Distributed Algorithms

Consensus
5th exercise session

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)

Exercises 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'.

Exercises 1 (Solution)

Proof of lemma 1: Assume no processes fail during round *r*, and let *I* be the set of not-failed processes after *r-1* and *r*. Since every process in *I* (and only these) send *W* to all other processes at round *r*, by the algorithm construction we have that at the end of round *r*, $W_i(r)$ is the union of all $W_j(r-1)$, $\forall j \in I$.

Proof of lemma 2: We use induction on the number of rounds. Base case is the assumption on the lemma. For the induction step, you just need to note that all processes are broadcasting the same set, and therefore do not make any changes to *W*.

Exercises 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.

Exercises 2 (Solution 1/2)

Proof of lemma 3: Since there at at most *f* failures, then there exists a round *r* such that no faults occur during that round. Then, Lemma 1 implies that $W_i(r) = W_j(r)$, $\forall i, j$ non-faulty after round *r*, and Lemma 2 implies that $W_i(f+1) = W_j(f+1)$, $\forall i, j$ non-faulty after round *f+1*.

Exercises 2 (Solution 2/2)

Proof of theorem:

- Termination: Follows from the termination rule of the algorithm.
- Validity: If all processes have the same initial value v, then any message ever transmitted will always contain $\{v\}$, therefore $W_i(r) = \{v\} \ \forall i$ non-faulty after $r \leq f+1$ rounds. This implies the decision value will be v.
- Agreement: Since all processes decide after $f+1$ rounds, Lemma 3 implies that any two processes that decide have the same value for W.

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 3 (Solution)

The communication complexity of the FloodSet algorithm in:

- number of messages: $O((f+1)n^2)$
- number of bits: $O((f+1)n^2)b$ in the binary case, or $O((f+1)n^3)b$ in the non-binary case

About the decision rule: Since FloodSet guarantees that all non-faulty processes obtain the same W after $f+1$ rounds, other decision rules would also work correctly, as long as all the processes apply the same rule.

For instance, if the value set V has a total ordering, the all processes could simply choose the minimum value in W. This rule guarantees the stronger validity. That is not necessarily true for the decision rule given in FloodSet, because v_0 might not be the initial value of any process.

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?

Exercise 4 (Solution 1/2)

a) The goal if the protocol is for every process to know the input of all the other processes. Each process internally allocates an array of size *w*h*, where each entry of the array represents the input of one process that initially is set to a sentinel value '?'. The process fills the array until no '?' are left, then the process can make its decision (e.g., the minimal value received). Whenever a process gets new information, it sends its updated array to all its four neighbors. At the beginning, the "new information" is its own input.

b) The number of required rounds is the length of the longest *shortest* path between any two processes. In the case of the grid, this is *w*+*h*.

Exercise 4 (Solution 2/2)

c) The problem reduces to building the longest path on a connected grid by *removing* a given number of nodes.