ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Autumn Term 2020 Prof. R. Wattenhofer

Distributed Systems
Exam

Thursday, 28th of January 2021, 14:30 - 16:00

Do not open or turn before the exam starts!
Read the following instructions!

The exam takes 90 minutes and there is a total of 90 points. The maximum number of points for each subtask is indicated in brackets. **Justify all your answers** unless the task explicitly states otherwise. Mark drawings precisely.

Answers which we cannot read are not awarded any points!

At the beginning, fill in your name and student number in the corresponding fields below. You should fill in your answers in the spaces provided on the exam. If you need more space, we will provide extra paper for this. Please label each extra sheet with your name and student number.

Family Name	**First Name**	**Student Number**

Task	**Achieved Points**	**Maximum Points**
1 - Choose the Correct Answer		24
2 - Asynchronous Byzantine Agreement		16
3 - Tree Quorum Systems		17
4 - Relationship Game		18
5 - Localization		15
Total		**90**

1 Choose the Correct Answer (24 points)

Each question has **exactly one correct** answer. Indicate your answer by checking the corresponding circle. A correct answer gives **2 points**, a wrong answer gives **0 points**, and no answer gives **0 points**. You do **not** need to explain your answers.

a) In Paxos, if some servers store a command:

- ○ The command will be executed by those servers.
- ○ The command will be executed by all servers.
- ○ The command might be executed by all servers.
- ○ The command might be executed by some servers.

b) What is the minimum number of byzantine clients f that will cause Paxos to fail?

- ○ $f = \lfloor \frac{n}{3} \rfloor$
- ○ $f = \lfloor \frac{n}{3} + 1 \rfloor$
- ○ $f = 1$
- ○ None of the above.

c) A consensus algorithm starting with input values of [0, 0, 0, 1, 0] terminated with node values of [1, 1, 1, 1, 1]:

- ○ Violated the agreement property.
- ○ Violated the validity property.
- ○ Violated the termination property.
- ○ Did not violate any consensus properties.

d) If in the first phase of the King algorithm the first king is honest, then:

- ○ Every honest node will decide for the initial input of the king.
- ○ An honest node will propose a value in the first phase.
- ○ The king himself can change its own value in the first phase.
- ○ The nodes might change the value in future phases if there are other honest kings later.

e) In a shared coin protocol:

- ○ It is impossible for some of the honest nodes to output 0 and for some honest nodes to output 1.
- ○ It is required that with constant probability all honest nodes output 0, but not necessarily for 1.
- ○ It is required that with constant probability all honest nodes output 1, but not necessarily for 0.
- ○ None of the above.

f) In best-effort broadcast it is impossible for two honest nodes to:

- ○ Accept different values if the sender is honest.
- ○ Accept different values if the sender is dishonest.
- ○ Accept the same value if the sender is honest.
- ○ Accept the same value if the sender is dishonest.

g) In reliable broadcast:

- ○ All honest nodes accept at most one message.
- ○ All honest nodes accept at least one message.
- ○ All honest nodes echo at least one message.
- ○ It might happen that no honest node accepts a message.

h) Which of the following is true?

- ○ Bitcoin is more efficient than PBFT (in communication complexity).
- ○ In a synchronous network, PBFT satisfies termination and agreement, while Bitcoin satisfies termination and probabilistic agreement.
- ○ In an asynchronous network, Bitcoin cannot guarantee the security of transactions (safety), while PBFT cannot guarantee progress (liveness).
- ○ All of the above.

i) In practice, a Bitcoin client executes a transaction tx when 6 blocks are built on top of the block that contains the transaction tx. Assuming the network is partitioned, Bitcoin satisfies:

- ○ Eventual consistency.
- ○ Availability.
- ○ All of the above.
- ○ None of the above.

j) As mentioned in the script, Bitcoin's proof-of-work function is given by

$$\mathcal{F}_d(c, x) \to \text{SHA256}(\text{SHA256}(c|x)) < \frac{2^{224}}{d},$$

where both challenge c and nonce x are bit strings, and difficulty d is a positive number. What is the maximum probability that a random attempt at the proof-of-work function is successful (at any difficulty)?

