



Distributed Systems Part II

Exercise Sheet 3

1 Consensus in a Grid

In the lecture you learned how to reach consensus in a fully connected network where every process can communicate with every other process. Now 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 , the height is h . The grid is big, meaning that $w + h$ is much smaller than $w \cdot h$. 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.

- Assume there is no faulty process. Write a protocol to reach consensus. Optimize your protocol according to speed.
- How many rounds does your protocol require?
- Assume there are $w + h$ faulty processes. The faulty processes may die any time, but may not send wrong messages. In a worst case scenario, how many rounds does the protocol require now?
- Assume there are $w/2$ Byzantine failures. How could they sabotage your protocol? Only a general idea is required, do not go into details.

2 Consensus in a Hypercube

Some networks are organized as a hypercube. There are $n = 2^m$ processes and each process can communicate with m other processes.

- Modify the King algorithm so that it works in a hypercube. Optimize the algorithm according to resilience.
- How many failures can your algorithm handle? (Assume Byzantine processes can neither forge nor alter source or destination of a message.)
- How many rounds does this algorithm require?

3 Consensus with Byzantine Failures

Does a 2-resilient algorithm without authentication for 6 processes exist? Write it down or sketch a proof for its non-existence.