Distributed systems I
Winter Term 2019/20

30/40

G2T1 – Assignment 3 (theoretical part)

Felix Bühler 2973410

Clemens Lieb 3130838

Steffen Wonner 2862123

Fabian Bühler 2953320

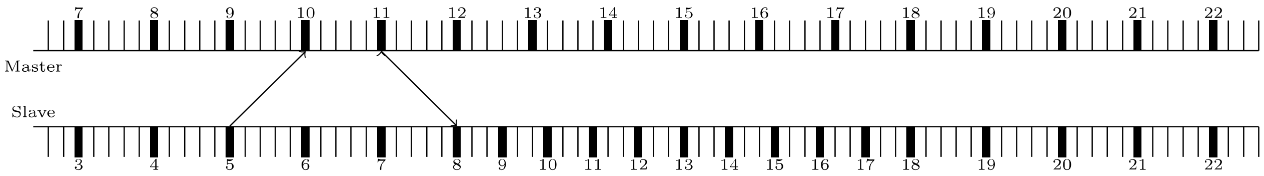
December 1, 2019

1 - Physical Clocks

4/14

a)

2/8



0/1.5

Figure 1: Solution for Figure 1

$$O = \frac{1}{2}((t_2 - t_1) - (t_4 - t_3)) = \frac{1}{2}((10-5) - (8-11)) = \frac{1}{2}(5+3) = 4$$

0.5

$$d = 1$$

0.5

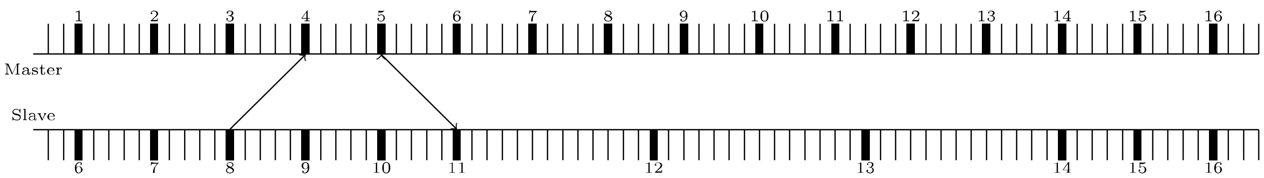
0

$$\Delta = 6, C_s(t_4 + \Delta) = 18$$

0

$$\frac{dC_S}{dt} = \frac{6+4}{6} = \frac{5}{3}$$

0



0/1.5

Figure 2: Solution for Figure 2

$$O = \frac{1}{2}((t_2 - t_1) - (t_4 - t_3)) = \frac{1}{2}((4-8) - (11-5)) = \frac{1}{2}(-4-6) = -5 \quad \boxed{0.5}$$

$$d = 1 \quad \boxed{0.5}$$

$$\boxed{0} \quad \Delta = 8, C_s(t_4 + \Delta) = 14 \quad \boxed{0}$$

$$\frac{dC_S}{dt} = \frac{8-5}{8} = \frac{3}{8} \quad \boxed{0}$$

b) 2/6

(i) We can use the difference in offsets to determine the clockrate of the master node in relation to the subservient node. For that purpose we calculate $O_1 = \frac{1}{2}((t_6 - t_5) - (t_8 - t_7))$ and $O_0 = \frac{1}{2}((t_2 - t_1) - (t_4 - t_3))$.

It's clear that $\frac{O_1 - O_0}{t_6 - t_2}$ is the rate at which the clocks are desynchronizing in reference to the master clock. If $O_1 - O_0$ is negative, the subservient clock is faster than the master, and vice versa. This means that for negative $O_1 - O_0$, the frequency ratio is $f = \frac{t_6 - t_2}{O_0 - O_1}$. For a faster master clock the frequency ratio is the desynchronization rate from before. 0

(ii) It makes no sense to synchronize the clock rates before the clock values, because to enable a "stable" value synchronization – i.e. monotonically increasing without large gaps in tick-time or clockvalues – the clock frequency needs to be adjusted again. As such, the synchronized clock frequency would need to be changed again to synchronize clock values, making it an unnecessary step. 0.5

(iii) Synchronization of the depicted clocks.

