Parallel and Distributed Computing DD2443 - Pardis24 Exercises for Lecture 11

Name: Casper Kristiansson

October 4, 2024

Exercise 1

Question

Explain carefully why it is theoretically justified to increase each n to the nearest multiple of 3 in the lower bound, Byzantine failures proof from the lecture.

Answer

In the Byzantine failure proofs increasing the number of processes to the nearest multiple 3 is justified because of how the Byzantine fault tolerance works. The Byzantine fault tolerance requires a specific relationship/connection between the number of faulty nodes and the total number of nodes. As stated in the lecture the system must require that $n > 3f$. This will help to make sure that the system can achieve consensus by making sure that we can reach the fault tolerance threshold while still being able to continue the program.

Exercise 2

Question

Does the correctness proof of the Pease-Shostak-Lamport algorithm still go through if we replace the condition $w \in QS^f$ with the condition $w \in Q^f$? Explain carefully (but briefly) why or why not the replacement works.

Answer

No the correctness proof for the Pease-Shostak-Lamport algorithm does not work if we replace the condition $w \in QS^f$ with the condition $w \in Q^f$. The condition that $w \in QS^f$ represents the quasi-silent faults in a program where a faulty node/process might not send out all messages. Replacing this part of the program with $w \in Q^f$ instead which only accounts if a faulty process sends no messages means that the algorithm will not be able to detect if any nodes are partially faulty (sent some but not all messages). Therefore the replacement of the condition $w \in QS^f$ with the condition $w \in Q^f$ does not work because it breaks the algorithm which requires the original condition.

Exercise 3

Question

For the problem of leader election, each node in a distributed system executes the same control program and is in one of the states following states:

$$\{undecided, notleader, leade\}$$

The task is to ensure that, on termination, exactly one node is in state leader, and all other nodes are in state not leader. A ring network topology is one in which the nodes are connected in a ring. A ring is anonymous if there is no feature (such as a node identifier) that allows telling the nodes apart. You can assume that all nodes are non-faulty (no Byzantine nodes, no crash failures).

1. Is leader election possible in an anonymous ring using a (round-based) synchronous and deterministic control program? Either give an algorithm (using suitable pseudocode) or prove that the task is impossible.
2. Is leader election possible in a synchronous, deterministic ring in which all but one of the processors have the same identifier? Either give an algorithm or prove that the task is impossible.
3. Consider a synchronous ring in which exactly two nodes have identifier A and all the other nodes have identifier B. Is deterministic leader election possible in this setting? Either give an algorithm or prove that the task is impossible.

Answer

(1)

The leader election is impossible in an anonymous ring because nodes cannot be individually identified because all nodes execute the same instructions each round. This type of symmetry in the protocol ensures that there is no single node that can break the ring and become the new leader of the ring.

(2)

With a leader election in a synchronous ring, it is possible that one specific node could have a unique identifier. The basic concept of this algorithm is that the node that has the unique identifier declares itself as a leader whereas the rest of the nodes do not become leaders after receiving the message of the new leader. Doing this will make a leader election possible.

(3)

A leader election for the given case is not possible. The reason for this is if two nodes with the same identifier A and the rest have identifier B each of the nodes in the groups will work identically with the rest of the nodes in the two groups. Doing this where they work together with the rest of the group will make it not possible to elect a new leader because if the nodes in A try to become a leader at the same time it will cause the protocol to break.