BLG 433E – Computer Communications, Fall 2012 Assignment#2

TA: Salih Serdar Güçlü (Research Lab. 3)

Announced: 06.12.2012

Due: 30.12.2012

In this assignment you are asked to write a network simulation program of which details are given below.

RULES TO GENERATE THE NETWORK'S TOPOLOGY:

1) Your network has exactly 20 bidirectional links. Cost of the link information from node A to node B is sent from node A to node B. Also, cost of the link information from node B to node A is sent from node B to node A. In this assignment, you will assume that bidirectional links are symmetrical.

2) Number of nodes (routers) in your network must be randomly chosen between 8 and 12.

3) Cost of the links in your network must be randomly chosen. A randomly chosen value between 1 and 10 represents cost and it is multiplied by 10^{-3} seconds, in order to find corresponding propagation delay.

4) You will create a random topology with respect to the previous rules. Make sure that your network is connected.

Following the establishment of the network, periodically, the routers will broadcast the packets related to the links they are connected to (every 10 sec). The routing tables will be updated by the routers according to **Link State Routing** as the link state packets reach them.

The fundamental idea is to use flooding to distribute the link state packets to all routers. In order to avoid infinite number of packets, you will assign a TTL value to each packet. Each router decreases value of the packet's TTL by 1. When TTL of the packet reaches zero, it will be discarded by routers. Hint: Initial value of TTL for each packet will be your network's diameter.

After the nodes have formed their routing tables (we will call this time as t_0), randomly chosen links will fail consecutively. The mean time **between two consecutive link failures** must be **negative exponentially distributed** with **a mean of 2 hours**.

Failed links will be recovered after some time from the failure. The mean time **between a link failure and its recovery** is also represented by **negative exponential distribution** with **a mean of 3 hours**.

Each packet experiences the propagation delay which is mentioned in 2^{nd} rule. (queueing delay is disregarded).

The chain of events should start at t_0.

PROGRAM FLOW:

1) Show random topology to user,
2) Allow user to select a router (by giving its number),
3) Show user how the selected router's routing table is completed,
4) Print a message to screen in every link failure and recovery (by giving link's number and attached routers' numbers to that link)
5) Show user the routing table of the node selected at Step 2 following the link failure and recovery (after 1 minute), wait for a response from the user when your program prints an updated routing table.
6) Your program should simulate events in the network for 40 hours,
7) Calculate and print the mean time between failures and the mean time for recoveries when your simulation is finished.

HOW TO GENERATE RANDOM NUMBERS ACCORDING TO NEGATIVE EXPONENTIAL DISTRIBUTION

First, you have to generate a uniformly distributed random number between (0,1). Let's call this number as "p". Then you will solve following equation and obtain "t":

$$1 - e^{-t/a} = p$$

Here, $1 - e^{-t/a}$ is **cumulative distribution function** of negative exponential distribution, where "a" is mean.

This "t" variable indicates that next event will occur "t" hours later.

PROGRAMMING LANGUAGE and DEVELOPMENT ENVIRONMENT

You can either choose JAVA or C++ as programming language.

1) JAVA

Your source files must be compiled with JDK 7u9 (which is the latest stable version of JDK by now). All your source files must be in a directory which is named as your student number. Inside the directory, your files must be compiled with this command:

javac *.java

And your program must run with this command:

java simulate

Your program will be tested under JRE 7u9.

2) C++

Your code must be compiled and tested in ITU's Linux Server using **g++**. If you have more than one source file, please attach your makefile.

REPORT

Give a brief explanation about your code. Please write your own well written comments about simulation results. It is much more important than explanation of your code.

Make sure that

1) Your code must be well commented.

2) The program must be written using object oriented principles.

Keep in mind that academic dishonesty including but not limited to cheating, plagiarism, collaboration is unacceptable and subject to disciplinary actions.