CS5320: Distributed Computing
Theory Assignment 4

Sagar Jain
CS17BTECH11034

April 14, 2020

Early Stopping Consensus Algorithm for Crash Failures

The following is the modified Algorithm 14.1 to design an early-stopping algorithm for consensus under failstop failures, that terminates within $f' + 1$ rounds, where f' , the actual number of stop-failures, is less than f.

The Algorithm

```
Process P_i (1 ≤ i ≤ n) executes the consensus algorithm for up to
f crash failures (early termination)
for round from 1 to f + 1 do
  Values = []
  ProcessIndices = []
  broadcast(x);
  y_j ← value (if any) received from process j in this round;
  append(j) to ProcessIndices;
  append(y_j to Values)
  x ← min_j(x, y_j);
  if (round > 1 and previousProcessIndices == ProcessIndices and PreiousValues
      == Values ) then
          break;
  previousValues = Values;
  previousProcessIndices = ProcessIndices;
Output x as consensus value
```

Approach & Proof of Correctness

The main idea for the algorithm is that if there are f' failures we would at most take $\leq f'+1$ rounds to terminate the algorithm. One key observation is the following:
If, there have been no failures between two subsequent rounds, call this round r, then there

would be no change in the number of values received and the values themselves for rounds r+1 and rounds r+2. This is the fact that we can exploit to make sure that we terminate within f+1 rounds or $f' + 2$ rounds. The **termination** is guaranteed by the fact that the number of failures is fixed and less than the number of processes and the **validity** of the algorithm follows from the algorithm ar 14.1. **Agreement** would certainly happen since the number of failures is $<$ f.