Project: CE 352 Computer Networks
German Jordanian University
Winter-2015

This project is about learning key concepts in computer networking through a series of network simulation tasks a using NS2 simulator.

In this course , you are going to perform three task assignments. At the end of each task assignment, you will need to submit the following by email (sahel.alouneh@gju.edu.jo):

1- The simulation script (TCL files)
2- The results of the simulation
3- A short description of what was done and analysis of the results

You will also be required to demonstrate through Q&A your knowledge of the task that was completed.

Network Simulation and the Ns2 Simulator

Performance evaluation is a critical component of systems research that allows the evaluation of new ideas, identification of problems and bottlenecks and optimization of existing systems. There are three general approaches to performance evaluation: (1) prototyping: build it (or a scaled down version of it) and see how it works; (2) analytical modeling: build a mathematical model of it and use it to analyze the system; and (3) Simulation: build a software model of the system.

Prototyping is often not feasible or time consuming especially for large scale systems; it also provides limited controllability and observability. Similarly, analytical modeling cannot capture highly complex systems. Thus, simulation has emerged as an attractive alternative that is heavily used in performance evaluation of computer systems. For this reasons, we are doing a common set of projects in a simulator, before we break off into research or non-research project (your choice) for the final project.

Network simulation enables us to predict behavior of a large-scale and complex network system such as the Internet at low cost under different configurations of interest and over long periods. Many network simulators, such as NS-2, SSFNet, Opnet, Qualnet, etc., are widely available. We will use NS-2 for this project. NS-2 is a discrete event simulator written in C++, with an OTcl interpreter shell as the user interface that allows the input model files (Tcl scripts) to be executed. Most network elements in NS-2 are developed as classes, in object-oriented fashion. The simulator supports a class hierarchy in C++, and a very similar class hierarchy in OTcl. The root of this class hierarchy is the TclObject in OTcl. Users create new simulator objects through the OTcl interpreter, and then these objects are mirrored by corresponding objects in the class hierarchy in C++. NS2 provides substantial support for simulation of TCP, routing algorithms, queuing algorithms, and multicast protocols over wired and wireless (local and satellite) networks, etc. It is freely distributed, and all source code is available.

Developing new networking protocols and creating simulation scripts are complex tasks, which require understanding of the NS-2 class hierarchy, C++, and Tcl programming. However, in this project, you only need to design and run simulations in Tcl scripts using the simulator objects without changing NS2 core components such as class hierarchy, event schedulers, and other network building blocks. You should put all your results in the files with given names. For your report, you do not need to rewrite the assumptions given in this document, but should provide any information not already given to you.

Task 1

The objective of task 1 is to learn how to install and run the Ns2 simulator and gain basic knowledge of the simulation scripts. For this task you will need to simulate a simple ethernet network with four nodes as shown in Figure 1.
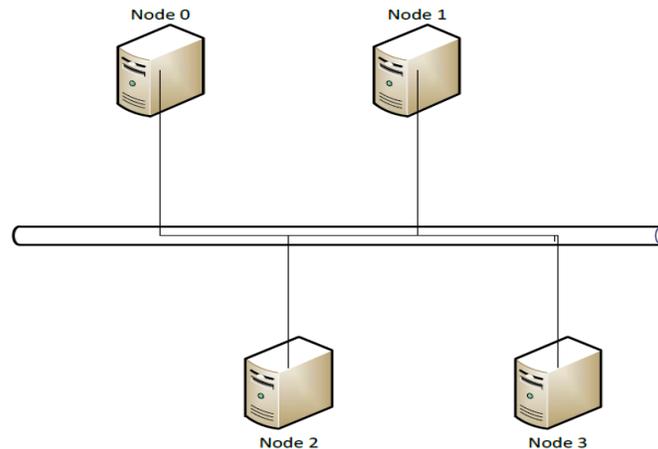
The Ethernet should have the following parameters:
- Bandwidth: 2Mbps.
- Delay: 10ms
- MAC: CSMA/CD
- Use default values for other Ethernet parameters.

Your simulation should create traffic flows from node 0 to node 2 and from node 1 to node 3 with the following parameters:
- Traffic type: Constant Bit Rate (CBR)
- Packet Size: 2000 Bytes
- Time between packets: 0.1ms

- You should run the simulation for 10 seconds and print the following:
  - Total number of bytes sent by nodes 0 and 1.
  - Total number of bytes received by nodes 2 and 3.

- Run two extra simulation runs by changing the packet Size to 20000 bytes and 200 bytes. (what can be the difference).
- Run the same experiment and change the bandwidth to be 100Kbps. How this change affect your results.

Due Date

This task is due before 9:30am on May 29. You must email the assignment before you come to class. No excuses about 'loss of work due to machine failures' or 'no internet service to email the completed assignments or install the software'. Any email received in my inbox after 9:30am will automatically be deleted. Start early to ensure you can run the software.

The new task will be announced on e-learning in beginning of December.

OSSCOM