Introduction to Distributed Systems
WT 19/20

Assignment 6 – Part I

Submission Deadline: Monday, 27.01.2020, 08:00

- *Submit the solution in PDF via Ilias (only one solution per group).*
- *Respect the submission guidelines (see Ilias).*

1 Replication with Atomic Multicast [17 points]

In the lecture, we have seen that Atomic Multicast can be used to ensure eventual consistency between replicated servers. In this implementation, the client proxy uses Atomic Multicast to multicast a request to the replicated server group, i.e., each client request is mapped to an Atomic Multicast. In addition, the client sends requests synchronously, i.e., before sending the next request it has to wait for the response of the previous request to arrive. This implementation is depicted in Figure 1, where the replicated server group has two replicas $R1$ and $R2$. In the figure, $C1$ and $C2$ represent two clients.

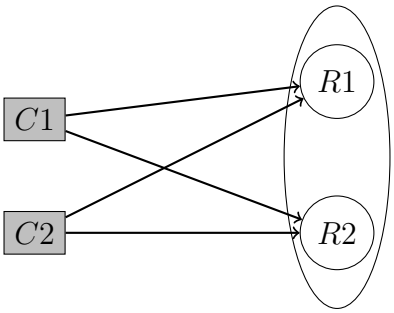
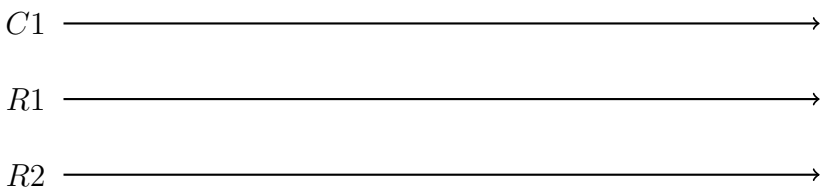
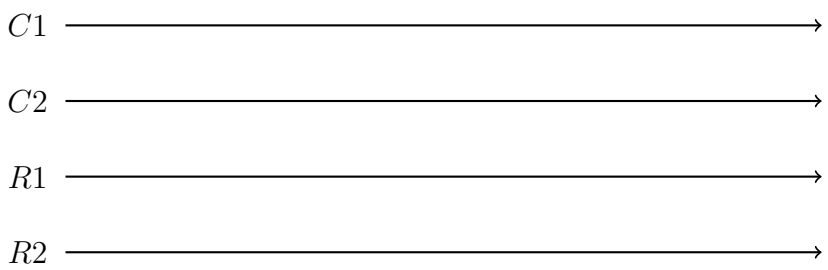


Figure 1: Replication with Atomic Multicast.

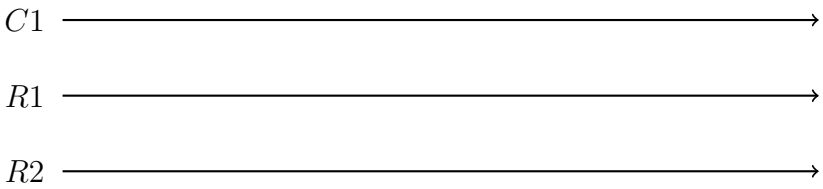
a) [5 points] Why is the *agreement property* of Atomic-Multicast essential for replication? Show an example using the following space-time diagram, where eventual consistency is violated without agreement.
State clearly the delivery of message(s) (using a dot •) and the content of message(s) between $C1$ and the two replicas $R1$ and $R2$, e.g., $M1 = read(x)$ or $M1 = write(x)$.



b) [5 points] Why is the *atomicity property* of Atomic-Multicast essential for replication? Show an example using the following space-time diagram, where eventual consistency is violated without atomicity.
State clearly the delivery of message(s) (using a dot •) and the content of message(s) between $C1$ and the two replicas $R1$ and $R2$, e.g., $M1 = read(x)$ or $M1 = write(x)$.



c) i. [5 points] What is the reason for restricting clients to call requests synchronously? Justify your answer with an example using the following space-time diagram. *State clearly the delivery of message(s) (using a dot •) and the content of message(s) between* $C1$ *and the two replicas* $R1$ *and* $R2$*, e.g.,* $M1 = read(x)$ *or* $M1 = write(x)$.



ii. [2 points] Would the usage of causal atomic Multicast enables the given implementation of atomic Multicast to use asynchronous calls?

2 Multicast Semantics [14 points]

a) [8 points] Figure 2 shows group communication between processes P_1, P_2 and P_3 belonging to the same multicast group. It is given that process P_2 is faulty. *Note*: An arrow indicates the transmission of a message. A dot indicates the delivery of a message, i.e., message m_1 will never be delivered to process P_2 while it is delivered to P_1 and P_3. Moreover, you can assume that no further events (multicast, deliver) occur before or after the depicted time frame.

