

Solution for exercise 7

7.1 Worst-case latency of eager reliable broadcast (2pt)

The *latency* of a broadcast protocol is defined as the maximum number of point-to-point message transmissions taken by the protocol, measured from when the sender broadcasts a payload message until every receiver has delivered it.

Consider the *eager reliable broadcast* protocol [CGR11, Algorithm 3.3]. What is its maximum latency? Describe a corresponding execution.

Solution. In the worst case, processes crash in sequence and the algorithm may incur $O(N)$ steps and $O(N^2)$ messages. A corresponding execution is shown in Figure 1. Process p *rb*-broadcasts a message m and immediately *beb*-broadcasts it. After process q *beb*-delivers this message, process p crashes and process r does not *beb*-deliver this message. In turn, process q *beb*-broadcasts the message and crashes: process r receives it but process p does not. Finally, process r *beb*-delivers the message, *beb*-broadcasts it again, and also *rb*-delivers it. From this execution, we see that in total 9 messages are transmitted, taking one communication step per process. Generalizing this argument, the maximum latency may be N message-transmission steps and incur $O(N^2)$ messages overall.

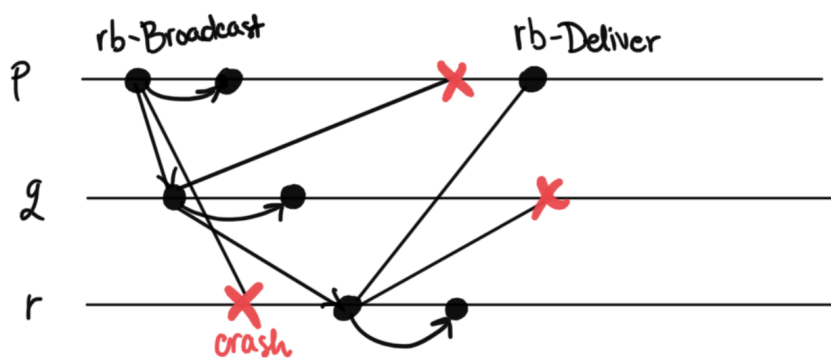


Figure 1. An example of the worst case execution of eager reliable broadcast.

7.2 Uniform reliable broadcast in the fail-stop model (4pt)

Algorithm 3.4 [CGR11, Sec. 3.4.2] implements uniform reliable broadcast using a perfect failure detector \mathcal{P} .

- What may happen if \mathcal{P} does not satisfy its *strong completeness* property?
- What may happen if \mathcal{P} violates its *strong accuracy* property?

In each case, explain whether this violates liveness or safety.

Solution.

- a) Consider a system of three processes: p , q , and r . Suppose that p *urb*-broadcasts a message m . If *strong completeness* is not satisfied then p might never *urb*-deliver m if either of q and r crashes and p never detects their crash or *beb*-delivers m from them. Process p might wait indefinitely for them to relay m and the algorithm may violate the *validity* property which is a liveness property.
- b) Assume that *strong accuracy* is violated and p falsely suspects q and r to have crashed. Process p eventually *urb*-delivers m . Assume that p crashes afterward. It might be the case that q and r have never *beb*-delivered m and have no way of knowing about m . Hence, they will not *urb*-deliver m , violating the *uniform agreement* property which is a liveness property.

7.3 FIFO broadcast from FIFO links (4pt)

An implementation of reliable broadcast with FIFO (first-in, first-out) order is found in Algorithm 3.12 [CGR11]. It runs on top of a reliable broadcast primitive. Here we consider a different implementation of FIFO reliable broadcast, without underlying reliable broadcast.

The usual specification of perfect point-to-point links (Module 2.3) does not impose any ordering. Assume, additionally, that the links between all servers are FIFO, that is, messages are delivered over each link *per process* in the order they were sent. A specification of FIFO links is available in Module 2.11 (Section 2.9.) Run the *basic broadcast* (Algorithm 3.1) using these FIFO links instead of perfect point-to-point links; this gives a FIFO best-effort broadcast (FIFO-BEB).

Now run the *eager reliable broadcast* protocol (Algorithm 3.3) with this FIFO-BEB (and the FIFO links). Does this implement FIFO-order reliable broadcast? Justify your answer.

Solution. Yes. Note that the *eager reliable broadcast* protocol uses best-effort broadcast and satisfies the properties of reliable broadcast. If it uses FIFO-BEB then its *FIFO delivery* property ensures that all DATA messages transmitted from one process to another one are delivered in FIFO order. Since a process *rb*-delivers every payload message immediately after receiving it for the first time through a DATA message, multiple messages from a specific sending process are also *rb*-delivered in the same order in which they were *rb*-broadcast. This, the modified implementation of reliable broadcast also satisfies the FIFO delivery order property.

References

- [CGR11] C. Cachin, R. Guerraoui, and L. Rodrigues, *Introduction to reliable and secure distributed programming (Second Edition)*, Springer, 2011.