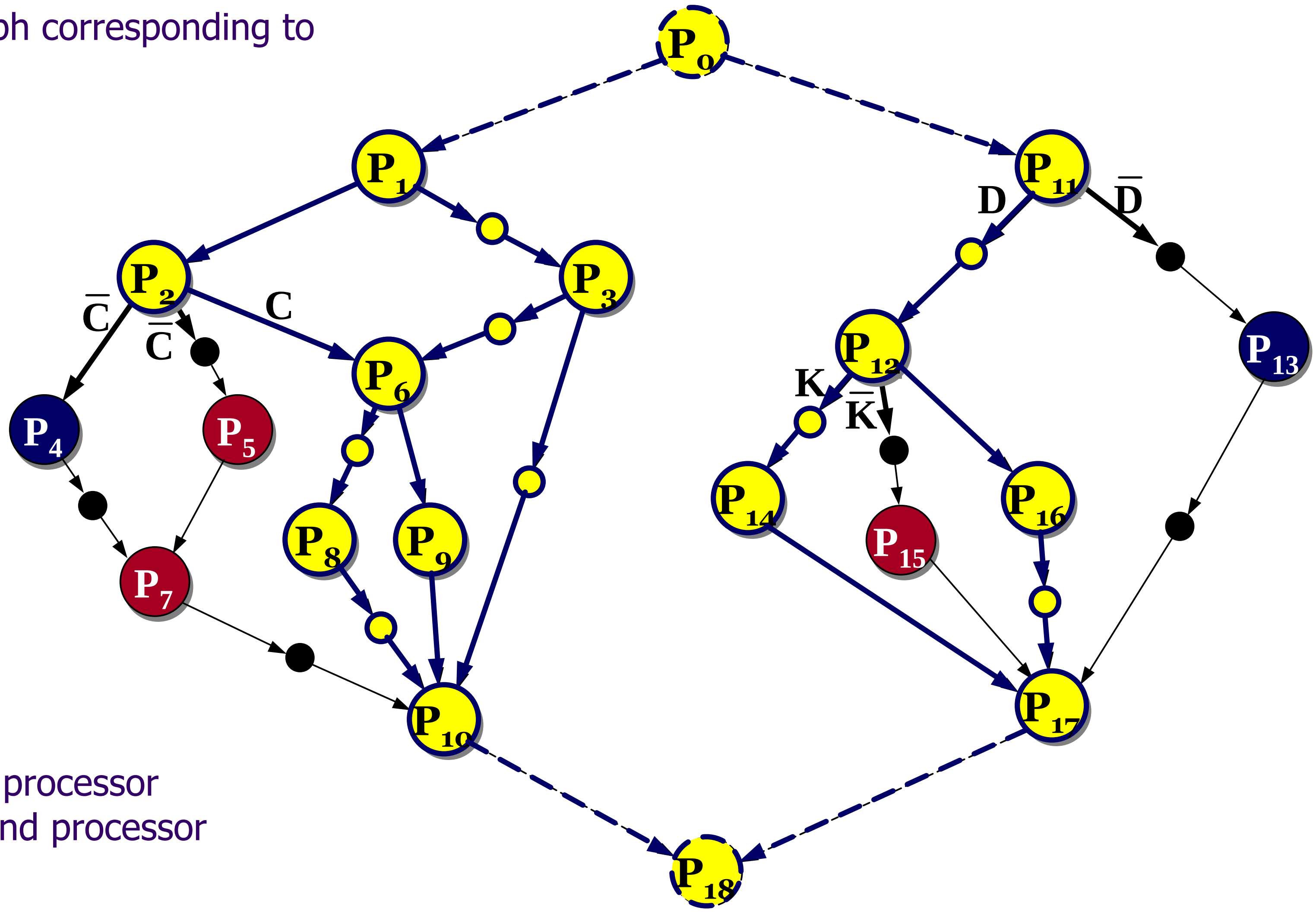
Application Modeling #1



- 🟦 First processor
- 🟥 Second processor
- ⬜ ASIC

Application Modeling #2

Subgraph corresponding to
$D \wedge C \wedge K$

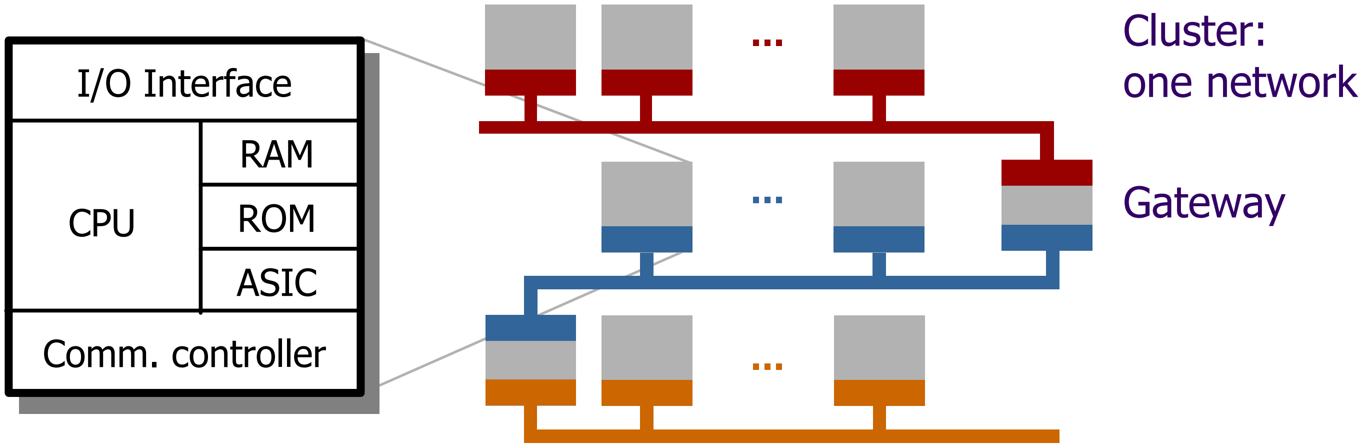


■ First processor

■ Second processor

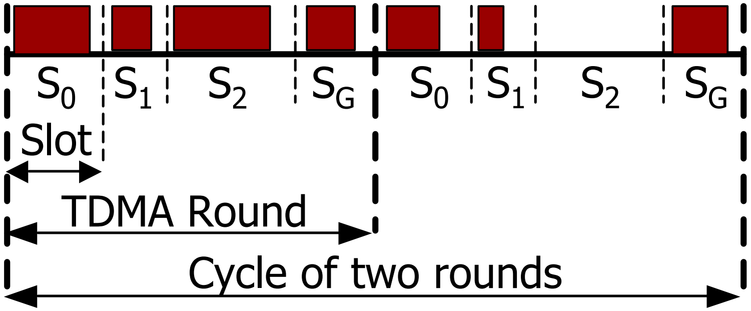
■ ASIC

Hardware Platform



Time Triggered Protocol (**TTP**)

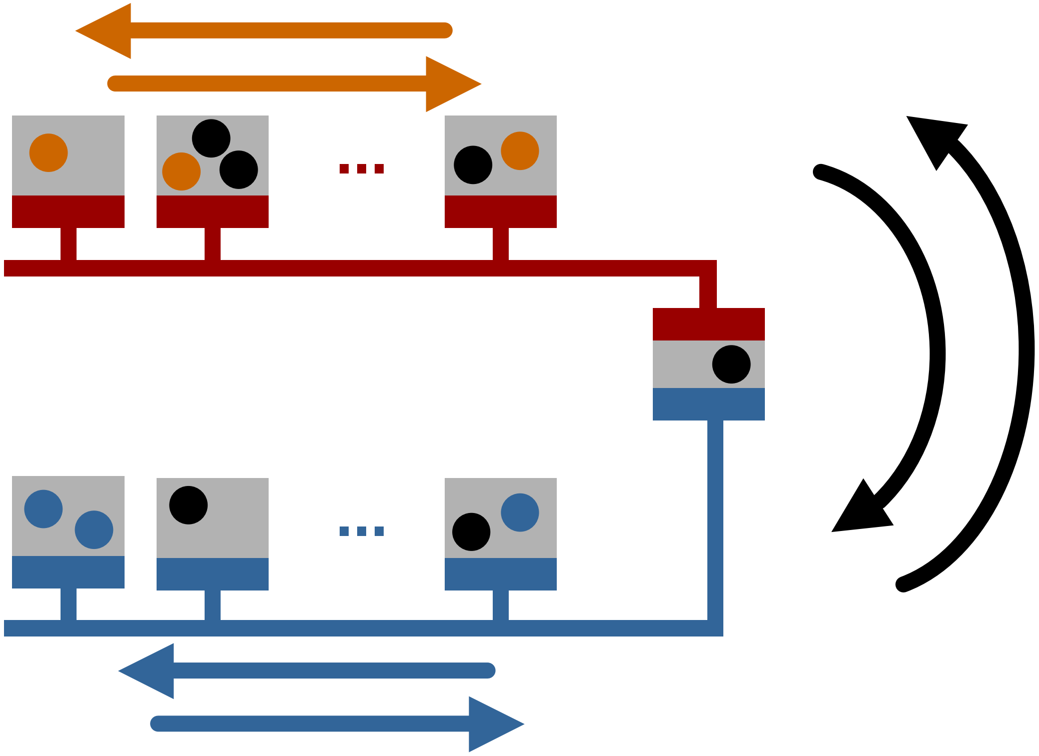
- Bus access scheme: time-division multiple-access (TDMA)
- Schedule table located in each TTP controller: message descriptor list (MEDL)



Controller Area Network (**CAN**)

