

## Exercise Session 3 Causal and Total Order Broadcast

### Exercise 1

Can we devise a broadcast algorithm that does not ensure the causal *delivery* property but only its nonuniform variant: no correct process  $p_i$  delivers a message  $m_2$  unless  $p_i$  has already delivered every message  $m_1$  such that  $m_1 \rightarrow m_2$ ?

### Exercise 2

Suggest an optimization of the garbage collection scheme of *Algorithm 1'* (slide 24).

### Exercise 3

Why is the condition on slide 30  $VC[pk] \geq VC_x[pk]$  and not just  $VC[pk] = VC_x[pk]$ ? Can you construct an execution where the local vector clock is greater than the received local clock for one place?

### Exercise 4

Can we devise a best-effort broadcast algorithm that satisfies the *causal delivery* property without being a causal broadcast algorithm, i.e., without satisfying the *agreement* property of a reliable broadcast?

### Exercise 5

The Uniform Reliable Broadcast Algorithm requires a process to receive an acknowledgment from all nonfaulty processes before it can deliver a message. The acknowledgment is needed because when a process invokes the underlying best-effort broadcast and then crashes, all components of the process are affected and stop (including the best-effort broadcast module and any further underlying modules, such as the modules that may implement perfect links). The unit of failure is a process, not a module.

For this exercise only, consider an idealized and nonrealistic system model, where some component may invoke infallible lower-level components. In this model, the unit of failure is not a process but a module. Describe an implementation of uniform reliable broadcast that uses an infallible perfect point-to-point links abstraction in this idealized model. Do not use failure detectors of any kind.

*Hint:* You may get some inspiration from the solution to last week's exercise 2.

### Exercise 6

Would it make sense to add the *total-order* property to the best-effort broadcast?

### Exercise 7

What happens in our consensus-based total order broadcast algorithm if the set of messages decided on is not sorted deterministically

- a) after the decision but is sorted prior to the proposal,
- b) neither a priori nor a posteriori?