# Selected solutions for

# TDDD07
# Real-time Systems

***Distributed systems, QoS, Real-time communication***

Massimiliano Raciti
Jordi Cucurull
Simin Nadjm-Tehrani

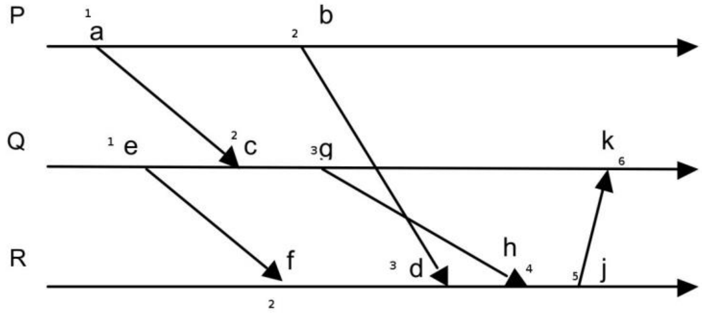
Real-Time Systems Laboratory
Department of Computer and Information Science
Linköping University, Sweden

November 2019

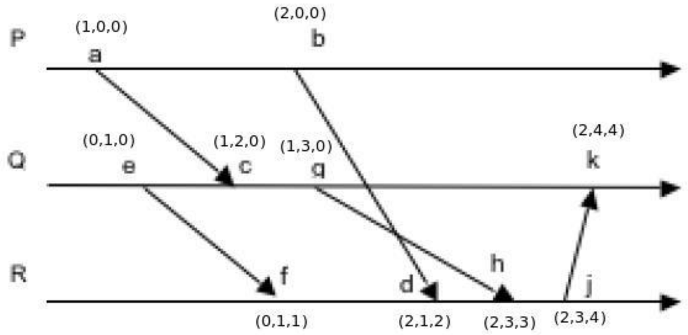
## Suggested Solutions

### Q2.1:

a)
Logic clock

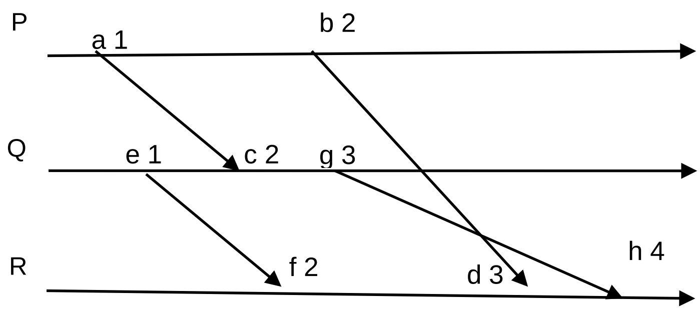



Vector clock




b)
(a,e) and (c,f) are concurrent event.
Let VC(a) and VC(e) be the vector timestamps at the time when a or e occurs.
The events a and e are concurrent since none of $VC(a) < VC(e)$ or $VC(e) < VC(a)$

### Q2.2:



In order to check whether there are events x and y in which LC(x) < LC(y) but is not the case that x happen before y, we should determine all the "happens before relationships".

a≺b (because from the point of view of P, a occurs before than b)

a≺c (because *c* is the event of reception of a message m and *a* is the event of sending m)

e≺c , e≺f, b≺d, c≺g, g≺h, f≺d, d≺h.

Using the transitive property (x ≺y and y ≺z -> x ≺z), we get all the possible relationships:

a≺b, a≺c, a≺d, a≺h, b≺d, b≺h, c≺g, c≺h, d≺h, e≺c, e≺g, e≺h, e≺f, e≺d f≺d, f≺h, g≺h.

Now, if we take the events b and g, we can see that LC(b)<LC(g), but we don't have b≺g. B and g were two "easy" examples, since graphically you can recognise that they are concurrent. There are other cases anyway that are not so easy to recognise: c and d, for example. LC(c)<LC(d) but we don't have c≺d.