- Priority bus, collision avoidance
- Highest priority message wins the contention
- Priorities encoded in the frame identifier

Distributed Safety-Critical Applications



- Distributed applications
  - On a single cluster
  - On several clusters
- Motivation
  - Reduce costs:
    use resources efficiently
  - Requirements:
    close to sensors/
    actuators

- Distributed applications are difficult to...
  - Analyze (e.g., guaranteeing timing constraints)
  - Design (e.g., efficient implementation)

Event-Driven vs. Time-Driven Systems

- **Event-driven** systems
  - Activation of processes is done at the occurrence of significant events
  - Scheduling event-triggered activities
    - Fixed-priority preemptive scheduling
    - Response time analysis:
      calculate worst-case response times for each process
    - Schedulability test: response times smaller than the deadlines

- **Time-driven** systems
  - Activation of processes is done at predefined points in time
  - Scheduling time-triggered activities
    - Static cyclic non-preemptive scheduling
    - Building a schedule table:
      static cyclic scheduling (e.g., list scheduling)

Outline

- Introduction
  - System-level design and modeling
    - Conditional process graph
  - The system platform
    - Time-triggered vs. event-triggered

- Communication-intensive heterogeneous real-time systems
  - ➔ Time-driven systems
    - Scheduling and bus access optimization
    - Incremental mapping
  - Event-driven systems
    - Schedulability analysis and bus access optimization
    - Incremental mapping
  - Multi-Cluster Systems
    - Schedulability analysis and bus access optimization
    - Frame packing

- Summary of contributions

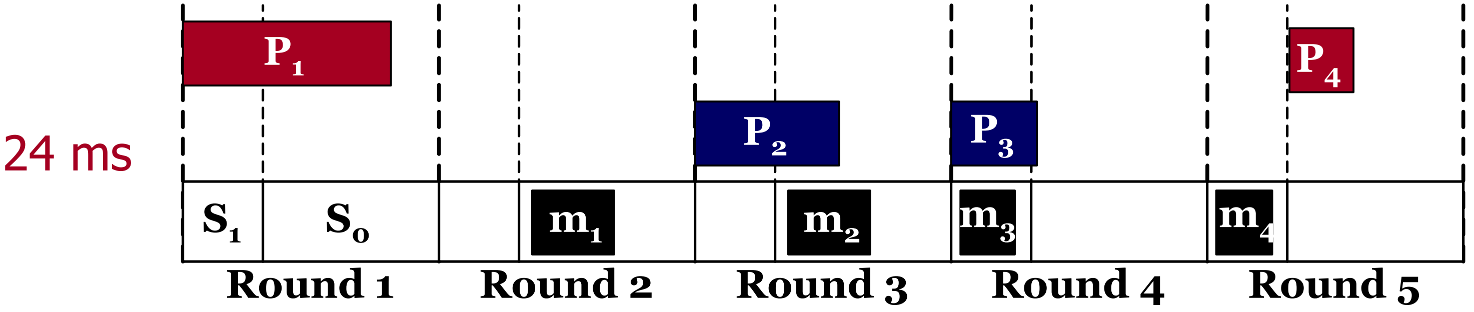
Scheduling and Bus Access Optimization

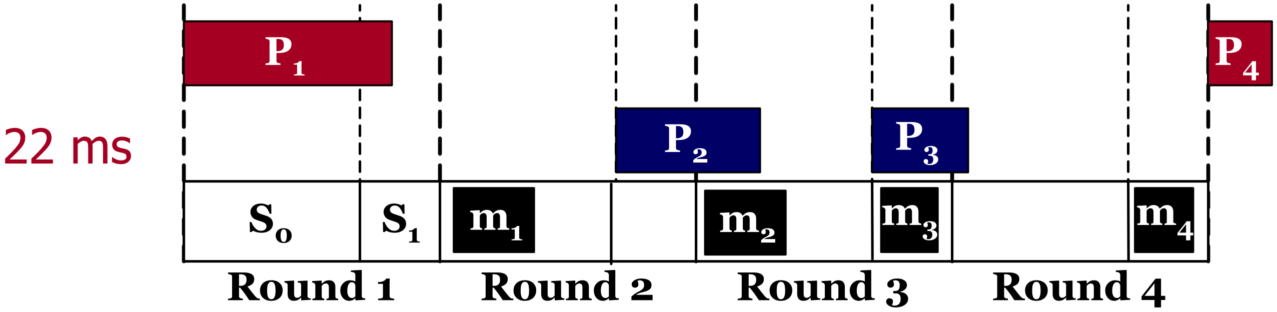
- Input
  - Safety-critical application: set of conditional process graphs
    - The worst-case execution time of each process
    - The size of each messages
  - The system architecture and mapping are given

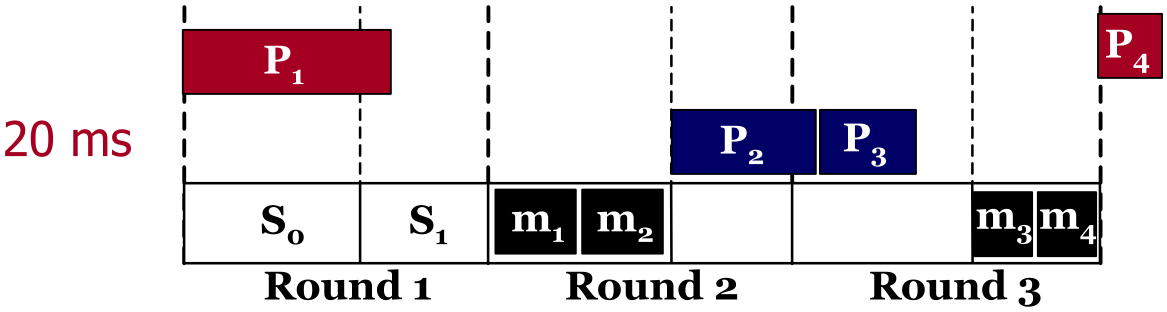
- Output
  - Design implementation
    such that the application is schedulable
    and execution delay is minimized
    - Local schedule tables for each node
    - The sequence and size of the slots in a TDMA round
    - The MEDL (schedule table for messages) for each TTP controller

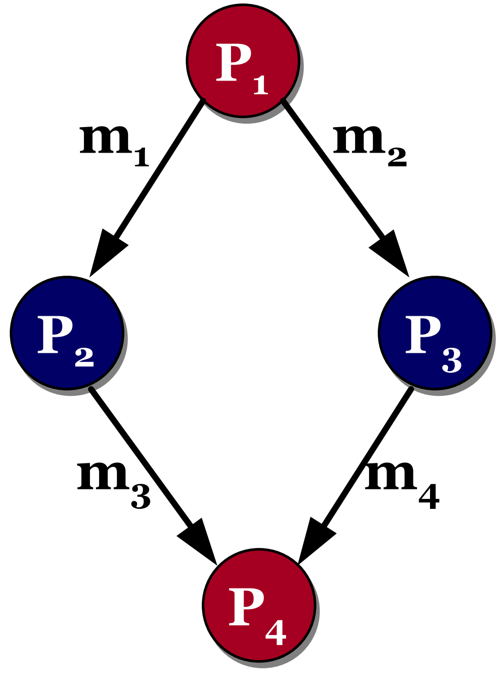
Communication infrastructure parameters

Scheduling and Optimization Example









Scheduling and Optimization Strategy

- List scheduling based algorithm
  - The scheduling algorithm has to take into consideration the TTP
  - Priority function for the list scheduling

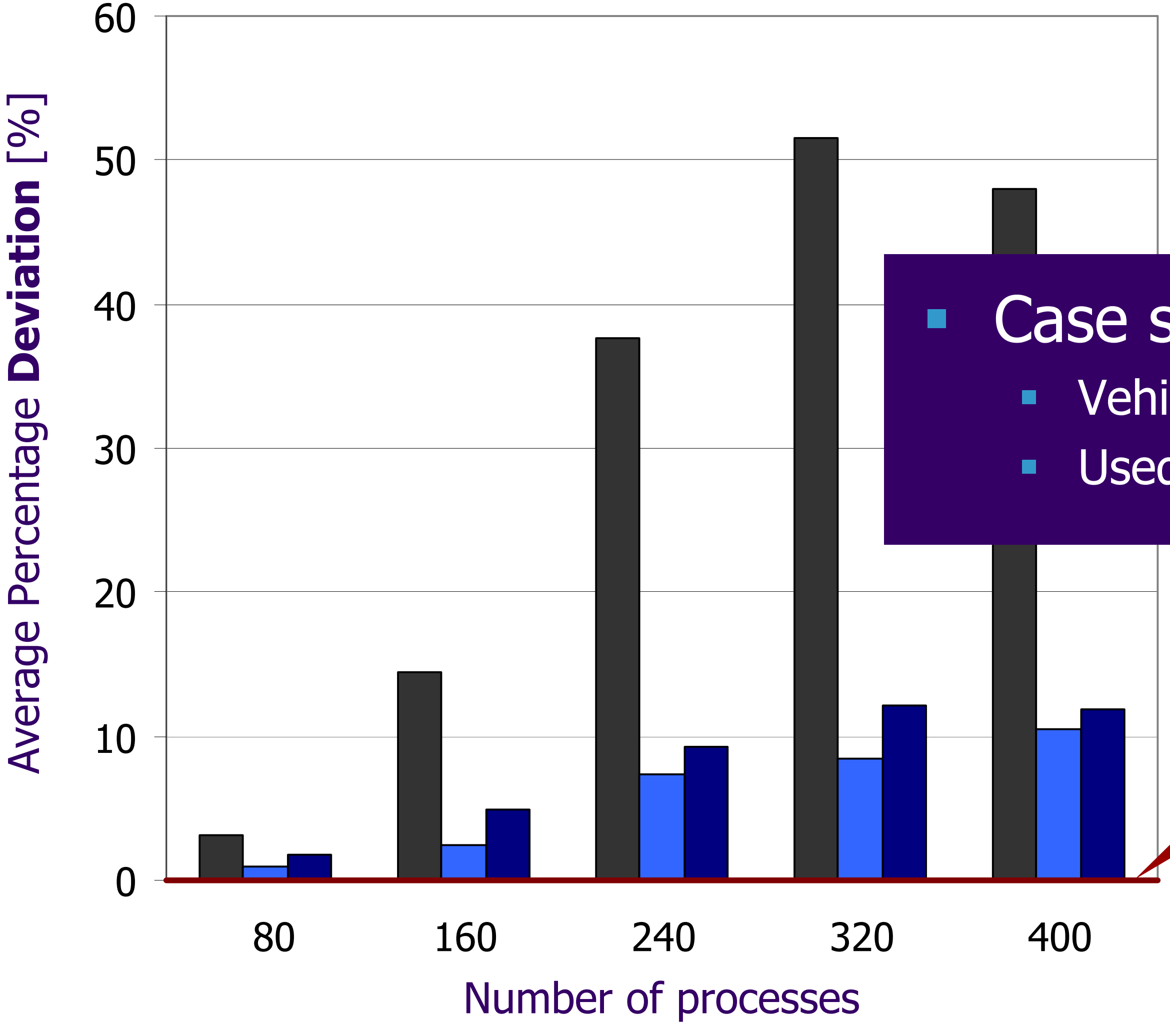
- Bus access optimization heuristics
  - Greedy heuristic, two variants
    - **Greedy 1** tries all possible slot lengths
    - **Greedy 2** uses feedback from the scheduling algorithm