$$O_0 = \frac{1}{2}((5-6) - (9-6)) = 1$$

$$O_1 = \frac{1}{2}((8-11) - (14-9)) = -4 \quad \boxed{0.5}$$

$$d = \frac{1}{2}((8-11) + (14-9)) = 1 \quad \boxed{0.5}$$

$$f = \frac{(8-5)}{1-(-4)} = \frac{3}{5} \quad \boxed{0.5}$$

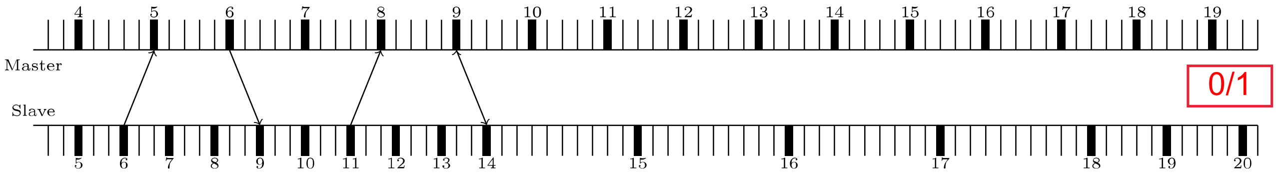
0 0

We choose $\Delta = 8, C_s(t_8 + \Delta) = 18$ modifying the clock frequency of C_s to $f * \frac{\Delta + O_1}{\Delta}$ of the original frequency. When $t_8 + \Delta$, we again modify the frequency by $\frac{\Delta}{\Delta - O_1}$ to return to the synchronized clock freqency.

This means $\frac{dC_s}{dt} = f * \frac{8-4}{8} = f * \frac{1}{2} = \frac{3}{10}$ 0

(iv) Completing the figure.

Note that the delay is not used in the timing schema proposed in (i). It has been included in the calculations for (iii) for completeness' sake.



0/1

Figure 3: Solution 1b)(iv)

10/10 2 - Logical Clocks

a)

(i) $C(e_1^1) = 2$
$C(e_1^2) = 3$
$C(e_1^3) = 4$
$C(e_1^4) = 5$

$C(e_2^1) = 5$
$C(e_2^2) = 6$
$C(e_2^3) = 7$
$C(e_2^4) = 8$

$C(e_3^1) = 1$
$C(e_3^2) = 2$
$C(e_3^3) = 3$
$C(e_3^4) = 9$ 1/1

(ii) $VC(e_1^1) = [1\ 0\ 1]$
$VC(e_1^2) = [2\ 0\ 1]$
$VC(e_1^3) = [3\ 0\ 1]$
$VC(e_1^4) = [4\ 0\ 1]$

$VC(e_2^1) = [3\ 1\ 1]$
$VC(e_2^2) = [3\ 2\ 2]$
$VC(e_2^3) = [3\ 3\ 2]$
$VC(e_2^4) = [3\ 4\ 2]$

$VC(e_3^1) = [0\ 0\ 1]$
$VC(e_3^2) = [0\ 0\ 2]$
$VC(e_3^3) = [0\ 0\ 3]$
$VC(e_3^4) = [3\ 4\ 4]$ 3/3

(iii) P_1: $e_1^1\ e_1^2\ e_1^3$
P_2: $e_2^1\ e_2^2$
P_3: $e_3^1\ e_3^2$

The vector event time of those events is smaller than the vector event time of e_2^3. A smaller vector time means events are causally related. 2/2

b)

(i) P_1: e_6 e_5 e_9 e_4 e_8 e_{10}
P_2: e_{12} e_3 e_{13}
P_3: e_{11} e_7 e_1 e_2 2/2

(ii) Local event: e_5 e_9 e_3 e_1 e_{11}
Send event: e_6 e_4 e_{10} e_{12}
Recieve event: e_8 e_{13} e_7 e_2 2/2

8/8

3 - Global State

a)

