Assignment 4: End-to-end timing analysis

Author: Sebastian Daag

Email: sdg20001@student.mdu.se

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 with by at least one task in each node. The interfering 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

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. Receiving tasks use polling policy to receive messages from the network
4. Receiving tasks “just miss” the read access of the messages

System information

Constraints

Constraint type	Time constraint
Age	15 ms
Reaction	30 ms

Taks

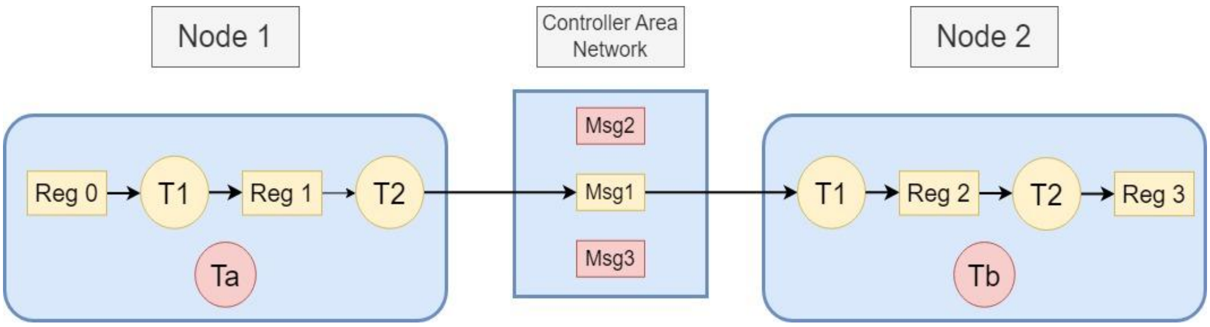
Periodic task	Type	Period	Priority	WCET
Task 1	Independently activated	10	High	1 ms
Task 2	Independently activated	5	Low	1 ms

Task 3	Independently activated	10	High	1 ms
Task 4	Independently activated	5	Low	1 ms
Task A	Interfering task	2	High	1 ms
Task B	Interfering task	2	High	1 ms

Messages

Message	CAN Speed	Bytes	Priority	Transm. Time
Msg 1	250 Kbit/s	8	Low	0.54 ms
Msg 2	250 Kbit/s	8	Medium	0.54 ms
Msg 3	250 Kbit/s	8	High	0.54 ms

Figure:

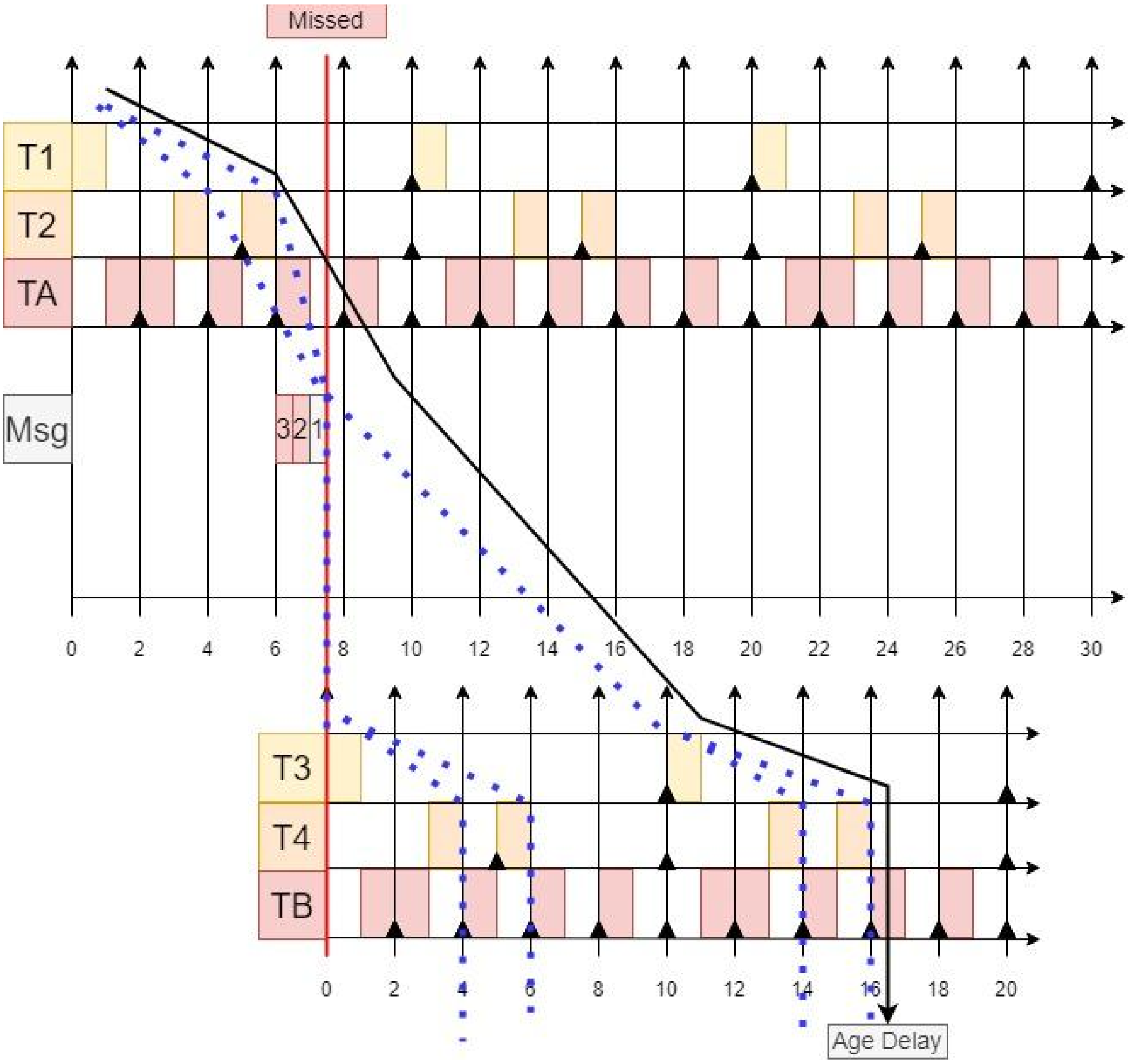


Where:

- The yellow blocks are part of the chain
- The red block are the interferences

(a) Identify the age and reaction delays using a timeline graph.

WCET timeline graph, age delay



Explanation:

- If T3 in Node 2 implements polling policy for receiving the message then in the worst-case scenario, T3 "just misses" Msg1. The missed message is read by the next instance of T3 has shown by the black arrow
- Task T3 starts just as the first message is sent so it misses it.

Info from graph:

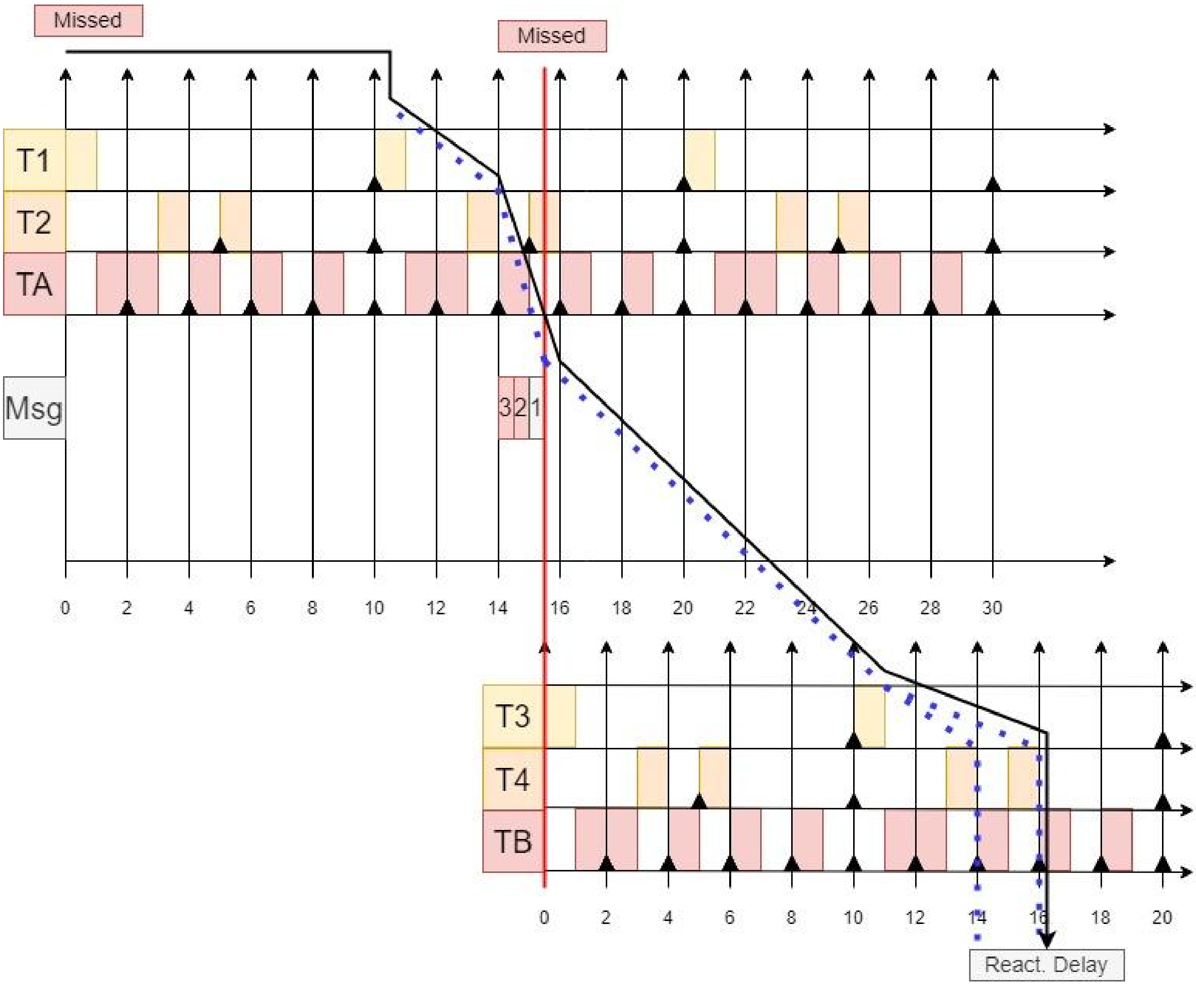
- Msg 2 and 3 will trigger just before Msg1.
  - Msg1: 0.54 + 0.54 + 0.54 = 1.62 ms
- The T1 or T3 misses data until the next period.
  - Missed delay = 10 ms