  - Simulated Annealing
    - Produces near-optimal solutions in a very large time
    - Cannot be used inside a design space exploration loop
    - Used as the baseline for comparisons

  - **Straightforward solution**
    - Finds a schedulable application
    - Does not consider the optimization of the design

Can We Improve the Schedules?

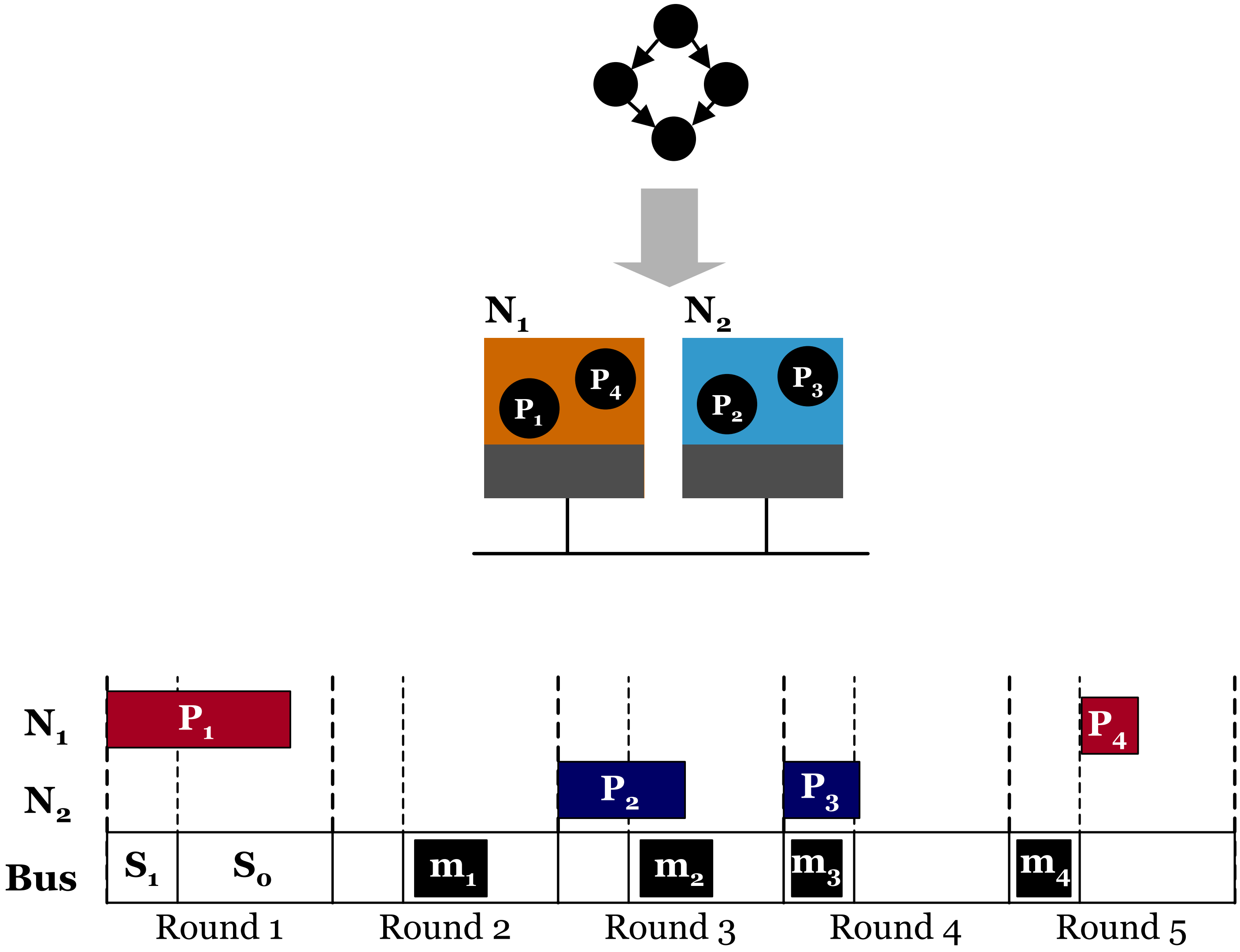
Cost function: schedule length



- Case study
  - Vehicle cruise controller
  - Used throughout the thesis

Baseline:
Simulated Annealing

“Classic” Mapping and Scheduling Example

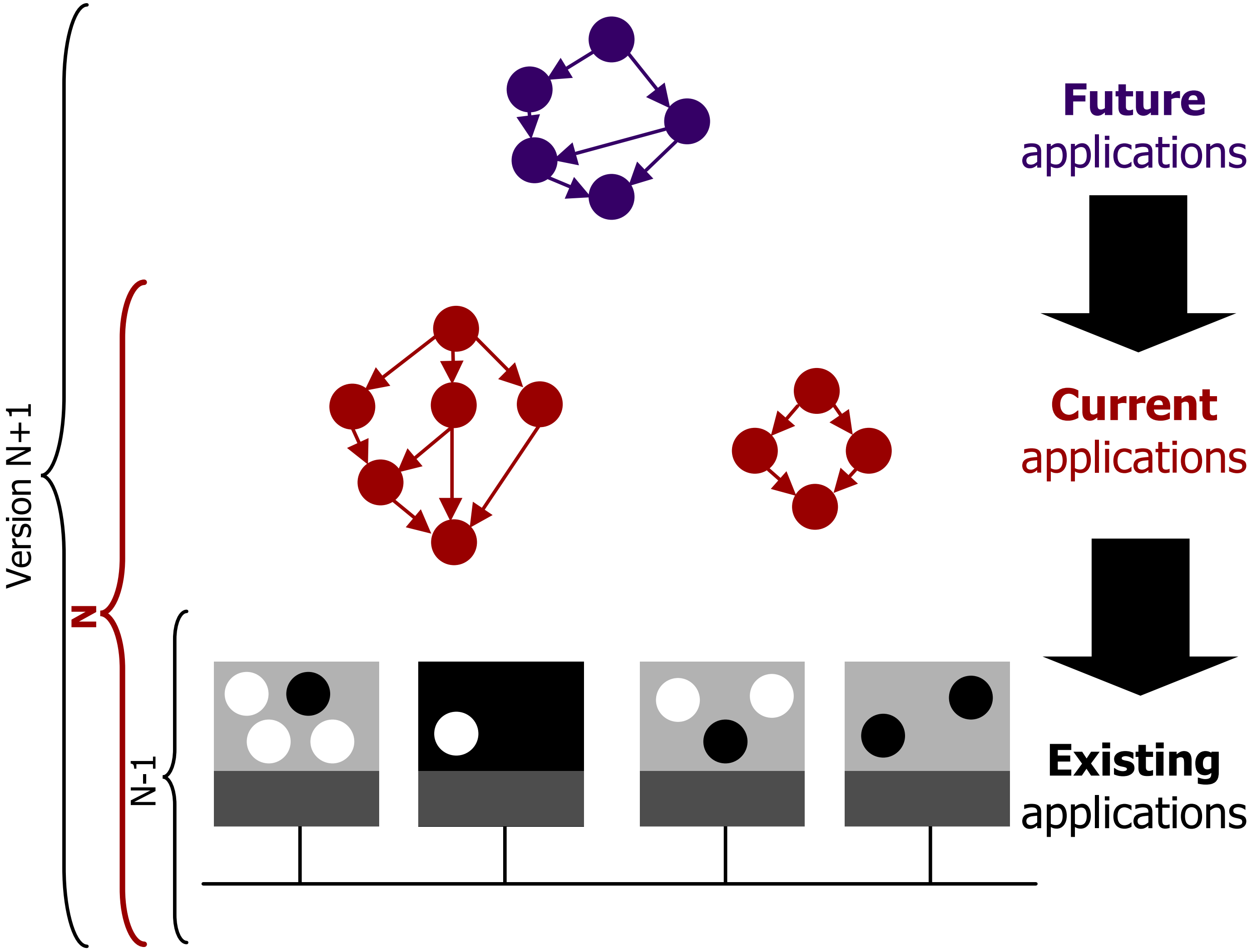


Incremental Design Process

- Start from an already existing system with applications
  - In practice, very uncommon to start from scratch

- Implement new functionality on this system (increment)
  - As few as possible modifications of the existing applications, to reduce design and testing time
  - Plan for the next increment:
    It should be easy to add functionality in the future

Incremental Mapping and Scheduling



Future applications

Current applications

Existing applications

Do not exist yet at Version N!