- ○ 1
- ○ $\frac{1}{21{,}000{,}000}$
- ○ 2^{-32}
- ○ 2^{-224}

k) For the same dimension d, a $(2, d)$-mesh $M(2, d)$, which has 2 nodes per dimension, has exactly the same number of edges as:

- ○ The $(2, d)$-torus $T(2, d)$.
- ○ The d-dimensional butterfly $BF(d)$.
- ○ All of the above.
- ○ None of the above.

l) Which statement about the NTP algorithm is true?

- ○ NTP becomes more accurate if more nodes participate.
- ○ NTP becomes less accurate if the network delays in both directions are not equal.
- ○ NTP uses UDP because packets could be lost with TCP.
- ○ NTP is not possible over wireless connections.

Solutions

a) (3). Only the command that is stored by the majority of servers will eventually be stored by all servers and executed. If a command is stored just on some servers it might or might not be picked for execution.

b) (3). In Phase 2 or 3 the sole byzantine client could send a different command c to every server.

c) (4). While the algorithm chose the minority value it satisfied all three consensus properties: agreement, termination and validity. Validity property only requires that the chosen value was one of the input values.

d) (3). If all other parties have input different from the king, one can think of as the case where the king is dishonest and other parties have pre-agreement. Thus they will propose this different value and the king himself will change his value.

where the bitstring is hardcoded in the protocol.

e) (4). A shared coin is a binary random variable shared among all nodes. It is 0 for all nodes with constant probability and 1 for all nodes with constant probability. The shared coin is allowed to fail (be 0 for some nodes and 1 for other nodes) with constant probability.

f) (1). Best-effort broadcast ensures that a message that is sent from a correct node u to another correct node v will eventually be received and accepted by v.

g) (4). In reliable broadcast, if the sender is dishonest, the honest nodes may accept none, one or multiple messages. On the one hand, if the sender does not send any messages, the honest nodes will wait forever and never accept a message. On the other hand, if the sender sends multiple messages, they will be echoed and accepted by all nodes.

h) (4). Reason: (1) Bitcoin requires only linear to the number of parties communication complexity because the miner winning the hash race simply transmits the block in the network, while PBFT requires at least quadratic communication complexity. (2) PBFT satisfies all BA properties: agreement, termination and validity. Bitcoin on the other hand satisfies termination because transactions are executed eventually, but the agreement is probabilistic since the chain may be reverted by byzantine nodes. (3) In asynchrony, Bitcoin cannot guarantee consistency hence loses safety, while BTC-PBFT cannot make progress thus loses liveness.

i) (2). In Bitcoin nodes keep producing blocks when the network is partitioned, so availability is satisfied. However, if the network is partitioned for long enough – more than 6 blocks are produced in the partitions – then consistency is not satisfied as the transactions are already executed.

j) (3). The SHA256 hash space is in the in $\{0, \ldots, 2^{256} - 1\}$. The minimum possible difficulty is 1. So, the right hand side of the PoW equation is 2^{224}. Any number in the SHA256 hash space being less than 2^{224} is $2^{-(256-224)} = 2^{-32}$.

k) (1). If $m = 2$ (m, d)-torus and (m, d)-mesh are d-dimensional hypercubes and have exactly the same structure. While the d-dimensional butterfly is said to effectively be an unrolled hypercube it does have $2^{n+1}n$ edges instead of the $2^{n-1}n$ edges the hypercube has.

l) (2). The calculation of the propagation delay assumes that both are equal.

2 Asynchronous Byzantine Agreement (16 points)

Let's consider the following protocol for solving asynchronous byzantine agreement.