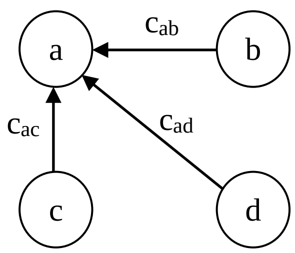
**Q2.3:**

An example of distributed real time system with hard deadlines can be the control system of car, where many elements are distributed and linked with a bus. This system has hard deadlines because the safety of the user depends on the correctly functionality of the service, i.e. no deadline misses are allowed.

An example of distributed real time system with soft deadlines can be a regular videoconference system, such as Skype. The system has soft deadlines because in case some of them are missed it will just affect the video/sound quality of the call.

**Q2.4:**

a) This is the scenario described:




Node *a* has an associated clock $c_a$ with time *t*. We represent the time calculated after synchronisation as $c_a'$. This new time is calculated as:

$$c_a' = \frac{c_a + c_{ab} + c_{ac} + c_{ad}}{4}$$

where $c_{ab} + c_{ac} + c_{ad}$ are the clocks the other nodes (*b*, *c*, and *d*) send to *a*.

In this scenario two of the nodes (*b* and *c*) will send a time that differ exactly δ from *t*. There are two cases to consider for clock d:

1) The other node (*d*) will send a time that differs γ from *t* and γ < δ. Given this information, three cases can be distinguished:

   i. $c_a = t \quad c_{ab} = t + \delta \quad c_{ac} = t - \delta \quad c_{ad} = t \pm \gamma$

new time is $c_a' = \dfrac{t + t + \delta + t - \delta + t \pm \gamma}{4} = t \pm \dfrac{1}{4}\gamma$

then $|c_a - c_a'| < \delta$.

   ii. $c_a = t \quad c_{ab} = t + \delta \quad c_{ac} = t + \delta \quad c_{ad} = t \pm \gamma$

new time is $c_a' = \frac{t+t+\delta+t+\delta+t\pm\gamma}{4} = t + \frac{1}{2}\delta \pm \frac{1}{4}\gamma$

then $|c_a - c_a'| < \delta$.

   iii. $c_a = t \quad c_{ab} = t - \delta \quad c_{ac} = t - \delta \quad c_{ad} = t \pm \gamma$

new time is $c_a' = \frac{t+t-\delta+t-\delta+t\pm\gamma}{4} = t - \frac{1}{2}\delta \pm \frac{1}{4}\gamma$

then $|c_a - c_a'| < \delta$.

Note that under all three cases (i..iii) the other three nodes will also get the same clock value equal to $c_a'$. Then the synchronisation requirement at node *a* is met, meaning that *a* remains within the allowed skew compared to all other nodes.

b) In the second case node *d* will send a time that differs γ from *t* and γ > δ.
   Under this assumption, the value of node *d* in the calculations above will be replaced by t.

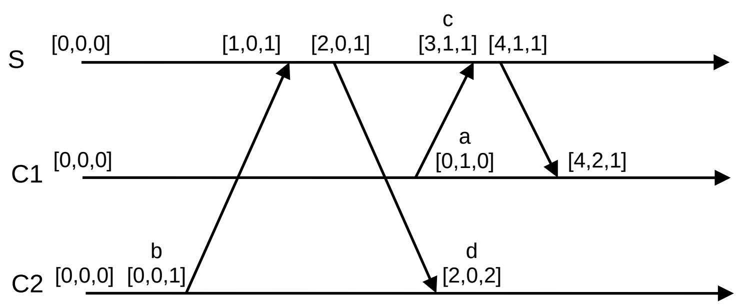
Case i: $c_a = t \quad c_{ab} = t + \delta \quad c_{ac} = t - \delta \quad c_{ad} = t$
Case ii: $c_a = t \quad c_{ab} = t + \delta \quad c_{ac} = t + \delta \quad c_{ad} = t$
Case iii: $c_a = t \quad c_{ab} = t - \delta \quad c_{ac} = t - \delta \quad c_{ad} = t$

Note that in this case we need not consider the new value for node *d*, since the algorithm requires to synchronise all non-faulty nodes, and the number of faulty nodes (f) here within a group of n (n=4) clocks satisfies the guarantee requirement (3f < n).