Map and schedule so that the future applications will have a chance to fit

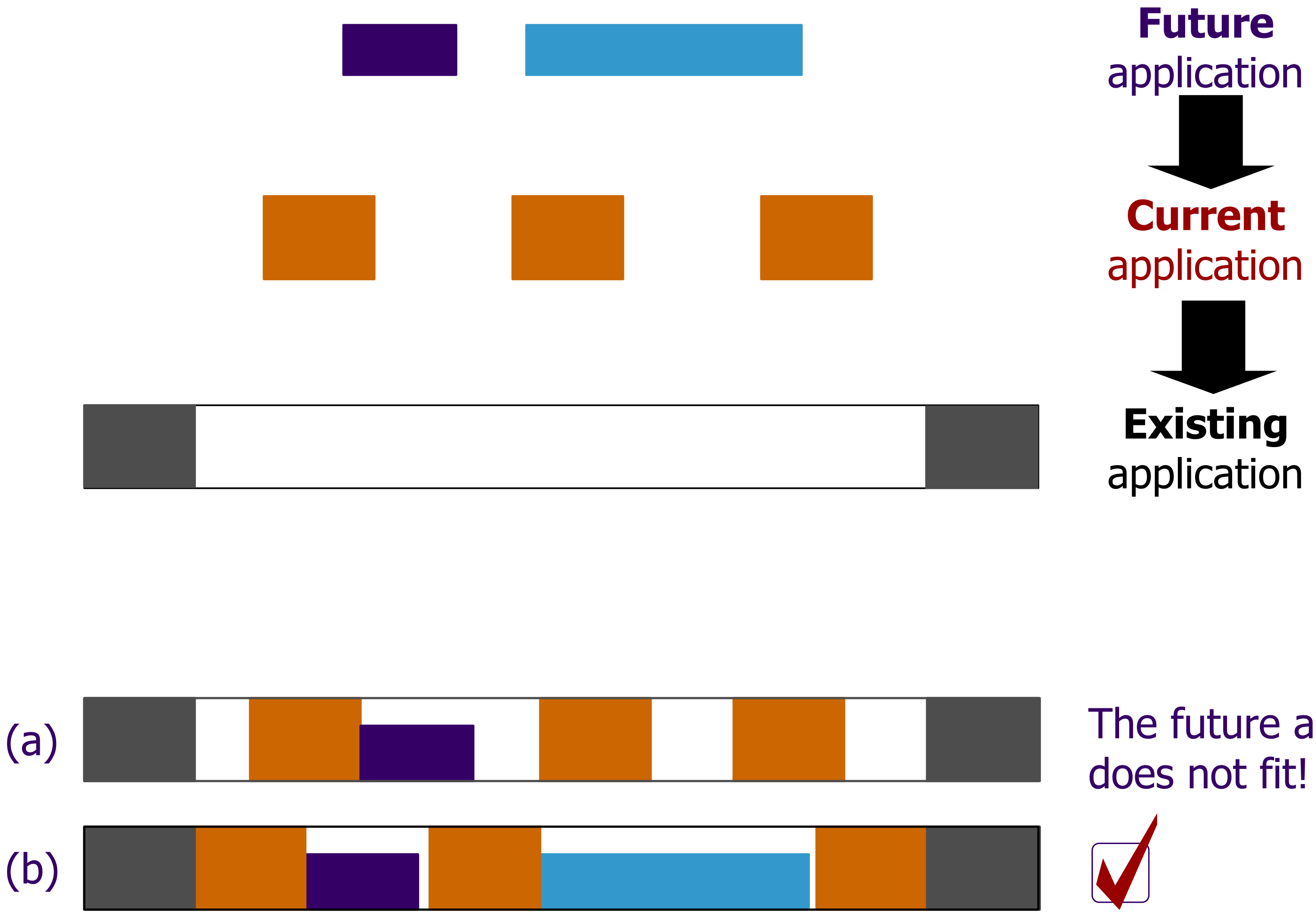
Minimal modifications are performed to the existing applications

Incremental Mapping and Scheduling

- Input
  - A set of existing applications modeled using process graphs
  - A current application to be mapped modeled using process graphs
    - Each process graph in an application has its own period and deadline
    - Each process has a potential set of nodes to be mapped on and a WCET
  - Characteristics of the future applications
  - The system architecture is given

- Output
  - A mapping and scheduling of the current application, such that:
    - **Requirement (a)**
      constraints of the current application are satisfied
      and minimal modifications are performed to the existing applications
    - **Requirement (b)**
      new future applications can be mapped on the resulted system

Mapping and Scheduling Example



Future application

Current application

Existing application

(a)

(b)

The future application does not fit!

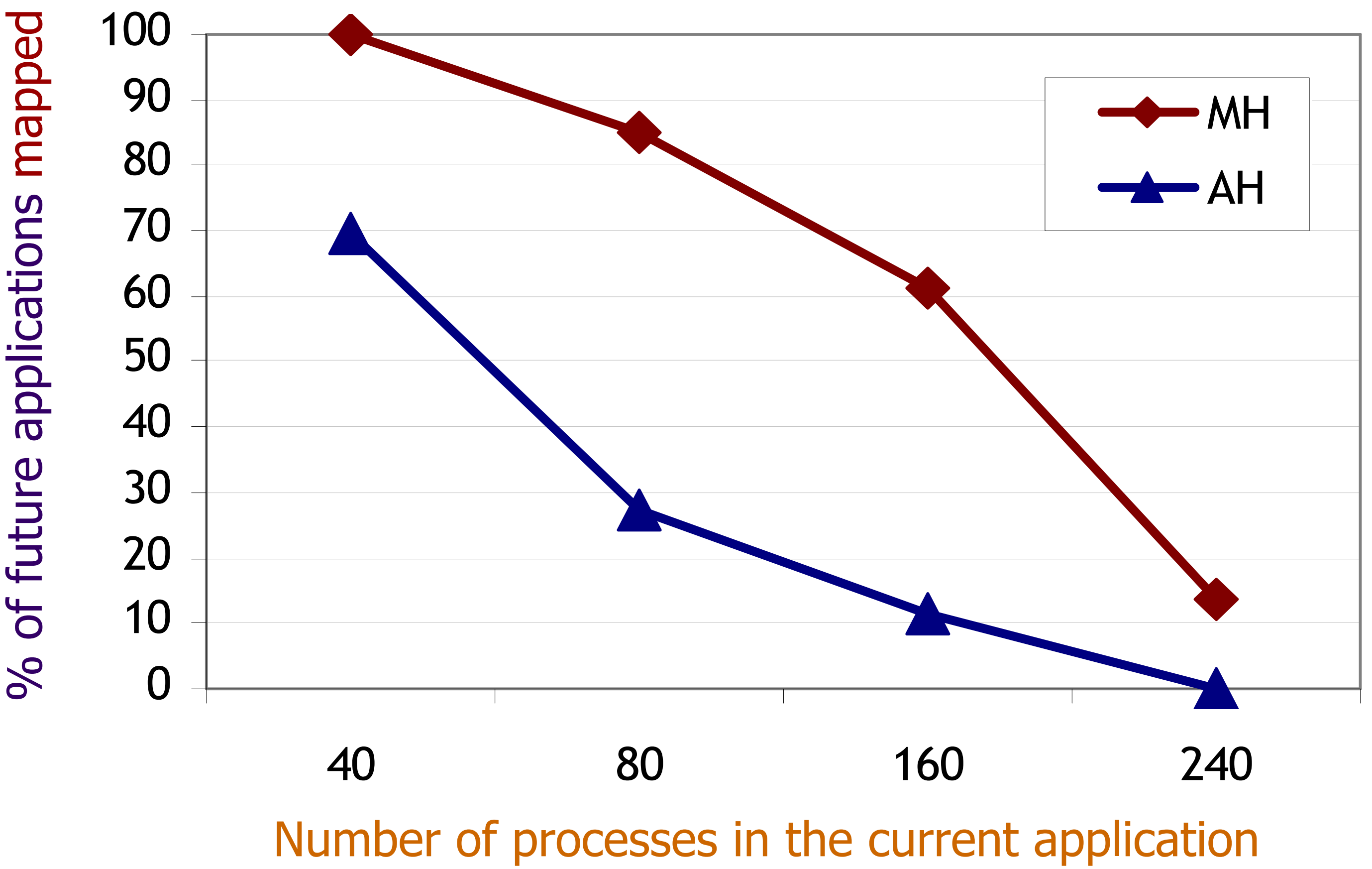
✓

Mapping and Scheduling Strategies

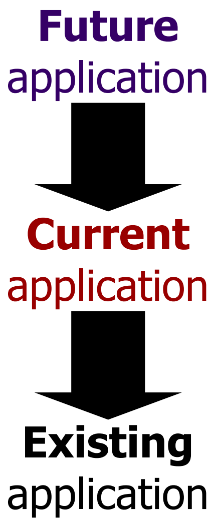
- Design optimization problem
  - Design criteria reflect the degree to which a design supports an incremental design process
  - Design metrics quantify the degree to which the criteria are met
- Heuristics to improve the design metrics
  - **Ad-hoc approach**
    - Little support for incremental design
  - **Mapping Heuristic**
    - Iteratively performs design transformations that improve the design

Can We Support Incremental Design?