Algorithm 1 Randomized Byzantine Agreement

1: $v_i \in \{0, 1\}$ input bit
2: round = 1
3: **while** true **do**
4: Broadcast `myValue`(v_i, round)

Propose

5: Wait until $n - f$ of `myValue` messages of current round arrived
6: **if** all received messages contain the same value v **then**
7: Broadcast `propose`(v, round)
8: **else**
9: Broadcast `propose`($\perp$, round)
10: **end if**

Update

11: Wait until $n - f$ of `propose` messages of current round arrived
12: **if** all received messages propose the same value v **then**
13: Broadcast `myValue`(v, round + 1)
14: Broadcast `propose`(v, round + 1)
15: Update $v_i = v$, decide for v and terminate
16: **else if** there are at least $f + 1$ proposals for v **then**
17: Update $v_i = v$
18: **else**
19: Update v_i randomly, with $Pr[v_i = 0] = Pr[v_i = 1] = 1/2$
20: **end if**
21: round = round + 1
22: **end while**

Let's assume four nodes P_1, P_2, P_3 and P_4 execute the above protocol, where P_4 is byzantine, i.e., we have $n = 4$ and $f = 1$.

a) `[9 points]` Continue the execution of the protocol in the following table where the nodes start with the same input 0 in such a way that they finally decide for 1, i.e., the protocol doesn't achieve all-same validity.

In the column "P_4 strategy", write down which values P_4 sends to each node. In the third column of the table, in each *propose* round write down three values each node *receives* and which value they *propose*. Similarly, in each *update* round write down three values that the nodes receive, the value and in which line (15, 17, or 19) they update v_i. Just fill in the table, no need for additional explanations.

Hint. You can assume the local coins to be in favor of the adversary.

Round	P_4 strategy	P_1	P_2	P_3
			broadcasts	
Initial Setup	sends 1 to all nodes	0	0	0
	P_4		*receives*	
Propose	proposes ___ to P_1	(___, ___, ___)	(___, ___, ___)	(___, ___, ___)
Round	proposes ___ to P_2		*proposes*	
	proposes ___ to P_3	___	___	___
			receives	
	P_4	(___, ___, ___)	(___, ___, ___)	(___, ___, ___)
Update	sends ___ to P_1		*updates v_i (and decides for)*	
Round	sends ___ to P_2	___	___	___
	sends ___ to P_3		*in line*	
		___	___	___
	P_4		*receives*	
Propose	proposes ___ to P_1	(___, ___, ___)	(___, ___, ___)	(___, ___, ___)
Round	proposes ___ to P_2		*proposes*	
	proposes ___ to P_3	___	___	___
			receives	
		(1,1,1)	(1,1,1)	(1,1,1)
Update	-		*updates v_i (and decides for)*	
Round		1	1	1
			in line	
		15	15	15

b) `[7 points]` Continue the execution of the protocol in the following table where two honest nodes decide for different values, i.e., the protocol doesn't achieve agreement.

Round	P_4 strategy	P_1	P_2	P_3
			broadcasts	
Initial Setup	sends 1 to all nodes	0	0	0
	P_4	___	*receives*	
Propose	proposes ___ to P_1	(___, ___, ___)	(___, ___, ___)	(___, ___, ___)
Round	proposes ___ to P_2		*proposes*	
	proposes ___ to P_3	___	___	___
			receives	
	P_4	(___, ___, ___)	(___, ___, ___)	(___, ___, ___)
Update	sends ___ to P_1		*updates v_i (and decides for)*	
Round	sends ___ to P_2	___	___	___
	sends ___ to P_3		*in line*	
		___	___	___
	P_4		*receives*	
Propose	proposes ___ to P_1	(___, ___, ___)	(___, ___, ___)	(___, ___, ___)
Round	proposes ___ to P_2		*proposes*	
	proposes ___ to P_3	___	___	___
			receives	
		(___, ___, ___)	(___, ___, ___)	(___, ___, ___)
Update	-		*updates v_i (and decides for)*	
Round		___	___	___
			in line	
		___	___	___

a)

