CNT5505 Programming Assignment No. 2: Performance of MAC Protocols

(This programming assignment can be done either individually, or by a group of two. Due November 6)

Purpose

- Implement simulation programs to evaluate MAC protocols including Aloha, Slotted Aloha, and CSMA 1-persistent.
- Get better understanding of the MAC protocols

Description

In this assignment, you will write programs to study the performance of MAC protocols. The first program is a traffic generator that generates synthetic **random** traffic and store the traffic in a file. The first line of the traffic file contains a number telling the total number of packets in the file. After that, the traffic file contains lines of the formats

pkt_id src_node dst_node pkt_size time

and are sorted based on the *time*. Each line represents a packet of size *pkt_size* in bits from *src_node* to *dst_node* that is ready to be sent at time *time*. The *pkt_id* must be unique for each packet in the file. The traffic generator should execute with the following parameters

./traffic_generator num_node pkt_size offered_load num_pkts_per_node [seed]

We will assume that our wireless system operates at a speed of 1Mbps. The first parameter *num_node* is the number of stations in the system that can send and receive packets. The value of this parameter is from 2 to 200. The second parameter *pkt_size* is the packet size in bits. The value of this parameter is from 100 to 2000. The third parameter the overall *offered_load.* It value is from 0.01 to 10. The last parameter is the number of packets sent from each station in the traffic file. All stations will send the same number of packets with exactly the same characteristics.

The unit time is set to be 1μs. The time starts from 0 (and only have integer values). A time value of 10000 means time 10000μs from the beginning. When generating the traffic, all stations should have the same probability to send packets throughout the time. For each packet sent from a station, the destination should be uniform randomly selected from the all nodes other than the source (all other nodes have equal probability to be the destination). For each station, since the unit time is 1μs and the bit rate is 1Mbps (each bit will take 1μs time to send) the average inter-packet *gap* (in the unit of μs) for each station can be computed from the *offered_load by solving the following:*

$$\frac{pkt\_size}{pkt\_size + gap} = \frac{offered\_load}{num\_node}$$

The gap may follow different probability distributions in practice. In this assignment, you only need to implement uniform distribution in the range [0, 2*gap], which is a random number between 0 to 2*gap (the average will be gap). The program has an optional *seed* parameter that is for setting the seed for random number generator. If this parameter is not provided, the program should generate a different traffic with the same statistical characteristics each time it runs. If the same seed is provided, the program should generate exactly the same file every time.

The second part of the program is to simulate Aloha, Slotted Aloha, and CSMA 1-persistent. You might write one program that can simulate all three schemes using a command line flag or write three programs, one simulating one MAC protocol. The program must simulate along the timeline, identify all important events including: packet starts sending; packet finishes sending; collisions, and mark each packet as successful or collision according to the protocol. A sample traffic file and sample outputs from some protocols will be provided. For the sample traffic file(s), your program's output must be exactly the same as the sample outputs (assuming no error is found). The simulator should be able to handle traffic file of any size.

After you finish your program, you must report a throughput study of the protocols. In particular, you must at least plot the relation between throughput and offered load, and compare your result to the results in Figure 4-4 in the book. Note that in order for the plots to be statistically sound, you must do experiments with a sufficiently large number of packets (e.g. 100000 or more).

Grading Policy

You can write the program in any language that is available on linprog, where your program will be graded. You must supply a README file describing how to run your program.

- Program compiles with no compiler error (10 points) -- programs with compiling error, get 0 points.
- Assignment demonstration (25 points, 5 points for each component: traffic generation, Aloha, Slotted Aloha, CSMA 1-persistent, CSMA 0.5-persistent).
- README file, make file, etc (5 points).
- Correct traffic generator (10 points)
- Correct Aloha simulator (10 points)
- Correct Slotted Aloha simulator (15 points)
- Correct CSMA 1-persistent simulator (15 points)
- Project report for performance study (10 points)

- Being the first one to report an error in the sample output (5 extra points per bug).
- For the simulators, your program will get 60% of the points for passing a basic test (similar to the sample traffic file). The reminding 40% will be subject to more extensive testing.
- Your report should resemble any scientific report of a Computer Science experiment: it first describes how things are done in the experiments (including how you simulate the protocols) and the setting of the experiments before reporting results. **Readers of a good report should be able to reproduce the results in the report (from the information given in the report) if they choose to repeat the experiments.**

Deadline and materials to be submitted

- Submit all files including readme and makefile through campus.fsu.edu by Monday November 6. You will sign up for a demo time in the next class.

Miscellaneous

All programs submitted will be checked by a plagiarism detection software. Honor code policy will be strictly enforced. Write the code by yourself and protect your code.

The sample outputs are provided as an example, and do not test the programs thoroughly. You must have more test cases to make sure that your code works correctly. And your code will be tested much more thoroughly when it is graded.