Are the mapping strategies proposed facilitating
the implementation of future applications?



existing applications: 400, future application: 80



Outline

- Introduction
  - System-level design and modeling
    - Conditional process graph
  - The system platform
    - Time-triggered vs. event-triggered
- Communication-intensive heterogeneous real-time systems
  - Time-driven systems
    - Scheduling and bus access optimization
    - Incremental mapping
  - ➔ Event-driven systems
    - Schedulability analysis and bus access optimization
    - Incremental mapping
  - Multi-Cluster Systems
    - Schedulability analysis and bus access optimization
    - Frame packing
- Summary of contributions

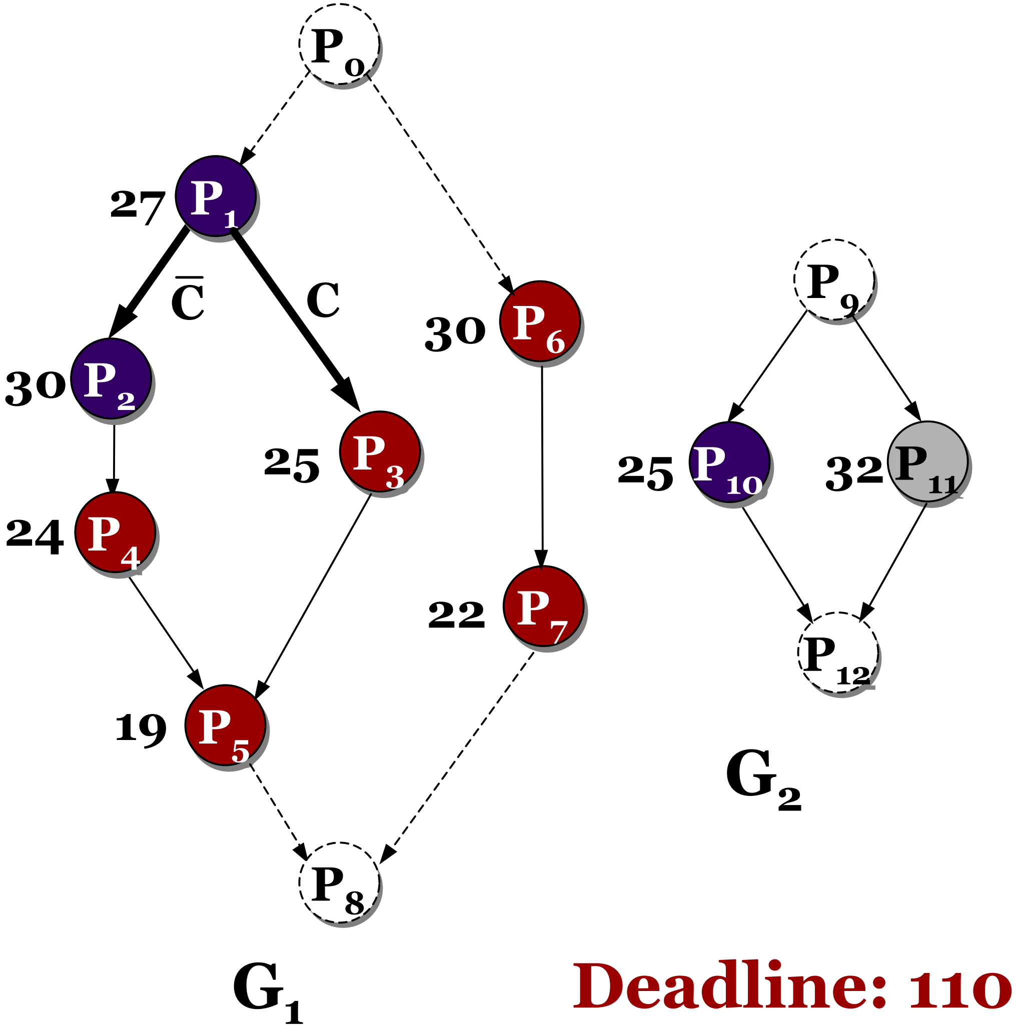
Scheduling and Bus Access Optimization

- Input
  - Safety-critical application: set of conditional process graphs
    - The worst-case execution time of each process
    - The size of each messages
  - The system architecture and mapping are given

- Output
  - Worst-case response times (schedulability analysis)
  - Design implementation
    such that the application is schedulable
    and execution delay is minimized
    - The sequence and size of the slots in a TDMA round
    - The MEDL (schedule table for messages) for each TTP controller

Communication infrastructure parameters

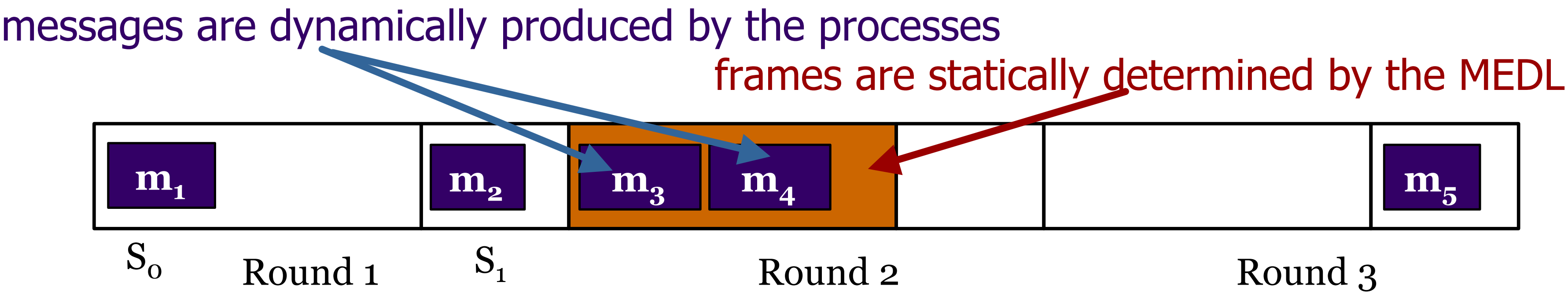
Conditional Process Graph Scheduling



CPG	Worst Case Delays	
	Not considering conditions	Considering conditions
G_1	120	100
G_2	82	82

Deadline: 110

Scheduling of Messages using the TTP


messages are dynamically produced by the processes
frames are statically determined by the MEDL
m_1
m_2
m_3
m_4
m_5
S_0
Round 1
S_1
Round 2
Round 3

1. Single message per frame, allocated statically:
   Static Single Message Allocation

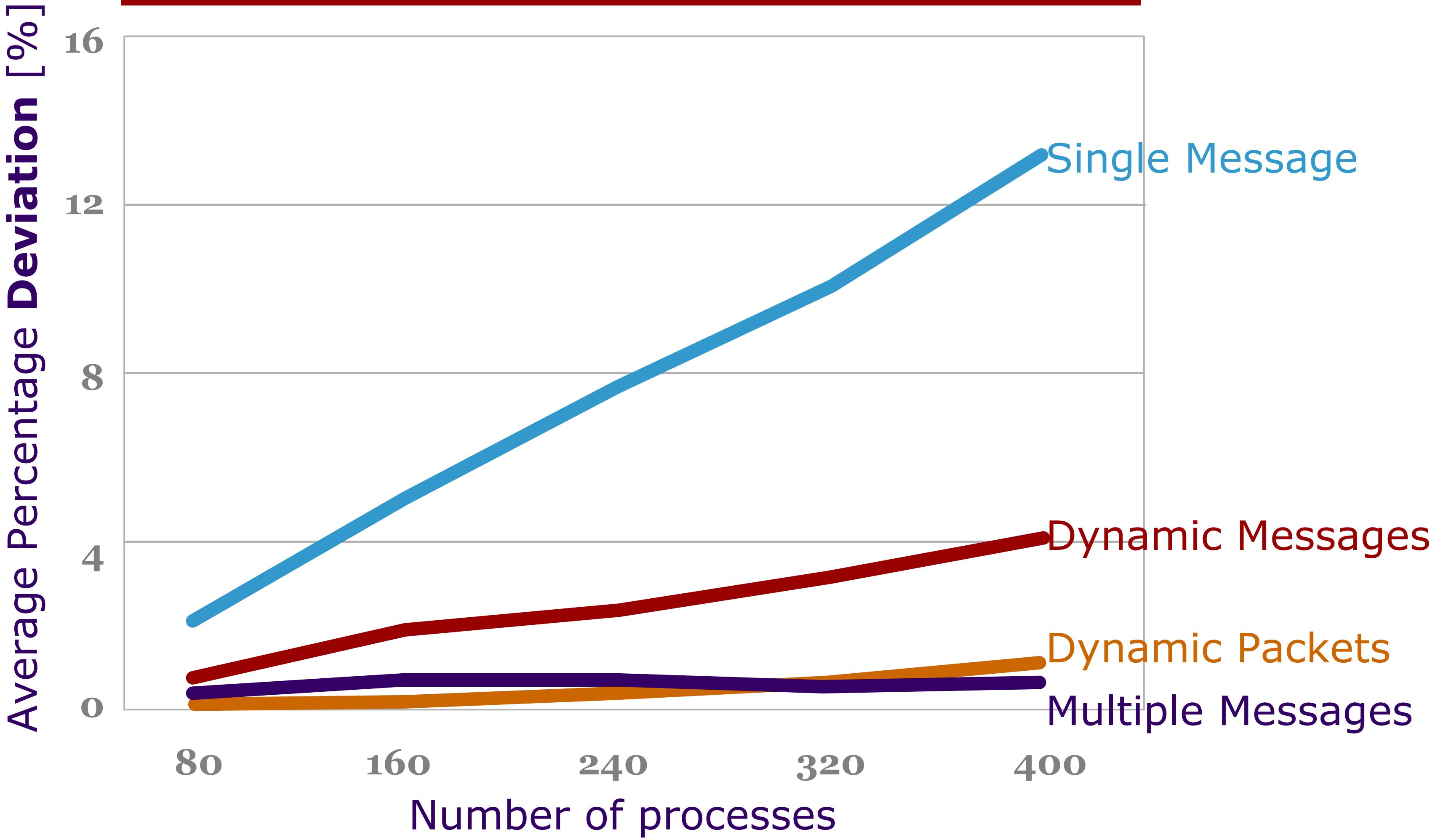
2. Several messages per frame, allocated statically:
   Static Multiple Message Allocation

