Distributed Algorithms

Solution for Project 1

Group 8

1. Alexandre Georges Rémi Martin, 406294
2. Benyamin Shafabakhsh, 406305
3. Yared Dejene Dessalk, 406228
4. Anubhav Guha, 406244
5. Jiaqiao Peng, 406038

November 23, 2018

Exercise 1.1

(i)

The complexity of echo algorithm is O(n)=2n because there are 2 **loops (line 29, and line 45)** that are iterating on the number of interfaces to send messages.

The first loop sends messages to its interfaces (so iterates on the number of interfaces it has) which triggers the nodes to send messages to their childs (child as the interface where the message is not coming from).

Those two-loop complexities added to each other is then equal to the number **n** of interfaces.

The loop line 45 will then not be used again as it is in an IF condition which checks if the node has sent a message (if color is not red, and line 49 in the IF sets the color to red preventing any more read of this part of the code).

Each message sent results to a count up of **rec**, which once equal to the number of interfaces the node has (and equal to the number of messages it has sent **minus** the one it received first) and sends a message to the father (the interface it has received the first message from). Each time the loop sends a message leads then to a second message sent back until it reaches the initiator.

The time complexity is then loop one in initiate()+ the one in receive() = n

And each time a message is sent, it is received and either leads to sending messages to the child which is counted in the complexity above, nothing if the count for messages received is not equal to the number of interfaces equal to a complexity of 1, or a send back to the father which is done once on each interface (as the number of messages sent has to equal the number of interfaces once) so a complexity of n

Therefore, the complexity is then 1+n+n = O(n) = 2n

The message complexity is n for n messages sent to the child and n for n messages sent back so 2n in total.

(ii)

A node knows only the number of messages it is supposed to receive (as shown in the question, only **rec** tells when to stop listening and send a confirmation message.

If a message is duplicated, the node will count it as a normal message and stop listening before receiving all the messages it should receive.

To avoid that, each node must listen to the first message of an interface only and ignore all the others.

Exercise 1.2

(I)

1. Worst-case Scenario

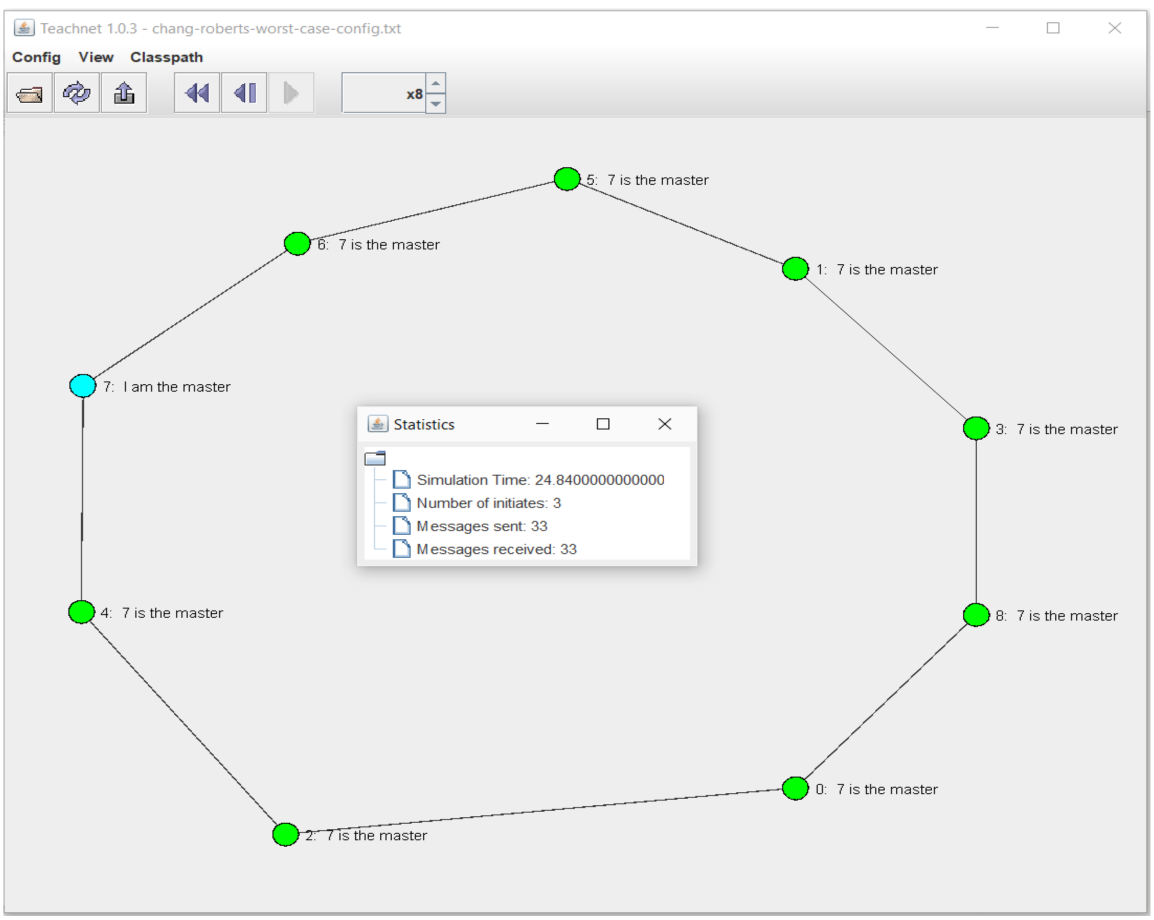
   The worst-case scenario for Chang and Roberts algorithm happens when the initiators are arranged on the ring in descending order and the election is initiated in in ascending order. In this case all the messages of the initiators need to traverses a long path until they get ceased by the master node.