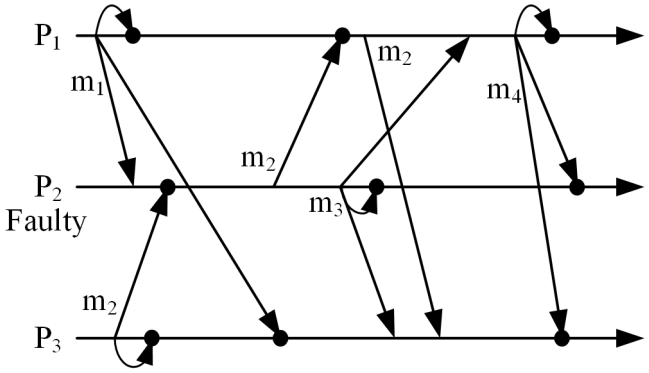


Figure 2: Group communication between processes P_1, P_2 and P_3

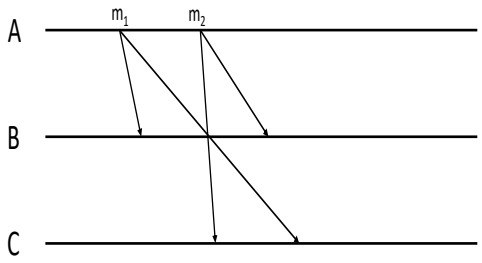
In the following, mark if the statements pertaining to this group communication are *true* or *false* and **justify your answer**.

Statement	**true**	**false**
The property of *Uniform Agreement* is satisfied.	☐	☐
All properties of *Reliable Multicast* (Validity, Agreement, Integrity) are satisfied.	☐	☐
The message deliveries follows FIFO semantics but not atomic semantics.	☐	☐
Uniform FIFO ordering is satisfied for the message deliveries.	☐	☐

b) Given are the processes A, B and C, which are member of the multicast group g. The following charts show the messages for a multicast to g. In each chart different multicast semantics for the multicast are fulfilled. Mark for each chart whether the given semantics are fulfilled or violated. You can assume that a process delivers a message immediately upon receiving it. Furthermore, a process that sends a message m to other processes also immediately delivers m to itself.

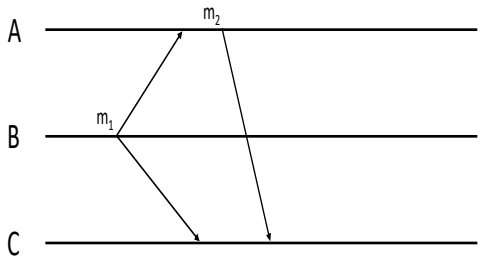
Note: Correct answer +0.5P, wrong answer –0.5P, missing answer 0P.

i. [2 points] Scenario 1:



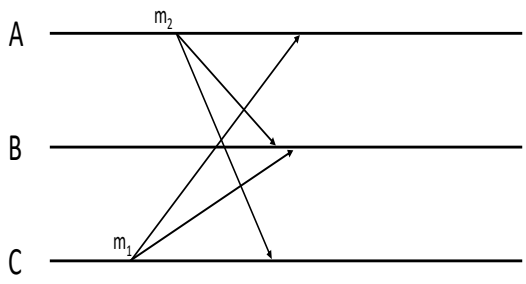
Semantics	Fulfilled/violated
Reliable Multicast	
FIFO Multicast	
Causal Multicast	
Atomic Multicast	

ii. [2 points] Scenario 2:



Semantics	Fulfilled/violated
Reliable Multicast	
FIFO Multicast	
Causal Multicast	
Atomic Multicast	

iii. [2 points] Scenario 3:



Semantics	Fulfilled/violated
Reliable Multicast	
FIFO Multicast	
Causal Multicast	
Atomic Multicast	

3 Causal Mulitcast [14 points]

a) In the lecture, we discussed the use of vector clocks (CBCAST) to achieve atomicity for reliable multicasts. Figure 3 shows message exchanges between a system of three processes P_1, P_2 and P_3. The processes begin with a vector clock of [0, 0, 0] at the start of the depicted time-frame.