3. Several messages per frame, allocated dynamically:
   Dynamic Message Allocation

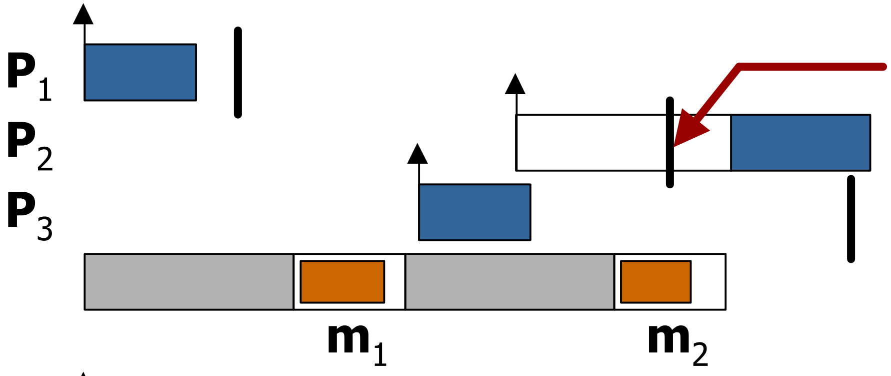
4. Several messages per frame, split into packets, allocated dynamically
   Dynamic Packets Allocation

Compartison...

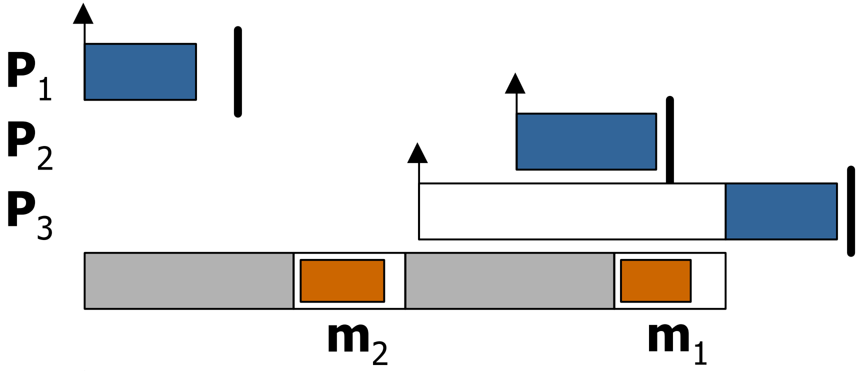
Cost function: degree of schedulability



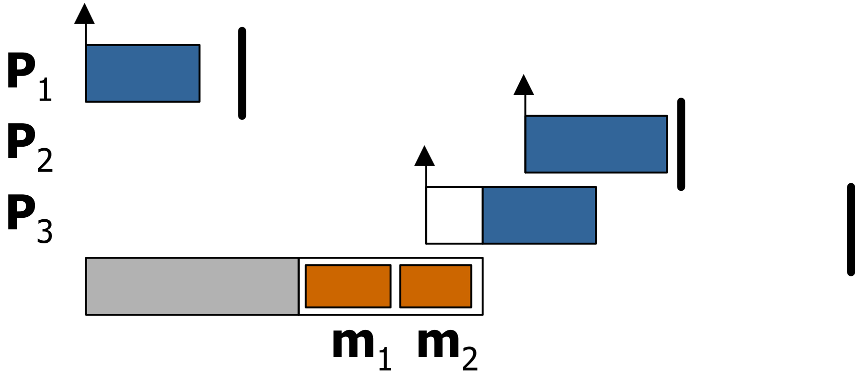
Optimizing Bus Access (Static Allocation)



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.

Outline

- Introduction
    - System-level design and modeling
        - Conditional process graph
    - The system platform
        - Time-triggered vs. event-triggered

- Communication-intensive heterogeneous real-time systems
    - Time-driven systems
        - Scheduling and bus access optimization
        - Incremental mapping
    - Event-driven systems
        - Schedulability analysis and bus access optimization
        - Incremental mapping
    - ➔ Multi-Cluster Systems
        - Schedulability analysis and bus access optimization
        - Frame packing

- Summary of contributions

Schedulability Analysis and Optimization

- Input
  - An application modeled as a set of process graphs
  - Each process has an worst case execution time, a period, and a deadline
  - Each message has a known size
  - The system architecture and the mapping of the application are given

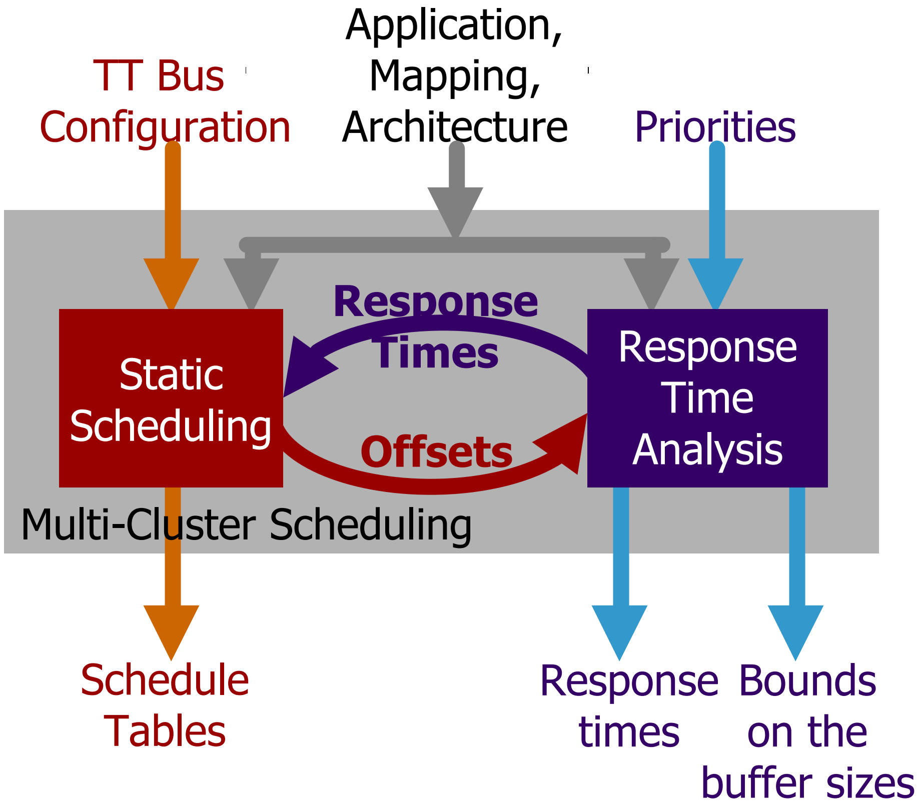
- Output
  - Worst case response times and bounds on the buffer sizes
  - Design implementation
    such that the application is schedulable and buffer sizes are minimized
    - Schedule table for TT processes
    - Priorities for ET processes
    - Schedule table for TT messages
    - Priorities for ET messages
    - TT bus configuration
      (TDMA slot sequence and sizes)

