# 18756 Project4 Report

Shan Gao

## Introduction

In this project, I implemented a switch with both input queueing and output queueing. The switch is designed as a non-blocking switch and each input/output nic has its own FIFO queue. The number of input nics is the same with that of ouput nics.

For input queueing, when computer send packet to one of the switch's input nic, the packet will be saved in the nic's input buffer if the buffer is not full, otherwise the packet will be dropped. Every tock the switch will  scan all the nics' input buffer to find out whether there is a packet to be sent. The packet will be moved to the output nic if its destination is not the same with the packets that are sent in this tock. Otherwise this packet will wait for next tock. And for every tock, each input queue should send at most one packet. In order to be fair, I implement a mechanism kind of like Round Robin to let different input queueing send packet first for every tock.

For output queueing, every packet will be sent to output nics from input nics directly. If the output queue is full, the packet will be dropped.

## Analysis of Performance of Input Queueing and Output Queueing

### Impact of size of switch

Firstly, we test the performance of input queueing and output queueing under different size of switch. We choose average delay and throughput as metrics to analyze these two queueing methods.

For test cases, we let all the computers send packets to random destinations every tock. The total number of packets are sent is 1000000. And the input queueing and output queueing buffer size is 20.

The results of throughput:

|  | 2X2 | 4X4 | 8X8 | 16X16 | 32X32 | 64X64 |
|---|---|---|---|---|---|---|
| Input queueing | 75.00% | 65.52% | 61.85% | 60.11% | 59.40% | 59.09% |
| Output queueing | 98.81% | 98.14% | 97.84% | 97.70% | 97.65% | 97.61% |

The results of average delay:

|  | 2X2 | 4X4 | 8X8 | 16X16 | 32X32 | 64X64 |
|---|---|---|---|---|---|---|
| Input queueing | 26.67 | 30.52 | 32.33 | 33.26 | 33.66 | 33.83 |
| Output queueing | 10.61 | 10.59 | 10.50 | 10.52 | 10.57 | 10.54 |

From results, we can see that the throughput of input queueing is lower than that of output queueing, and the average delay of input queueing is bigger than that of output queueing.

For input queueing, if several packets from different input buffer have the same destination, only one of them can be sent to output nic, the others should wait for the next tock to be processed. Thus, the packet that is sent is blocking packets from other queues, which is called head-of-line blocking. This will lead to the input buffer overflow. So the throughput of input queueing is low because of head-of-line blocking, and so is the average delay. From the results, we can see that the throughput of 64X64 switch is 59%, which is very close to 58%.

For output queueing, all the packets are sent to the output queue from input nics, so there is no head-of-line blocking. The output buffer overflow only when the most of the packets' destination are the same for a period of time. The possibility of this phenomenon  is small because each output queue can definitely send one packet every tock if its buffer is not full, unlike the input queueueing. Thus, the throughput of output queueing is very high, which is close to 100%. For the same reason, the average delay of output queueing is low.

For switch size, the larger the switch is, the lower the throughput is. This trend is obvious in input

queueing. The reason is that when the switch size becomes larger, all the computers send packets at the same time, the input queue will be more likely blocked due to large traffic. The output queueing is nearly not affected because it doesn't have head-of-line blocking.

**Impact of distributions of input packets**

Then we test the performance of input queueing and output queueing under different distrbutions of input packets from computers. We also choose average delay and throughput as metrics to analyze these two queueing methods.

For test cases, we let half of the computers send packets to half of the destinations randomly, and let the remaining computers send packets to all the destinations randomly. The total number of packets are sent is 1000000. And the switch size is 64X64.

The results of throughput:

|  | Uniform distribution | Test case |
|---|---|---|
| Input queueing | 59.08% | 29.75% |
| Output queueing | 97.67% | 48.76% |

The results of average delay:

|  | Uniform distribution | Test case |
|---|---|---|
| Input queueing | 33.83 | 33.59 |
| Output queueing | 10.56 | 10.51 |

There are various kind of distribution of input packets, we only choose one to test the performance of input/output queueing when the traffic is not uniform. As we can seen from the results, the throughput of test case is lower than uniform distribution for both input queueing and output queueing. This is because that when the traffic is not uniform, the buffers that receive more packets would be more likely to overflow and drop packests, which leads to lower throughput. As for the average delay, it mainly has relationship with switch size and buffer size, so it doesn't change in these two cases.

In conclusion, when the traffic is not uniform, both of the throughput of input queueing and output

queueing will decrease.

**Impact of buffer size**

Finally, we test the performance of input queueing and output queueing under different size of input/output buffer. We also choose average delay and throughput as metrics to analyze these two queueing methods.

For test cases, we let all the computers send packets to random destinations every tock. The total number of packets are sent is 1000000. And the switch size is 64X64.

The results of throughput:

|  | 5 | 20 | 50 | 100 |
|---|---|---|---|---|
| Input queueing | 59.00% | 59.08% | 59.21% | 59.60% |
| Output queueing | 90.76% | 97.61% | 99.04% | 99.59% |

The results of average delay:

|  | 5 | 20 | 50 | 100 |
|---|---|---|---|---|
| Input queueing | 8.47 | 33.83 | 84.33 | 167.38 |
| Output queueing | 3.06 | 10.52 | 24.94 | 43.80 |

For input queueing, the throughput is not affected by the buffer size because of head-of-line blocking. But the average delay is increasing with the buffer size. This is because when the buffer size gets bigger, more packets will wait in the buffer blocking for longer time, so the average delay of input queueing becomes very big. This phenomenon is enlarged by head-of-line blocking.

For output queueing, the throughput is increasing with the buffer size. Because when the buffer size

gets bigger, the packets would less likely be dropped, so the throughput gets bigger too. The reason of increasing average delay is the same with the reason of that of input queueing.