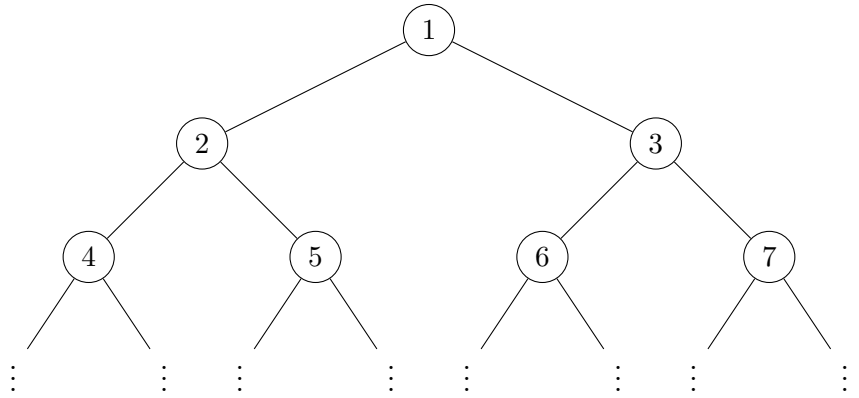
Round	P_4 strategy	P_1	P_2	P_3
			broadcasts	
Initial Setup	sends 1 to all nodes	0	0	0
	P_4		*receives*	
Propose	proposes $\perp$ to P_1	(0,0,1)	(0,0,1)	(0,0,1)
Round	proposes $\perp$ to P_2		*proposes*	
	proposes $\perp$ to P_3	$\perp$	$\perp$	$\perp$
			receives	
	P_4	($\perp$, $\perp$,$\perp$)	($\perp$,$\perp$,$\perp$)	(0,$\perp$,$\perp$)
Update	sends 1 to P_1		*updates v_i (and decides for)*	
Round	sends 1 to P_2	1	1	1
	sends 1 to P_3		*in line*	
		19	19	19
	P_4		*receives*	
Propose	proposes 1 to P_1	(1,1,1)	(1,1,1)	(1,1,1)
Round	proposes 1 to P_1		*proposes*	
	proposes 1 to P_1	1	1	1
			receives	
		(1,1,1)	(1,1,1)	(1,1,1)
Update	-		*updates v_i (and decides for)*	
Round		1	1	1
			in line	
		15	15	15

b)

Round	P_4 strategy	P_1	P_2	P_3
			broadcasts	
Initial Setup	sends 1 to all nodes	0	0	0
	P_4		*receives*	
Propose	proposes 0 to P_1	(0,0,0)	(0,0,0)	(0,0,1)
Round	proposes $\perp$ to P_2		*proposes*	
	proposes $\perp$ to P_3	0	0	$\perp$
			receives	
	P_4	(0,0,0)	(0,$\perp$,$\perp$)	(0,$\perp$,$\perp$)
Update	sends 1 to P_1		*updates v_i (and decides for)*	
Round	sends 1 to P_2	0	1	1
	sends 1 to P_3		*in line*	
		15	19	19
	P_4		*receives*	
Propose	proposes 1 to P_1	-	(1,1,1)	(1,1,1)
Round	proposes 1 to P_2		*proposes*	
	proposes 1 to P_3	-	1	1
			receives	
		-	(1,1,1)	(1,1,1)
Update	-		*updates v_i (and decides for)*	
Round		-	1	1
			in line	
		-	15	15

3 Tree Quorum Systems (17 points)

In the following we study the tree quorum system, where the servers are ordered in a binary tree.



Let's assume we have $n = 2^k - 1$ servers. A quorum is chosen as follows: Start from the root of the tree (the server with number 1) and follow some path towards a leaf. For example, for $n = 7$, a possible quorum is the set $\{1, 2, 5\}$. We define a tree quorum system as a set of all possible paths from the root to some leaf.

a) [3] Is the tree quorum system valid?

b) [6] What are load, work and resilience of the tree quorum system?

c) [8] Assume that every server works with probability 2/3 (and fails with probability 1/3). What is the asymptotic failure probability of the tree quorum system?

a) Yes. All paths start from the root, so any two quorums intersect, i.e. they have the root server in common.

b) Since the root server is part of any quorum, the load of the system is 1. Furthermore, since any path from the root to a leaf has $k = \log(n+1)$ nodes, the work is $k = \log(n+1)$. Finally, the resilience of the system is 0: if the root fails, then there is no quorum without a failed node.

c) For depth 1 (only the root server), the failure probability is $f = 1 - p = 1/3$. For depth 2, if the root doesn't fail, then both the left subtree and the right subtree must fail so that the whole system fails. Hence we have $f = 1/3 + 2/3 \cdot (1/3)^2 \approx 0.407$. Hence, for $n \to \infty$, we can calculate the asymptotic failure probability f recursively as follows:

$$
\begin{aligned}
f &= 1/3 + \frac{2}{3} f^2 \\
\Leftrightarrow \frac{2}{3} f^2 - f + \frac{1}{3} &= 0 \\
2f^2 - 3f + 1 &= 0 \\
f &= \frac{3 \pm \sqrt{9 - 4 \cdot 2 \cdot 1}}{4} \\
f &= 1 \wedge f = \frac{1}{2}
\end{aligned}
$$

The series $0.333, 0.407, \ldots$ converges to $f = 1/2$.