Communication infrastructure parameters

System configuration parameters

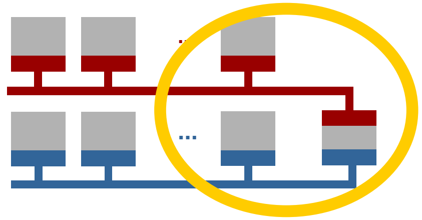
Multi-Cluster Scheduling

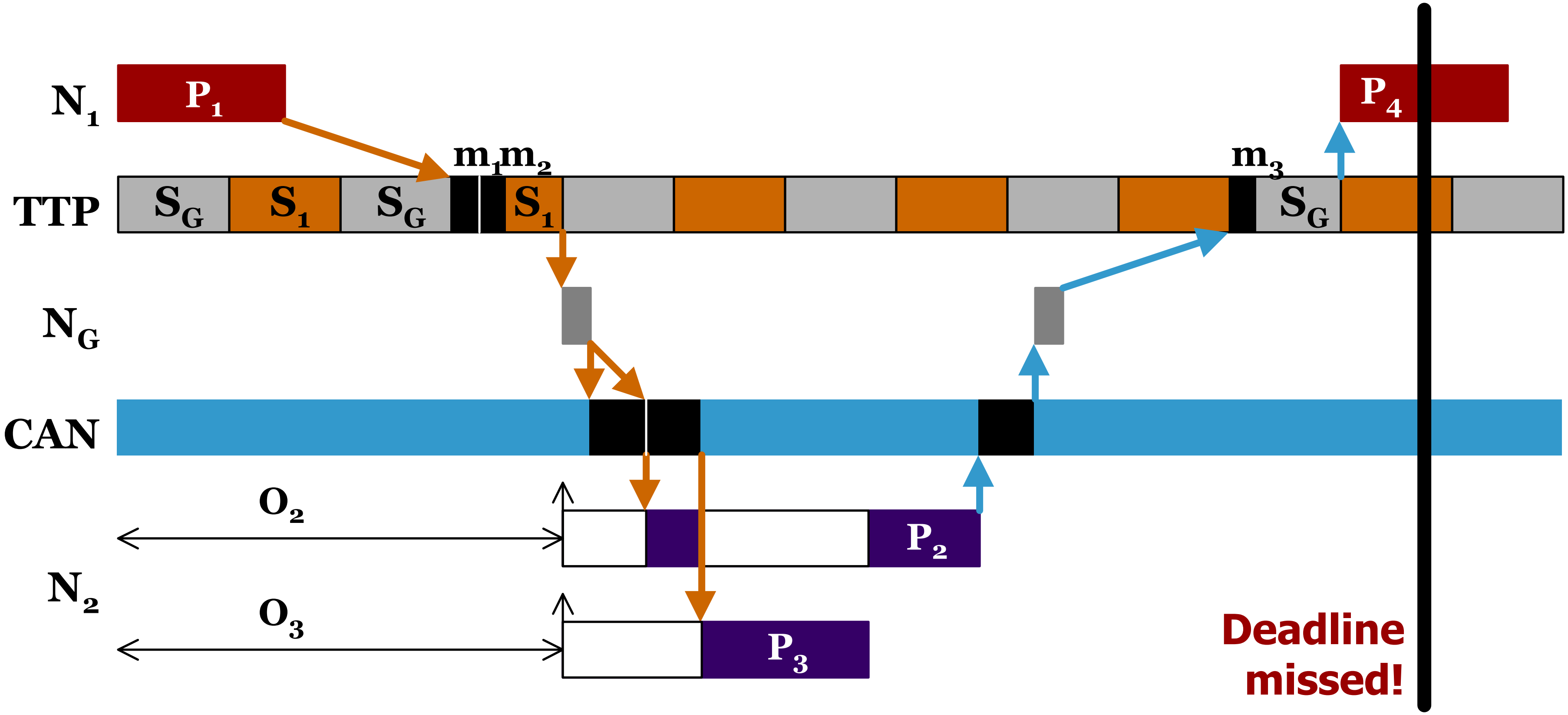
- Scheduling cannot be addressed separately for each type of cluster
- The inter-cluster communication creates a **circular dependency**:
  - TTC static schedules (offsets) Þ ETC response times
  - ETC response times Þ TTC schedule table construction



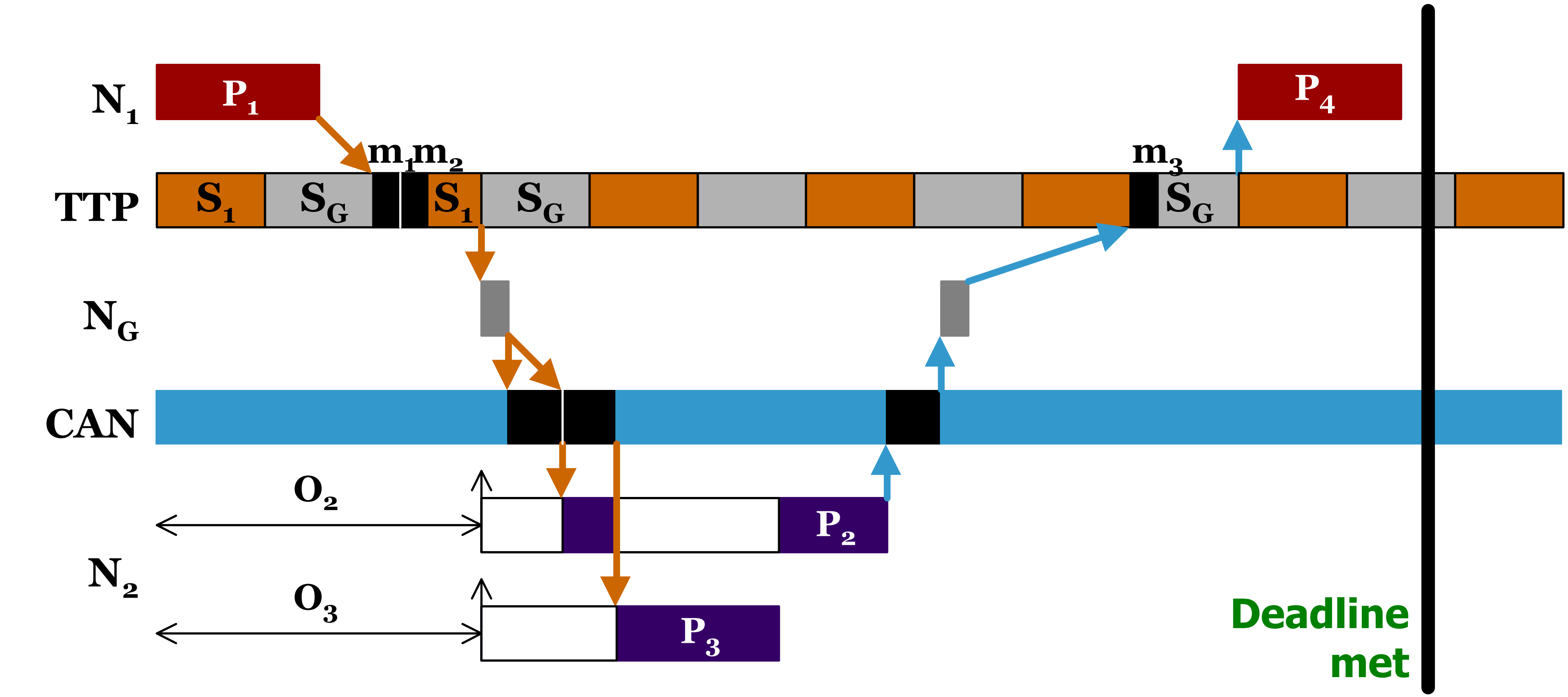
Offset
earliest possible start time for an **event-triggered** activity

