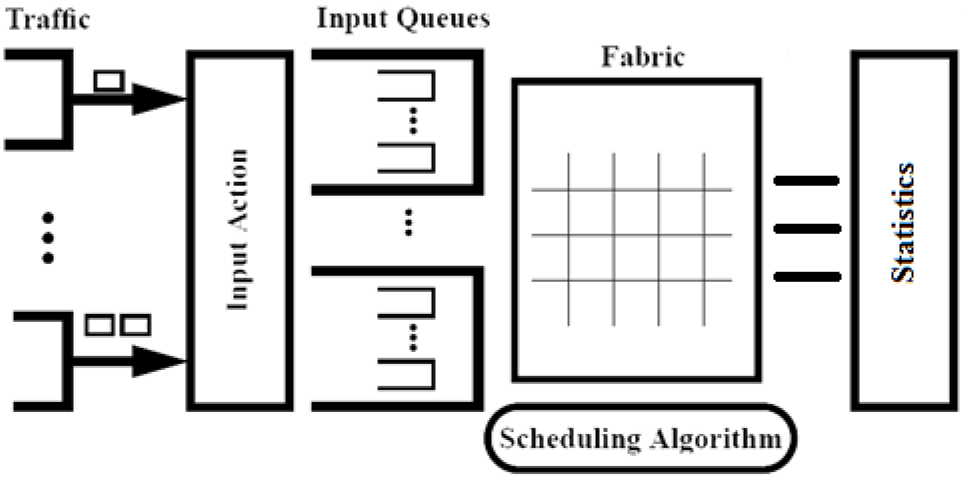
CSIT 5600 Programming Project (Spring 2015)

Due: 5pm, May 8th

Introduction

The aim of this programming assignment is to implement a Network Switch Simulator (i.e. `SwitchSim`), which emulates the behavior of switch and allows the experimentation on the performance analysis of crossbar scheduling algorithms.

Your simulator should emulate a VOQ crossbar switch structure as shown below:



The simulated switch works in a time slot manner. The whole system progresses one slot time (i.e. ATM cell) at a time. For offered simulation load in every time slot system checks for new arrivals of cells to the inputs and decides which cells will be forwarded from inputs to outputs trough switching fabric. Simulator should have a modular architecture. Every module has different functionality. The functionality modules are: traffic model, queuing policy, switch fabric, and scheduling algorithm. It provide overall statistics (i.e. average latency per cell, throughput) after simulation.

Switch Configuration

Users should have flexibility to configure the switch, for example, including switch size, fabric, input traffic, scheduling algorithm and buffer size. Such switch configuration is specified via an input configuration file, or command line argument. To simplify your simulator, we assume 1) numbers of input and output ports are same, 2) the traffic is admissible, no input and/or output is overloaded, 3) only unicast traffic is considered.

Input Traffic

Real life traffic are modelled to different traffic types, including

Uniform traffic: Bernoulli i.i.d. uniform traffic
Bursty traffic: two-state Markov-modulate traffic
Non-uniform traffic: unbalanced, Chang's diagonal and hot spot non-uniform traffic

You simulator should be able to generate uniform traffic as minimum requirement. Experimentation examines the switch performance under different traffic load and pattern.

Queuing Policy

By default, the input and output queues are first-come-first-served (FCFS). Note virtual-output-queues (VOQs) reside at input ports, meaning there're N VOQs in each input for NxN switch. The simulator should be able to support queues of arbitrary size; however the user can set the maximum queue capacity via configuration file, or command line argument if needed.

Switch Fabric

The switching fabric is used for the input and output ports interconnection. The crossbar is a switching fabric which is modeled in this project because it is one of the most used fabrics for high performance switches. By default the crossbar fabric may setup the input and output interconnection once per time slot (speedup = 1).

Scheduling Algorithm

The switching fabric may use a specific scheduling algorithm to decide how to setup the input and output port connection, and therefore which cells to transfer in each slot time. You simulator should be able to support round-robin (RR) and iSLIP scheduling algorithm as minimum requirement.

Statistics

When the cell is generated, it carries information such as source input, destination output, and time tag. When it is finally switched out from the switch, at the output side, one can collect information such as the queuing delay. Collection of all cell status gives the performance statistic of the switch, such as average latency per cell, worst case latency, and system throughput. You may also monitor the average VOQ lengths during the simulation. Your simulator should be able to collect average latency per cell as minimum requirement.

Tasks

- Design and implement a complete Network Switch Simulator as described above. You're free to use any programming language you feel comfortable with, i.e. C/C++, Java, Python. But we do recommend a modular design which allows further extension in functionalities. The code implementation counts 50% of the final project score. Code that fails to execute receive 0 points.

- Run performance analysis of crossbar scheduling algorithms under different switch configurations and input traffic. Round-robin (RR) and iSLIP performance (in terms of average cell delay) with Bernoulli i.i.d. traffic is must. But you can implement and test more algorithms (i.e. FIRM, iLQF, Maximize) under variety of traffic pattern if interested. You may evaluate and compare the performance via different aspects (i.e. switch size, throughput, and average latency, synchronization of priority pointers). Refer to related papers to decide what kinds of statistic you may wish to collect and report.

Write a short report (less than five A4 pages excluding cover page) based on the observation above to give:

- General design and implementation of the simulator at high level
- Very simple user manual of the simulator
- Challenges met during the project and how you came up with solutions
- Performance analysis and/or comparison of different crossbar scheduling algorithms, preferably with experiment settings, and supported pilot graphs

The report and performance analysis counts 50% of the final project score.

Submission

Submit the executable program of the simulator and full set of source code, together with the report via LMES on or before **5pm, May 8th**. Please zip all files into a single .zip file.

Each group only needs to submit one copy. Make sure all group member names, student ids and emails appear on the cover page of the report.

Reference Papers:

Three papers on PIM, iSLIP and FIRM are uploaded in LMES. Read and use those papers as a starting point for your programming assignment. You may explore more papers related to the topic.

Grading Rubric

	Adequate	Excellent
Switch simulator	Program always works and meets the minimum requirement as specified. Input traffic, queues, switch fabric, scheduling algorithm, and statistic modules are correctly implemented.	Program allows supporting multiple scheduling algorithms in addition to RR and iSLIP. Various input traffic models are implemented (uniform, non-uniform, bursty). With modular design, it allows easy further extension, e.g. adding another scheduling algorithm, support fabric with speedup 2, etc. Statistic data are well thought and collected.
Report	Cover all required topics. The description of the general design of the simulator, user manual is clear and easy to understand. Challenges and solutions are well described. Pilot and analyze RR and iSLIP performance under different uniform traffic load and switch size. Pilot graphs which shows the simulation results.	All topics are well covered. The general design and user manual are abstracted at high level; while still emphasize the important details. Thoughtful simulations under different traffic patterns. Insightful and complete analysis on different factors that influence the performance. Nice pilot graphs that visualize the observation.