Institute of Computer Science, University of Bern
Distributed Algorithms, Spring 2020

Prof. Christian Cachin
Jovana Mićić

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.

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 $\mathcal{P}$.

a) What may happen if $\mathcal{P}$ does not satisfy its *strong completeness* property?

b) What may happen if $\mathcal{P}$ violates its *strong accuracy* property?

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

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.

References

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