CS607: Technical Report
Assignment 2 - Queueing in a Packet Switch

B Siddharth Prabhu
`200010003@iitdh.ac.in`

22 October 2023

1 Introduction

The objective of this assignment is to understand the basic concepts of the working of an input queue (INQ), an output queue (KOUQ) and ISLIP protocol which uses the concepts of Virtual Output Queue (VOQ). The algorithms were briefed in the PDF file provided, so their working need not be elaborated here. Some important parameters are mentioned below:

- `-N` (switchportcount) : Number of input/output ports (assumed to be equal). [default = 8]
- `-B` (buffersize) : Size of buffer in each port. [default = 4]
- `-p` (packetgenprob) : Probability of a packet being generated at some port, at some timeslot. [default = 0.5]
- `-queue` (qType) : Scheduling algorithm used (INQ/KOUQ/ISLIP). [default = INQ]
- `-K` (knockout) : Maximum packets that can be queued to an output port in KOUQ, as K$\times$N. [default = 0.6]
- `-out` (outputfile) : Output file to append output string to. [default = output.txt]
- `-T` (maxtimeslots) : Number of timeslots to run the simulation for, before terminating and computing performance. [default = 10000]

2 Running the Program

First, the user must compile the program `main.cpp` using `g++ main.cpp -o main.o` . The created executable is then used to run the scheduling algorithms for different parameter values. To run for a specific set of parameters, the syntax is:

```
./main.o -N switchportcount -B buffersize -p packetgenprob -queue INQ | KOUQ | ISLIP -K
knockout -out outputfile -T maxtimeslots
```

This process has been semi-automated in various helper script files that have been used to obtain data for the several graphs plotted. To run a few sample cases, try out `bash run.sh` to run the provided script file on a few different inputs. The output will be created in the file `output.txt` in the same directory as the program file. Also, note the following:

- Sample output related to the helper scripts are all included in the folder `output/` .
- Images are all included in this report with some inferences.
- `.ipynb` files that generated these graphs are documented in the folder `helper_scripts/` .

3 Deliverables: Graphs

(The analysis for each graph has been done just above it, in most cases.)

3.1 Graph 1: INQ

In Figure (1), we observe that average link utilization generally decreases with an increase in N. If buffer size is higher, then link utilization shifts upwards for some fixed N. The increase for B=4 between N=[16,32] appears to be the anomaly, possibly due to randomness, as a result of packet generation or choice of packet for transmission.

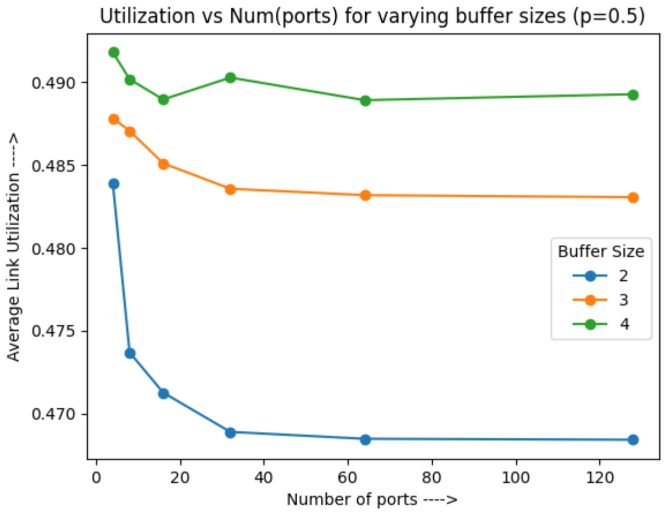

Figure 1: Graph 1

3.2 Graph 2: KOUQ

In Figure (2), we can see the average packet delay increasing alongside N, initially with a sharp slope, gradually levelling out. It makes sense for packet delays to increase then (somewhat) saturate, since the delay values initially get large, eventually levelling out with establishment of a 'steady state' in terms of how long a packet waits.

