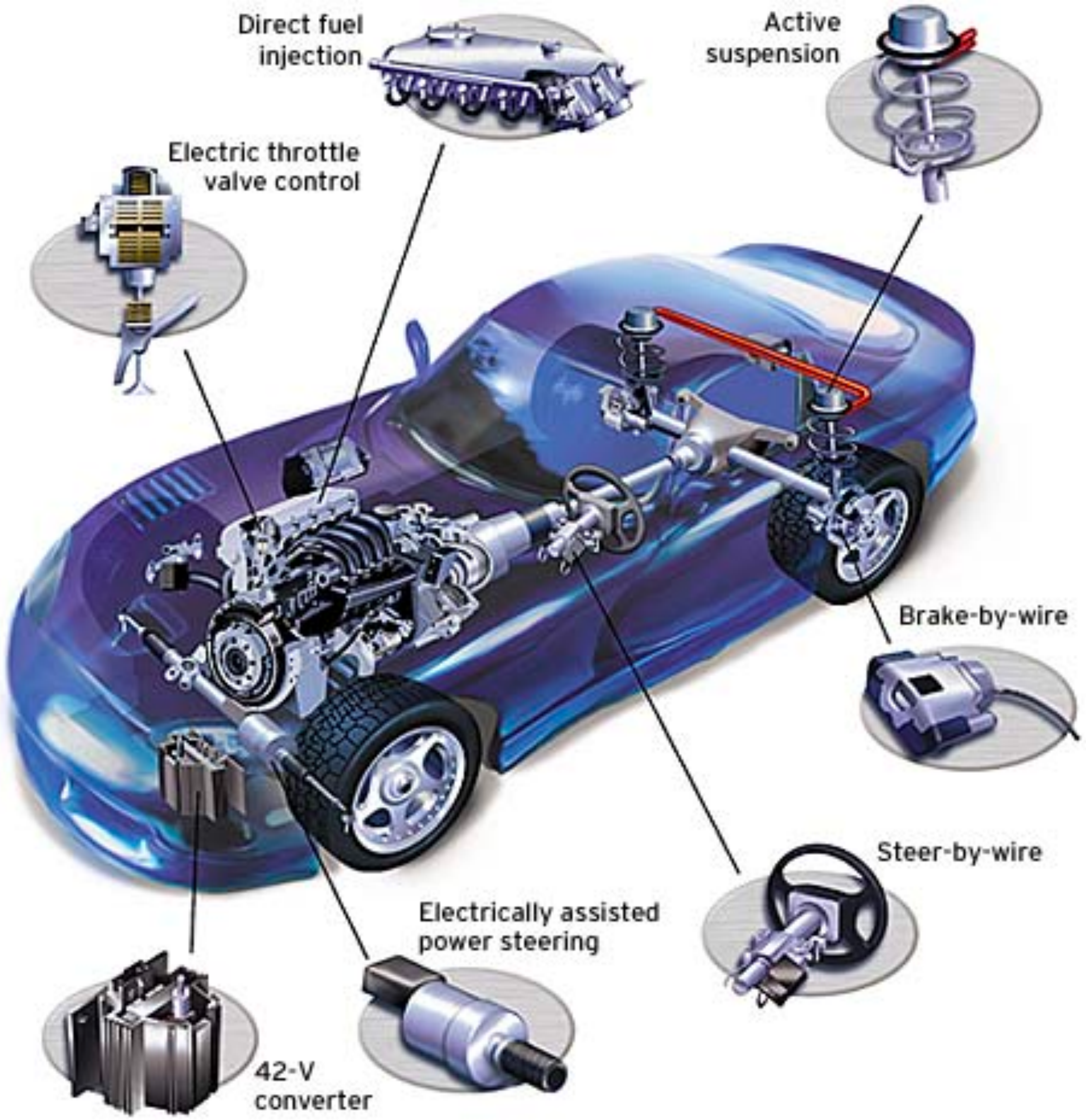
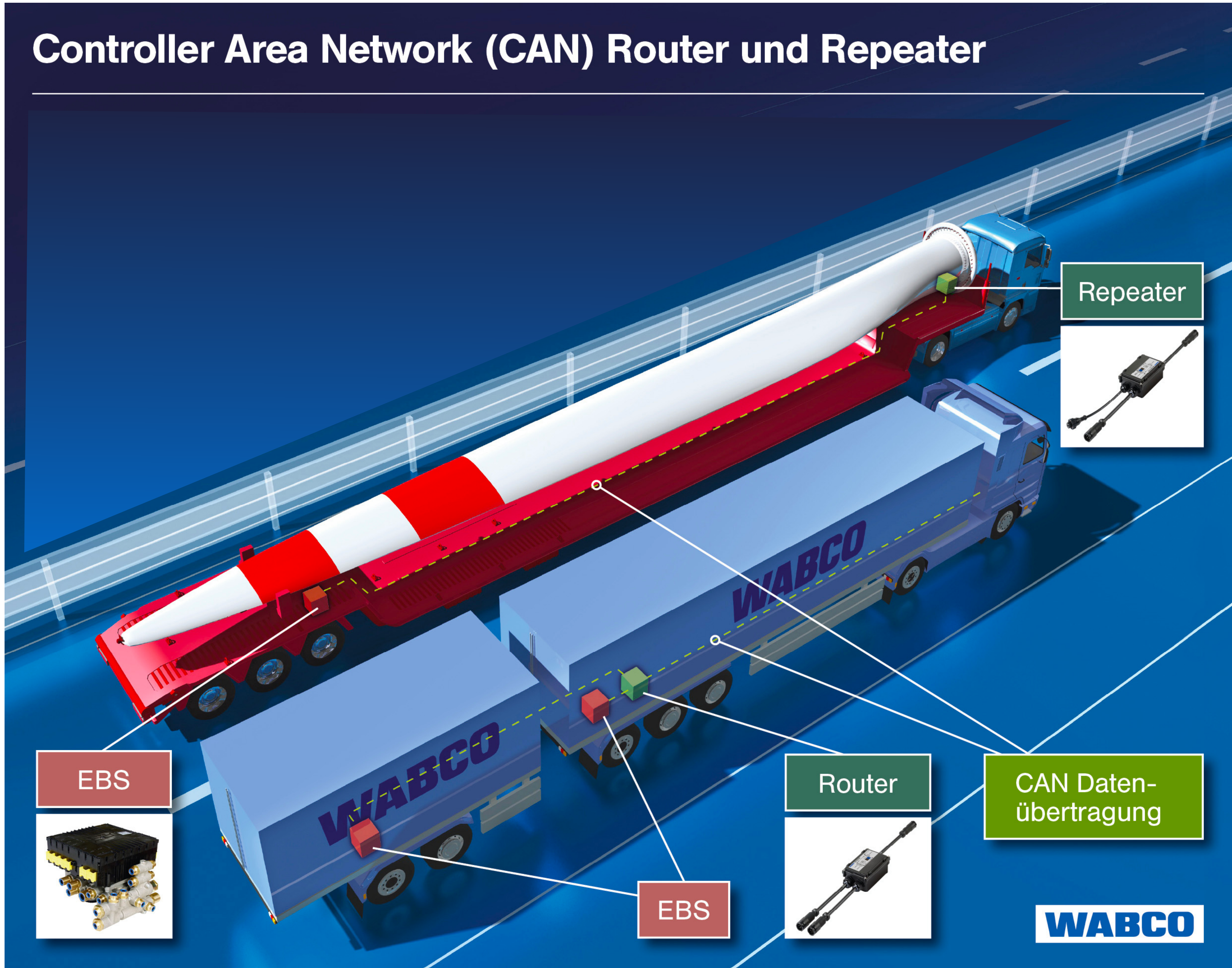
Using **Schedule-Abstraction Graphs** for the Analysis of **CAN Message Response Times**

Mitra Nasri, **Arpan Gujarati**, and Björn B. Brandenburg
Max Planck Institute for Software Systems (MPI-SWS), Germany

Controller Area Network (CAN) is widely used in real-time embedded systems





Message transmission over CAN is predictable

CAN's bitwise arbitration method → Messages are transmitted in order of their priorities

CAN messages timing analysis ≈ **Uniprocessor fixed-priority non-preemptive scheduling**