In all three cases $|c_a - c_a'| < \delta$ (with similar calculations to case 1. above) and since *b* and *c* were synchronised with *a* before the synchronisation, they remain synchronised after the replacement of *d*'s clock value with $t + \delta$ or $t - \delta$ and dividing their respective sums by 4.

The worst case for deviation from $c_a'$ (at *b* or *c*) is $|c_a' - (4t+3\delta)/4|$ or $|c_a' - (4t-3\delta)/4|$.

b)




<u>Concurrent events</u>

a: C1-Sending [0,1,0]
b: C2-Sending [0,0,1]

$VC(a) \nless VC(b)$ and $VC(b) \nless VC(a)$

c: Reception C1 request [3,1,1]
d: Reception S response in C2 [2,0,2]

$VC(c) \nless VC(d)$ and $VC(d) \nless VC(c)$

**Q2.5**

Consider the following two terms used in quality of service (QoS) requirements descriptions, and identify for each term whether it is an application level description or an enforcement level indication: loss ratio, video quality.

Loss ratio is an enforcement level indication, since it is a generic quality of service parameter applicable to any underlying service to support applications. Video quality, instead, is an application level description since it refers to a quality of service parameter specific of a video application that it is directly noticeable by the user.

**Q2.6:**

a) See course literature.

b) We have the following parameters

| Node | Period $T_i$ (ms) | Priority (π) |
|---|---|---|
| 1 | 5 | 3 |
| 2 | 7 | 2 |
| 3 | 10 | 1 |

The formulas required to calculate the worst-case response time of each message are the following

$$w_i^{n+1}(q) = B_i + qC_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i^n + J_j + \tau_{bit}}{T_j} \right\rceil C_j \quad \text{where} \quad w_i^0(q) = B_i + qC_i$$

$$R_i(q) = J_i + w_i(q) - qT_i + C_i$$

$$R_i = \max_{q=0..Q_i-1} (R_i(q))$$

In addition, before using them, the number of possible messages, $Q_i$, that become ready before the end of the busy period must be calculated with the following formulas:

$$t_i^{n+1} = B_i + \sum_{\forall j \in hep(i)} \left\lceil \frac{t_i^n + J_j}{T_j} \right\rceil C_j \quad \text{where} \quad t_i^0 = C_i$$

$$Q_i = \left\lceil \frac{t_i + J_i}{Ti} \right\rceil$$

The maximum blocking time for respective messages is $B_1 = B_2 = 135\mu s$ and $B_3 = 0\mu s$. Frames sent by $n_3$ have the lowest priority, reason why their blocking factor is defined to zero. The transmission time is given by $C_i = 135\tau_{bit} = 135\mu s$.

In this first case we assume $J_i = 0\mu s$, i=1..3.

**Response time R3:** We calculate first the length of the busy period for message 3:

$$t_3^0 = C_3 = 135\mu s$$

$$t_3^1 = B_3 + \left\lceil \frac{t_3^0 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_3^0 + J_2}{T_2} \right\rceil C_2 + \left\lceil \frac{t_3^0 + J_3}{T_3} \right\rceil C_3 = 405\mu s$$

$$t_3^2 = B_3 + \left\lceil \frac{t_3^1 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_3^1 + J_2}{T_2} \right\rceil C_2 + \left\lceil \frac{t_3^1 + J_3}{T_3} \right\rceil C_3 = 405\mu s$$

$$t_3^2 = t_3^1 = t_3 = 405\mu s$$

Then the number of instances Q3 that become ready:

$$Q_3 = \left\lceil \frac{t_3 + J_3}{T_3} \right\rceil = 1$$

The response time must be calculated for each of the $Q_3$ instances:

$$w_3^0(0) = B_3 + 0C_3 = 0\mu s$$

$$w_3^1(0) = B_3 + 0C_3 + \left\lceil \frac{w_3^0 + J_1 + \tau_{bit}}{T_1} \right\rceil C_1 + \left\lceil \frac{w_3^0 + J_2 + \tau_{bit}}{T_2} \right\rceil C_2 = 270\mu s$$

$$w_3^2(0) = B_3 + 0C_3 + \left\lceil \frac{w_3^1 + J_1 + \tau_{bit}}{T_1} \right\rceil C_1 + \left\lceil \frac{w_3^1 + J_2 + \tau_{bit}}{T_2} \right\rceil C_2 = 270\mu s$$

$$w_3^2 = w_3^1 = w_3 = 270\mu s$$

$$R_3(0) = J_3 + w_3(0) - 0T_3 + C_3 = 405\mu s$$

$$R_3 = \max_{q=0..Q_3-1}(R_i(q)) = R_3(0) = 405\mu s$$

c)

**Response time R3 with Jitter:**

$$J_1 = J_2 = 5ms, \quad J_3 = 8ms$$

$$t_3^0 = C_3 = 135\mu s$$

$$t_3^1 = B_3 + \left\lceil \frac{t_3^0 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_3^0 + J_2}{T_2} \right\rceil C_2 + \left\lceil \frac{t_3^0 + J_3}{T_3} \right\rceil C_3 = 540\mu s$$

$$t_3^2 = B_3 + \left\lceil \frac{t_3^1 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_3^1 + J_2}{T_2} \right\rceil C_2 + \left\lceil \frac{t_3^1 + J_3}{T_3} \right\rceil C_3 = 540\mu s$$

$$t_3^2 = t_3^1 = t_3 = 540\mu s$$

Then the number of instances $Q_3$ that become ready:

$$Q_3 = \left\lceil \frac{t_3 + J_3}{T_3} \right\rceil = 1$$

The response time must be calculated for each of the $Q_3$ instances:

$$w_3^0(0) = B_3 + 0C_3 = 0\mu s$$

$$w_3^1(0) = B_3 + 0C_3 + \left\lceil \frac{w_3^0 + J_1 + t_{bit}}{T_1} \right\rceil C_1 + \left\lceil \frac{w_3^0 + J_2 + t_{bit}}{T_2} \right\rceil C_2 = 405\mu s$$

$$w_3^2(0) = B_3 + 0C_3 + \left\lceil \frac{w_3^1 + J_1 + t_{bit}}{T_1} \right\rceil C_1 + \left\lceil \frac{w_3^1 + J_2 + t_{bit}}{T_2} \right\rceil C_2 = 405\mu s$$

$$w_3^2 = w_3^1 = w_3 = 405\mu s$$

$$R_3(0) = J_3 + w_3(0) - 0T_3 + C_3 = 8540\mu s$$

$$R_3 = \max_{q=0..Q_3-1}(R_i(q)) = R_3(0) = 8540\mu s$$

**Q2.7:**

| Message | Period $T_i$ (ms) | Priority ($\pi$) | Jitter (ms) |
|---|---|---|---|
| m1 | 20 | high | 1 |
| m2 | 10 | middle | 2 |
| m3 | 5 | low | 0 |

**Response time $R_3$:**

$B_3 = 0ms$, since m3 is the lowest priority process.