   The formula to get the message complexity for n, number of nodes and k, number of initiators is

   $$= n.k - \frac{k \cdot (k-1)}{2}$$

   and additional n messages for win notification

   for n=9 and k=3, we expect total number of messages = **33** and it is what we got in the simulation.



2. Best-case Scenario

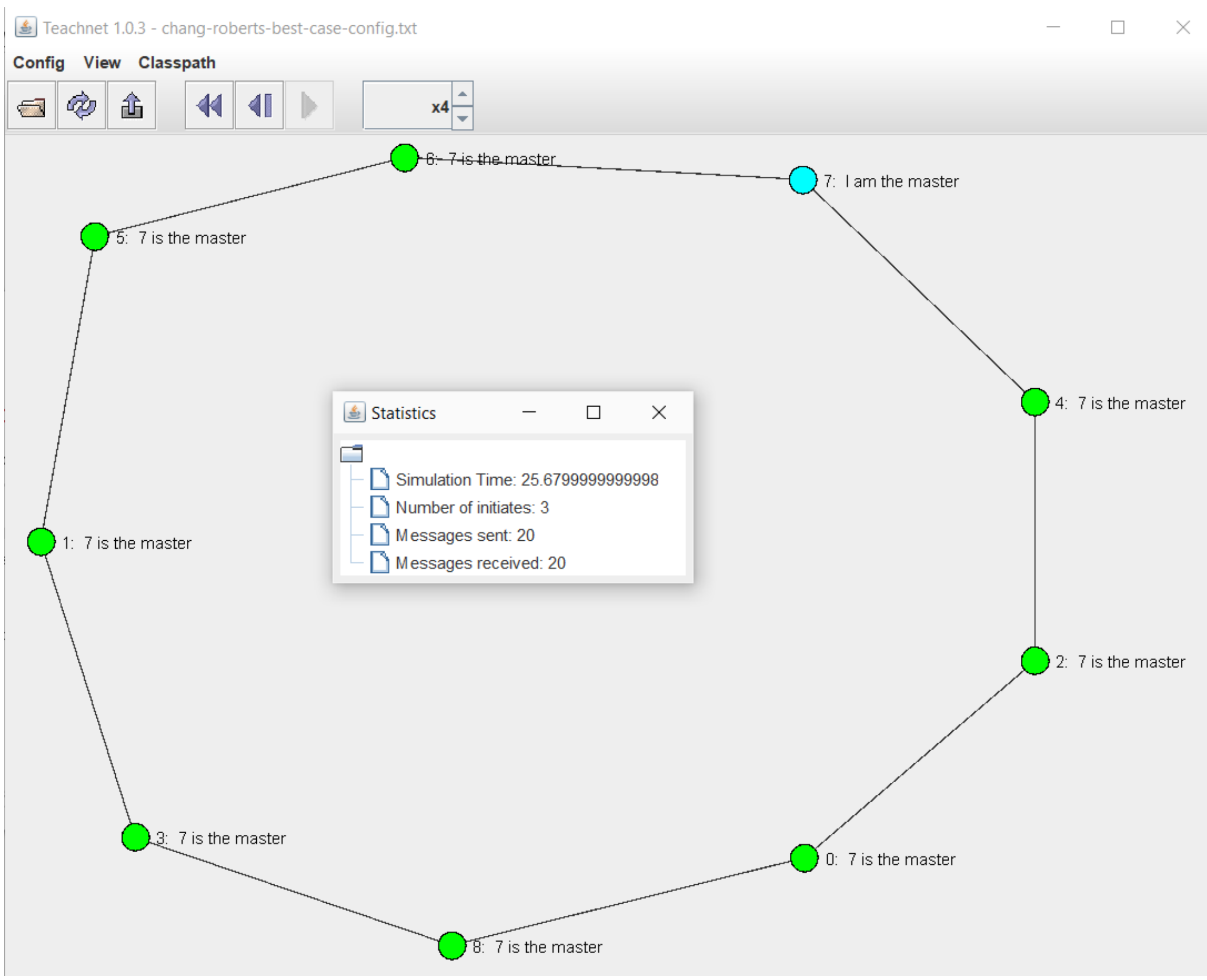
The best case scenario for the algorithm occurs when the initiators are arranged in ascending order. All the non winning messages will be trapped by the master node, which is going to be the winner, without traveling longer path.

