Model-based Development of Embedded Systems: The Rubus Approach

Saad Mubeen

Learning Objectives of this Lecture

After this lecture you should be able to explain and reflect on

- Model- and component-based software development of embedded systems using a state-of-the-practice approach
  - Modeling of software architecture
  - End-to-end timing analysis

Outline

There are six parts of this lecture

1. Modeling of software architecture and end-to-end timing analysis
2. Group work 1: Solve a proplem collaboratively using the FiSH model
3. Discussion
4. Group work 2 (class assignment): End-to-end timing analysis of a simple distributed system
5. Discussion
   - Class assignment
   - Assignment 5
6. Rubus-ICE demonstration

10 minutes

20 minutes

20 minutes

45 minutes

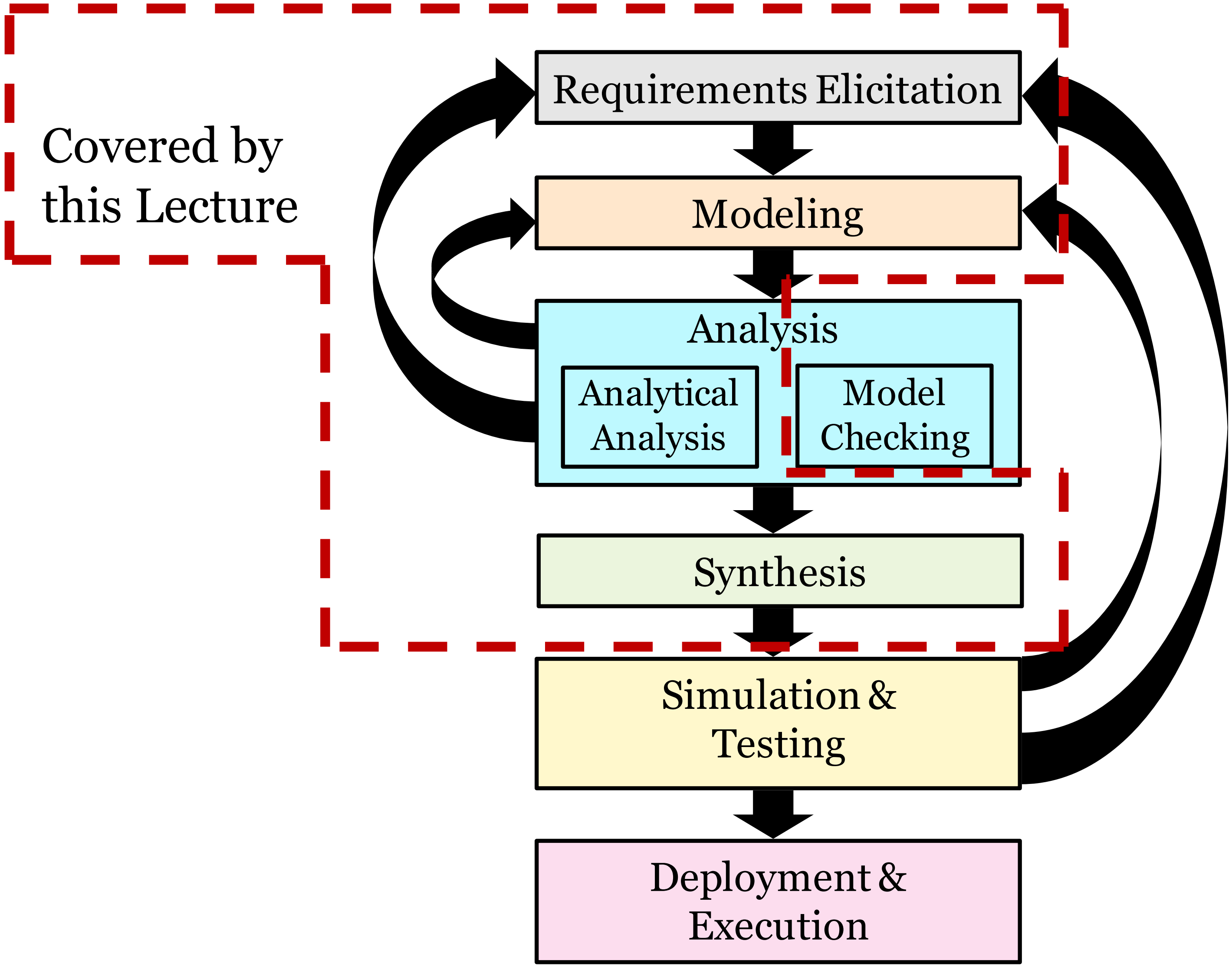
30 minutes

20 minutes

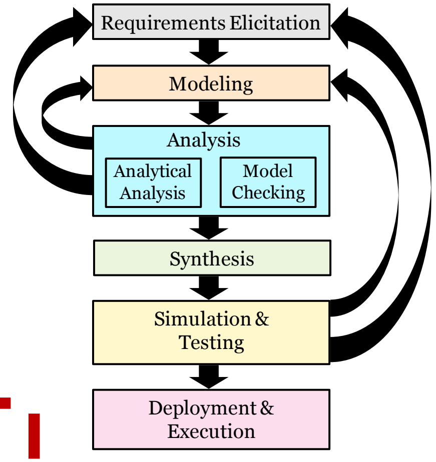
Part 1:

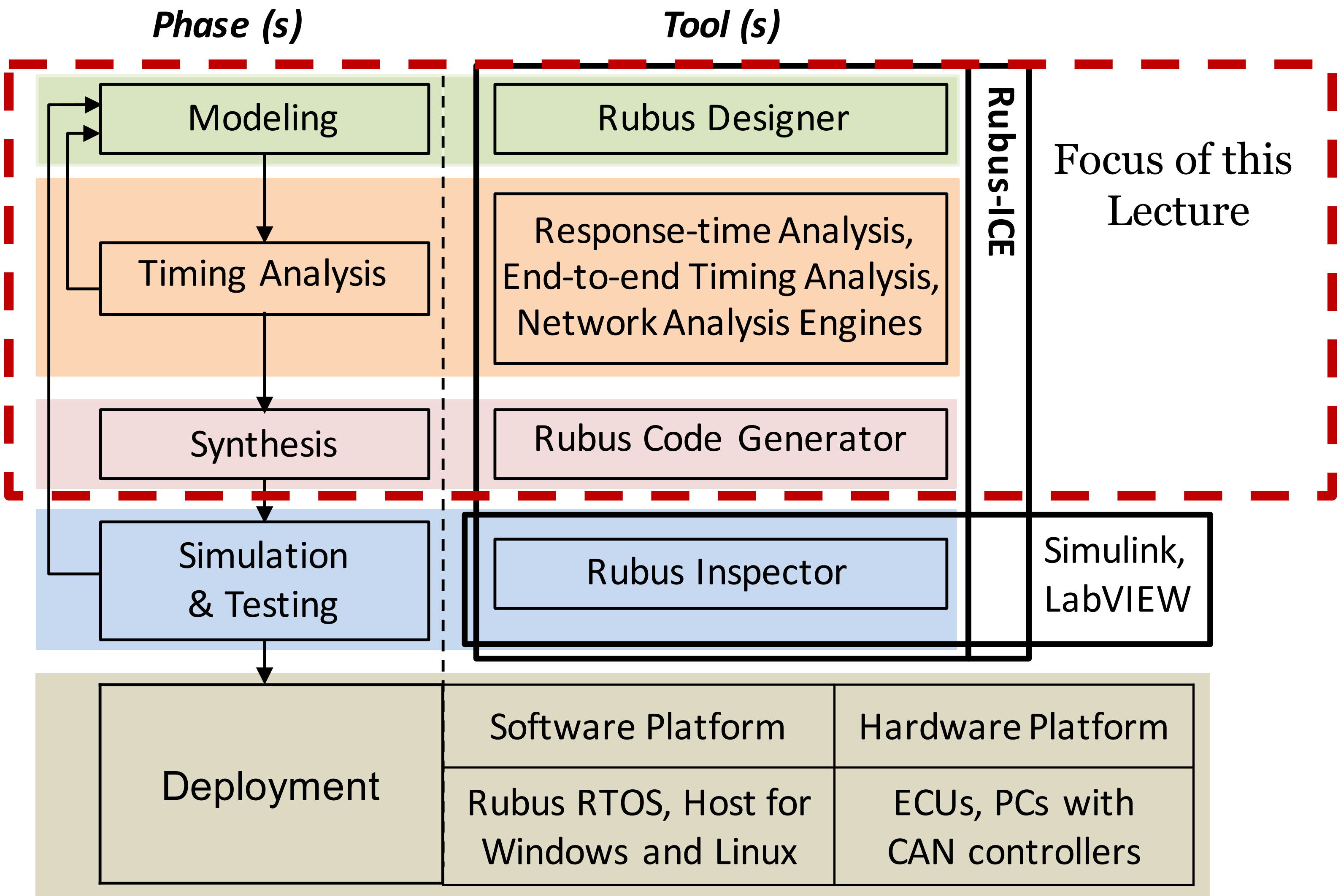
Modeling and Timing Analysis

Model-based Software Development Process (Reap from Lecture 1)