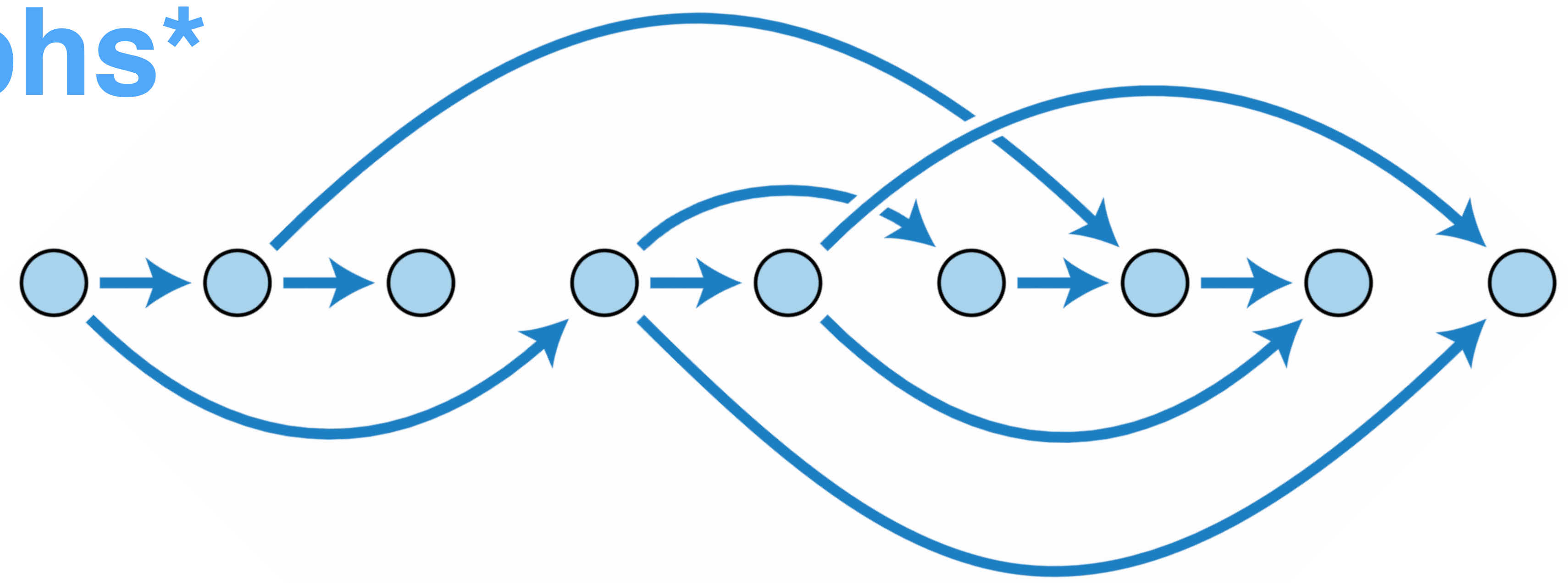
TV tower **Radio** → **Electromagnetic Interference (EMI)** → **Error detection & retransmission**

This paper

Fine-grained timing analysis of CAN messages in presence of **EMI-induced retransmissions**

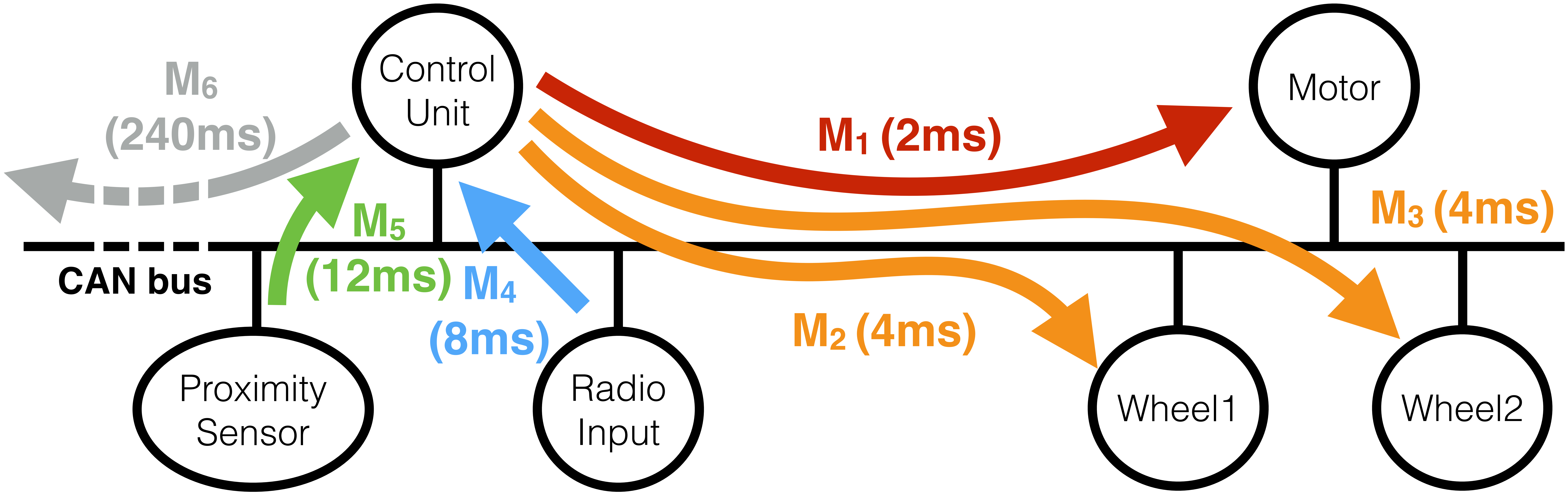
How?

Schedule-abstraction graphs*



* *Mitra Nasri and Bjorn B. Brandenburg. "An exact and sustainable analysis of non-preemptive scheduling." RTSS, 2017.*

