22C: 166 Distributed Systems and Algorithms

Homework 1, Total points = 50

Assigned 9/6/12 due 9/13/12

Question 1 (10 points): In 1978, Tony Hoare introduced the language CSP (Communicating Sequential Processes) that had a profound influence on the early days of distributed algorithm design (According to Citeseer, his book on CSP was the third most cited work in Computer Science till 2006). In CSP, processes communicate using synchronous message passing (i.e. handshaking): *the sender cannot complete the send operation unless the receiver is ready to receive it, and vice versa*. CSP used the symbols ! and ? to denote output and input operations respectively. To send a value v to Q, the sending process P uses the instruction Q!v, and to receive this value and assign it to a local variable x, process Q uses the instruction P?x. Here is an example:

program for process COPY
define c: character
do west?v → east!x **od**

It represents a process COPY that repeatedly receives a character from the process named **west**, and then sends that character to process named **east**, which assigns that value to its local variable x.

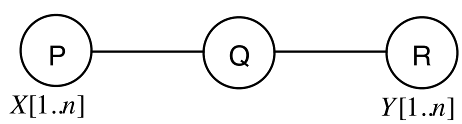

Figure 1. A system of three processes P, Q, R

Write a program for each of the three processes P, Q, R (Figure 1). The goal of the program is to copy an array X[1..n] of integers from P to R via Q. The final recipient process R will assign these values to a local array Y[1..n]. (*Feel free to dig out and read the original paper on CSP*)

Question 2 (10+15=25 points): Consider a population of size n in a town, and the task of multicasting a message m to every resident of that town. One resident starts the multicast, and residents communicate using Twitter. Communication is bi-directional (i.e. the communication graph is undirected) and round-based: in each round, a person can tweet a message to exactly one other person. The goal is to complete the multicast as quickly as possible.

Part 1. Assume that each sender has full knowledge of those who have not yet received the message. Propose an algorithm using which the broadcast is completed in the *fewest number of rounds* (only the main idea using pseudo-codes is needed here). Calculate the time complexity in rounds.

Part 2. Now assume that senders have no knowledge of who has already received the message (senders are lazy and no one maintains the list of the residents to whom s/he has already tweeted the message in the previous rounds). So each sender randomly picks a resident, and tweets him/her. Calculate the expected number of rounds E(n) needed for the message to reach every resident. You are encouraged to run a simulation experiment to compute this for 10 different values of n ranging from 1000 to 1 million and by repeating each experiment 50 times, Show the graph E(n) vs n. For a complete answer, you must explain your approach.

As an alternative, you have the option to analyze the algorithm come up with a closed form formula.

Question 3 (15 points): Consider an *anonymous* distributed system consisting of **N** processes. The topology is a completely connected network, and the links are bidirectional. Propose an algorithm using which processes can acquire *unique identifiers*. (Hint: use coin flipping, and organize the computation in rounds). Justify why your algorithm will work.