Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

FS 2009

Prof. R. Wattenhofer / C. Lenzen / T. Locher

Principles of Distributed Computing
Exercise 6: Sample Solution

1 Deterministic Consensus

a) Take three nodes c, v, and w with input values 0, 1, and 1, respectively. Let c crash after sending its value to, e.g., v. Then v will decide on 0, but w will only receive the 1 from v and decide on 1. Thus no agreement between the non-faulty nodes v and w is reached.

b) We can simply run Algorithm 1 exactly $f+1$ times, where each node replaces its value after each run by the computed value. All nodes finally decide on the value computed in the last round.

Obviously, this algorithm terminates after $f+1$ rounds at all non-faulty nodes, and nodes will always decide on values from the initial set of inputs. As in a), the agreement condition may be violated at the end of a single round. However, since at most f nodes may crash, in at least one round no node crashes. In this round, all non-faulty nodes will decide on the same value x. After this round it is irrelevant if further nodes crash, since in each round all nodes will receive only the value x. Hence, after at most $f+1$ rounds the non-faulty nodes reach consensus.

2 Randomized Consensus

a) First, we show that the algorithm tolerated $f < n/8$ crash failures. If all nodes have the same input x, then every node receives x $n-f$ times and all nodes decide on x. Let us assume now that the input consists of both 0s and 1s. In this case, any result is fine, i.e., the validity condition is satisifed trivially. Assume that a node decides on a value x because it received x $n-2f$ times. In this case, any other node v must have received at least $n-3f$ BID messages with value x. Hence it follows that every non-faulty node will bid for x in the next round, and then decide on x. Analogously to the proof of Theorem 6.6, the algorithm terminates after a finite number of rounds (in expectation) for the following reason. If the nodes bid for 0 or 1 with equal probability, they will all bid on the same value with probability $1/2^n$. It is not possible that some nodes propose 0 and others propose 1 in Line 8 because this would imply that there are $n-4f$ nodes bidding 0 and $n-4f$ different nodes bidding 1, i.e., we must have that $n-4f \leq n/2$, a contradiction to the assumption that $f < n/8$. Thus, the expected running time is upper bounded by 2^n.

We will now show that the algorithm may not terminate if $f \geq n/8$. Assume that $n = 8$ and $f = 1$. Four nodes bid 0 and the other four nodes bid 1. Obviously, no node receives $n-f = 7$ messages with either 0 or 1, but all the nodes with 0 may receive $n-4f = 4$ bids for 0 and the nodes with 1 receive 4 bits for 1, i.e., they continue to bid for the same value in the next round and so on. Since the nodes neither decide on a value deterministically nor choose a different value randomly, the algorithm never terminates.

b) Since there are no Byzantine nodes, we decide on the value x if all $n-f$ received values are x, i.e., we change $n-2f$ to $n-f$ in Line 5. Any other node receives at least $n-2f$ of

Algorithm 1 Crash failure resistant randomized consensus.

1: node u starts with input bit $x_u \in \{0, 1\}$, round:=1.
2: broadcast BID(x_u, round)
3: **repeat**
4:     wait for $n - f$ BID messages of current round
5:     **if** at least $\mathbf{n - f}$ messages have value x **then**
6:         $x_u := x$; decide on x
7:     **else if** at least $\mathbf{n - 2f}$ messages have value x **then**
8:         $x_u := x$
9:     **else**
10:         choose x_u randomly, with $Pr[x_u = 0] = Pr[x_u = 1] = 1/2$
11:     **end if**
12:     round := round + 1
13:     broadcast BID(x_u, round)
14: **until** decided

these values. Thus, we can change $n - 4f$ to $n - 2f$ in Line 7. The new algorithm is given in Algorithm 1.

Analogously to the above arguments, if all nodes have the same input, then every node receives x $n - f$ times and all nodes decide on x. In the following, we assume that the input consists of both 0s and 1s (and the validity condition is satisfied). Assume that a node decides on a value x because it received $n - f$ BID messages with x. In this case, any other node v must have received at least $n - 2f$ BID messages with x. Hence it follows that every non-faulty node will bid for x in the next round, and then decide on x. Again, it remains to verify that the algorithm terminates. Assume again that some nodes propose 0 and others propose 1 in Line 8. In this case, there are $n - 2f$ nodes bidding 0 and $n - 2f$ different nodes bidding 1, i.e., there is a total of $n + (n - 4f) > n$ nodes, a contradiction. Hence, in the worst case all nodes have to choose the same bit randomly, which results in an expected time complexity of 2^n.