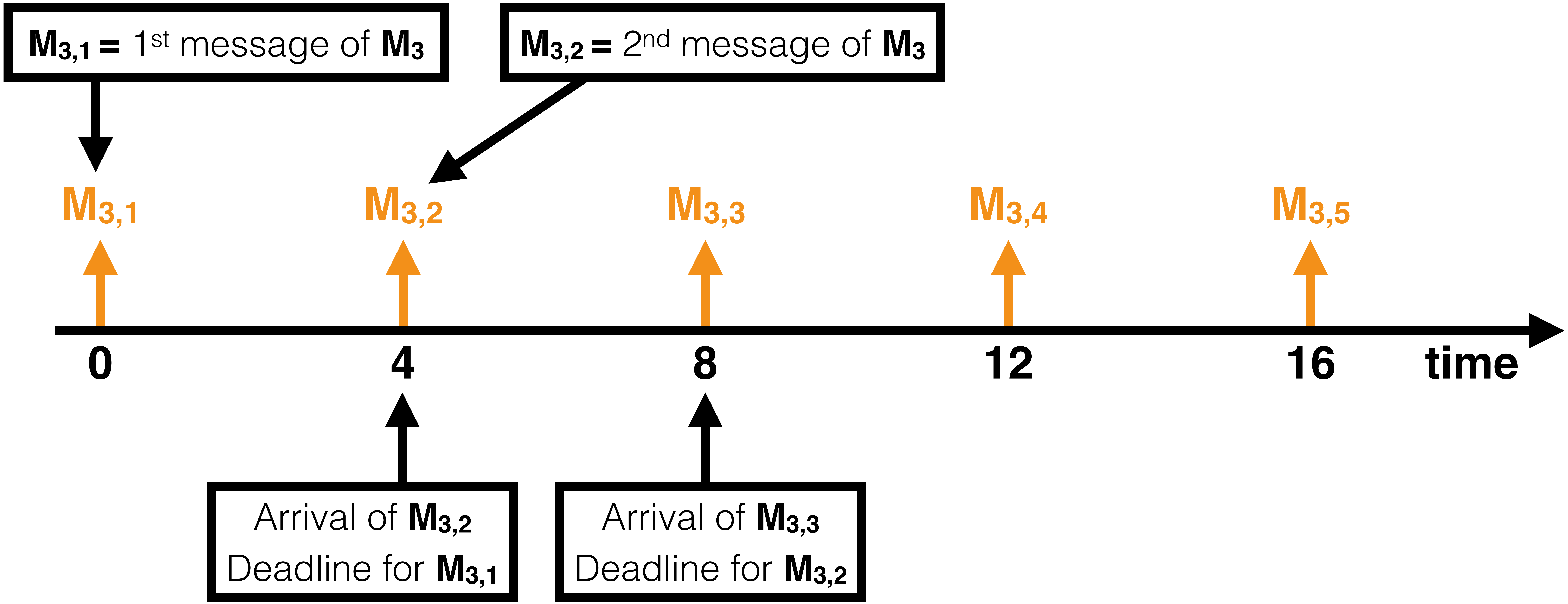
Example: CAN-based mobile robot



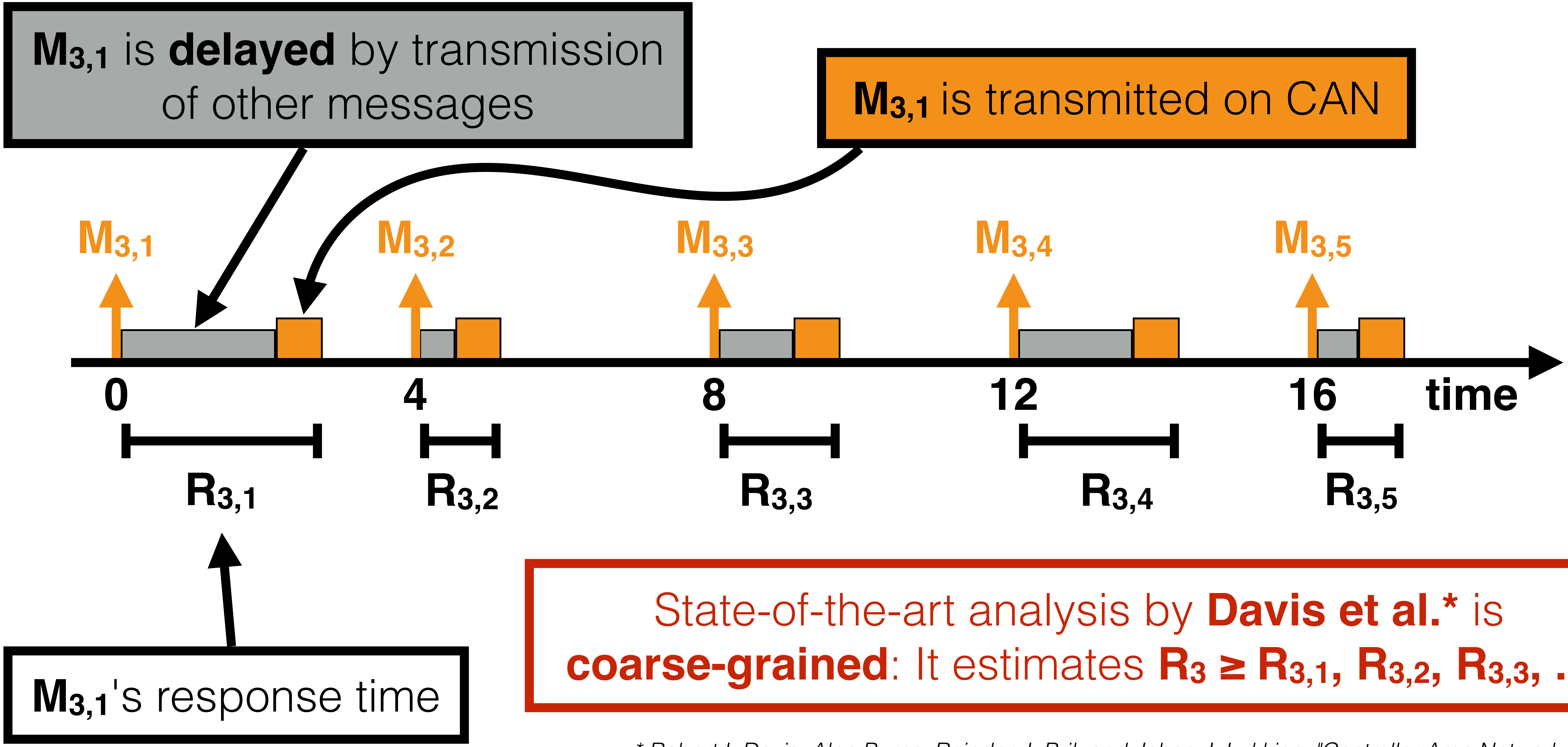
Periodic message streams

Priority order: $\mathbf{M_1 > M_2 > M_3 > M_4 > M_5 > M_6}$

Consider the periodic message stream M_3(4ms)



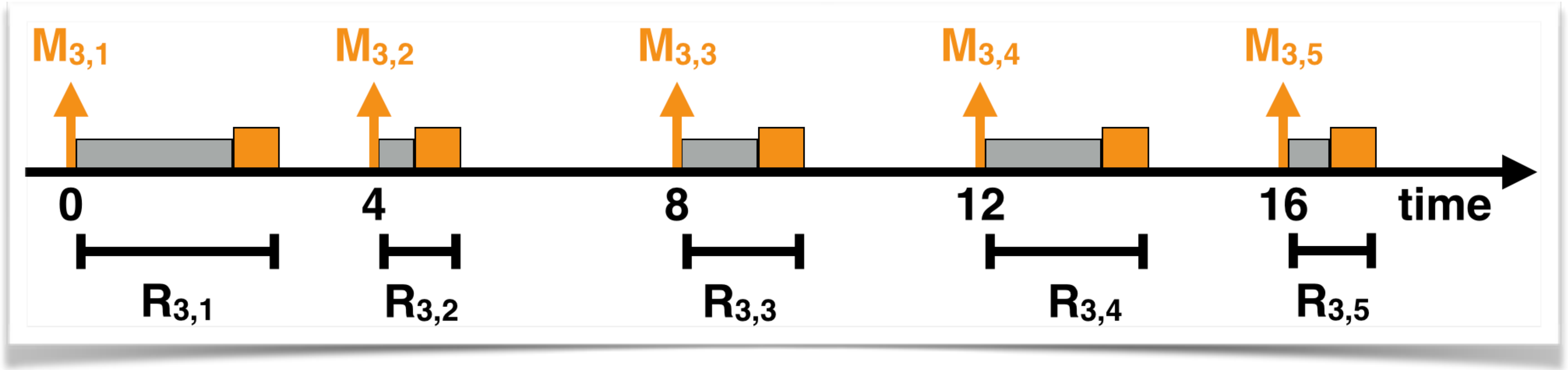
Consider the periodic message stream M_3(4ms)



State-of-the-art analysis by **Davis et al.*** is **coarse-grained**: It estimates $\mathbf{R_3 \geq R_{3,1}, R_{3,2}, R_{3,3}, \ldots}$

* *Robert I. Davis, Alan Burns, Reinder J. Bril, and Johan J. Lukkien. "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised." Real-Time Systems 35, no. 3 (2007): 239-272.*

Need for a fine-grained analysis with retransmissions



Individual bounds on $\mathbf{R_{3,1}}$, $\mathbf{R_{3,2}}$, $\mathbf{R_{3,3}}$, **...**

Analysis of weakly-hard real-time systems

➡ (1, 3) system needs only one out of three consecutive deadlines to be satisfied