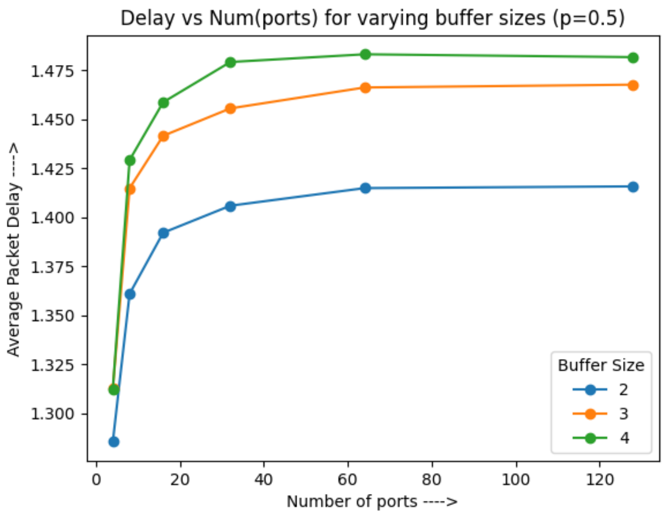

Figure 2: Graph 2

3.3 Graph 3: KOUQ

As we increase the number of ports, the link utilization in KOUQ appears to increase a bit before stabilizing/gradually decreasing, as observed in Figure (3). However, for buffers of very small size, like 2, the decrease in link utilization appears to begin from the start. It's possible that the initial increase for the larger buffer sizes is related to the increase anomaly in Figure (1).

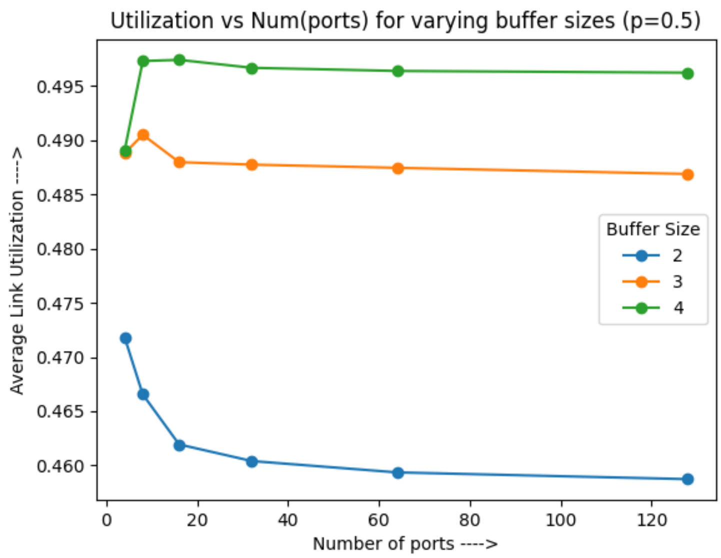

Figure 3: Graph 3

3.4 Graph 4: KOUQ

All of the knockout values in Figure (4) appear to result in a similar trend for average packet delay, and it looks like the delays for higher knockout values will (if the graph is extrapolated,) exceed that of the lower ones. In general, there is a sharp initial increase, followed by slower and slower increases in packet delay, as the number of ports increases.

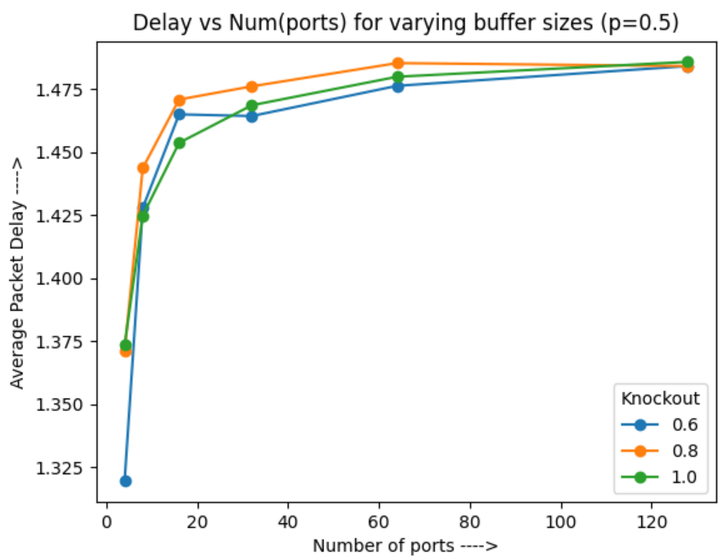

Figure 4: Graph 4

3.5 Graph 5: KOUQ

With an assumption made that the question here asks us to recreate *Graph 3*, but for different knockout values, the graphs shown in Figures (5), (6), and (7) demonstrate trends in link utilization with varying number of ports, over different buffer sizes. The three graphs seem to have the same trends as described in Section (3.3). The observables appear to also suggest that the initial increases observed thus far are a product of randomness. The sources of this randomness lie in the packet generation, as well as the choice of which packets to send/drop in case of a knockout excess.