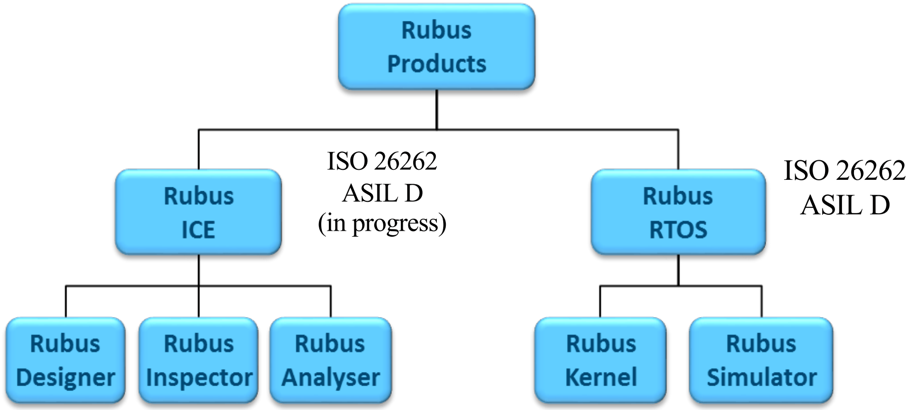
Rubus Development Process

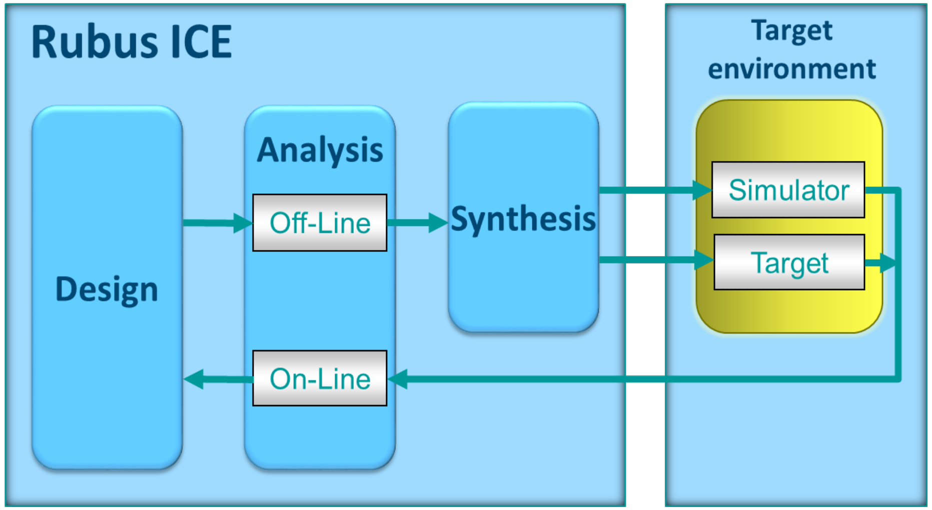




Rubus Tool Suite

Developed by Arcticus Systems





BAE SYSTEMS

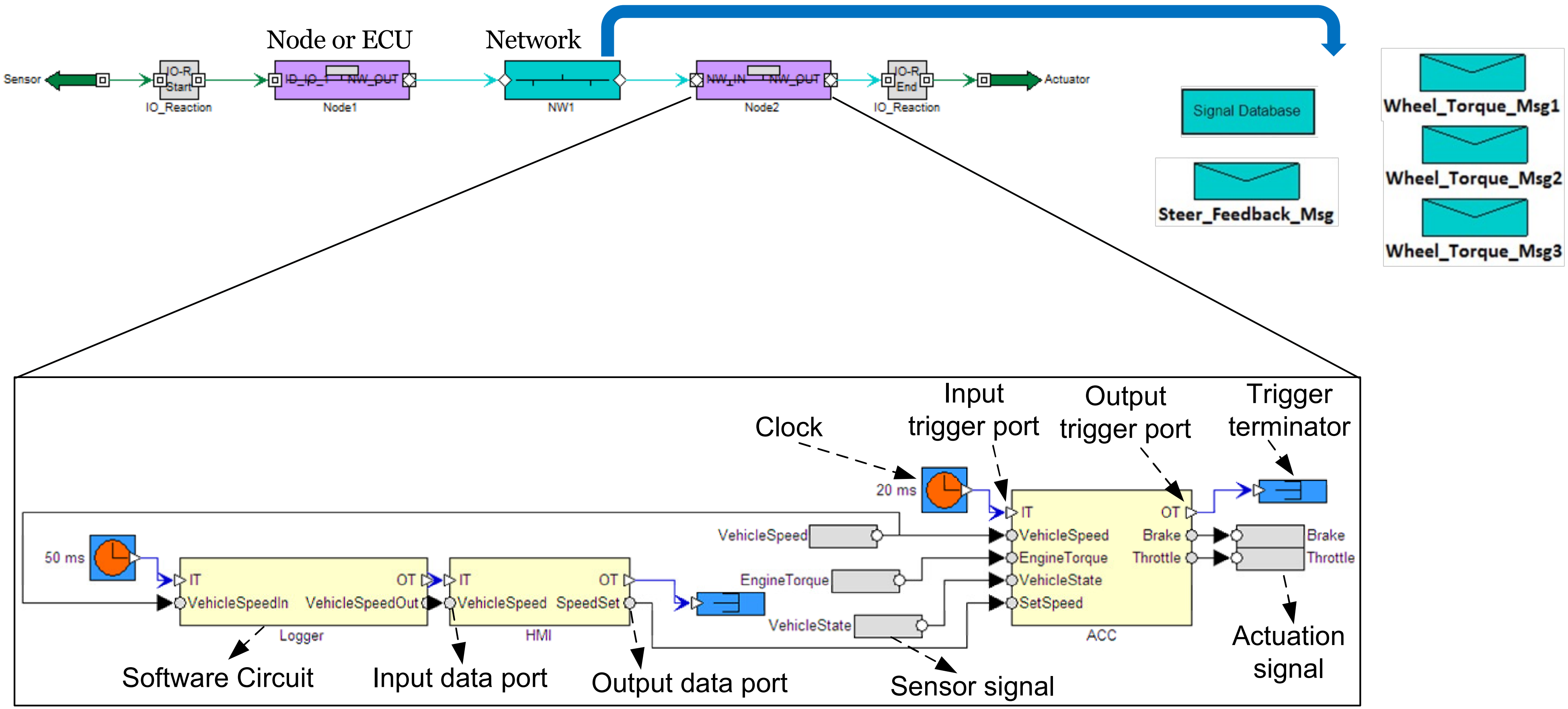
HOERBIGER

KNORR-BREMSE

BorgWarner

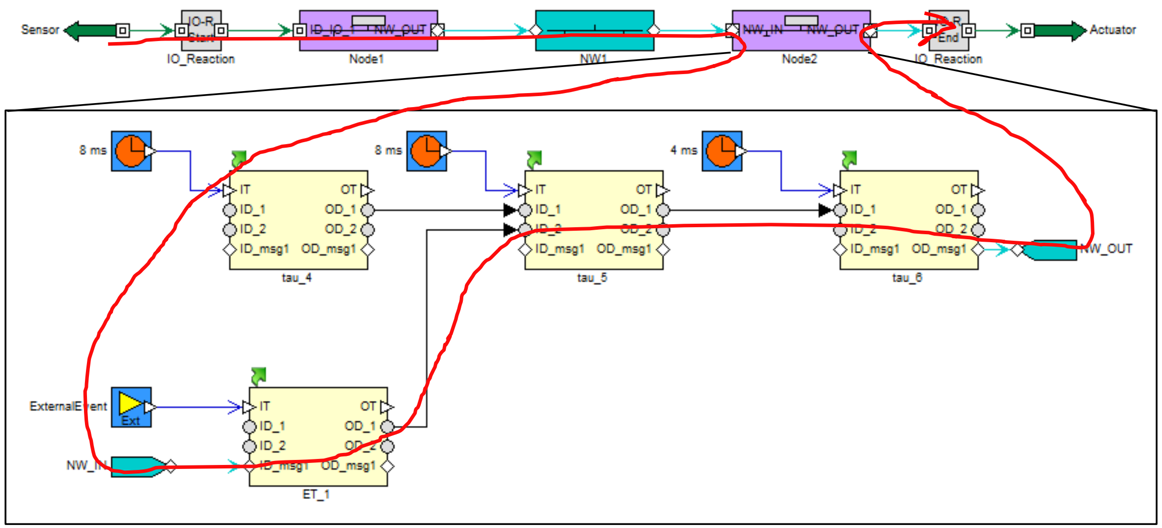
Modeling of Software Architecture with the Rubus Component Model (RCM)

Example of a two-node Distributed Embedded System



Note that control and data flows are separated in RCM

Analytical Analysis (End-to-end Timing Analysis)

