Aix-Marseille Université
Master Informatique
Algorithmique Distribuée
2014

TP 4: Randomized Election in Ring Networks

1 Simulating Randomized Algorithms for Election

In the class, we have seen several algorithms for election using randomized algorithms. We will simulate the following algorithms for election on ring networks.

Algorithm 1:
Each node selects a random number between 1 and n. The nodes with maximum value becomes candidates and all other nodes become passive. If there is only one candidate it becomes the leader. If there are more than one candidates, the candidate nodes repeat the previous step (i.e. each candidate select another random number and the nodes with maximum value remains a candidate for the next round.)

Algorithm 2:
All nodes are candidates in the first round.
In each round each candidate node selects either 0 or 1 with equal probability. If only one candidate chooses '0' it becomes the leader. Else if all candidates choose '1' they all remain candidates for the next round. Otherwise, the nodes that choose '0', become candidates and all other nodes become passive. While there are more than one candidates, the candidate nodes repeat the previous step.

Algorithm 3:
All nodes are candidates in the first round.
In each round each candidate node selects '0' with probability $1/n$ and selects '1' with probability $(1 - 1/n)$. If only one candidate chooses '0' it becomes the leader and all other nodes become passive. Otherwise, all nodes remain candidates for the next round.

Exercice

1. We will use *JBotSim* for simulating the randomized algorithms on the ring. (See last week's TP on how to use *JBotSim* or check the documentation at: `http://jbotsim.sourceforge.net/javadoc/`).

   Download the file "RingElection.zip" from the following link :
   `http://pageperso.lif.univ-mrs.fr/~arnaud.labourel/AD2014/RingElection.zip`
   This contains a program that constructs an anonymous ring network of a given size. Compile the java files and execute the program using JBotSim on your computer.

   ```
   java -cp.:jbotsim.jar RingMain <size-of-ring>
   ```

2. Change the program written in "RingNode.java" to implement the randomized algorithm for election. Each node has two links `left` and `right` to communicate with the neighbor on the left or the right. In each round, color the candidate nodes as "green". At the end of the election the program should change the color of the elected node to "red".
   Use the java class `java.util.Random` for generating random numbers. You must initialize the random number generator in the main file and use the `Random.nextInt()` function to obtain random numbers.

3. Add a counter to the program to count the number of rounds. Run the program on rings of different sizes and check how many rounds are required to achieve election. Add a global counter to compute the number of messages sent during the execution of the algorithm. Compare the different algorithms.