Offset assignment for improved schedulability

Case study at the end

Reliability analysis

➡ is a function of individual message failure probabilities

Sampling jitter analysis

➡ requires both best-case and worst-case response-time analysis

Response-Time Analysis

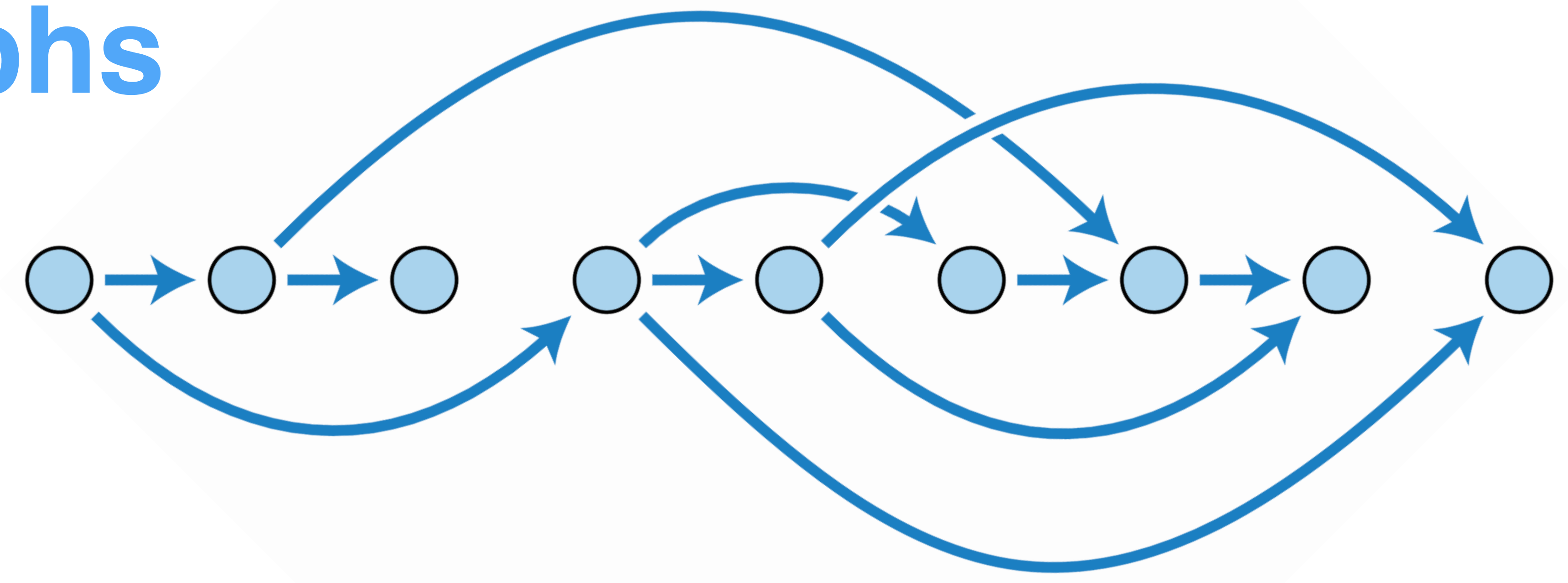
Background: Exact uniprocessor analysis

- ➡ Possible arrival times & execution times
- ➡ Deadline & priority

Finite sequence of jobs

Job_1, Job_2, Job_3, ⋮, Job_N

Schedule-abstraction graphs



Job_1 WCRT & BCRT

Job_2 WCRT & BCRT

Job_3 WCRT & BCRT

…

Job_N WCRT & BCRT

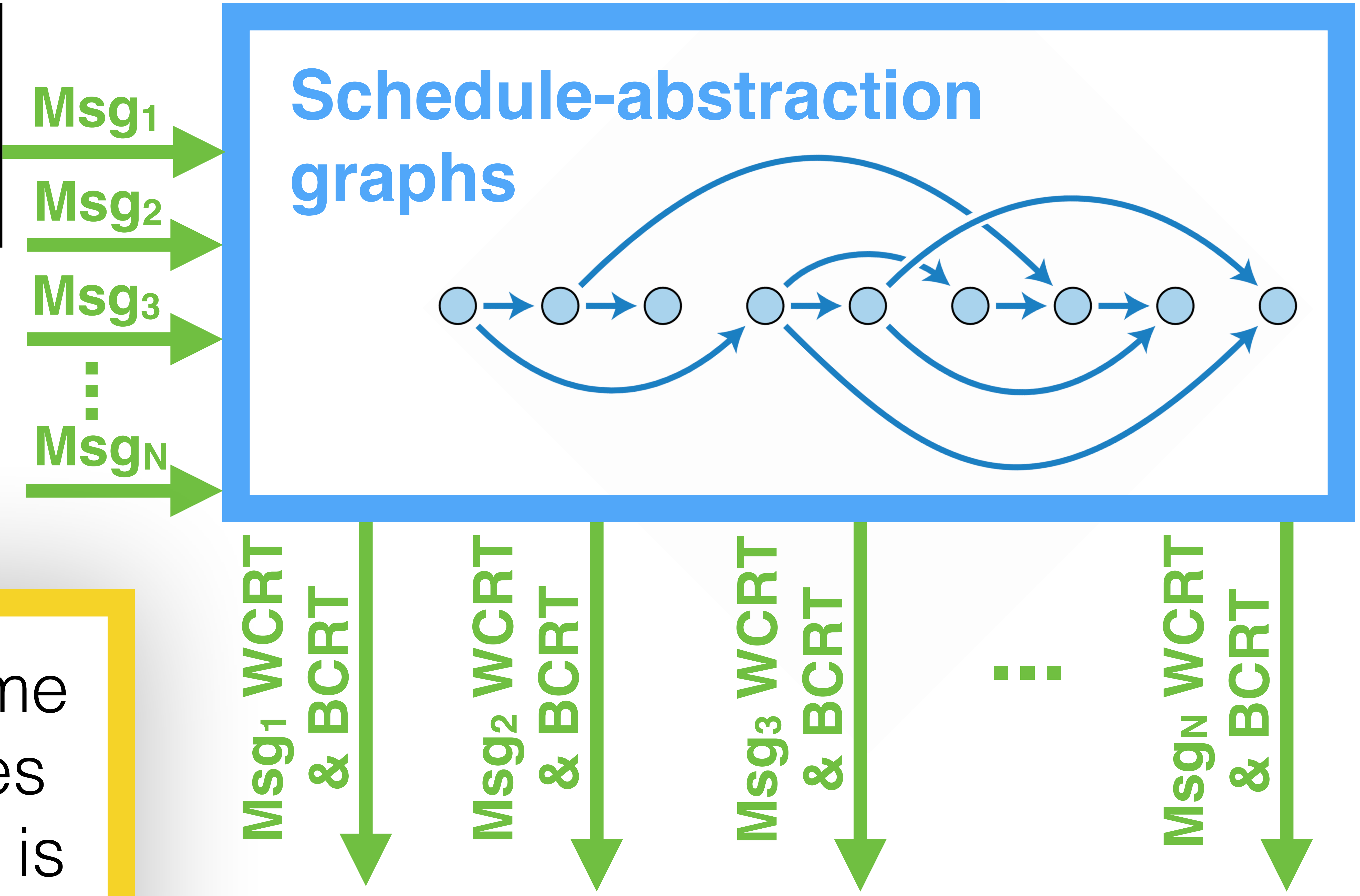
Periodic tasks
- ➡ Infinitely many jobs
- ➡ Consider the finite sequence of jobs in the hyperperiod

Exact worst-case response-time (WCRT) & best-case response-time (BCRT) of each job on a **uniprocessor** platform

Using schedule-abstraction graph analysis as a black-box

- ➡ Possible arrival times & **transmission** times
- ➡ Deadline & priority

Finite sequence of CAN messages



Fine-grained response-time analysis of CAN messages **without retransmissions** is **trivial!**

How to account for retransmissions?

Step 1: Suppose that all jobs are affected by up to **f** retransmissions

- ➡ A safe bound on **f** can be determined based on the hyperperiod
- ➡ For a probabilistic analysis, the analysis can be repeated for multiple values of **f**

How to account for retransmissions?

Step 2: Consider two sets of messages as input to the black-box

Messages that are successfully transmitted

Msg_1 Msg_2 Msg_3 Msg_4 ... Msg_N

$\underbrace{\qquad\qquad\qquad\qquad}$

Same parameters as the original set of CAN messages

Erroneous transmissions of up to **f** messages

Msg_{N+1} Msg_{N+2} Msg_{N+2} ... Msg_{N+f}

$\underbrace{\qquad\qquad\qquad\qquad}$

Since any message can be affected by EMI:

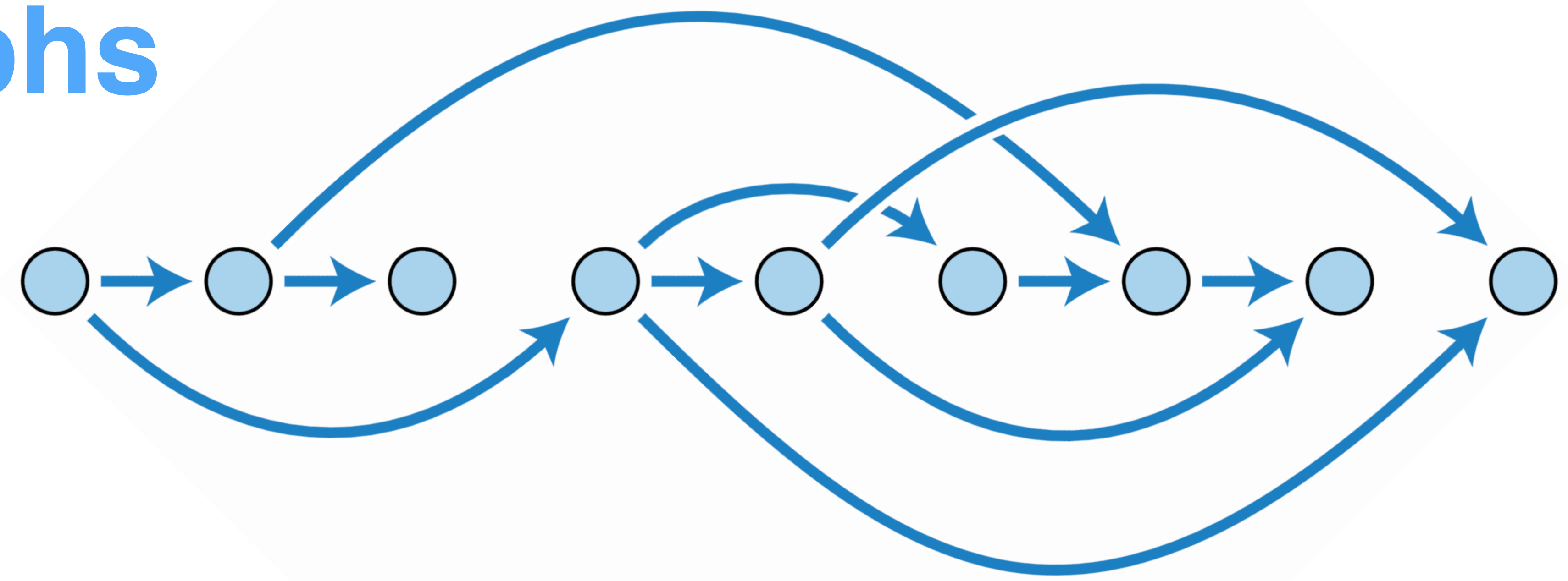
- ➡ Possible release times = **Union of the possible release times of Msg_1, Msg_2, ... Msg_N**
- ➡ Transmission times similarly assigned
- ➡ Priority = **Highest Priority**
- ➡ Deadline = **∞ (irrelevant)**

