

total:40/45

# Distributed systems I

## Winter Term 2019/20

### G2T1 – Assignment 6 (theoretical part)

Felix Bühler  
2973410

Clemens Lieb  
3130838

Steffen Wonner  
2862123

Fabian Bühler  
2953320

January 26, 2020

## 17 1 - Replication with Atomic Multicast

5 a)

As shown in figure 1 message M1 is delivered by R1 but not R2. This can happen if the sender is faulty as the validity property is only enforced for correct senders. If the agreement property is not enforced R2 may never receive the message M1. This means that x is not replicated and R2 has a stale value even though it is still correct.

R2 will however not deliver any other message since it still expects a message with the sequence number of M1 to arrive before delivering any message with a higher sequence number.

Note: C1 does not have to be faulty

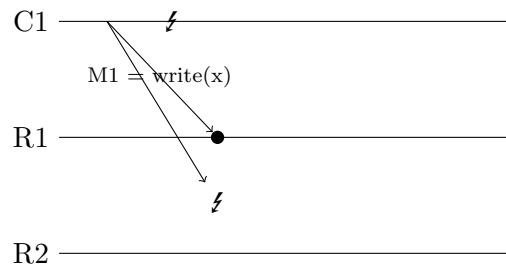


Figure 1: Atomic Multicast without Agreement

5 b)

Assuming the atomicity property is the total order property from the lecture.

As seen in figure 2 if the total order of messages is not enforced the actual order messages are delivered at each replica may be different. Now both replicas have a different value for x.

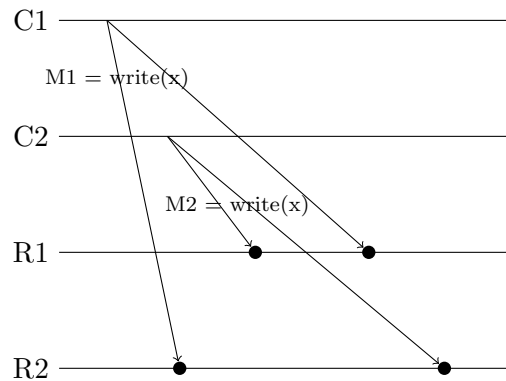


Figure 2: Atomic Multicast without Atomicity/Total Order

c)

5 i.

As seen in figure 3 if the client does not wait for an acknowledgement for its messages a newer message may get overwritten by an older message that was not yet delivered. The total order property only guaranties that the order of delivered messages is the same for all replicas.

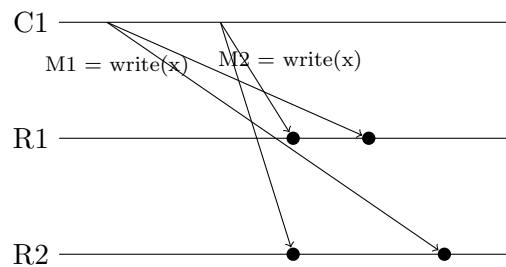


Figure 3: Atomic Multicast without synchronous clients

2 ii.

As causal atomic multicast also includes fifo ordering the wrong arrival order of messages when sending them asynchronously like in figure 3 cannot happen. Messages from the same sender will arrive in the order they were sent.

## 14 2 - Multicast Semantics

a)

i.

2 False

$m_3$  is delivered to  $P_2$  but not to  $P_1$  and  $P_3$  which are correct.

**ii.**

2 True

$m_1$ ,  $m_2$  and  $m_4$  are send by correct processes and delivered exactly once by  $P_1$  and  $P_3$

**iii.**

True

2 Only  $P_1$  sends more than 1 message. The second one is send after all processes recieved the first one. So it is FIFO multicast.

$P_1$  and  $P_2$  are delivering  $m_1$  and  $m_2$  in different orders. So no atomic multicast.

**iv.**

2

False

$P_2$  does not deliver  $m_1$  but  $m_4$ , that violates Uniform FIFO.

**b)**

2

**i.**

Reliable Multicast: fulfilled

FIFO Multicast: violated

Causal Multicast: violated

Atomic Multicast: violated

2

**ii.**

Reliable Multicast: violated

FIFO Multicast: violated

Causal Multicast: violated

Atomic Multicast: violated

2

**iii.**

Reliable Multicast: fulfilled

FIFO Multicast: fulfilled

Causal Multicast: fulfilled

Atomic Multicast: violated

9

### 3 - Causal Multicast

3 a)

i.