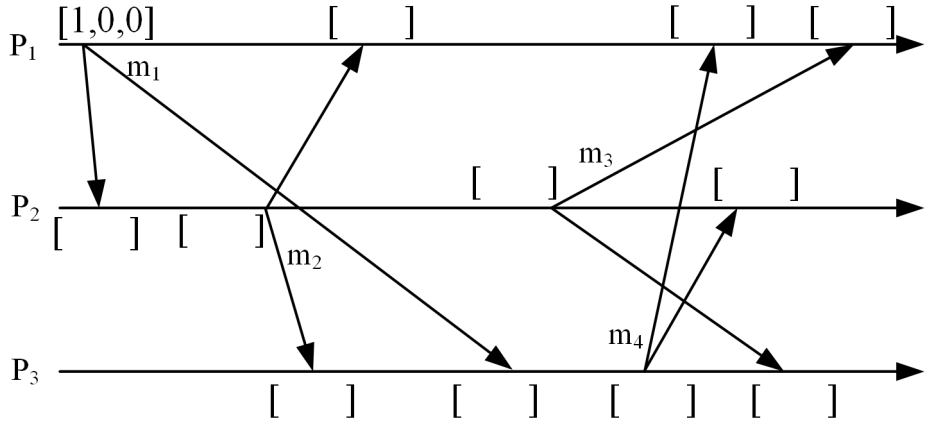


Figure 3: Execution of CBCAST Algorithm

i. [3 points] Show the evolution of vector time-stamps for each of the processes whenever they send or receive any message in the Figure 3.
Hint: Note that only multicast events are relevant.

ii. [2 points] As in CBCAST, delay the delivery of required message(s) in Figure 3 so as to achieve causal semantics. Justify the choice of your message(s) based on the vector timestamps.

b) In the lecture, we discussed a second algorithm based on F-multicast. Now, we modify this algorithm so that it is based on R-multicast instead of F-Multicast (see Algorithm 1)

i. [5 points] Figure 4 shows a system of three processes (P_1, P_2 and P_3) from a multicast group. Apply Algorithm 1 and determine what is included in the multicast messages and what sequence of messages is delivered to each of the group members. Discuss the result.

Algorithm 1 Causal Multicast Algorithm for a process P.

```
1: Initialisation:
2:    prevDlvrs_g :=⊥ ▷ /*messages that P C-delivered since its previous C-multicast for g */
3: To execute C-multicast(g, m):
4:    R-multicast(g, < prevDlvrs_g || m >)
5:    prevDlvrs_g :=⊥
6: C-deliver(m) occurs as follows:
7:    upon R-deliver(< m_1, m_2, ..., m_l >) for some /do with g = group(m)
8:    for i := 1..l do
9:       if P has not previously executed C-deliver(m_i) then
10:         C-deliver(m_i)
11:         prevDlvrs_g := prevDlvrs_g || m_i
```

Note: We have omitted the extra messages sent by R-multicast to achieve agreement.

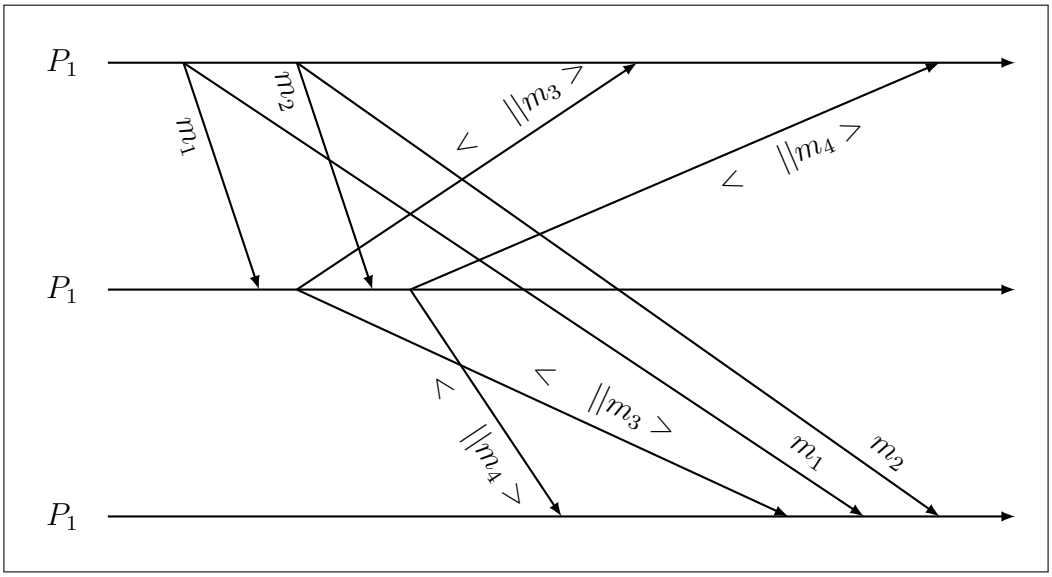


Figure 4: Execution of Algorithm 1

ii. [2 points] Explain why FIFO multicast is needed as the basis of this protocol to ensure correctness.

iii. [2 points] How would we have to modify Algorithm 1 to achieve causal delivery even if R-multicast is used as a basis.