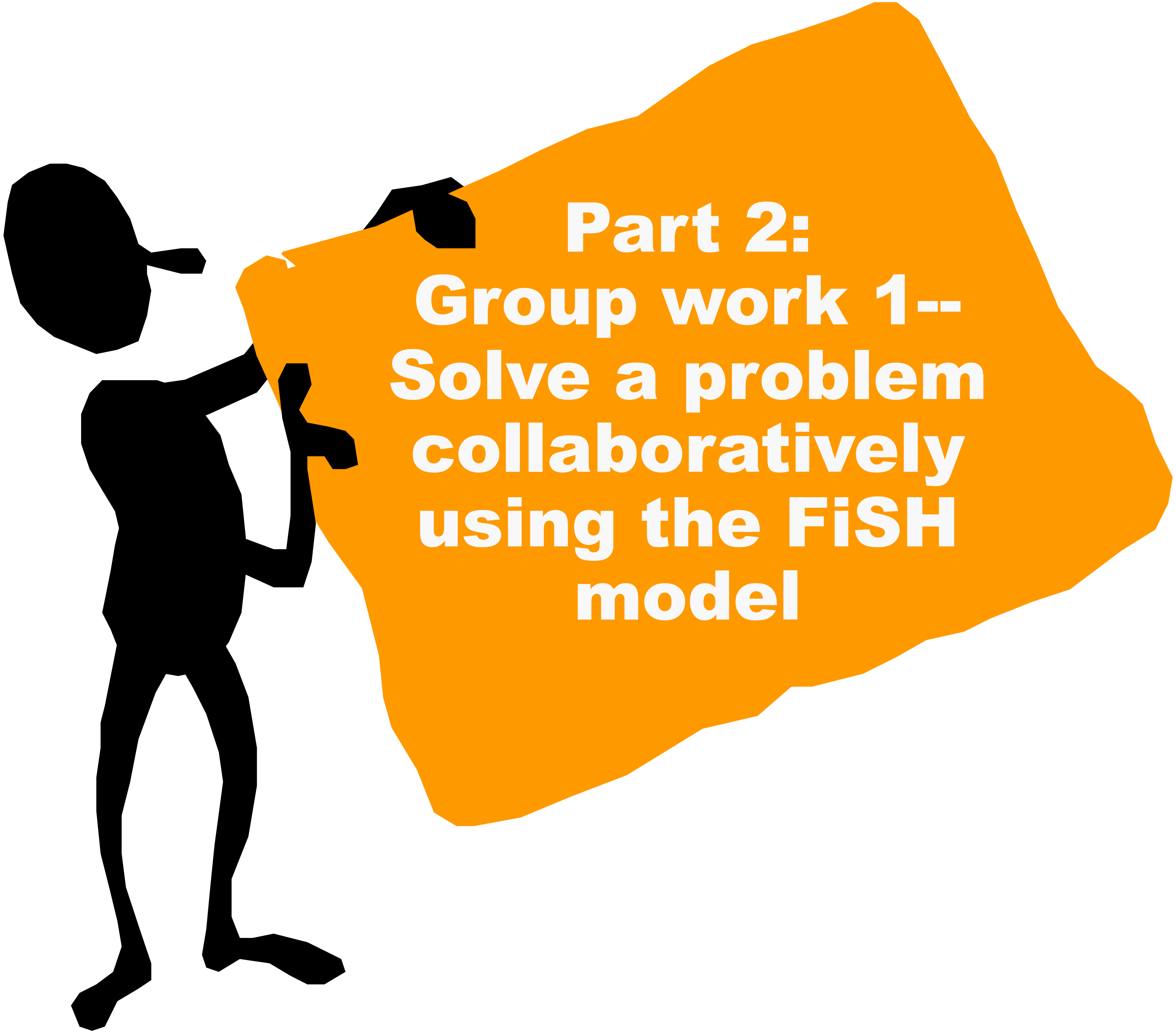


Timing analysis engines

- Response-time analysis
  - Tasks
  - Network messages (e.g., CAN)
  - Distributed chains of tasks and messages (end-to-end)
- End-to-end delay analysis

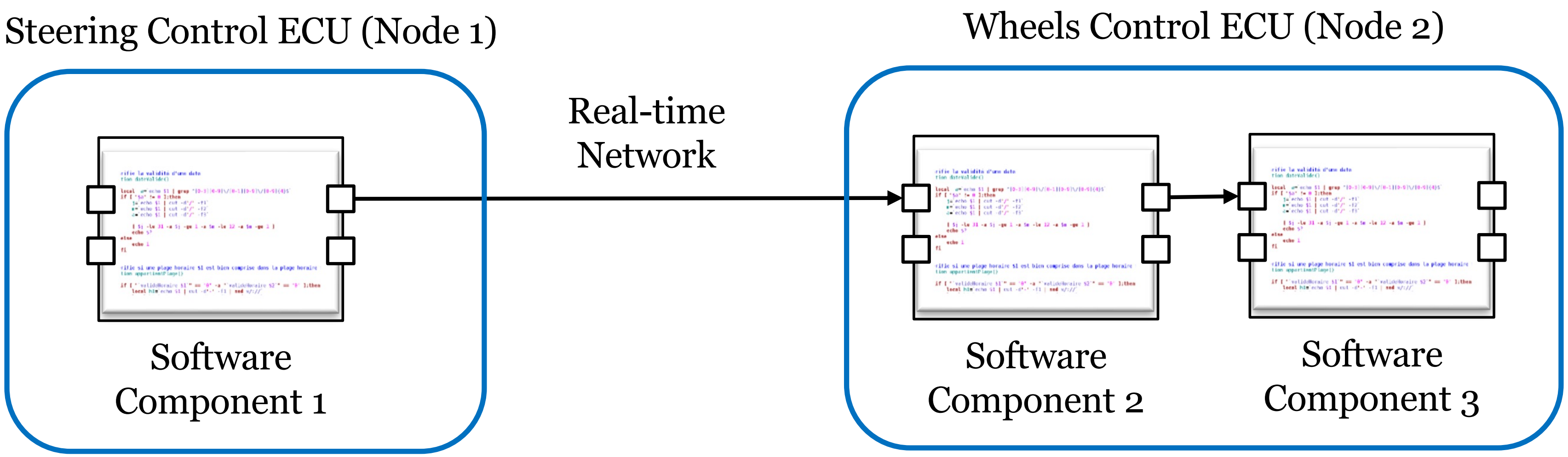
Focus of this Lecture

MÄLARDALEN UNIVERSITY SWEDEN



End-to-end Timing Analysis of a Software Architecture of a Distributed Embedded System

Example of the steer-by-wire distributed embedded system use case



What do you need to know to show that the above software architecture of a distributed real-time system is predictable with respect to its end-to-end timing?

- What more information do you need in this software architecture?
- How can you verify this?

Group Assignment: End-to-end Data-propagation Delay Analysis of a Distributed Real-time System

- Making groups
- Give a cool name to your group
- We will solve an assignment in groups during the lecture
- Once completed, you will write the results/summary of discussion/assumptions etc. on a white board
- We will compare the results and discuss the solution(s)

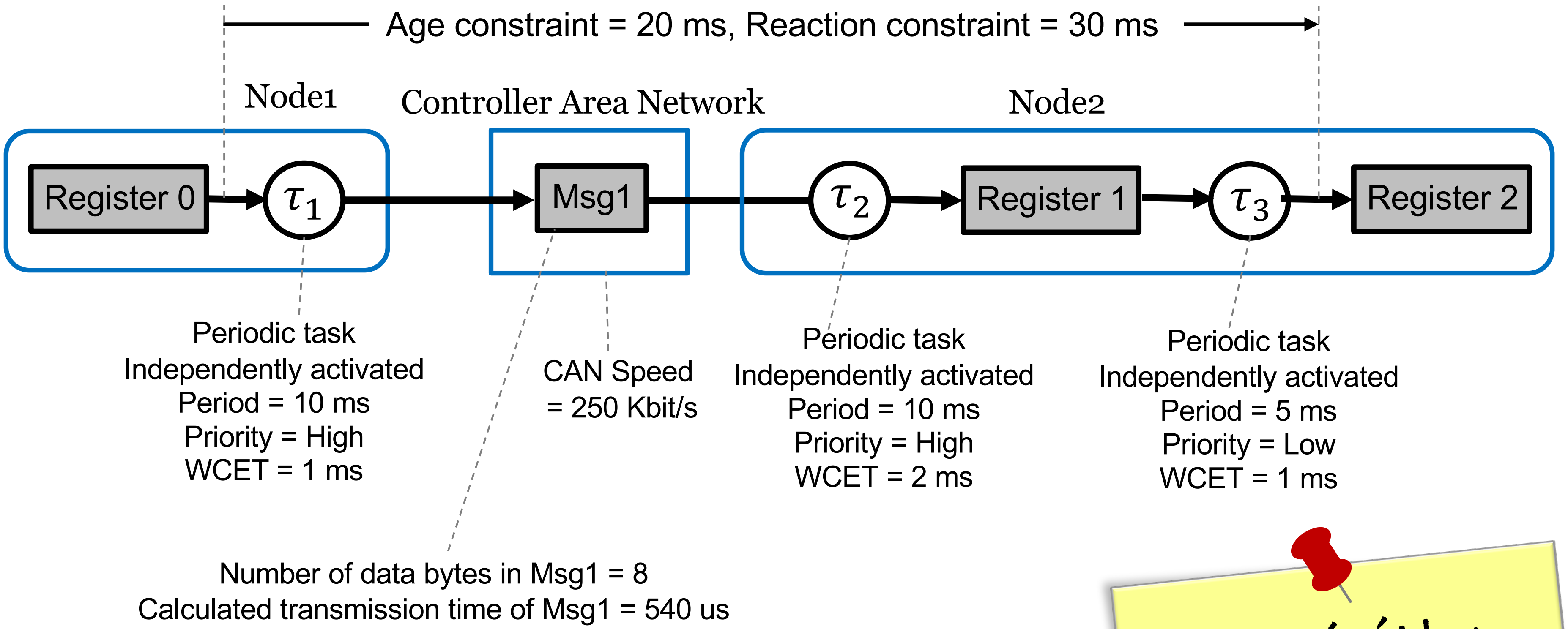
MÄLARDALEN UNIVERSITY
SWEDEN

Part 4:

