CS6040 – Router Architectures and Algorithms
Jul.-Nov. 2024, Prof. Krishna M. Sivalingam
Lab 3 - Queuing in a Packet Switch
Due Date: Oct. 6, 2024, 11 PM, on Moodle
Programming Language of your choice.
NO Copying/Code-Sharing allowed from anywhere. Code written should be your own, except for use of Standard Templates. Students who violate this policy will receive an 'U' grade in the course.

1 Objective

The purpose of this lab experiment is to understand the performance of queuing in a packet switch. The program will implement different types of queuing mechanisms, as described below.

2 Inputs

The command line will specify the following parameters:

```
% ./routing -N switchportcount -B buffersize -p packetgenprob \
 -q NOQ | INQ | CIOQ -K knockout -L inpq -o outputfile -T maxslots
```

- All packets are taken to be of the same length. Further, slotted time is assumed, where one time slot equals the transmission time of one packet.
- The switch is assumed to have N input and N output ports. The ports are numbered 0 through $N-1$; Default value is 8.
- Each port will hold up to B fixed-length packets; Default value of B is 10.
- *packetgenprob* denotes the probability that an input port will generate a packet in a given slot; Default value is 0.5.
- The '-q' argument specifies the queue type; the output file will contain the output generated by the program. Default is: IQ.
- The *maxslots* argument specifies the simulation time, in slots; Default value is : 10,000.

3 Switch Operation

The (simulator) program will consist of three phases, repeated in a loop until termination: Phase 1 corresponds to traffic generation, Phase 2 corresponds to packet scheduling, and Phase 3 corresponds to packet transmission. All the three phases will take place at the beginning of each time slot, and processed one after the other.

Initially, all the packet queues are assumed to be empty.

3.1 Traffic Generation

In this phase, each port will generate a packet with probability *packetgenprob*. The destination port for each packet is randomly selected, with uniform probability, from the set of all output ports. Let t denote the current time slot. The start time of each packet is randomly set between $t + 0.001$ and $t + 0.01$. For delay purposes, ignore this offset and assume that the arrival time is t.

3.2 Scheduling

In this phase, the packets generated in the previous phase are handled.

NOQ: There is no buffer in the system. In each slot, contention for the same output port is resolved by randomly choosing any one of the input ports with equal probability.

INQ: For each packet generated, if there is no contention for its desired output port, it is selected for transmission and placed in the corresponding output port's buffer. For packets contending for the same output port, the packet from **the input port that has the lowest port number** is randomly selected for transmission and placed in the corresponding output port's buffer; the other packets are queued at the corresponding input port. For this case, $L = 1$, even if a different value is input on the command line.

CIOQ : L packets can be buffered at each input Q; the switch backplane is K times faster than the input lines. In each slot, a set of L packets per input port are considered, and an iterated matching algorithm (of your choice) is executed to generate the set of possible packets for transmission. The packets not selected for transmission stay in the corresponding input queue. A maximum of K packets (per output port) that arrive in a given slot are queued (based on packet arrival time) at the corresponding output port. If two or more packets have the same arrival time, the packets can be queued in any order. If more than K packets arrive in a slot for a particular output port, then K packets are randomly selected for buffering, and the remaining packets are dropped at their corresponding input ports. The default values are: $L = K = \lceil 0.4N \rceil$.

3.3 Transmission

At each output port, the packet at the head of the queue is transmitted. Packet delay is calculated as the difference between transmission completion time and the packet arrival time.

4 Outputs

The performance metrics to be measured are as follows. These are calculated at simulation termination.

- **Average Packet Delay:** This value represents the mean of packet delay computed for all transmitted packets, across all ports.
- **Average Port Utilization:** For each port, the utilization is defined as the fraction of the time that the output port has been used for transmitting a packet, with respect to the entire simulation duration.

  The average port utilization is the mean value over all the output port utilizations.

- **CIOQ Drop Probability:** The probability, per slot, that more than K packets were generated for a output port. For example, if in a given slot, more than K packets were generated for 3 out of 8 output ports, then the probability for that slot is 0.375. Report the average probability over all slots for the simulation duration.

The program, upon termination, will output a line with the specified values (separated by one TAB) as given in the tables below, that will be **appended** to the specified output file given on the command line.

Output format for NOQ and INQ:

N	p	Q Type	Avg PD	Avg LU

Output format for CIOQ:

N	p	L	K	Q Type	Avg PD	Avg LU	Avg Drop Prob

Report: A technical report (only in PDF) that compares the performance of the above schemes needs to be generated. The specific algorithm used for selecting up to N packets from LN packets at the input ports will be clearly documented in an algorithmic format. You must also include an illustrative example and approximate time complexity analysis. If you are implementing an existing algorithm, you should provide the reference to the same.

The report will consist of: performance graphs comparing average packet delay and average link utilization for different values of $N \in \{32, 64\}$, $p \in \{0.4, 0.6, 0.8, 1.0\}$, and $K \in \{0.4, 0.7, 1.0\} \times N$ (for CIOQ). Repeat this for different values of L, $L \in \{0.4, 0.7, 1.0\} \times N$ (for CIOQ).

The quality of the report is very important, i.e. correct English sentences, easily viewable performance graphs and technically detailed explanation of graphs (include comparison to known theoretical switch utilization values).

Please submit a tar-gzipped file including the source code files, sample output files, technical report, README and COMMENTS files.

5 Grading Criteria

- NOQ: 15 points
- INQ: 20 points
- CIOQ: 45 points
- Report: 15 points
- Demo and Viva Voce: 5 points; (No Demo; No grade)

6 Hints

- The use of random variables and related functions will be needed. You may use the functions discussed in 'man 2 random'. Remember to initialize the random number generator using the "srandom()" function. Alternatives such as arc4random() can also be considered.

- You can use graph drawing packages based on Excel, gnuplot, R, matlab, xgraph, etc. to generate the performance graphs.
- Optional: Compare the simulation results to the theoretical results discussed in class, for better understanding of the mechanisms.
- Start your work early.