# Distributed System I Wintersemester2020/21 Assignment 1

*Ciheng Zhang (3472321) zch3183505@gmail.com*
*Chenxi Li(3502796) cli216@outlook.com*
*Leqi Xu(3556962) st176119@stud.uni-stuttgart.de*
*Yaosheng Zheng (3563285) zhengyaosheng312@icloud.com*
*Team 19*

Total: 22/36

November 22, 2020

Distributed System I
Wintersemester2020/21
Assignment 1

Ciheng Zhang (3472321)
Chenxi Li(3502796)
Yaosheng Zheng (3563285)
Leqi Xu(3556962)

## Transparency Levels 8/12

### a) 1,5/3

Concurrency transparency: in this question we use http protocol. HTTP allow Allow multiple people to access at the same time. 0,5 yes, but why is it transparent?

Replication transparency: here users never know how many copies exist in the server. 0 This is not about the copies in the server but rather about the number of servers that host this file.

Scaling transparency: The server can extend the storage of the server but dont need to change the structure. 1

### b) 4,5/5

Concurrency transparency, Replication transparency, Scaling transparency. The reason is same as the question a because of http protocol. 2,5 (not the same for replication transparency)

Location transparency: In the question we use domian not IP address. we need to use DNS to got the real address of the server.So we actually dont know where its the server. 1

Migration transparency: we use DNS to locate. So if we change our real location of our server, there will be no affect to application. 1

### c) 2/4

We use write all and read one methode. 0 locking all copies locks more than only a majority of the copies
Write operation: the protocol lock all copys. Then the client write the change to all copys. Then all lock release.
Read operation: At first lock all copys, and then the client read only one copy. after reading the locks release.
In this protocal the changes will be write in all the copys. And when the client is reading one copy.the server is locked no one can change the copys. so it ensure delivers the most recent copy. 2

## System Models

### a) 6/12

2/2 In this question, The necessary assumptions are: there is no Failure in the system(No communication Failure and no Process Failure). This mean if a requset is sent, after a period of time P2 always will recived it.

Ciheng Zhang (3472321)
Chenxi Li(3502796)
Yaosheng Zheng (3563285)
Leqi Xu(3556962)

2/3
drift rate isn't relevant
+ no communication and process failures

### b)

The transmission delay and excute time is always less than a $t_{max}$ also the drift of reciver and sender should be same.

0/3
Sorry, but I don't understand your answer. You would need the following assumptions to guarantee the property:
• No process failures
• No communication failures
• Upper bound on message delay
• Upper bound on clock drift

### c)

Both P1 and P2 w.r.t a real time, and after the requset can be reached anyway. Justify: a requset is send and the transmission need some time. So $t_{P2}^{recive}(m_{req})$ and $t_{P1}^{send}(m_{req})$ must have a time delay. and we want to compare this two time. We need to make the tow local clock with reference with a same global time.

2/4
+ no process failures
+ no communication failures
+ communication channel must be FIFO

### d)

The P2 process after process send responses in order recived, that can make $t_{P1}^{recive}(m_{res1}) < t_{P1}^{recive}(m_{res2})$ and the P2 send responses of first msg after send second msg from P1. So $t_{P1}^{send}(m2) - t_{P1}^{send}(m1) < t_{delay} + t_{excution} + t_{drift}$.This mean the time between P1 send first message and second message should less than the send delay time and excution time in P2 and drift time of P2.

clock drift doesn't matter here, because we are just looking at the local time of P1.

## Three-Army-Problem 4/6

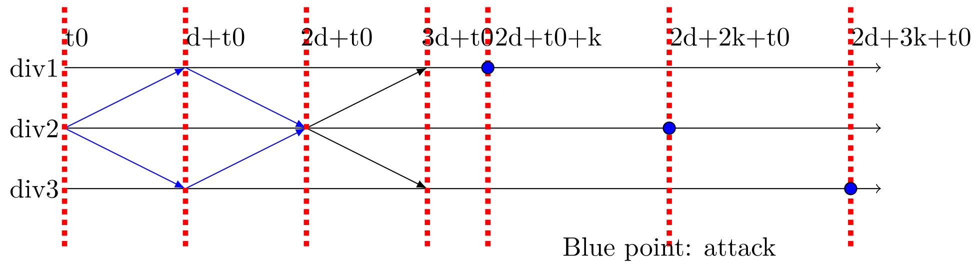
1/2
+ no process failures (death of a leader)

### a)

The assumptions is there should be no Failure in communication, this mean all the message can always send to the reciver. also the delay time between send and recive should not be too long($d > k$). It could still work if d > k
Three army in the same time send the number of division. And then the largest send message to other 2 divisions about the attack time. Then at the attack time the largest division attack, after k mins the second-largest attack. And k mins later the smallest division attack. This is task b)

### b)

3/4 you could improve the duration of phase 1 from 3 message delays to 2 message delays by using broadcasts. So the first attack could be sooner.



in this diagram. we assum the largest division is div1 and second-largest is div 2 and smallest is div3.At time t0 div2 want to attack, and send message to div1 and div2. Then at time d div1 and div2 got the message and send back message about their size to div2. At 2d time div2 know all three divisions size, and send

Ciheng Zhang (3472321)
Chenxi Li(3502796)
Yaosheng Zheng (3563285)
Leqi Xu(3556962)

to div1 and div3 message about attack time. At 2d+t0+k time the bigest div1 start attack. After k time div2 and then div3.

## System Availablity 4/6

1

### a)

$$A_X = \frac{60}{100} = 0.6$$

$$A_Y = \frac{80}{100} = 0.8$$

1

### b)

$$A_{DS} = 1 - \bar{A_X}\bar{A_Y} = 0.92$$

1/2 What about the independency?

### c)

$$P(X = up|Y = up) = \frac{60}{80} = 0.75$$

$$P(X = up|Y = down) = 0$$

### d)

1/2 Comparison to question 4 b)?

accroding question c. if Y fail X must be fail:

$$P = 1 - 0.2 = 0.8$$