Group Work 2 --

E2E Timing Analysis

Group Assignment: End-to-end Data-propagation Delay Analysis of a Distributed Real-time System



Calculate the Age and Reaction delays

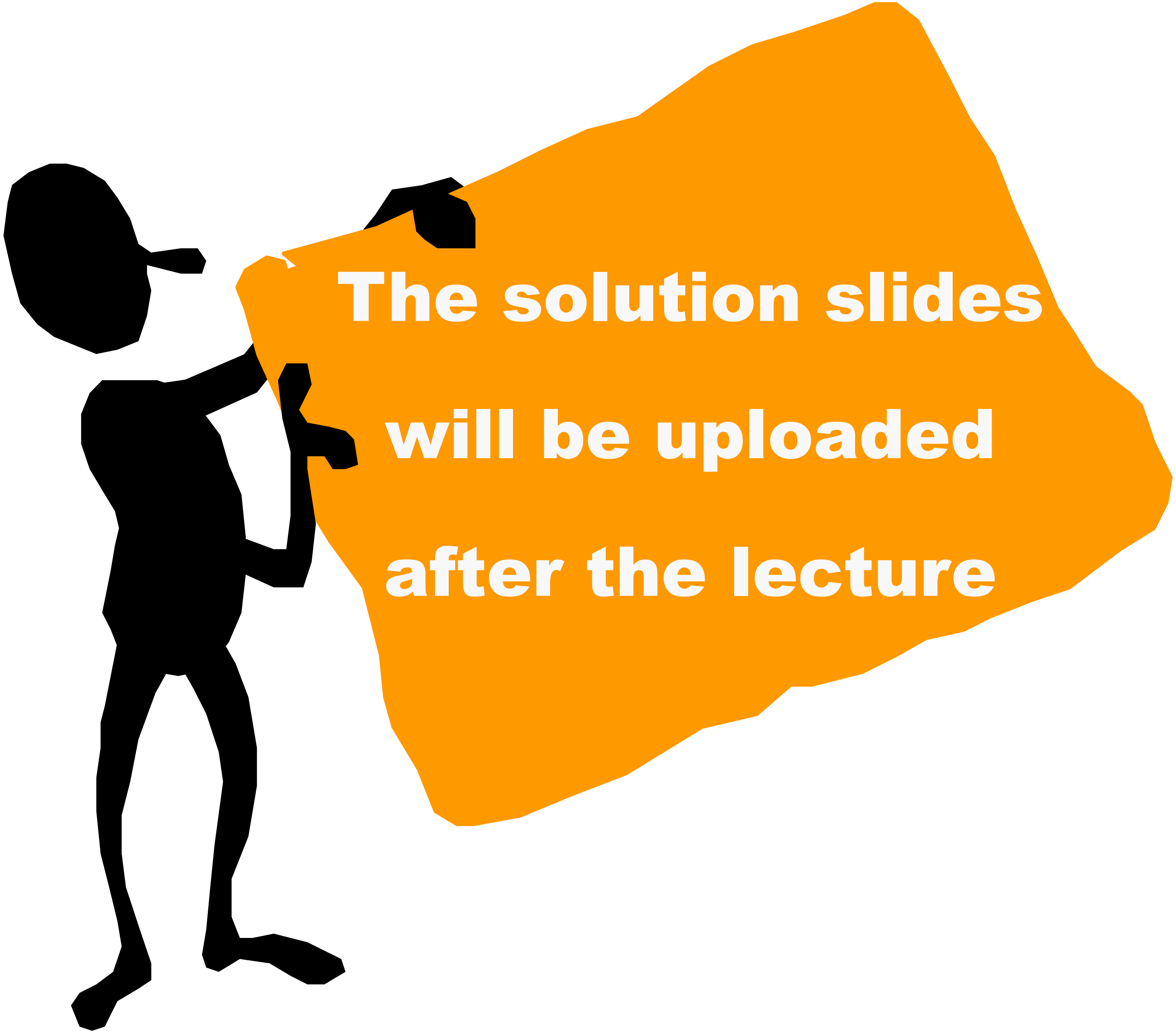
1. Graphically
2. Mathematically (if possible, within the given time)

Explicitly mention your assumptions

Group Assignment: End-to-end Data-propagation Delay Analysis of a Distributed Real-time System

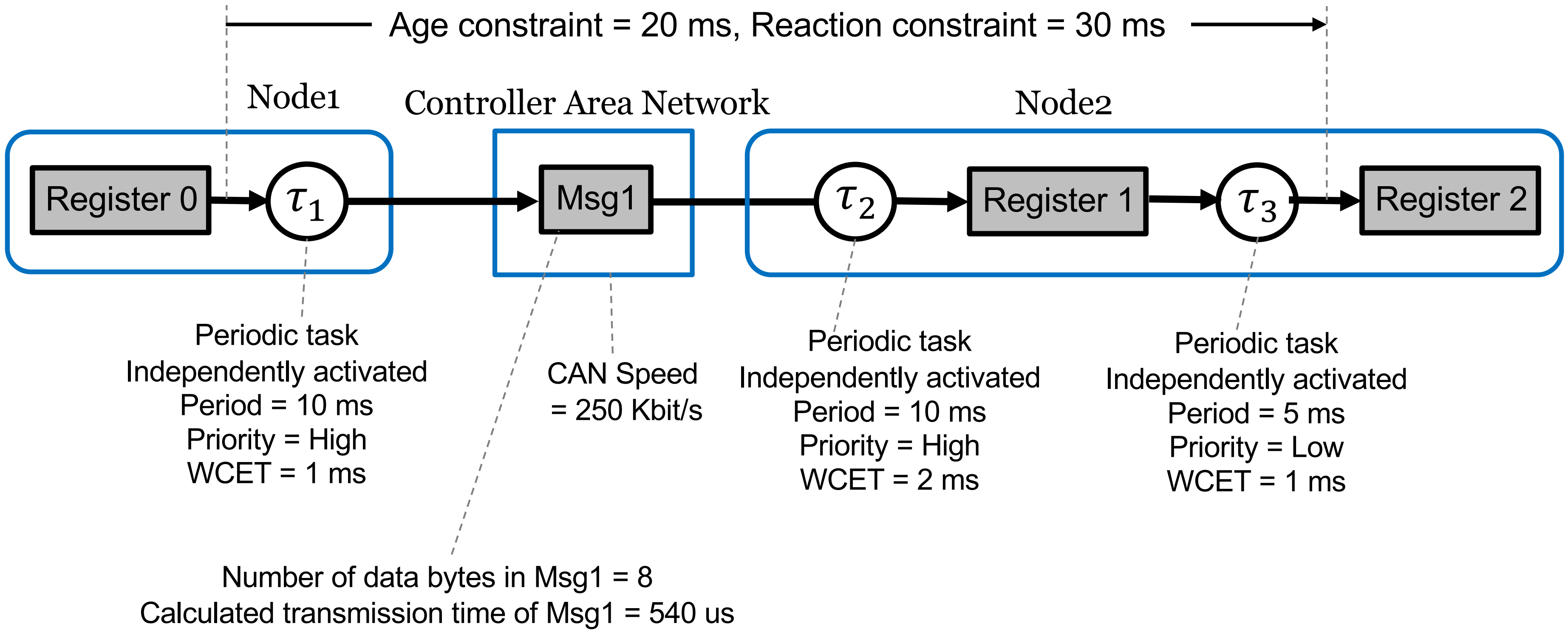
DVA482-Discussion of Class Assignment

Group Name	Age Constraint (ms)	Calculated Age Delay (ms)	Reaction Constraint (ms)	Calculated Reaction Delay (ms)	Assumptions
	20		30		


The solution slides
will be uploaded
after the lecture

Part 3: Discussion

Group Assignment Solution: End-to-end Data-propagation Delay Analysis of a Distributed Real-time System



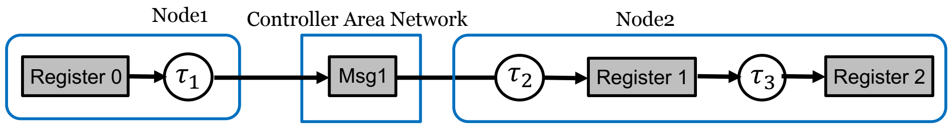
Extra Reading:

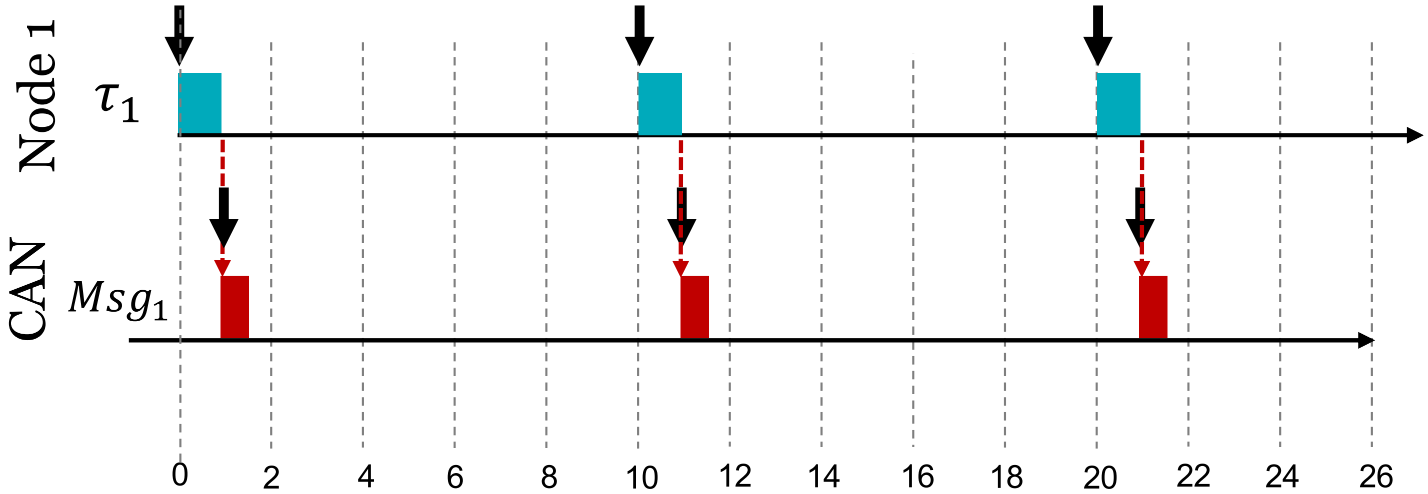
- End-to-end timing analysis in Rubus-ICE
- Effect of Synchronised and Unsynchronised nodes on the Age delay
- Job-level dependencies and end-to-end delays
- End-to-end timing analysis of cause-effect chains in automotive embedded systems

Assumptions

- The nodes are not synchronized
- The receiving task, τ_2, in Node2 implements polling policy for message reception, i.e., τ_2 periodically checks if Msg_1 is received or not.

