

Department of Computer Science

PURDUE
UNIVERSITY

CS505: Written Assignment 1

Channels and Broadcast

Reliable Channels

1. Consider the following definition of a Semi-Reliable Channel:

I. No duplication

- A message is *received* at most once

II. No creation

- No message is *received* unless some process did send it

III. Validity

- One of two messages that a correct process p_i sends to a correct process p_j is eventually *received* by p_j

► Consider two variants of semi-reliable channels

- a. “One of two” is defined deterministically, i.e., by the order of invocation of `send` (every other)
- b. “One of two” is defined randomly, i.e., any `send` succeeds independently with probability .5

► Compare each of the two channels to reliable channels

- Remember: an algorithm/abstraction A is *stronger than* A' if A' can be implemented with (on top of) A . Inversely, A' is then *weaker than* A . If A and A' are each stronger than the other then they are *equivalent*. If A is stronger than A' but not inversely then A is *strictly stronger than* A' , and A' is *strictly weaker than* A .

Reliable Broadcast

2. Is it possible to get a better message complexity than $O(n^2)$, e.g. $O(n)$, with Reliable Broadcast?

▶ You can

- piggy-back information on messages
- presuppose a static predefined 'overlay' structure on the group
- ...

▶ By e.g. exhibiting an algorithm or arguing why it is not possible

FIFO Broadcast

3. Can we use FIFO reliable channels to build a FIFO Reliable Broadcast from the Reliable Broadcast algorithm seen in class?

- If yes, describe the algorithm and prove that it implements the specification. Compare the two solutions
- If not, argue why not

Uniformity

4. Take the Causal Atomic Broadcast seen in class

- a. Does it implement Uniform Causal Order? Why (not)?
- b. Suppose the underlying FIFO Atomic Broadcast algorithm provides Uniform Agreement instead of Agreement (i.e., I, II, III, V, IX, XIII instead of I, II, III, IV, IX, XIII). Does the resulting algorithm also provide Uniform Agreement? Can we simplify it?