First the length of the busy period for message 3 is calculated:

$$t_3^0 = C_3 = 1ms$$

$$t_3^1 = B_3 + \left\lceil \frac{t_3^0 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_3^0 + J_2}{T_2} \right\rceil C_2 + \left\lceil \frac{t_3^0 + J_3}{T_3} \right\rceil C_3 = 3ms$$

$$t_3^1 = B_3 + \left\lceil \frac{t_3^1 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_3^1 + J_2}{T_2} \right\rceil C_2 + \left\lceil \frac{t_3^1 + J_3}{T_3} \right\rceil C_3 = 3ms$$

$$t_3 = 3ms$$

Then the number of instances $Q_3$ of message 3 that become ready before the end of the busy period:

$$Q_3 = \left\lceil \frac{t_3 + J_3}{T_3} \right\rceil = 1$$

The response time must be calculated for each of the $Q_3$ instances, in this case just one.

$$w_3^0(0) = B_3 + 0C_3 = 0ms$$

$$w_3^1(0) = B_3 + 0C_3 + \left\lceil \frac{w_3^0 + J_1 + \tau_{bit}}{T_1} \right\rceil C_1 + \left\lceil \frac{w_3^0 + J_2 + \tau_{bit}}{T_2} \right\rceil C_2 = 2ms$$

$$w_3^2(0) = B_3 + 0C_3 + \left\lceil \frac{w_3^1 + J_1 + \tau_{bit}}{T_1} \right\rceil C_1 + \left\lceil \frac{w_3^1 + J_2 + \tau_{bit}}{T_2} \right\rceil C_2 = 2ms$$

$$w_3(0) = 2ms$$

$$R_3(0) = J_3 + w_3(0) - 0T_3 + C_3 = 3ms$$

$$R_3 = \max_{q=0..Q_3-1}(R_i(q)) = R_3(0) = 3ms$$

**Response time $R_2$:**

$$B_2 = 1ms$$

$$t_2^0 = C_2 = 1ms$$

$$t_2^1 = B_2 + \left\lceil \frac{t_2^0 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_2^0 + J_2}{T_2} \right\rceil C_2 = 3ms$$

$$t_2^2 = B_2 + \left\lceil \frac{t_2^0 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_2^0 + J_2}{T_2} \right\rceil C_2 = 3ms$$

$$t_2 = 3ms$$

Then the number of instances $Q_2$ that become ready:

$$Q_2 = \left\lceil \frac{t_2 + J_2}{T_2} \right\rceil = 1$$

The response time must be calculated for each of the $Q_2$ instances:

$$w_2^0(0) = B_2 + 0C_2 = 1ms$$
$$w_2^1(0) = B_2 + 0C_2 + \left\lceil \frac{w_2^0 + J_1 + t_{bit}}{T_1} \right\rceil C_1 = 2ms$$

$$w_2^2(0) = B_2 + 0C_2 + \left\lceil \frac{w_2^1 + J_1 + t_{bit}}{T_1} \right\rceil C_1 = 2ms$$

$$R_2(0) = J_2 + w_2(0) - 0T_2 + C_2 = 5ms$$

$$R_2 = \max_{q=0..Q_2-1}(R_i(q)) = R_2(0) = 5ms$$

**Response time $R_1$:**

$$B_1 = 1ms$$
$$t_1^0 = C_2 = 1ms$$
$$t_1^1 = B_1 + \left\lceil \frac{t_1^0 + J_1}{T_1} \right\rceil C_1 = 2ms$$
$$t_1^2 = B_1 + \left\lceil \frac{t_1^1 + J_1}{T_1} \right\rceil C_1 = 2ms$$
$$t_1 = 2ms$$

Then the number of instances $Q_1$ that become ready:

$$Q_1 = \left\lceil \frac{t_1 + J_1}{T_1} \right\rceil = 1$$

The response time must be calculated for each of the Q1 instances:

$$w_1(0) = B_1 + 0C_1 = 1ms$$
$$R_1(0) = J_1 + w_1(0) - 0T_1 + C_1 = 3ms$$

**Q2.8:**

| Message | period (ms) | Jitter |
|---|---|---|
| m1 (high priority) | 30 | 5 |

| | | |
|---|---|---|
| m2 (middle priority) | 15 | 0 |
| m3 (low priority) | 5 | 0 |

In this case only the response time for the lowest priority message is required, i.e. m3.
$\tau_{bit}$ is considered to be smaller than 1ms