Safe response-time analysis with retransmissions

Msg$_1$ Msg$_2$ Msg$_3$ Msg$_4$ ... Msg$_N$

Msg$_{N+1}$ Msg$_{N+2}$ Msg$_{N+2}$ ... Msg$_{N+f}$

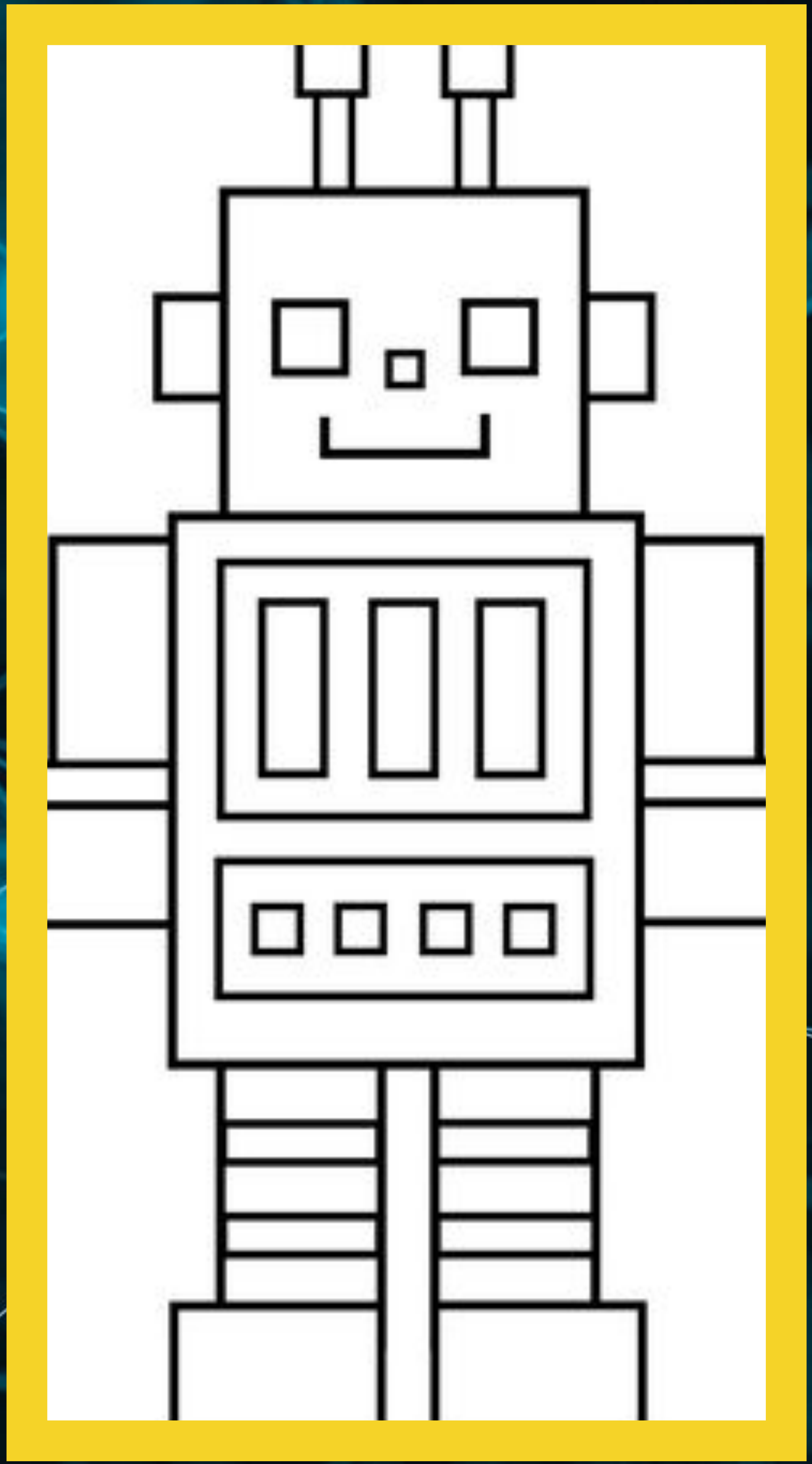
Schedule-abstraction graphs



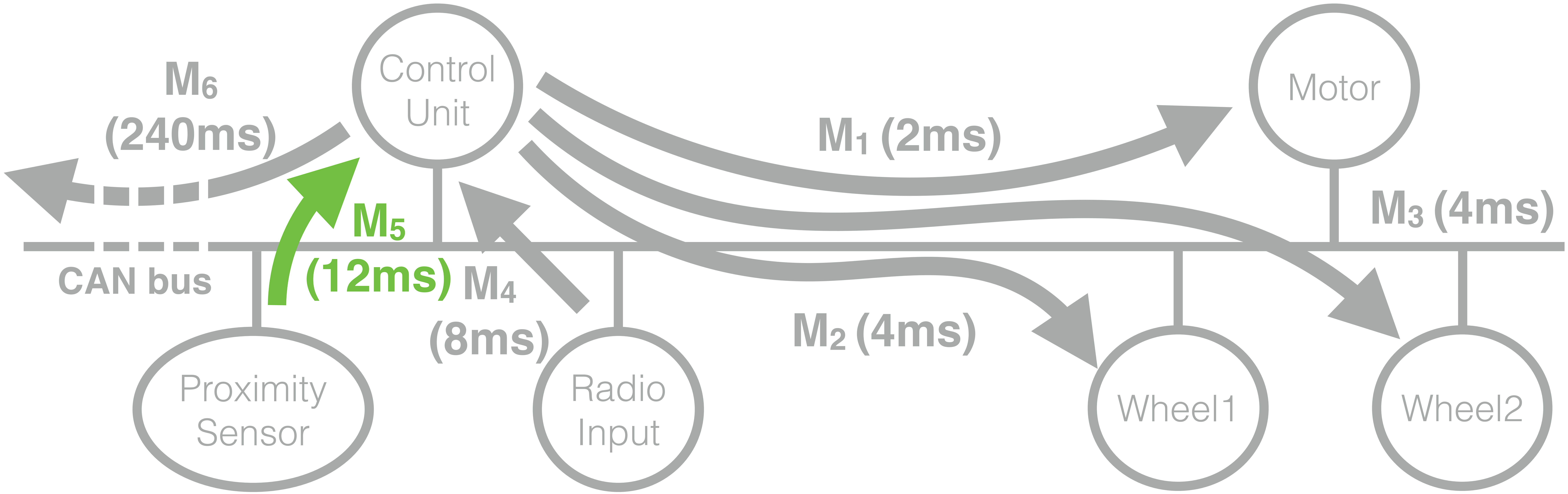
Exact BCRTs and upper bound on the WCRTs of successful transmissions