Answer:

Age: Node1 + Message + Missed delay + Node2 = Age delay

Age: 6 + 1.62 + 10 + 6 = <u>23.62 ms</u>

WCET timeline graph, reaction delay



Explanation

- The first miss
  - “Task T1 just missed the new data”, The missed data is read by the next instance of T1
- If T3 in Node 2 implements polling policy for receiving the message then in the worst-case scenario, T3 “just misses” Msg1. The missed message is read by the next instance of T3.
- Task T3 starts just as the first message is sent so it misses it.

Info from graph:

- Msg 2 and 3 will trigger just before Msg1.
  - Msg1: 0.54 + 0.54 + 0.54 = 1.62 ms
- The T1 or T3 misses data until the next period.
  - Missed delay = 10 ms

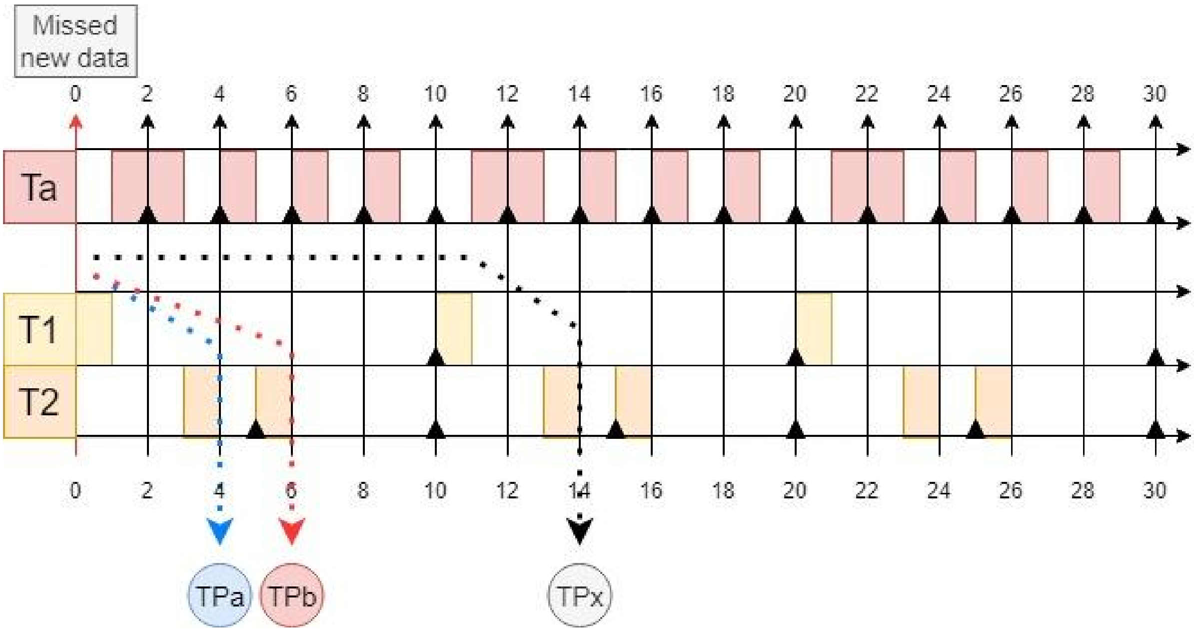
Answer:

Reaction: Missed delay + Node 1 + Message + Missed delay + Node 2 = Reaction delay

Reaction: 10 + 4 + 1.62 + 10 + 4 = 14 + 1.62 + 14 = <u>29.62 ms</u>

(b) Calculate the age and reaction delays mathematically.

Time graph node 1:



Age delay

Element	Formula	Value
TPa	(T1(1), T2(1))	{α1(1) = 0, R1(1) = 1} , {α2(1) = 2, R2(1) = 2}
TPb	(T1(1), T2(2))	{α1(1) = 0, R1(1) = 1} , {α2(2) = 4, R2(2) = 2}
Delay(TPa)	α2(TPa) + R2(TPa) - α1(TPa)	2 + 2 - 0 = 4
Delay(TPb)	α2(TPb) + R2(TPb) - α1(TPb)	4 + 2 - 0 = 6
Age delay	max{Delay(TPa), Delay(TPb)}	max{4, 6} = 6

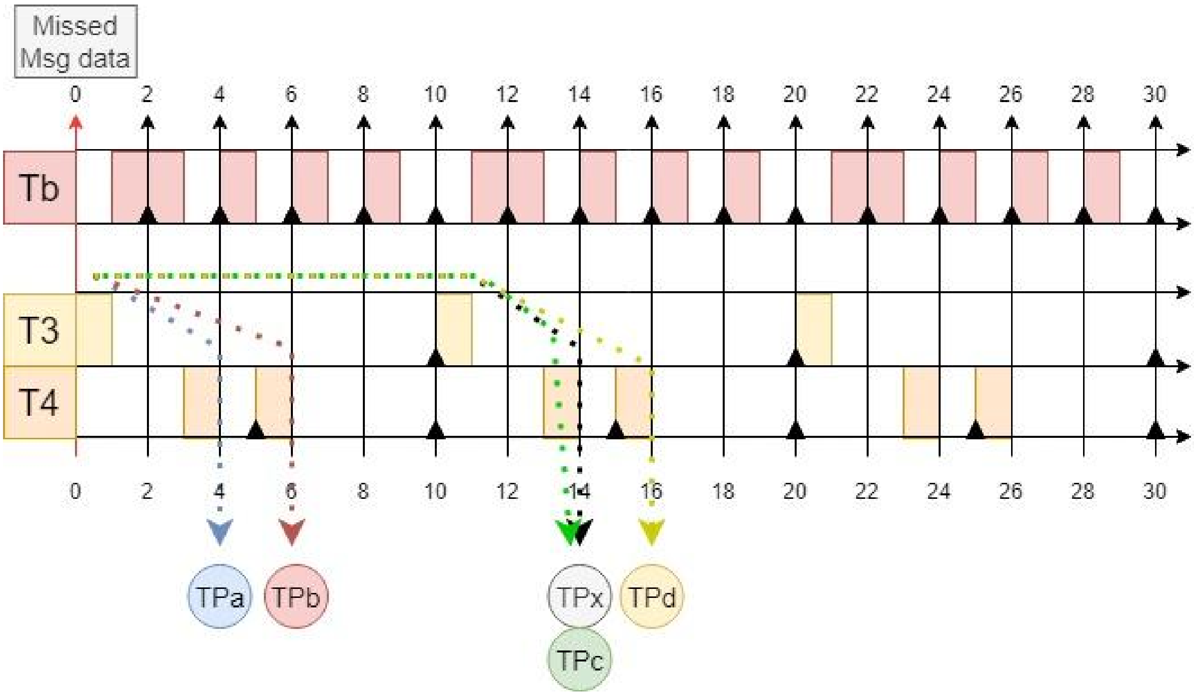
Reaction delay

Element	Formula	Value
TPx	(T1(2), T2(3))	{α1(2) = 10, R1(2) = 1} , {α2(3) = 12, R2(3) = 2}
Delay(TPx)	α2(TPx) + R2(TPx) - α1,Pred(TPx)	12 + 2 - 0 = 14
Reaction delay	max{Delay(TPx)}	max{14} = 14