**Response time R3:**

$B_3 = 0ms$, since m3 is the lowest priority process.
First the length of the busy period for message 1 is calculated:

$$t_3^0 = C_3 = 1ms$$

$$t_3^1 = B_3 + \left\lceil \frac{t_3^0 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_3^0 + J_2}{T_2} \right\rceil C_2 + \left\lceil \frac{t_3^0 + J_3}{T_3} \right\rceil C_3 = 0 + \left\lceil \frac{1+5}{30} \right\rceil 1 + \left\lceil \frac{1+0}{15} \right\rceil 1 + \left\lceil \frac{1+0}{5} \right\rceil 1 = 3ms$$

$$t_3^1 = B_3 + \left\lceil \frac{t_3^1 + J_1}{T_1} \right\rceil C_1 + \left\lceil \frac{t_3^1 + J_2}{T_2} \right\rceil C_2 + \left\lceil \frac{t_3^1 + J_3}{T_3} \right\rceil C_3 = 0 + \left\lceil \frac{4+5}{30} \right\rceil 1 + \left\lceil \frac{4+0}{15} \right\rceil 1 + \left\lceil \frac{4+5}{5} \right\rceil 1 = 3ms$$

$$t_3 = t_3^2 = t_3^1 = 3ms$$

Then the number of instances Q3 of message 3 that become ready before the end of the busy period:

$$Q_3 = \left\lceil \frac{t_3 + J_3}{T_3} \right\rceil = \left\lceil \frac{3+0}{5} \right\rceil = 1$$

The response time must be calculated for each of the Q3 instances, in this case just one.

$$w_3^0(0) = B_3 + 0C_3 = 0ms$$

$$w_3^1(0) = B_3 + 0C_3 + \left\lceil \frac{w_3^0 + J_1 + \tau_{bit}}{T_1} \right\rceil C_1 + \left\lceil \frac{w_3^0 + J_2 + \tau_{bit}}{T_2} \right\rceil C_2 = 0 + 0 + \left\lceil \frac{0+5+\tau_{bit}}{30} \right\rceil 1 + \left\lceil \frac{0+0+\tau_{bit}}{15} \right\rceil 1 = 2ms$$

$$w_3^2(0) = B_3 + 0C_3 + \left\lceil \frac{w_3^1 + J_1 + \tau_{bit}}{T_1} \right\rceil C_1 + \left\lceil \frac{w_3^1 + J_2 + \tau_{bit}}{T_2} \right\rceil C_2 = 0 + 0 + \left\lceil \frac{2+5+\tau_{bit}}{30} \right\rceil 1 + \left\lceil \frac{2+0+\tau_{bit}}{15} \right\rceil 1 = 2ms$$

$$w_3^2(0) = w_3^1(0) = 2ms \rightarrow w_3(0) = 2ms$$

$$R_3(0) = J_3 + w_3(0) - 0T_3 + C_3 = 0 + 2 - 0 + 1 = 3ms$$

$$R_3 = \max_{q=0..Q_3-1}(R_i(q)) = R_3(0) = 3ms$$

**Q2.9:**

a) The priorities for messages in a CAN network must be the same fixed priorities used in the priority-based scheduling of the processes in the nodes that are connected to the bus.

**False**. The CAN network is completely independent of the processes that run in each node that it is connected to the bus, therefore the priorities can be different

b) If one node that is connected to a TTP bus crashes, this can be detected easier by the other nodes than if the system would use a CAN bus.

**True**. TTP /C has a membership service that is in charge of monitoring the “health” of the nodes, excluding them in case of failure.

c) If a process exceeds its assumed worst case execution time (WCET) at some point in time, it is stopped from sending its final output on a TTP buss.

**True**. If the process exceeds the WCET and misses its assigned slot to transmit, it cannot send the final output at that time.

d) On a CAN bus a high priority message can only be delayed (blocked) once by messages of lower priority.

**True**. A high priority message can only be delayed by a lower priority messages that is already being transmitted, since no preemption exists in the CAN bus.