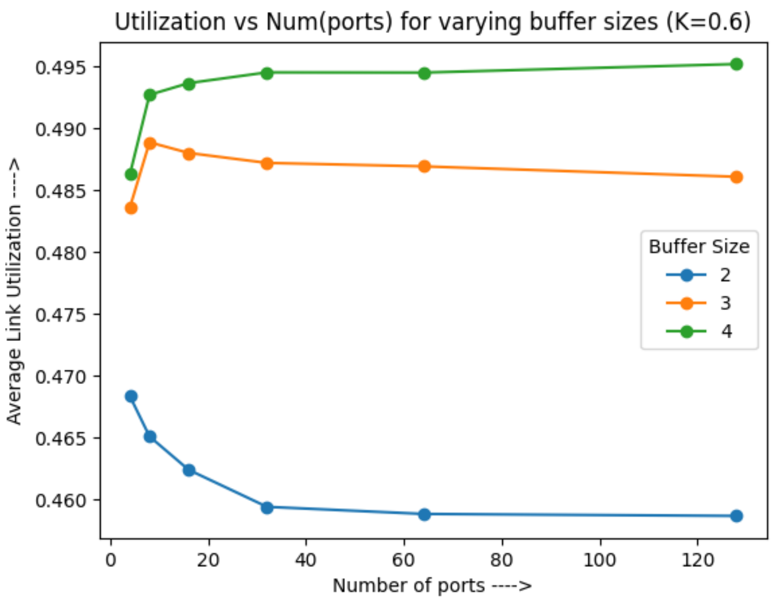

Figure 5: Graph 5 ($K = 0.6$)

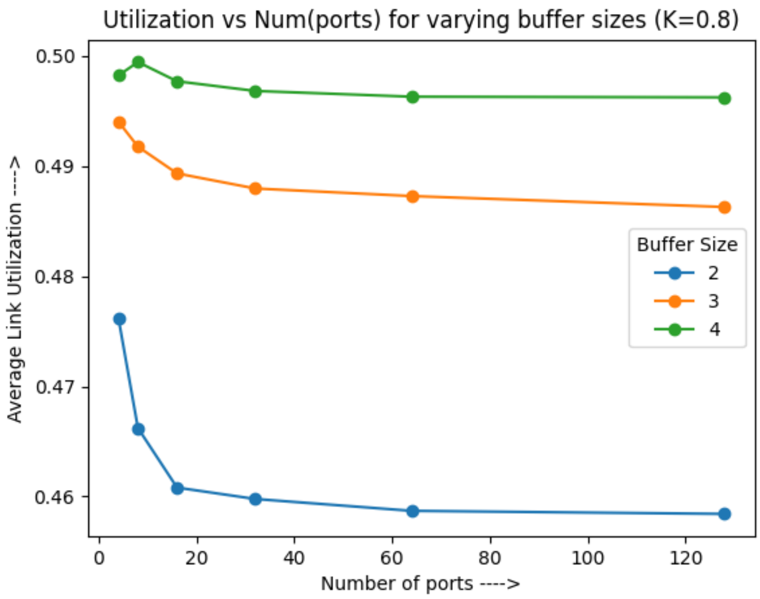

Figure 6: Graph 5 ($K = 0.8$)

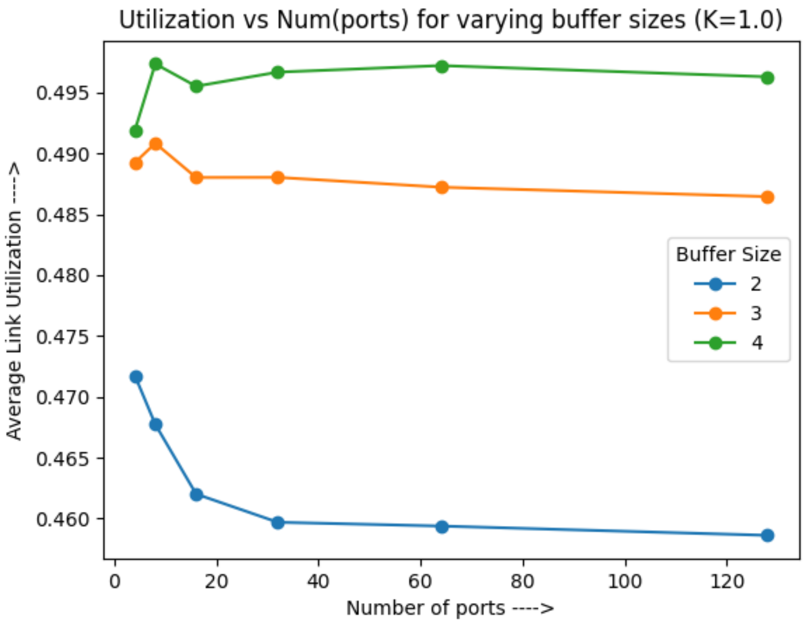

Figure 7: Graph 5 ($K = 1.0$)