The formula to get the message complexity for n, number of nodes and k, number of initiators is

$$= n + k - 1$$

And additional n messages for win notification

For n=9 and k=3, we expect **20** total number of messages. From simulation we also got the same number of total messages.



3. Average-case Scenario

In order to get the to total number of messages for average case scenario, we configured the simulator to randomly assign IDs. For 9 nodes and 3 initiators, we got the following total number of messages in 10 runs.

No.	Total Number of Messages
1	25
2	29
3	24
4	26
5	26
6	24
7	22
8	26
9	30
10	23
	Average ~ **26**

Using the formula given in the lecture, we can calculate the number of messages:

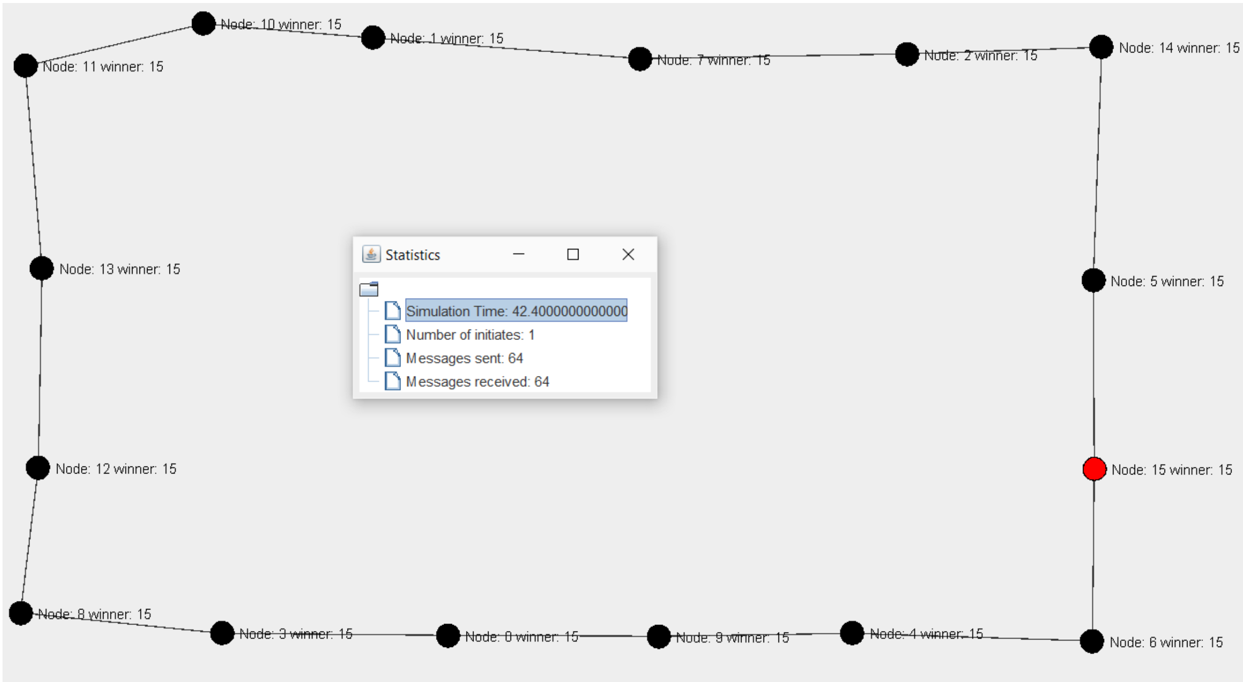
= $n \cdot \ln k$

And additional n messages for win notification

For n=9 and k=3, the above formula approximately gives **26** total messages which is the same with what we provides above in the table.

(II) Hirschberg Sinclair Algorithm

When a red node appears the election is done. But sometimes the simulation is continuing for a little while because some message is still transmitting. If we count the time more precisely and try a lot of times to get the average time consumption, the time complexity is around O(nlogn).


Node: 10 winner: 15
Node: 1 winner: 15
Node: 7 winner: 15
Node: 2 winner: 15
Node: 14 winner: 15
Node: 11 winner: 15
Node: 13 winner: 15
Statistics
Simulation Time: 42.4000000000000
Number of initiates: 1
Messages sent: 64
Messages received: 64
Node: 5 winner: 15
Node: 12 winner: 15
Node: 15 winner: 15
Node: 8 winner: 15
Node: 3 winner: 15
Node: 0 winner: 15
Node: 9 winner: 15
Node: 4 winner: 15
Node: 6 winner: 15