Age Delay





CAN: Controller Area Network

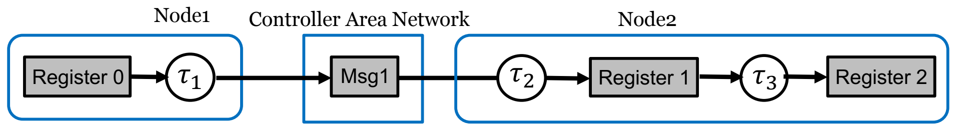
τ : Task

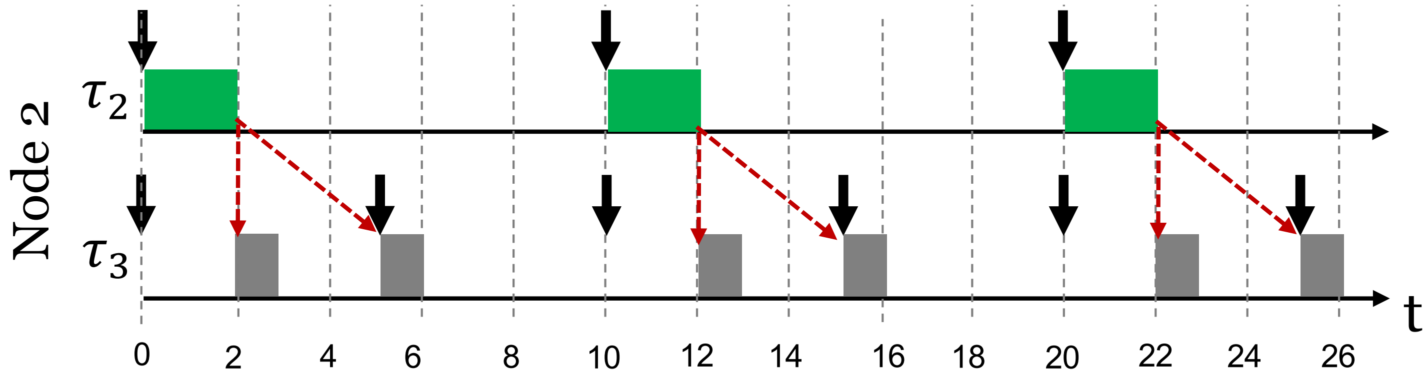
DP: Data path

Task arrival

Msg: Message

Age Delay



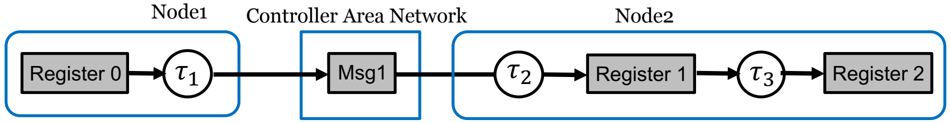


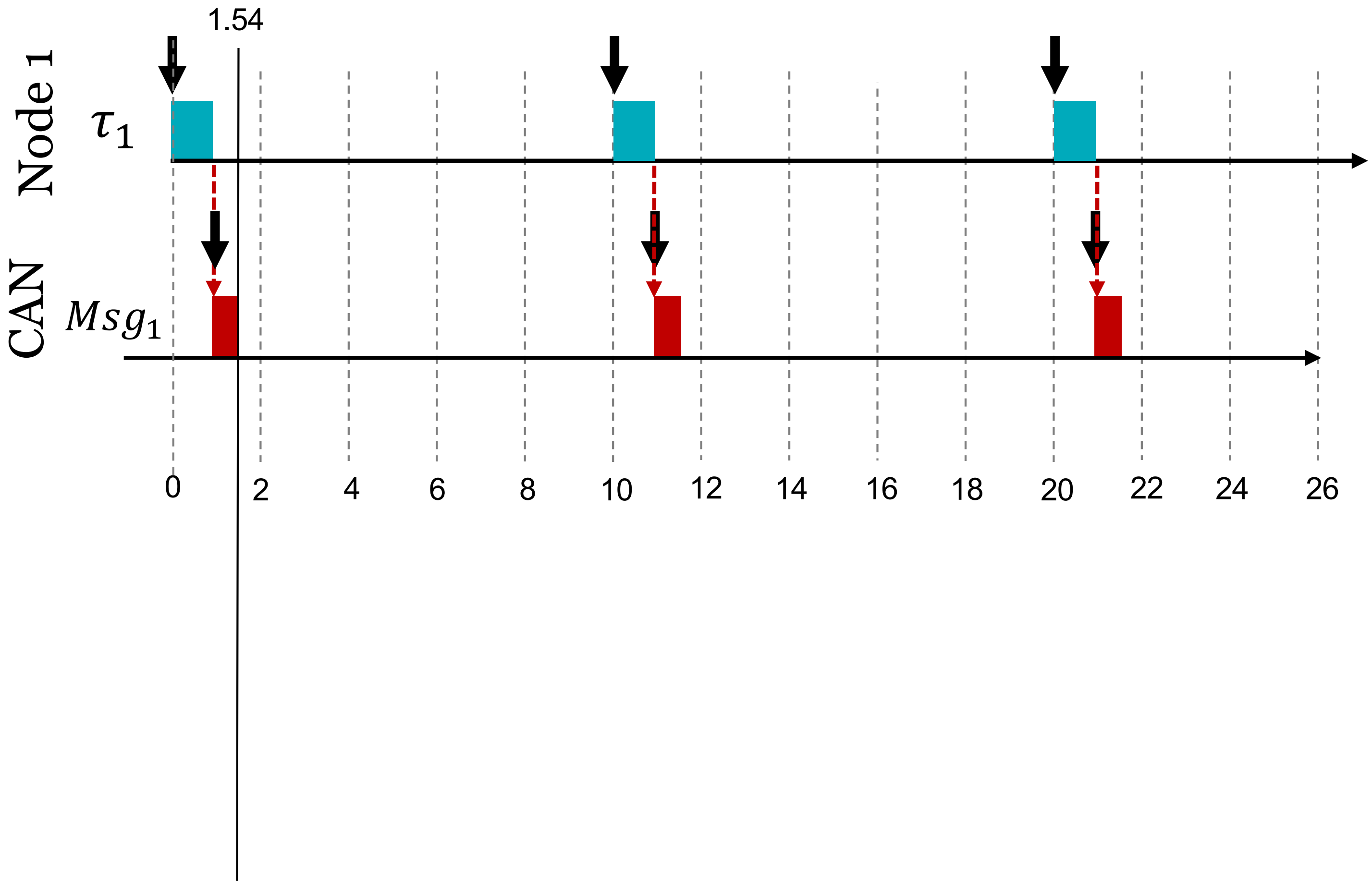
CAN: Controller Area Network

τ : Task DP: Data path

Task arrival Msg: Message

Age Delay





CAN: Controller Area Network

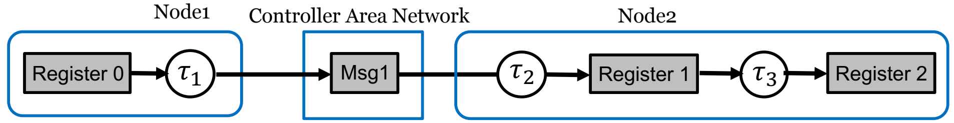
τ : Task

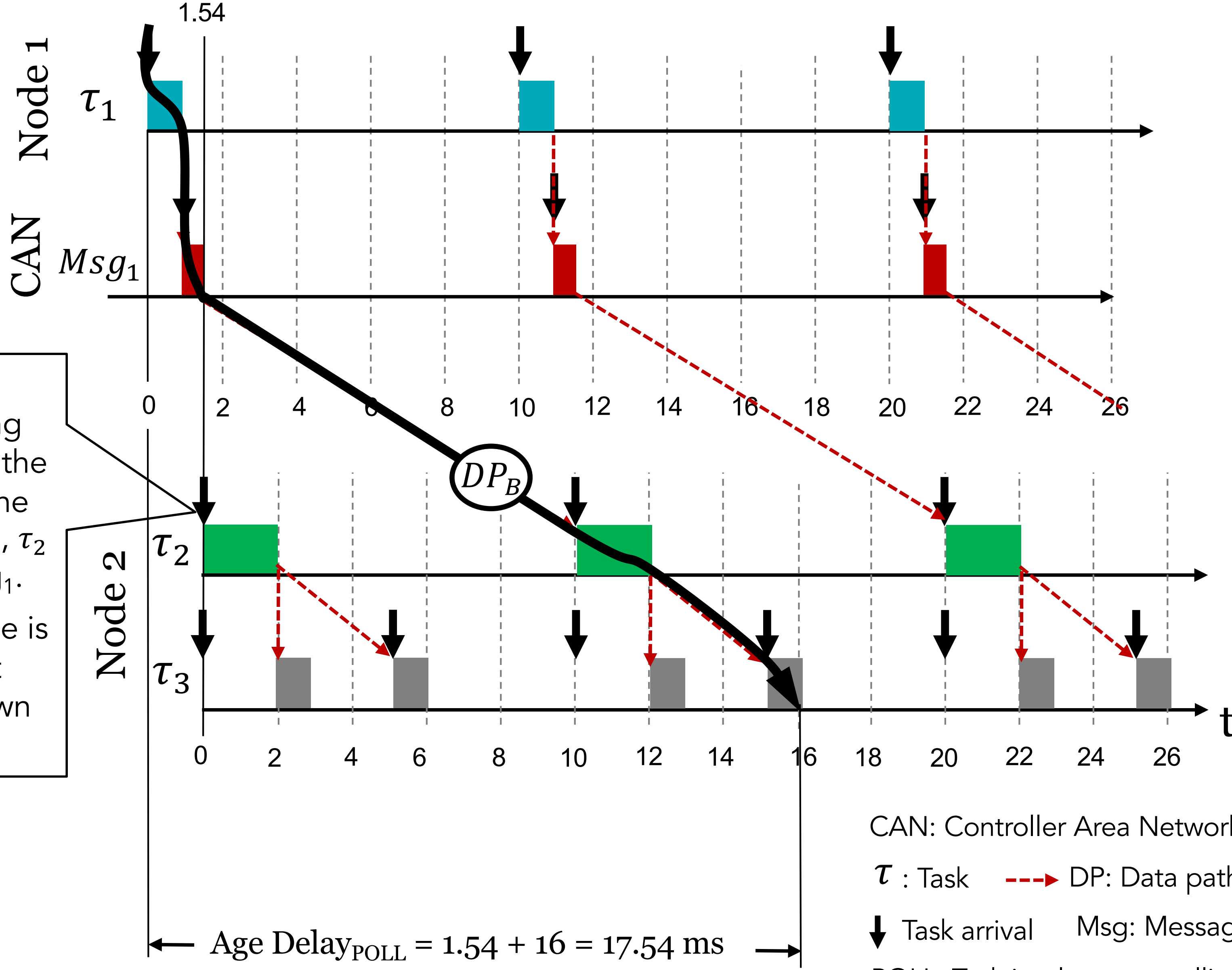
DP: Data path

Task arrival

Msg: Message

Age Delay





If τ_2 in Node 2 implements polling policy for receiving the message then in the worst-case scenario, τ_2 "just misses" Msg_1.

The missed message is read by the next instance of τ_2 shown by DP_B

$\text{Age Delay}_{\text{POLL}} = 1.54 + 16 = 17.54$ ms

CAN: Controller Area Network

τ : Task

DP: Data path

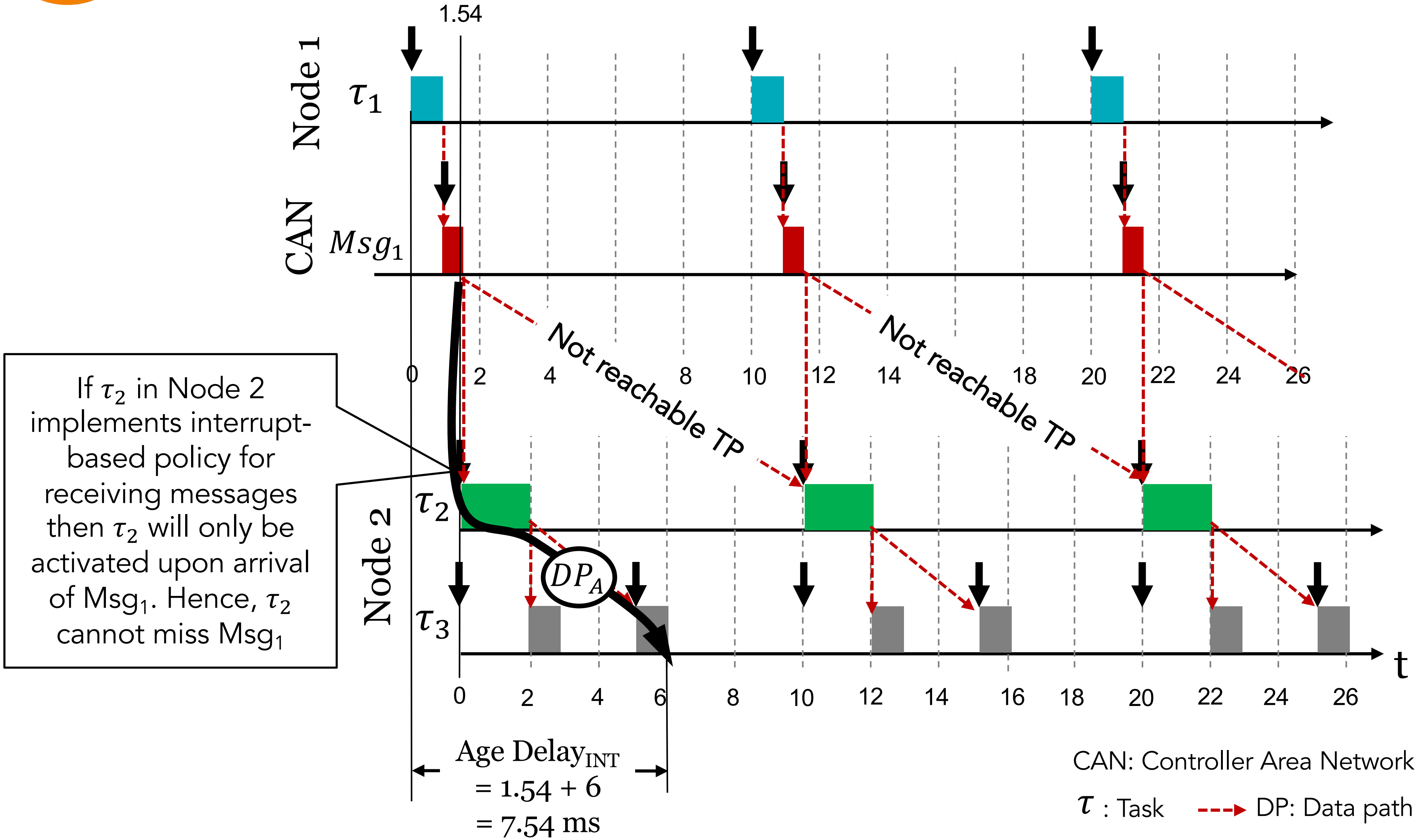
Task arrival

Msg: Message

POLL: Task implements polling policy to receive a message

Age Delay

What if τ_2 in Node 2 is not periodic? Instead, it is activated by an interrupt (minimum inter-arrival time = 10 ms). τ_3 remains the same (i.e., periodic).