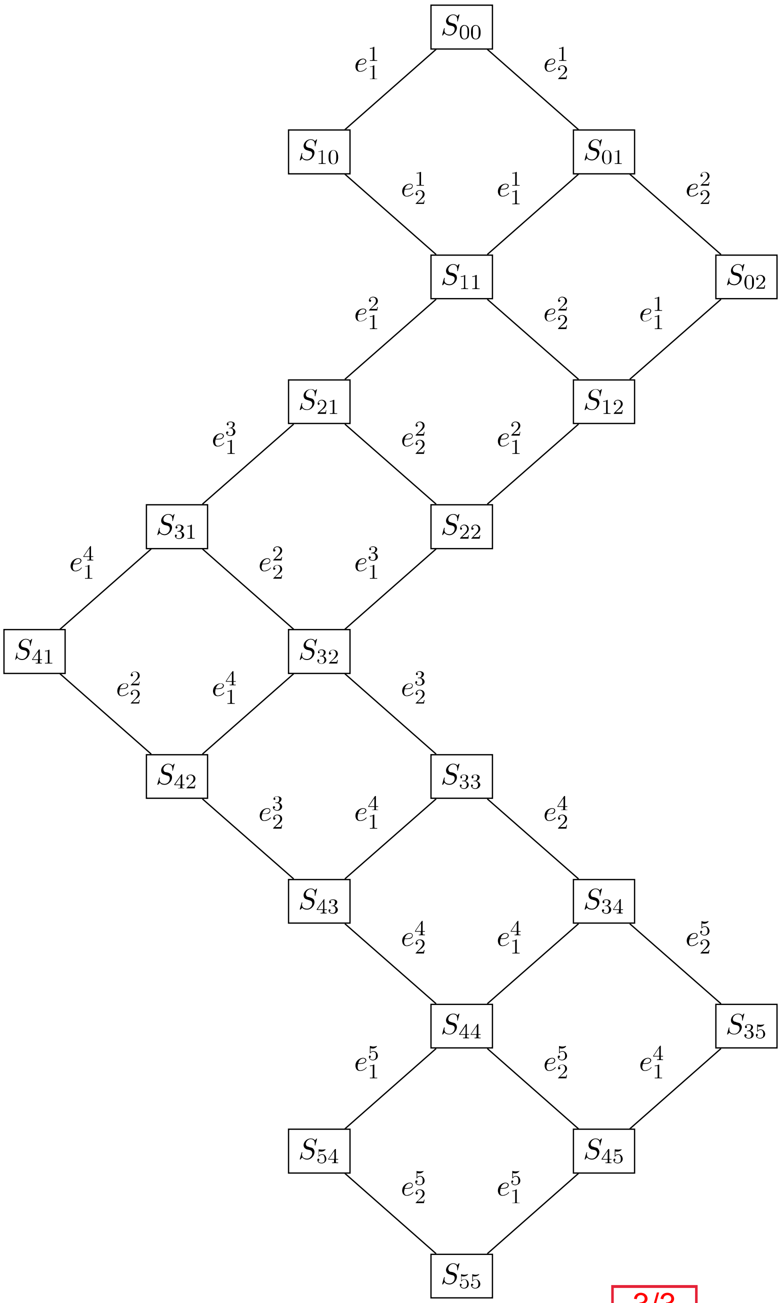
$\{(e_1^1, e_2^1), (e_1^2, e_2^2), (e_2^2, e_1^3), (e_1^4, e_2^3), (e_1^4, e_2^4), (e_1^5, e_2^5), (e_1^1, e_2^2), (e_2^2, e_1^4), (e_1^4, e_2^5)\}$ 3/3

b)

1. yes, as it is a valid linearization.
2. no, because e_2^3 can not happen before e_1^3 2/2

c)

Felix Bühler
Clemens Lieb
Steffen Wonner
Fabian Bühler



3/3

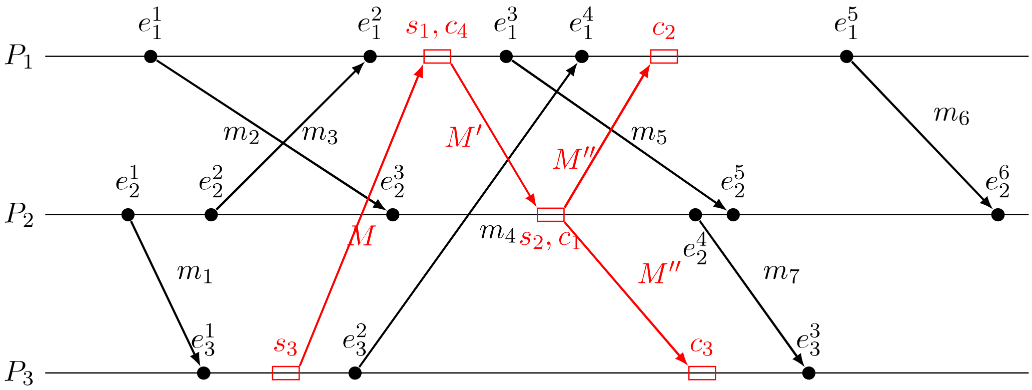
Figure 4: solution for 3.c)

8/8

4 - Snapshot Algorithm

a)

See Figure 5 6/6



The red boxes represent where a process saves some state. A process state save is denoted by s_i and a channel state save by c_j where i is the process number and j is the channel number.

Figure 5: solution for 4.a)

b)

No channel received a message between the process starting recording and the first/second marker arriving. All channel states are empty sets.

$$c_1 = \{\}$$
$$c_2 = \{\}$$
$$c_3 = \{\}$$
$$c_4 = \{\}$$

2/2