See figure 4

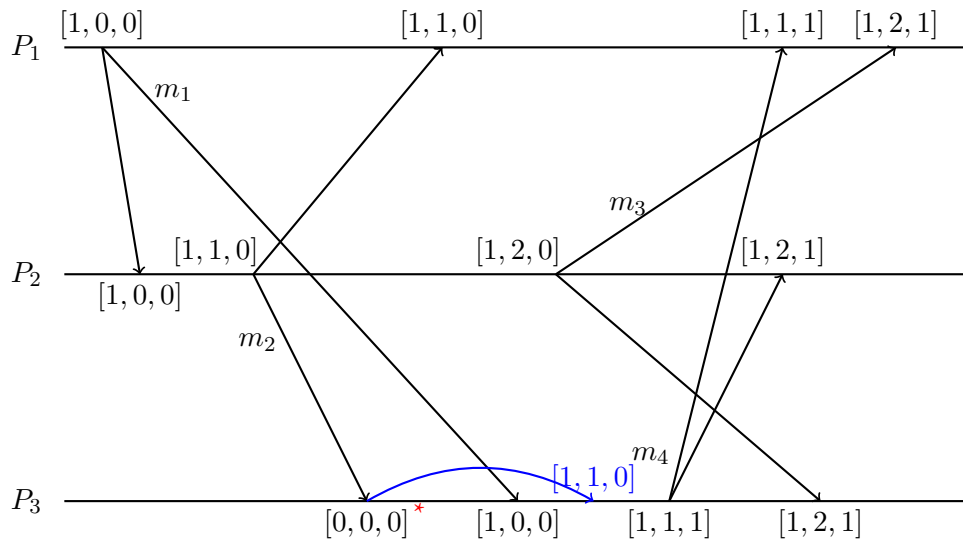


Figure 4: Execution of CBCAST Algorithm

ii.

- 2 The delivered message  $m_2$  on  $P_3$  has to be delayed. The blue vector clock would be updated to  $[1, 1, 0]$  after the received  $m_1$  message.

b)

2 i.

See figure 5 **Discuss the result missing!**  
**Delivery order missing!**

ii.

- 0 As seen in figure 5 the fifo ordering can be violated, and therefore the correctness.

iii.

- 2 Append the last sent message as causal dependency to the new message. The receiver can then guarantee fifo ordering again.

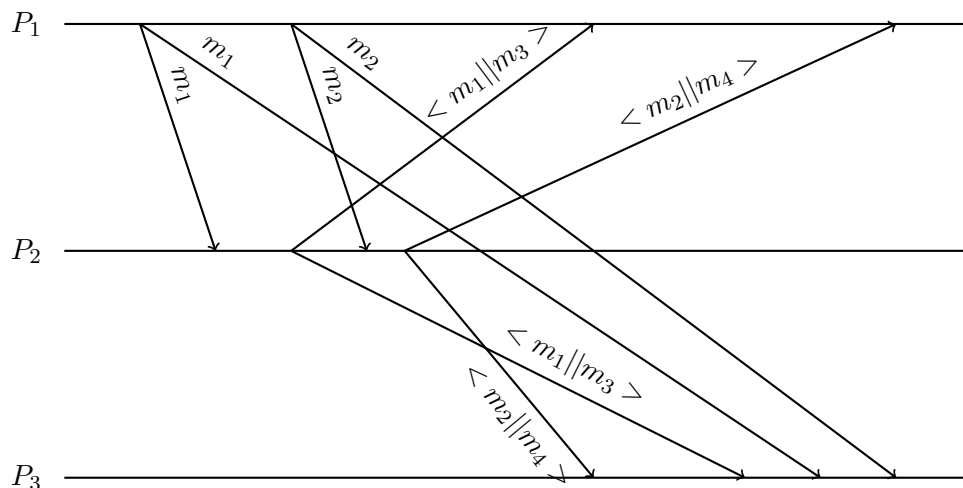


Figure 5: Execution of Algorithm 1