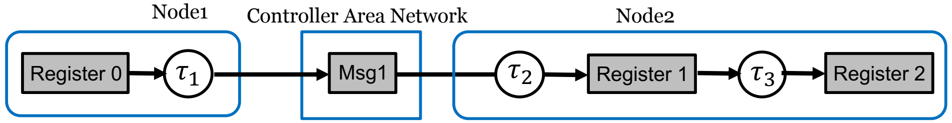
CAN: Controller Area Network

τ : Task DP: Data path

Task arrival Msg: Message

INT: Task implements interrupt-based policy for receiving message

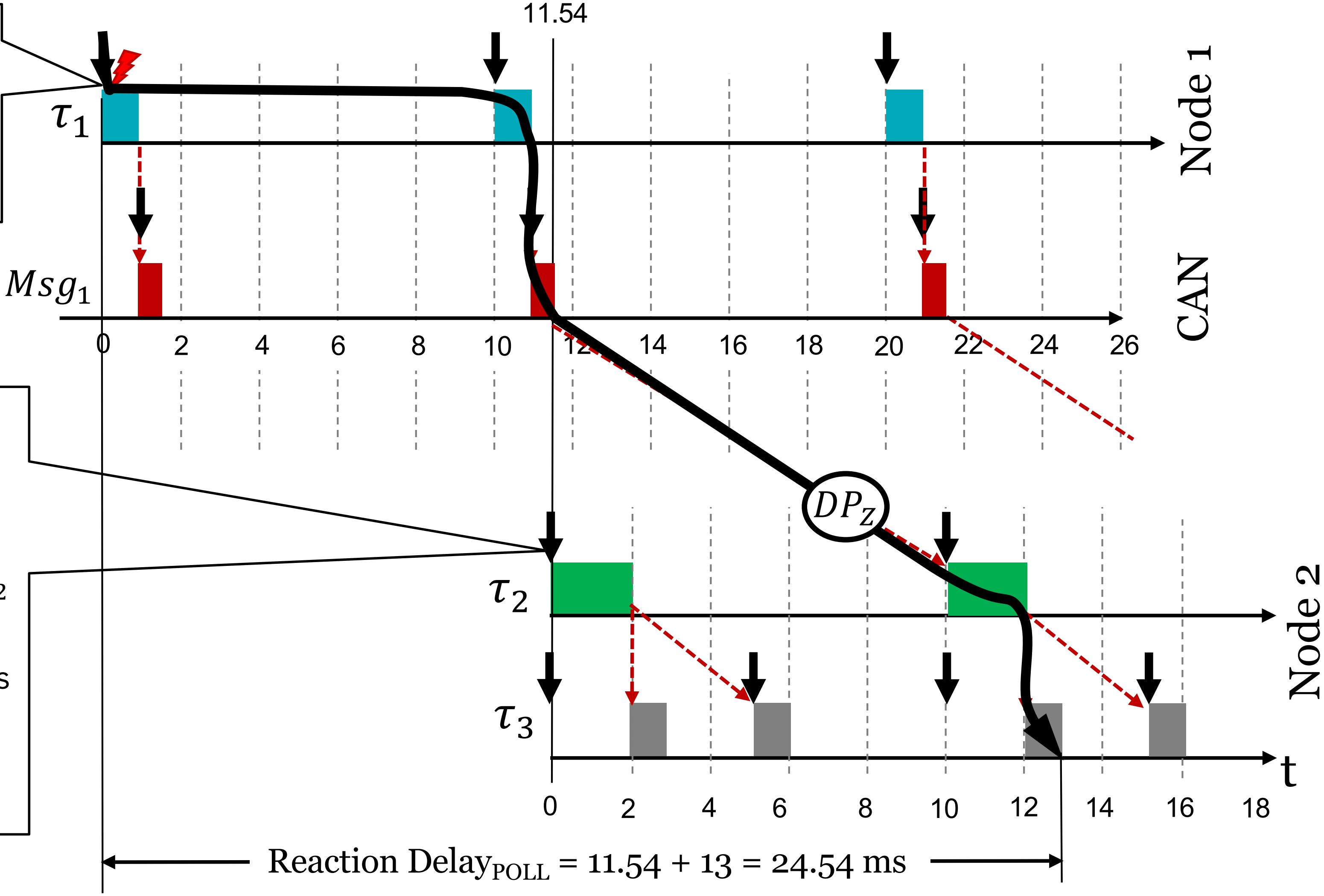
Reaction Delay



"Task τ_1 just missed the new data"
The missed data is read by the next instance of τ_1

If τ_2 in Node 2 implements polling policy for receiving messages then in the worst-case scenario, τ_2 "just misses" Msg_1.

The missed message is read by the next instance of τ_2 shown by DP_B

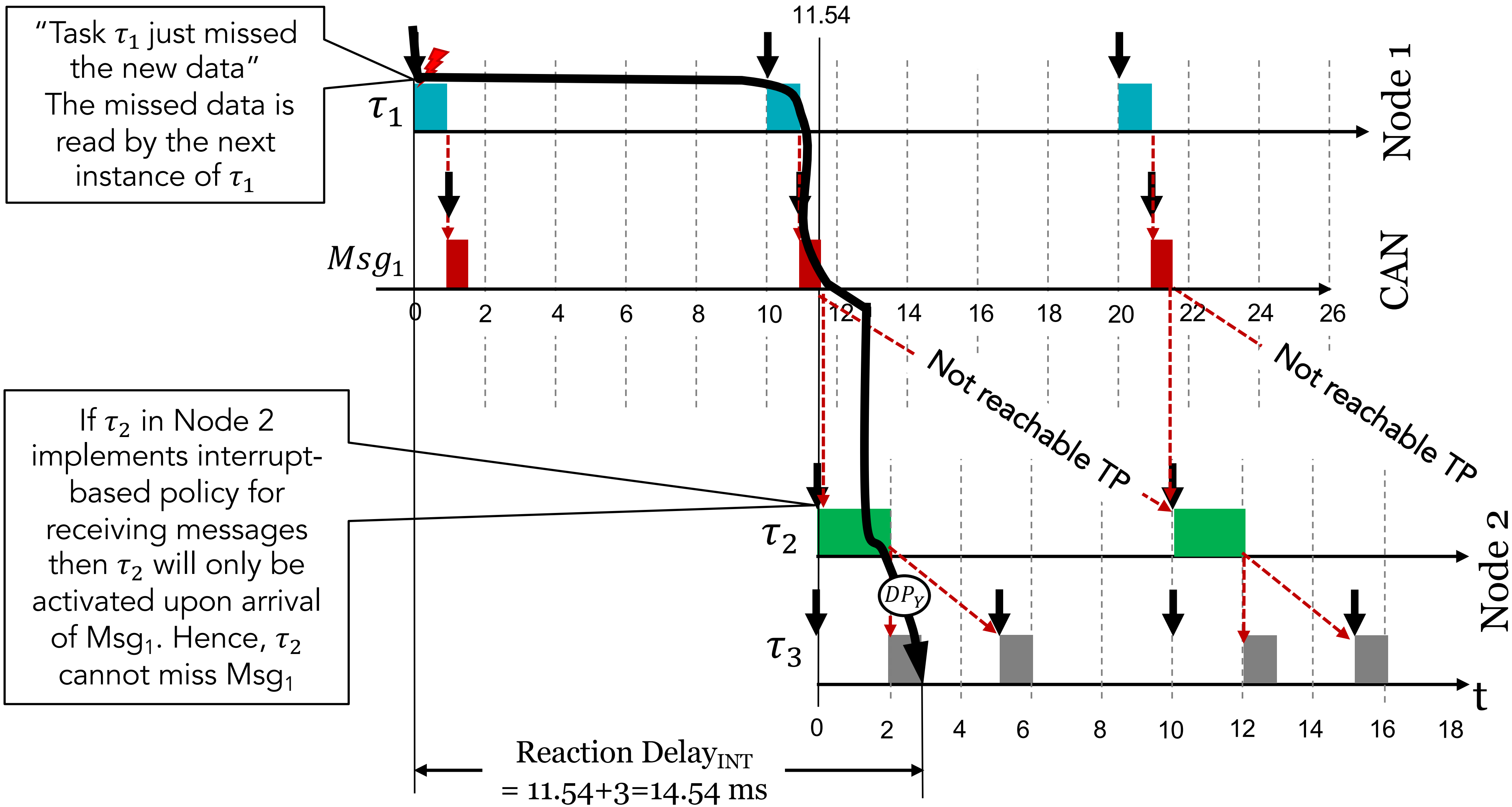


τ : Task Msg: Message CAN: Controller Area Network Task arrival DP: Data path

POLL: Task implements polling policy for receiving message

Reaction Delay

What if τ_2 in Node 2 is not periodic? Instead, it is activated by an interrupt (minimum inter-arrival time = 10 ms). τ_3 remains the same (i.e., periodic).