We don't care about RTs for erroneous transmissions

Case Study



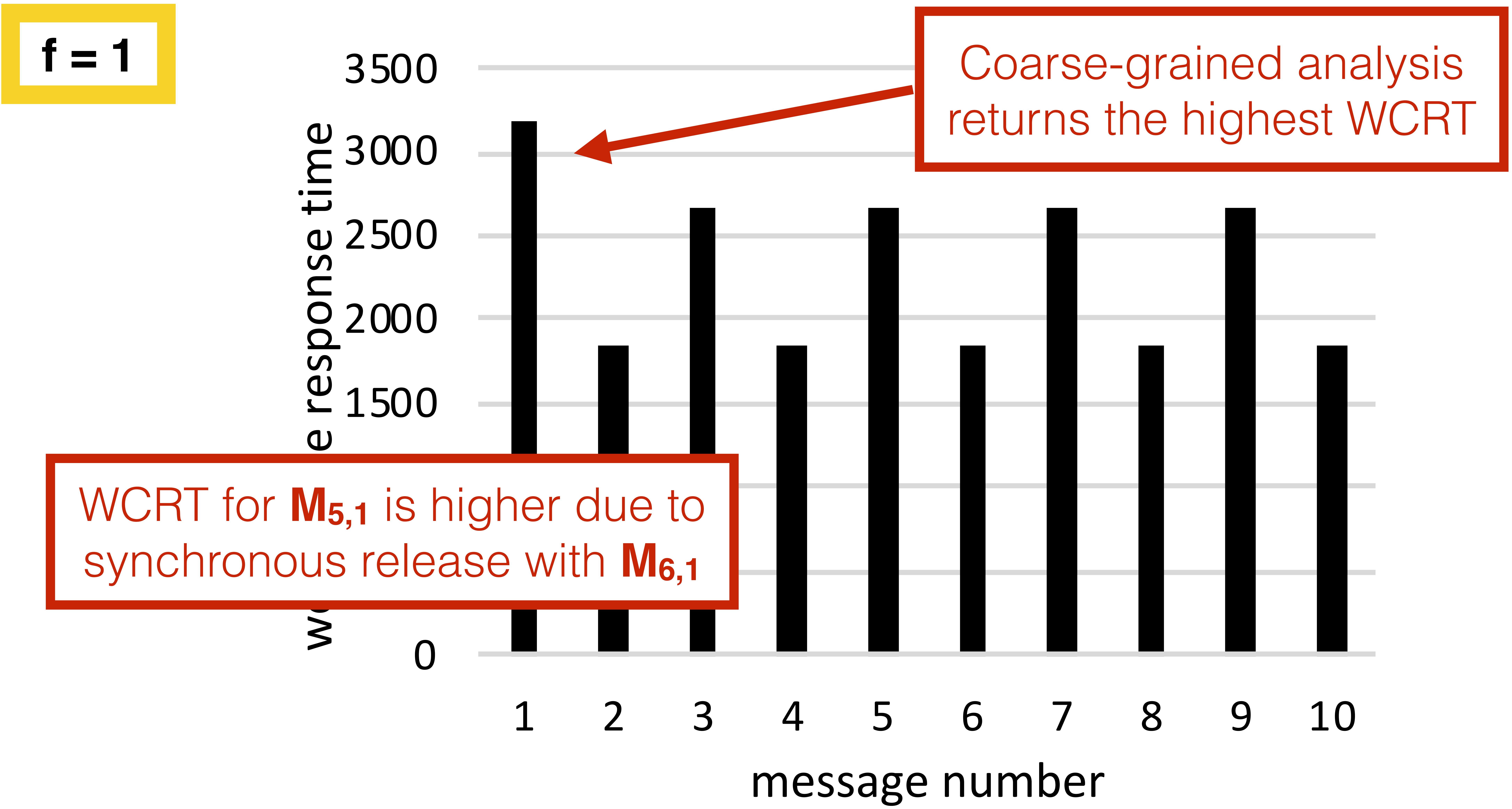
Example: CAN-based mobile robot



Objectives:

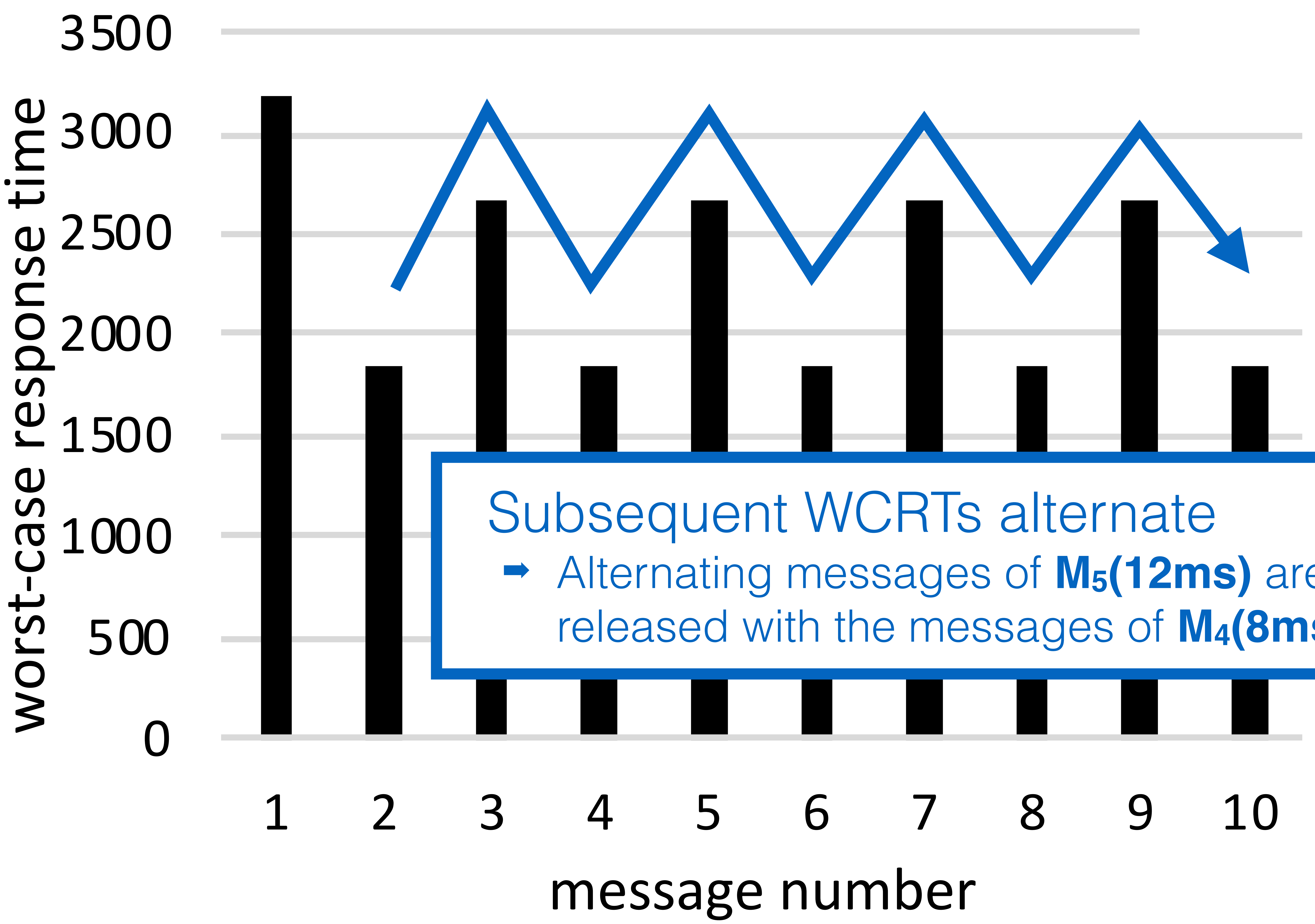
- ➡ Fine-grained WCRT analysis of $\mathbf{M_5}$
- ➡ Offset-assignment to improve $\mathbf{M_5}$'s maximum WCRT

