Distributed Algorithms

Fall 2020

Consensus
6th exercise session, 26/10/2020

Matteo Monti <matteo.monti@epfl.ch>
Jovan Komatovic <jovan.komatovic@epfl.ch>

FloodSet Algorithm (1/2)

The agreement problem for crash failures has a very simple algorithm, called FloodSet. Processes just propagate all the values in the input set V that they have ever seen and use a simple decision rule at the end.

The flooding algorithm, due to Dolev and Strong, gives a straightforward solution to synchronous agreement for the crash failure case. It runs in f+1 rounds assuming f crash failures.

FloodSet Algorithm (2/2)

The algorithm:

- Each process maintains variable $W_i \subseteq V$, initially $\{v_i\}$.
- f+1 rounds, consisting of:
  - Broadcast W_i,
  - Add all received elements to W_i.
- After that, if $W_i = \{v_i\}$ (i.e. a singleton) decide v_i, otherwise decide v_0 (default value).

The goal:

Prove the algorithm is correct (guided proof)

Exercise 1

Definition: Let $W_i(r)$ be the value of variable W at process i after r rounds.

Lemma 1: If no process fails during a particular round r, $1 \leq r \leq f+1$, then $W_i(r) = W_j(r)$, $\forall$ i, j non-faulty after round r.

Lemma 2: Suppose that $W_i(r) = W_j(r)$ $\forall$ i, j non-faulty after round r. Then, $\forall$ r', $r \leq r' \leq f+1$, the same holds, that is, $W_i(r') = W_j(r')$ $\forall$ i, j non-faulty after round r'.

Exercise 2

Lemma 3: If processes i, j are non-faulty after f+1 rounds, then $W_i(f+1) = W_j(f+1)$.

Theorem: Floodset solves consensus, i.e., Agreement, Validity and Termination hold.

Note:

The property of validity in the consensus problem comes in two flavors.

- A: If all processes have the same initial value v, then v is the only possible decision value.
- B (stronger): The decision value is the initial value of some process.

Exercise 3

What is the communication complexity of the FloodSet algorithm in:

- number of messages,
- bits, given that b is the number of bits required to represent the elements of V?

Is the decision rule (last bullet of the algorithm) so critical? In other words, is there any alternative decision rule we can have? If so, name one.

Exercise 4 (1/2) - Bonus

A strange setting for consensus (*synchronous case)*

Consider a network that is organized as a 2-dimensional grid, such that every process has up to 4 neighbors. The width of the grid is w and the height is h. The grid is big, meaning that w+h is much smaller than w*h. While there are faulty and correct processes in the network, it is assumed that two correct processes are always connected through at least one path of correct processes. In every round processes may send a message to each of its neighbors, the size of the message is not limited.

Exercise 4 (2/2) - Bonus

a) Assume there is no faulty process. Write a protocol to reach consensus. Optimize your protocol according to speed.

b) How many rounds does your protocol require?

c) Assume there are w+h faulty processes. In the worst-case scenario, how many rounds does the protocol require now?