Optimization Example #1



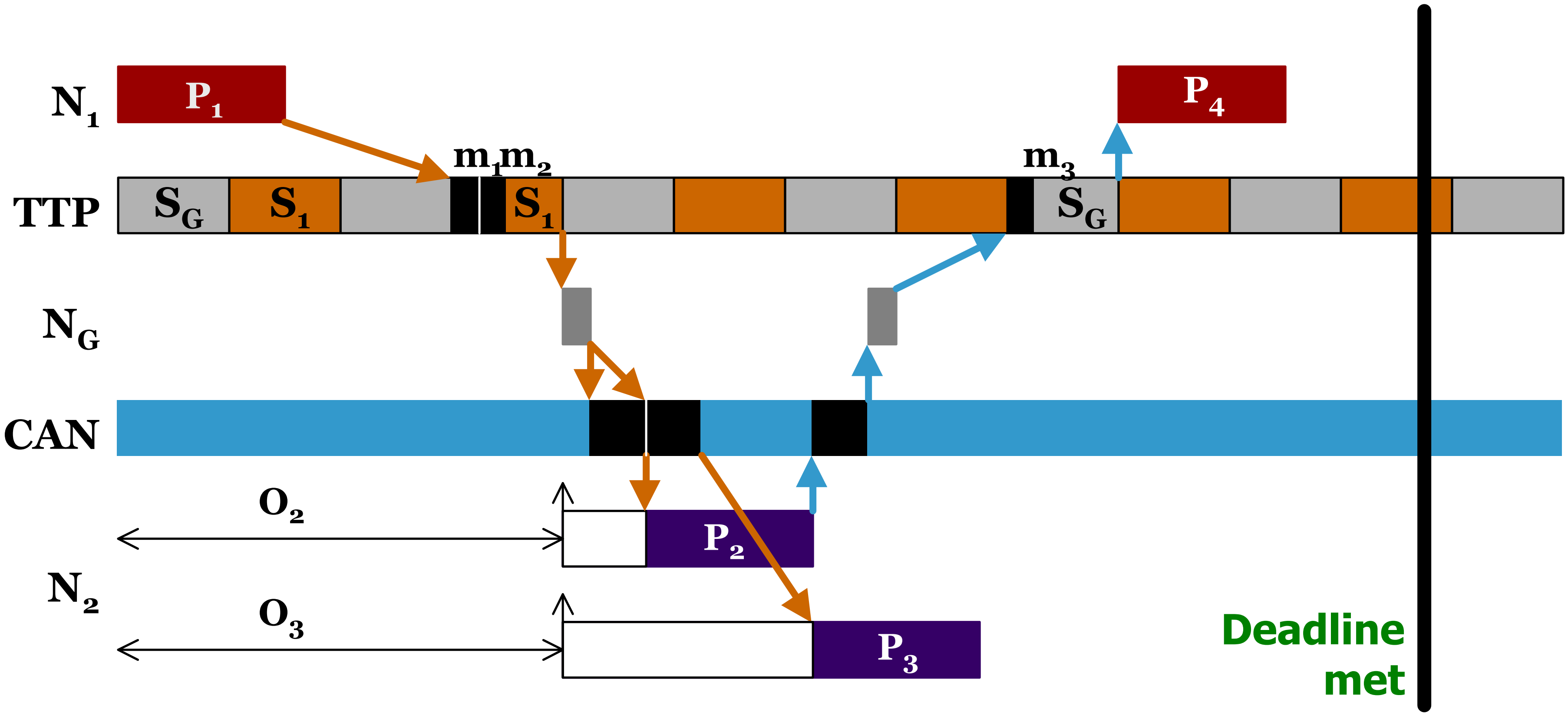


Optimization Example #2



Transformation: S_1 is the first slot, m_1 and m_2 are sent sooner

Optimization Example #3



Transformation: P_2 is the high priority process on N_2

Optimization Strategies

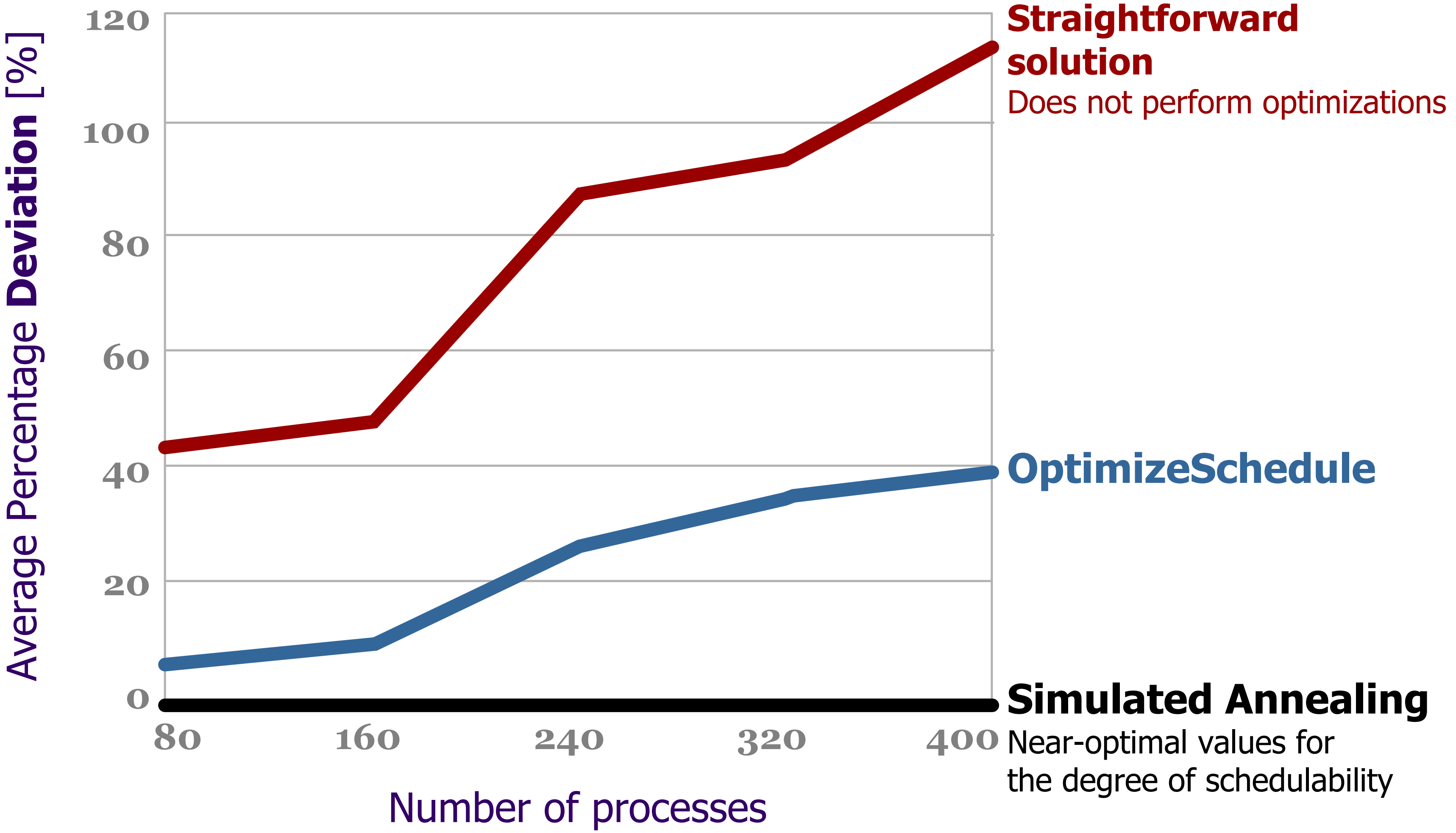
- **OptimizeSchedule**
  - Synthesizes the communication and assigns priorities to obtain a schedulable application
  - Based on a greedy approach
    - Cost function: degree of schedulability

- **OptimizeBuffers**
  - Synthesizes the communication and assigns priorities to reduce the total buffer size
  - Based on a hill-climbing heuristic
    - Cost function: total buffer size

- **Straightforward solution**
  - Finds a schedulable application
  - Does not consider the optimization of the design

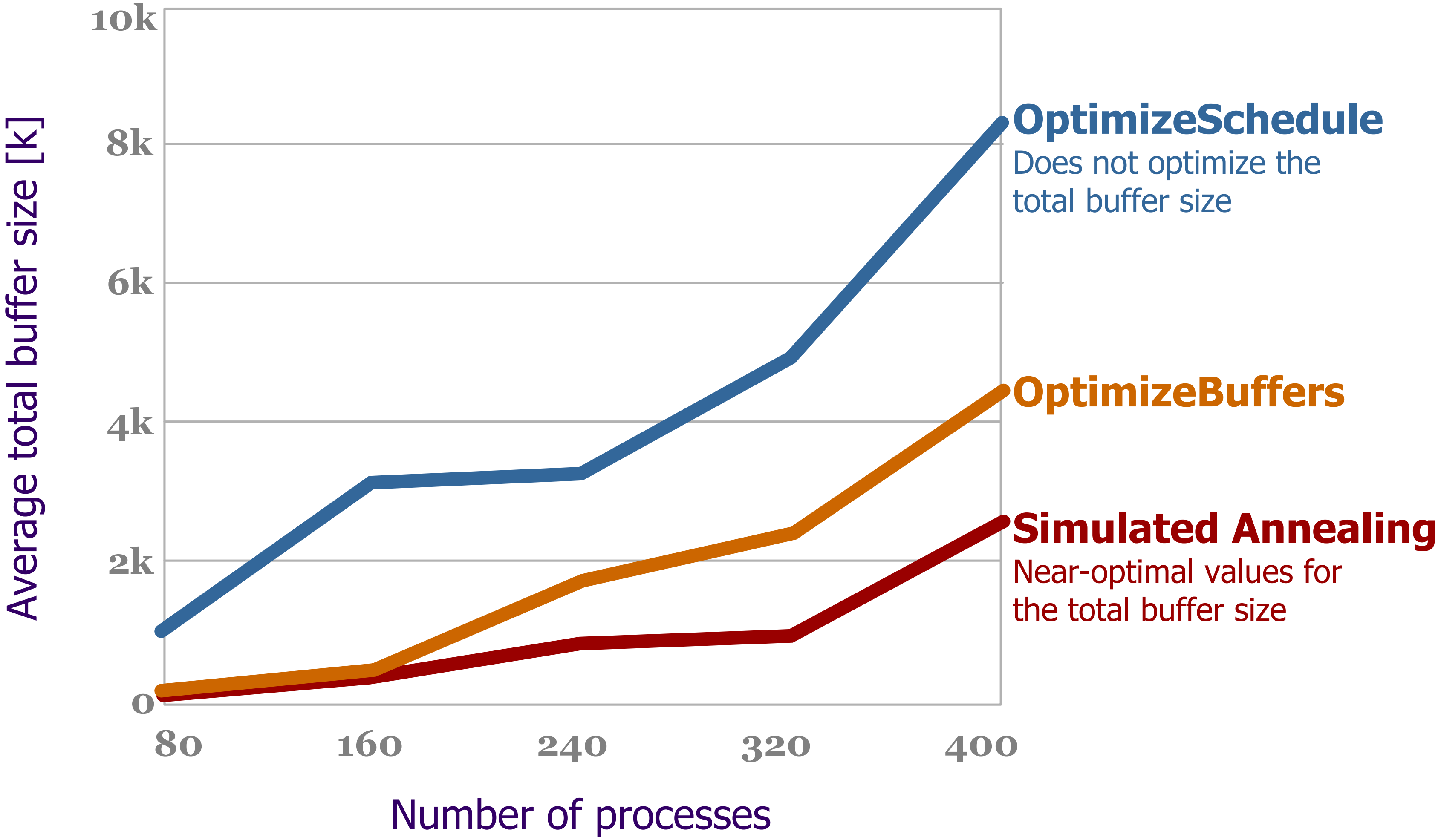
Can We Improve Schedulability?

Cost function: degree of schedulability



Can We Reduce Buffer Sizes?

Cost function: total buffer size



OptimizeSchedule: Does not optimize the total buffer size

Simulated Annealing: Near-optimal values for the total buffer size

Outline

- Introduction
  - System-level design and modeling
    - Conditional process graph
  - The system platform
    - Time-triggered vs. event-triggered

- Communication-intensive heterogeneous real-time systems
  - Time-driven systems
    - Scheduling and bus access optimization
    - Incremental mapping
  - Event-driven systems
    - Schedulability analysis and bus access optimization
    - Incremental mapping
  - Multi-Cluster Systems
    - Schedulability analysis and bus access optimization
    - Frame packing

➔ Summary of contributions

Thesis Contributions

- **Time-driven** systems
  - Static scheduling strategy
  - Optimization strategies for the synthesis of the bus access scheme
  - Mapping and scheduling within an incremental design process

- **Event-driven** systems
  - Schedulability analysis
  - Optimization strategies for the synthesis of the bus access scheme
  - Mapping and scheduling within an incremental design process

- **Multi-cluster** systems
  - Schedulability analysis for multi-cluster systems
  - Optimization heuristics for system synthesis:
    minimal buffer sizes needed to run a schedulable application
  - Frame-packing optimization heuristics