Fine-grained WCRTs of M_5(12ms) in a synchronous release scenario



Fine-grained WCRTs of M_5(12ms) in a synchronous release scenario

f = 1

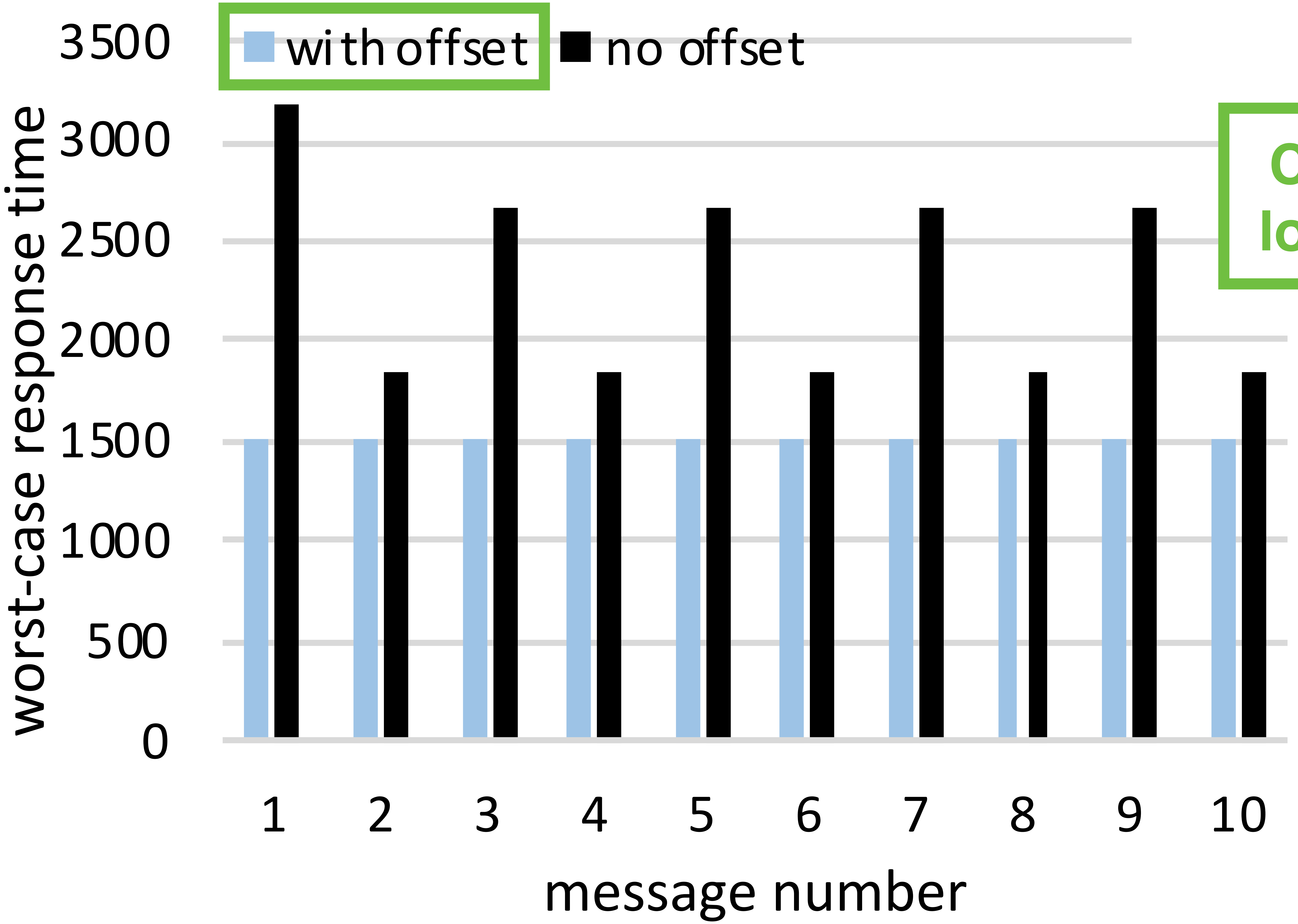


Subsequent WCRTs alternate

➡ Alternating messages of **M_5(12ms)** are synchronously released with the messages of **M_4(8ms)**

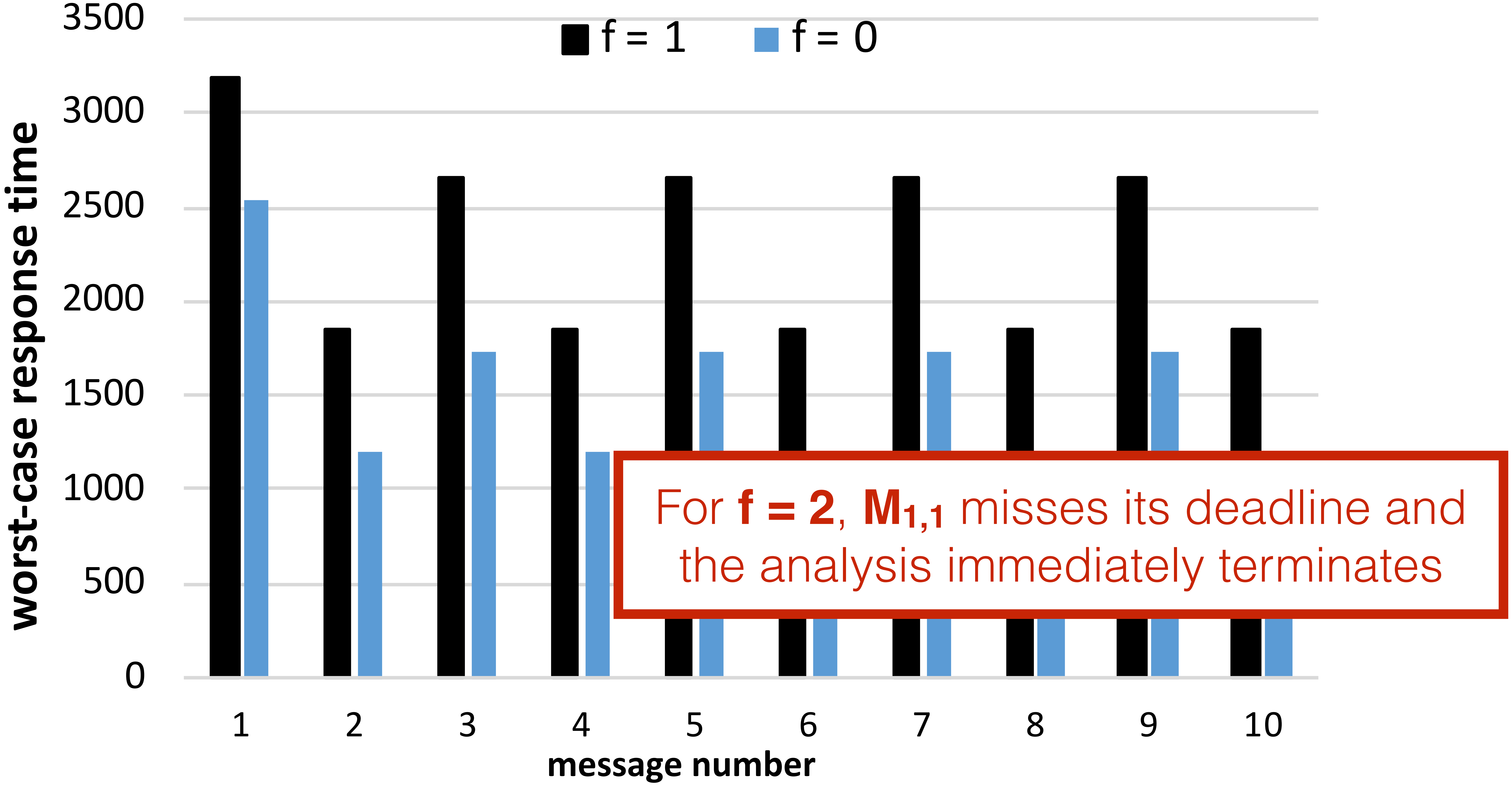
Improving WCRT profile of M_5(12ms) through offset assignment

f = 1



Consistently lower WCRTs

WCRT profile of M_5(12ms) for different values of f



Summary

Fine-grained analysis of CAN message response times **with retransmissions**

The analysis can estimate both **exact BCRTs** and **upper bounds on the WCRTs**

Future work: White-box analysis for exact WCRTs; **probabilistic analysis**

Backup Slides

Model

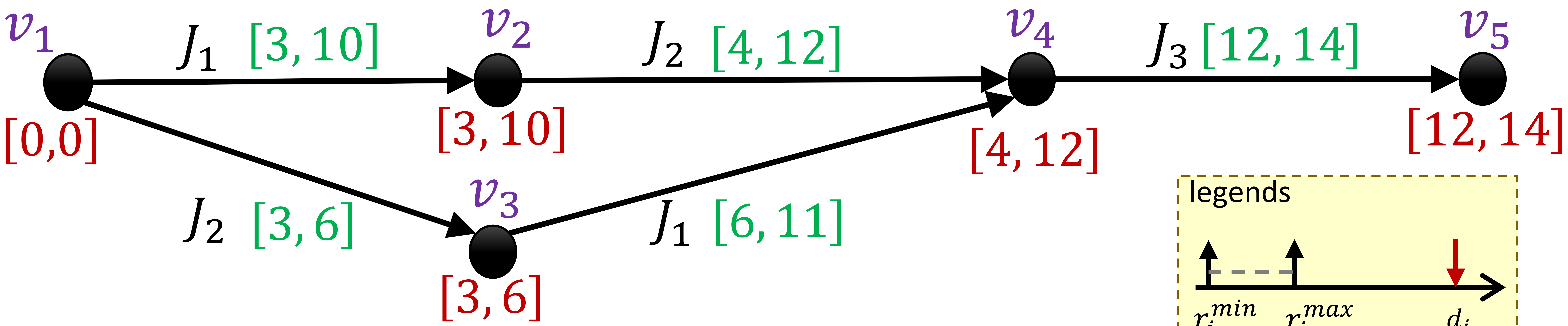
Given a finite set of jobs $\mathbf{J = \{J_1, J_2, J_3, \ldots \}}$ for a uniprocessor

- Each job $\mathbf{J_i = \{[r_i^{min}, r_i^{max}], [C_i^{min}, C_i^{max}], d_i, p_i\}}$
  - $\mathbf{[r_i^{min}, r_i^{max}]}$**:** Release interval accounting for release jitter
  - $\mathbf{[C_i^{min}, C_i^{max}]}$**:** Execution time interval
  - $\mathbf{d_i}$**:** Deadline
  - $\mathbf{p_i}$**:** Priority

The analysis returns the **exact** BCRT and the WCRT for each job in **J**

For periodic tasks resulting in infinitely many jobs, analysing a finite sequence of jobs in the hyperperiod is sufficient