Node 1 data:

Age delay = 6 ms
Reaction delay = 14 ms

Time graph node 2:



Age delay

Element	Formula	Value
TPa	(T3(1), T4(1))	{α1(1) = 0, R1(1) = 1}, {α2(1) = 2, R2(1) = 2}
TPb	(T3(1), T4(2))	{α1(1) = 0, R1(1) = 1}, {α2(2) = 4, R2(2) = 2}
TPc	(T3(2), T4(3))	{α1(1) = 10, R1(1) = 1}, {α2(1) = 12, R2(1) = 2}
TPd	(T3(2), T3(4))	{α1(1) = 10, R1(1) = 1}, {α2(2) = 14, R2(2) = 2}
Delay(TPa)	α2(TPa) + R2(TPa) - α3(TPa)	2 + 2 - 0 = 4
Delay(TPb)	α2(TPb) + R2(TPb) - α1(TPb)	4 + 2 - 0 = 6
Delay(TPc)	α4(TPc) + R4(TPc) - α1(TPc)	12 + 2 - 0 = 14
Delay(TPd)	α4(TPd) + R4(TPd) - α1(TPd)	14 + 2 - 0 = 16
Age delay	max{Delay(TPa), Delay(TPb), Delay(TPc), Delay(TPd)}	max{4, 6, 14, 16} = 16

Reaction delay

Element	Formula	Value
TPx	(T3(2), T4(3))	{α1(2) = 10, R1(2) = 1} , {α2(3) = 12, R2(3) = 2}
Delay(TPx)	α2(TPx) + R2(TPx) - α1,Pred(TPx)	12 + 2 - 0 = 14
Reaction delay	max{Delay(TPx)}	max{14} = 14

Node 2 data:

Age delay = 16 ms
Reaction delay = 14 ms

Messages:

Msg 2 and 3 will trigger just before Msg1, so Msg1 takes all their time.
Msg transmission time = 0.54 ms
Msg1: 0.54 + 0.54 + 0.54 = 1.62 ms

Summary:

Constraint type	Formula	Value
Age delay	Node1 + Msg + Node2	6 + 1.62 + 16 = 23.62 ms
Reaction delay	Node1 + Msg + Node2	14 + 1.62 + 14 = 29.62 ms

Explanation of formulas:
Tk(n) = task k of instance n
α = activation time, all interference/delayed time
R = response time, include all previous tasks, assuming no delays
Delay TPm = α,Last(TPm) + R,Last(TPm) - α,First(TPm)
Max delay = Max{Delay(TP1), …, Delay(TPm)}
α,Pred(TPx), means the activation time of the instance that is predecessor to the instance of the First task.

α,Last(TPm) is the activation time of the instance of the Last task which is in path TPm
R,Last(TPm) is the response time of the instance of the Last task which is in path TPm
α,First(TPm) is the activation time of the instance of the First task which is in path TPm

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

Constraint type	Time constraints	Time delay	Satisfied
Age	15	23.62	No
Reaction	30	29.62	Yes

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

1. No offsets
   - If any offsets were to be added to the end-to-end data-propagation delay analysis, it would experience both jitter and unpredictability.
   - All time delays would increase, age, reaction, message time and delays. Impacting the synchronization between tasks.
   - The WCET becomes more complex and uncertain.

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
   - It would lead to unpredictable task Interference.
   - If the T2 in the node had a higher priority than T1, it could be called before T1 has received any data. Thus potentially causing the node to miss data packages and cause longer delays.
   - It would make the entire time graph more complex

3. Receiving tasks use polling policy to receive messages from the network
   - It would mean having to use another or no policy for receiving messages:
   - If an interrupt based policy was to be implemented it might actually improve the time delay and satisfy the age constraint.
     - This is because Node 2 would not be able to miss a message because T3 would only start up in the arrival of Msg1.
     - Which could actually reduce the complexity.
   - With no policy it would quickly become complex and chaotic, if not just impossible.

4. Receiving tasks “just miss” the read access of the messages
   - If receiving tasks do not "just miss" the read access, they would not have to wait the extra hyper period before processing the data.
     - Which would mean reduced age and reaction delays in node 2 and reduced age and reaction delays overall.
   - The system would probably become less complex