

CS 425/ECE 428 Distributed Systems
Spring 2016
Homework 4 Solutions

(1) In this case $n = 4$ and $f = 1$, thus, $n - f = 3$. Suppose that nodes 1, 2, 3 and 4 have inputs 1, 2, 3 and 4, respectively.

The smallest y value is obtained when a node receives values 1, 2, 3 in round 1, respectively, from nodes 1, 2, 3, and computes the new value of y using these three (i.e., $n - f$) values. Thus, the smallest possible y value after one round is $(1+2+3)/3 = 2$. Similarly, the largest possible y value after round 1 is $(2+3+4)/3 = 3$.

Observe that the original values are spread in interval $[1,4]$, of length 3, whereas the values after one round are guaranteed to be in the interval $[2,3]$, of length 1.

(2) $(f/2)+1$ rounds suffice. In each round, 0 or 2 processes fail. Since at most f processes fail, there is at least one round during which no new process will fail. In this round, all processes that haven't yet failed will receive identical set of values. This ensures that consensus is achieved.

(3) Yes, consensus is achievable. Here is an example algorithm that achieves consensus:

- Process P_0 sends its input to all other processes, and sets its output to equal its own input, and then terminates.
- Each process P_k (where P_k is not same as P_0) waits to receive a message from P_0 , sets its output equal to the value received from P_0 , and then terminates.

Termination in finite time is guaranteed since each non-faulty P_k will receive P_0 's message in finite time.

Exact agreement is guaranteed since all non-faulty processes use P_0 's input as their output.

When all processes have same input, that input equals P_0 's input, and becomes the output of the algorithm -- thus validity is achieved.