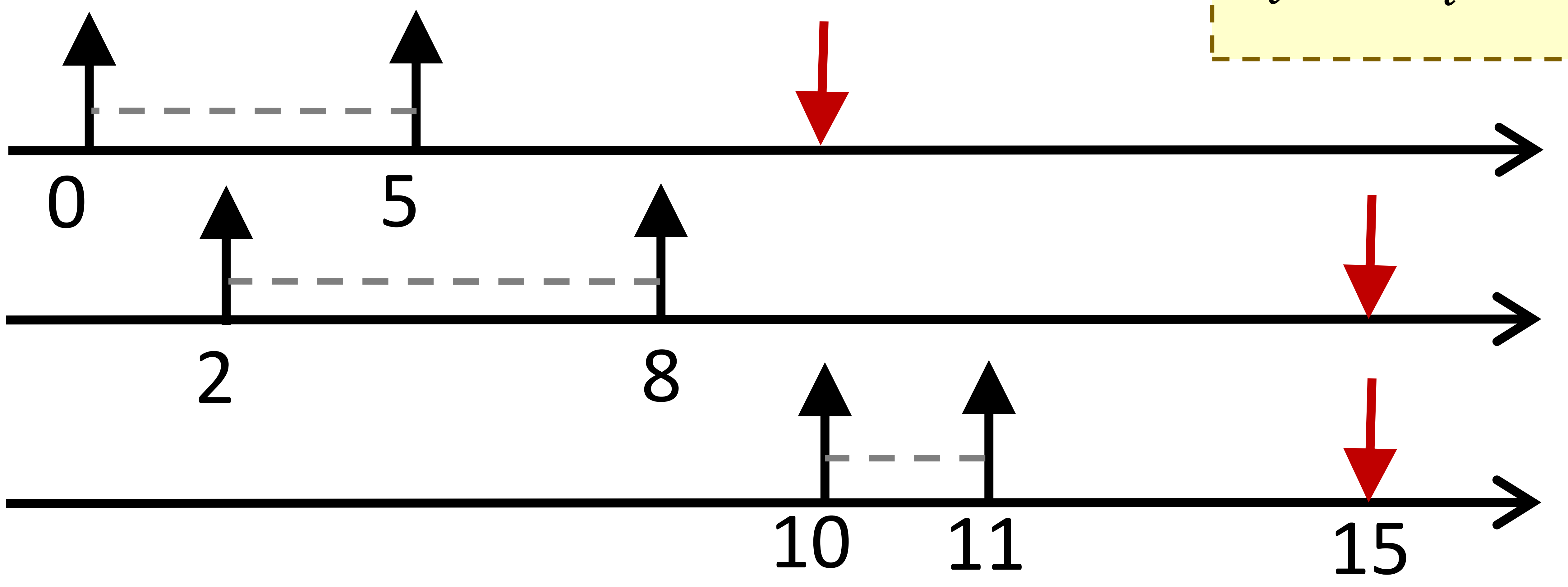
Example



$$J_1 = ([0,5],[3,5],10,1)$$

$$J_2 = ([2,8],[1,2],15,2)$$

$$J_3 = ([2,8],[2,4],15,3)$$

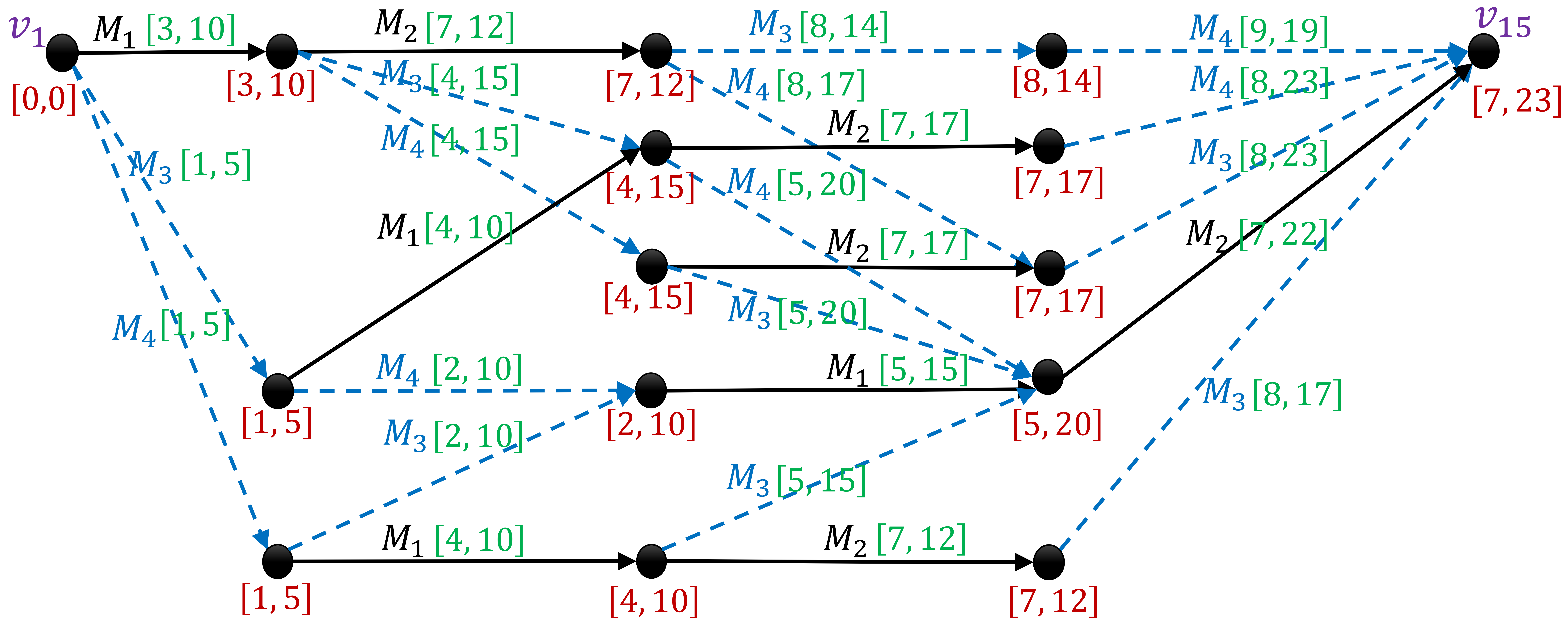


Accounting for f retransmissions

We consider a new message set $\mathbf{M' = M} \cup \mathbf{M^f}$

- $\mathbf{M^f = \{M_{n+1}, ..., M_{n+f}\}}$ denotes the set of **erroneous** transmissions
- Each erroneous message $\mathbf{M_{n+i} = \{[r^{min}, d^{max}], [C^{min} + \varepsilon, C^{max} + \varepsilon], \infty, 0\}}$
  - Since messages can be corrupted at any time,
    $\mathbf{r^{min} = min\{ r_i^{min} \mid M_i} \in \mathbf{M \}}$ and $\mathbf{d^{max} = max\{ d_i \mid M_i} \in \mathbf{M \}}$
  - Since any message in **M** can be corrupted,
    $\mathbf{C^{min} = min\{ C_i^{min} \mid M_i} \in \mathbf{M \}}$ and $\mathbf{C^{max} = max\{ C_i^{max} \mid M_i} \in \mathbf{M \}}$
  - $\boldsymbol{\varepsilon}$ denotes the error frame transmission overhead
  - To model corruption of the highest-priority message, $\mathbf{p_i = 0}$
  - Since deadline of an erroneous message is irrelevant, $\mathbf{d_i = \infty}$

Example (f = 2)



v_1
v_{15}
M_1 [3, 10]
M_2 [7, 12]
M_3 [8, 14]
M_4 [9, 19]
[0,0]
[3, 10]
[7, 12]
[8, 14]
[7, 23]
M_3 [4, 15]
M_4 [8, 17]
M_4 [8, 23]
M_4 [4, 15]
M_2 [7, 17]
M_3 [1, 5]
[4, 15]
M_4 [5, 20]
[7, 17]
M_3 [8, 23]
M_1 [4, 10]
M_2 [7, 17]
M_2 [7, 22]
[4, 15]
[7, 17]
M_4 [1, 5]
M_3 [5, 20]
M_4 [2, 10]
M_1 [5, 15]
[1, 5]
[2, 10]
[5, 20]
M_3 [8, 17]
M_3 [2, 10]
M_3 [5, 15]
M_1 [4, 10]
M_2 [7, 12]
[1, 5]
[4, 10]
[7, 12]


(a)
$\bar{M}_1$
with offset
no offset
WCRT
message number
(b)
$\bar{M}_2$
with offset
no offset
WCRT
message number
(c)
$\bar{M}_3$
with offset
no offset
WCRT
message number
(d)
$\bar{M}_4$
with offset
no offset
WCRT
message number
(e)
$\bar{M}_5$
with offset
no offset
WCRT
message number
(f)
with offset
no offset
WCRT
messages in one hyperperiod sorted by their WCRT
(g)
$\bar{M}_5$ (no offset)
WCRT
message number
(h)
with offset
no ofset
% of jobs in hyperperiod
WCRT