Niroojen Thambimuthu(10153928)

CPSC 441 Assignment 4

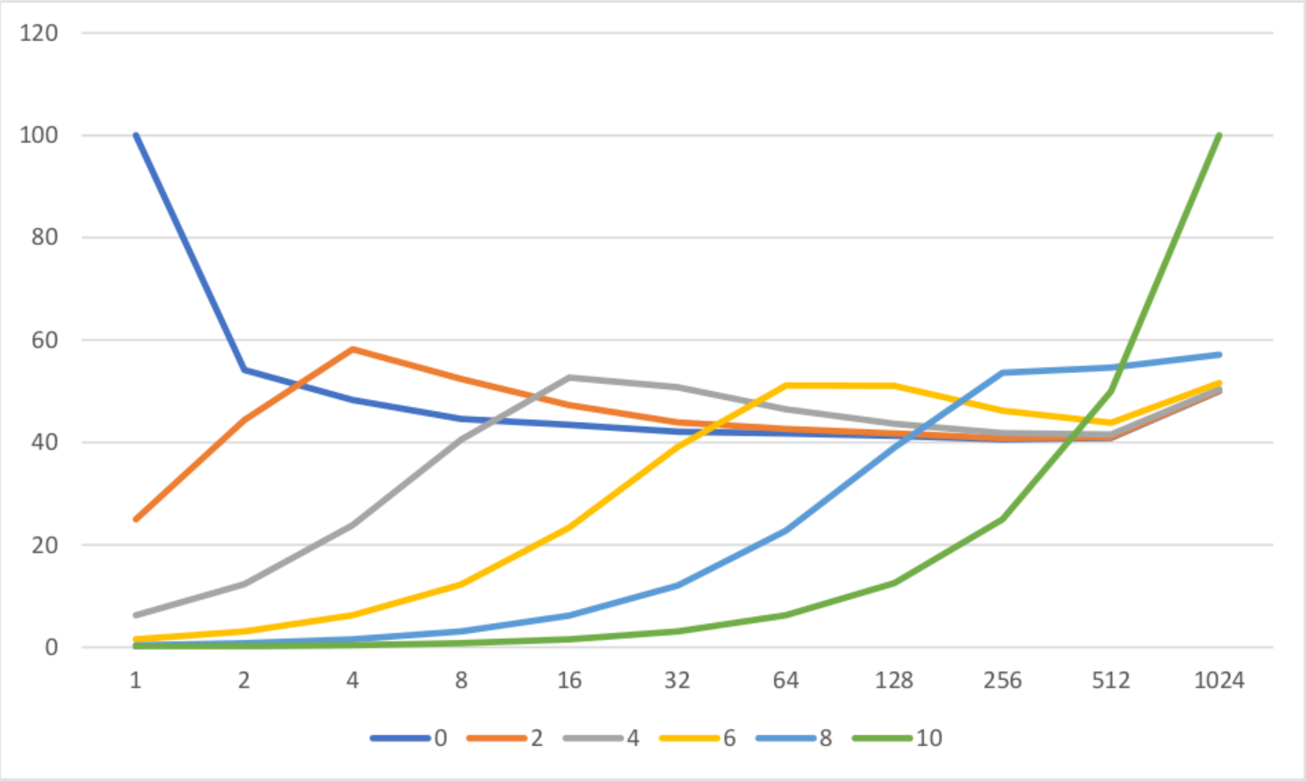
- How to compile: - gcc A4.c
- How to run: - ./a.out

By running the program, the simulation will automatically print the results into a text file within the directory folder. For this assignment, I used 10 arrays to simulate a makeshift binary tree with 10 levels, also the array for the level 10 is a 2 dimensional array so that I can store randomly generated samples into 100 test cases. I also implemented a loop function so that it can recursively call itself from any starting levels. Afterwards, it would calculate the average percentages and print into the file. It uses 2 for loops for each i and k values. I used my mac and school Linux computer to program and test.

For my testing part, I used a combination of the print statements, in conjunction with hand written examples such the example given in the description and by randomly making some sample binary trees and comparing them.

From these results, whenever both sending stations and starting level are either at the lowest(i=0,k=1) or at the highest(i=10,k=1024), it’s at its most successful point within the simulation. In between k values 2 and 256-512, we notice around 50% success rates for a few of the starting levels and less than 2-% success rate for a few others.

- Graph showing the performance characteristics of the Adaptive Tree Walk channel access protocol for N = 1024. Colored lines represent starting levels for the algorithm, i = 0,2,4,6,8,10 ; Y-axis represent success percentage ; X-axis represent sending stations k = 1,2,4,8,16,32,64,128,256,512,1024.



- a numerical table of your simulation results for N = 1024. The table should have columns showing the percentage of idle, successful, and collision probes for each load level and starting level of the probe algorithm.

The program will write results into text file “simulation.txt” in the directory folder.