

CS9668

Assignment 2

Linxiao Wang
250888611

October 15, 2018

PROBLEM 1

DESCRIPTION OF THE ALGORITHM

PROBLEM 2

DESCRIPTION OF THE ALGORITHM

In the following description, we refer to the processor who doesn't have a left neighbor the leftmost node, the processor who doesn't have a right neighbor the rightmost node, all the processors in between the middle nodes. Also, we refer to the ID in the received message id_m , and the ID of the current processor id_p .

- 1 At the beginning of the algorithm, every processor determines whether it is the rightmost node or leftmost node or a middle node.
- 2 Then the leftmost node sends its ID to its right neighbor.
- 3 For every middle node, after receiving a message from its left neighbor, the processor send the larger one between id_m and id_p to its right neighbor.
- 4 When the rightmost node receives a message, it compares the id_m with id_p , if id_m is larger, then the processor knows it is not the leader, else it is the leader. Then the processor send the larger one between id_m and id_p to its left neighbor, and return.

- 5 For every middle node, after receiving a message from its right neighbor, the processor compare id_m with id_p , if id_m is larger, then the processor knows that it is not the leader, else it is the leader. Then the processor forward the message to its left neighbor and return.
- 6 When the leftmost node receives a message, it compares the id_m with id_p , if id_m is larger, then the processor knows it is not the leader, else it is the leader, then return.

PROVE OF TERMINATION

- Because its a line network, so there will be a leftmost processor. Start from the leftmost processor, every processor will send a message to its right neighbor, so the right most processor will receive a message and terminate according to step 4 above.
- The rightmost processor will send a message to its left neighbor, and the message will be forwarded to all the left neighbors by all the middle processors.
- According to step 5 and 6, all the middle processors and the leftmost processor will receive a message from their right neighbors, so they all will terminate.

PROVE OF CORRECTNESS

- Start from the leftmost processor, every processor will send a message contains the largest ID so far. So when the rightmost processor receive a message, it will contain the largest ID from all the processors except the rightmost one.
- The rightmost one can compare the largest one from all the processors from its left with its own, so it will know whether it is the leader or not.
- Then the rightmost processor will send the largest ID among all the processors to its left neighbor, and the largest ID will be forwarded to all the processors. All the other processors will compare that largest ID with their own and know whether they are the leader or not.

COMPLEXITY COMPUTATION

Here we assume n is the number of the processors in the network.

Start from the leftmost processor, one message will be send per round, until the right most processor get an message. This will take $n - 1$ round and $n - 1$ messages are sent.

Then a message will be sent from the rightmost processor to the leftmost processor, it will take $n - 1$ rounds and $n - 1$ messages are sent.

Time complexity: $2(n - 1) = 2n - 1$ which is $O(n)$.

Communication complexity: $2(n - 1) = 2n - 1$ which is $O(n)$.