4 Relationship Game (18 points)

Alex and Billy are dating, but Alex likes hiking while Billy likes to go to the cinema. Doing the preferred activity is enjoyable and is worth 1 happiness point to each of them ($a = 1$). As they also like spending time together, if they end up doing the same activity each gets an additional reward of 2 ($t = 2$). Both types of happiness rewards add up to $a + t$.

a) [4] Write this game in matrix form.

b) [2] Identify the pure Nash equilibiria in this game (no explanation needed).

c) [7] What are the mixed Nash equilibria?

d) [2] What is the Optimistic Price of Anarchy (OPoA)?

e) [3] Assume non-negative a and t. What should be the relationship between a and t so that each player would have a dominant strategy?

a)

Alex \ Billy	**Hiking**	**Cinema**
Hiking	3,2	1,1
Cinema	0,0	2,3

b) There are two pure Nash equilibria (Hiking, Hiking) and (Cinema, Cinema).

c) There is one mixed Nash equilibrium. Let's say Billy picks Hiking with probability p and Cinema with probability $1 - p$. For the mixed Nash equilibrium Billy has to make Alex indifferent to picking a particular action:

$$3p + 1(1 - p) = 0p + 2(1 - p)$$

$$p = \frac{1}{4}$$

For Alex, the situation is symmetric. Thus, the resulting mixed Nash equilibria are (($\frac{3}{4}$ - Hiking, $\frac{1}{4}$ - Cinema), ($\frac{1}{4}$ - Hiking, $\frac{3}{4}$ - Cinema))

d) OPoA $= \frac{NE_+}{SO} = \frac{5}{5} = 1$.

e) If $a \geq t$ always choosing the preferred activity becomes the dominant strategy for each player because the other activity always results in less points regardless of the action of the other player. (It is not expected to explain why $a \leq t$ is not a dominant strategy).

5 Localization (15 points)

We build a localization system using transmitters on the ground instead of satellites. We have three transmitters A (at coordinates $x = 0$km, $y = 0$km), B (0km, 30km) and C (60km, 0km). Each transmitter sends a strong and short radio transmission once per second at a random time to avoid collisions with other transmitters. We know that radio signals can be received very far due to signal reflections at the ionosphere. Assume messages travel with the speed of light $c = 3 * 10^8 m/s$.

a) [3] What information must the radio transmission contain to allow the self-localization of a receiver that does not know its time and location.

b) [6] Write down the system of equations to solve for the position (x, y) of a mobile handset receiving these 3 messages.

Transmitter	Measured transmission delay
A	53.10ms
B	53.14ms
C	53.10ms

c) [3] Below are two possible solutions for the equation system of b). Which solution is more plausible for a receiver?

Position (x, y)	Offset θ
(35km, 0km)	53ms
(-30km, -35km)	52.9ms

d) [3] We now deploy this system in all of Europe with a total of five transmitters. For locations far from the transmitters we experience large localization errors. Why?

a) Needs a send timestamp and location of the transmitter or some identification of the transmitter.

b) $\|(x, y)\|/c = 53.1ms - \theta$
$\|(0km, 30km) - (x, y)\|/c = 53.14ms - \theta$
$\|(60km, 0km) - (x, y)\|/c = 53.1ms - \theta$

c) The measured transmission delay is the same for A and B. The receiver has to be the same distance from both. So position (30km, 0km) with offset 53ms would be correct. (35km, 0km) is close to it.

Or the residuals for both solutions can be calculated.

d) The signal is reflected on the ionosphere. So over long distances the signal does not travel on a straight path. Therefore the transmission delay is longer than the direct path (that might even go through the ground).