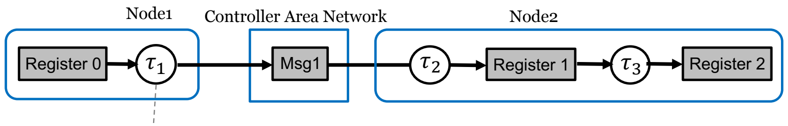


τ : Task Msg: Message CAN: Controller Area Network Task arrival DP: Data path

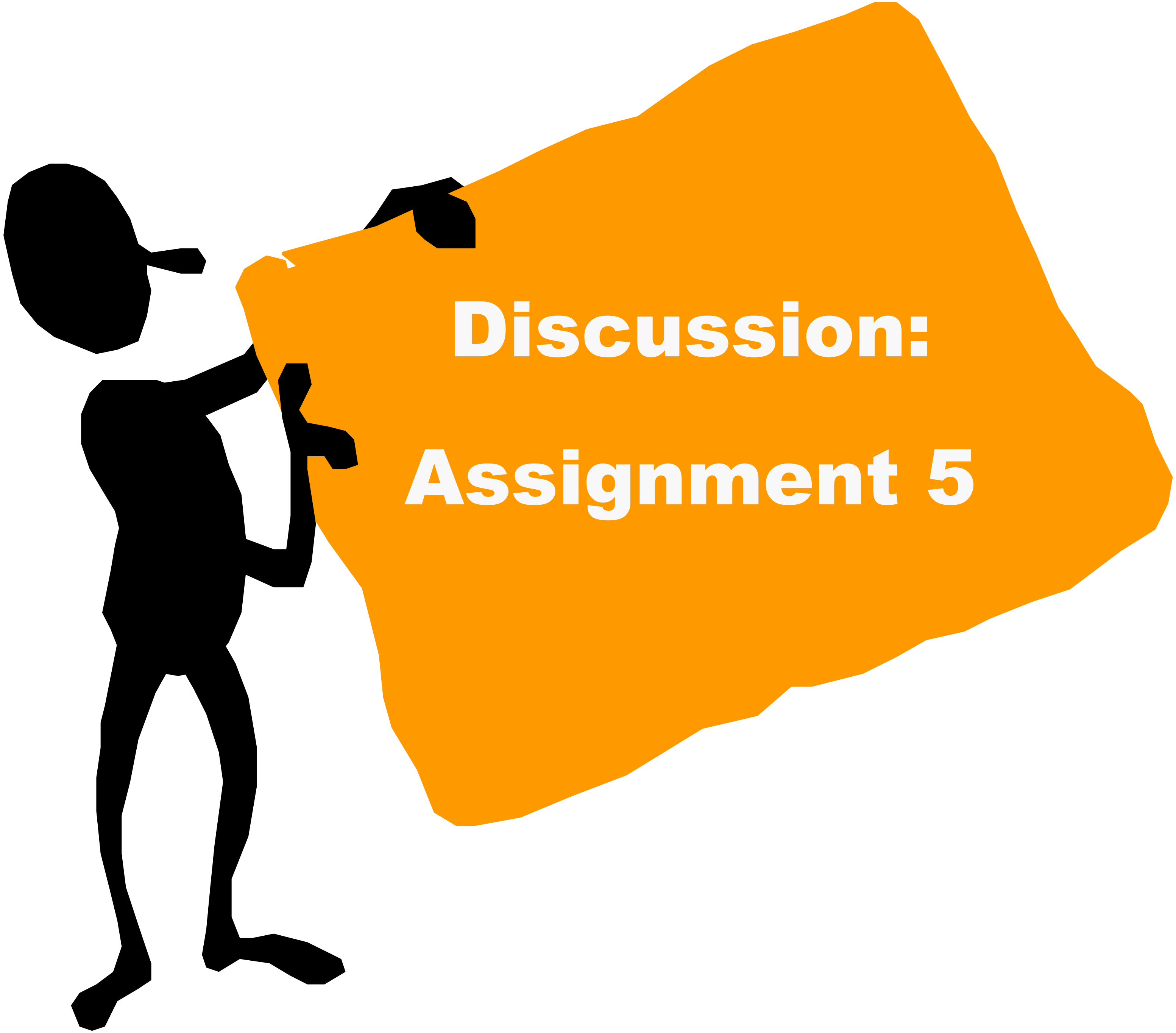
INT: Task implements interrupt-based policy for receiving message

Calculations for the Age and Reaction Delays in Distributed Real-time Systems

Calculations for the age and reaction delays in a distributed real-time system can be performed in a compositional way as follows:



- Calculate the delays in nodes separately
- The delay in the network is equal to the worst-case response time of the message in the chain
- Since the nodes are not synchronized in the case of CAN network, consider the effect of "receiving task just missing the read access of the message" for both Age and Reaction delays. What about the case when nodes are synchronized?
    - Don't forget to consider the effect of "just missing the new data" at the start of the chain in the case of the Reaction delay.
- Add all nodes and network delays to get the end-to-end delays
    - Note that the response time of the message should not include its release jitter as this has been included as part of the delay in the sending node (sending task).


Discussion:
Assignment 5

End-to-end timing analysis of a distributed embedded system

Assignment 5: End-to-end timing analysis of a distributed embedded system

The requirements are as follows.

- The system should consist of two nodes that are connected by one CAN network
- Select the network speed of your choice conforming to the standard CAN protocol
- The network must contain at least 3 messages
- There must be at least one multi-rate distributed chain in the system
  - The chain must have at least two tasks in each node
  - The chain must be interfered by at least one task in each node. The interferring task(s) should not be part of the chain
  - The message in the chain must be the lowest priority message in the network
  - Use one precedence constraint between any two tasks in the system
  - Specify one age and one reaction constraint from the start to the end of the chain
- Assume message and task parameters yourself

Assignment 5: End-to-end timing analysis of a distributed embedded system

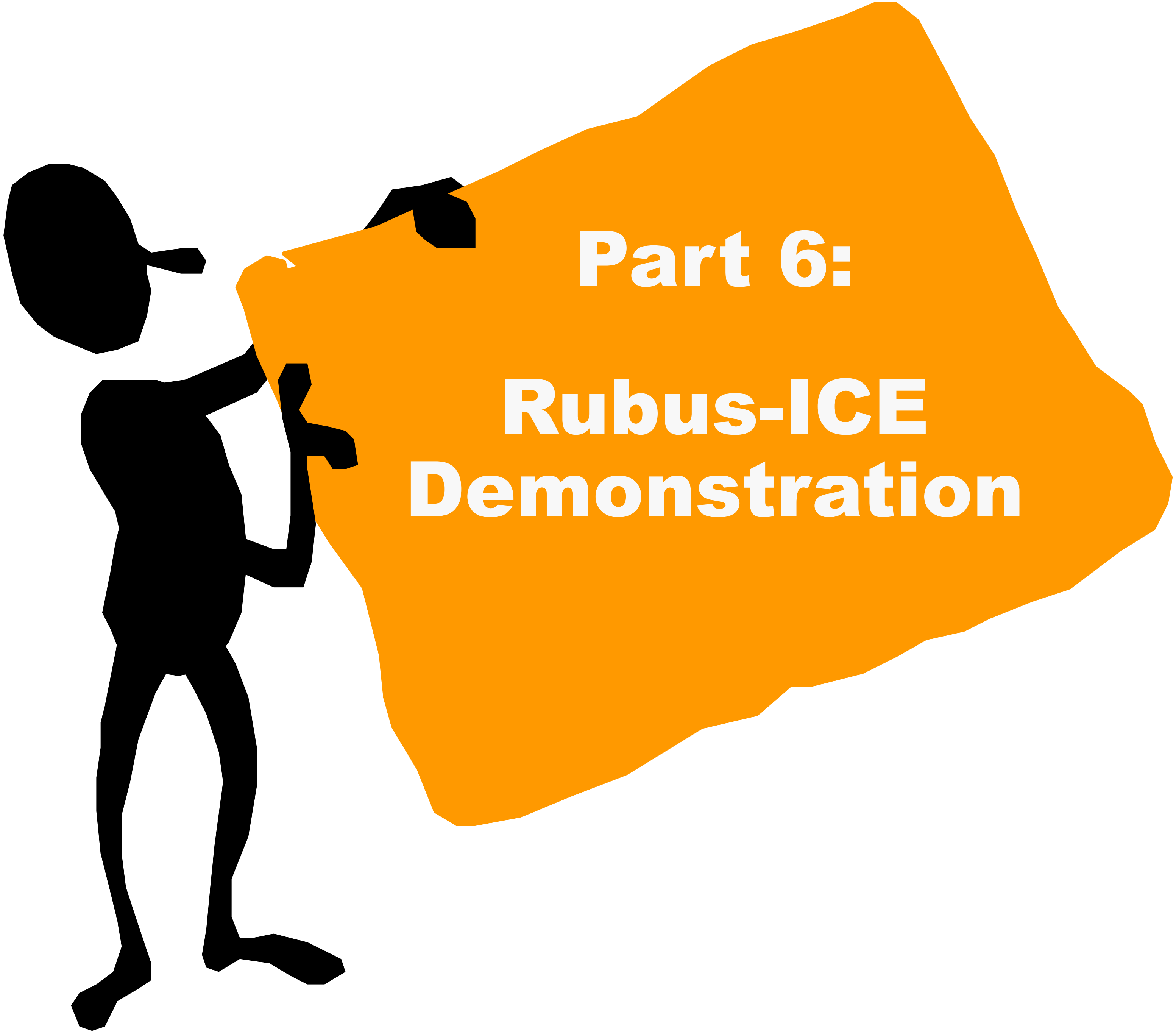
- Use the assumptions discussed in the corresponding lecture as follows
  1. No offsets
  2. Within a multi-rate chain, the priority of any task is not higher than the priority of its predecessor task within the same node
  3. Receving tasks use polling policy to receive messages from the network
  4. Receiving tasks "just miss" the read access of the messages

(a) Identify the age and reaction delays using a time line graph

(b) Calucate the age and reaction delays mathematically

(c) Are the specified age and reaction constraints satisfied?

(d) What could be the consequences or complexities of not using the above assumptions? Explain for each assumption (1-4) separately.


Part 6:
Rubus-ICE
Demonstration