3.6 Graph 6: ISLIP

As can be seen in Figure (8), there is initially a very slight rise in packet delay with an increase in the number of ports, followed by a slow and gradual decrease. For smaller buffer sizes, the average packet delays are smaller, since the larger buffers are likely to contain older packets. In general, average packet delay appears to increase as the buffer size is increased.

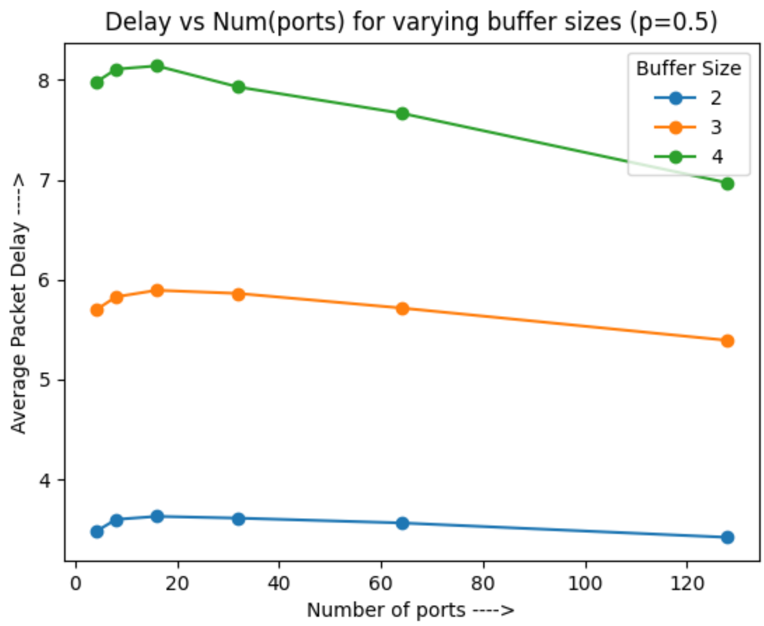

Figure 8: Graph 6

3.7 Graph 7: ISLIP

In Figure (9), we observe an initial sudden drop in link utilization with increase in number of ports for the ISLIP algorithm. After $N > 40$, the values of link utilization seem to rise again, with the smaller buffer size having the slowest increase in link utilization. The larger buffer seems to lead to lower utilization for the smaller cases ($N < 20$), but this changes as it overtakes the other buffer sizes as the ports increase in number.

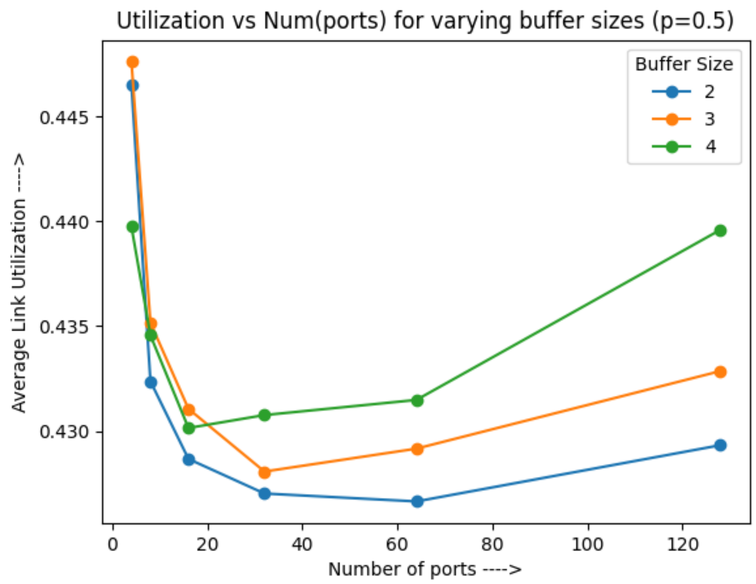

Figure 9: Graph 7

3.8 Graph 8: Comparing all algorithms

(An explanation for this section will be presented after its three figures.)

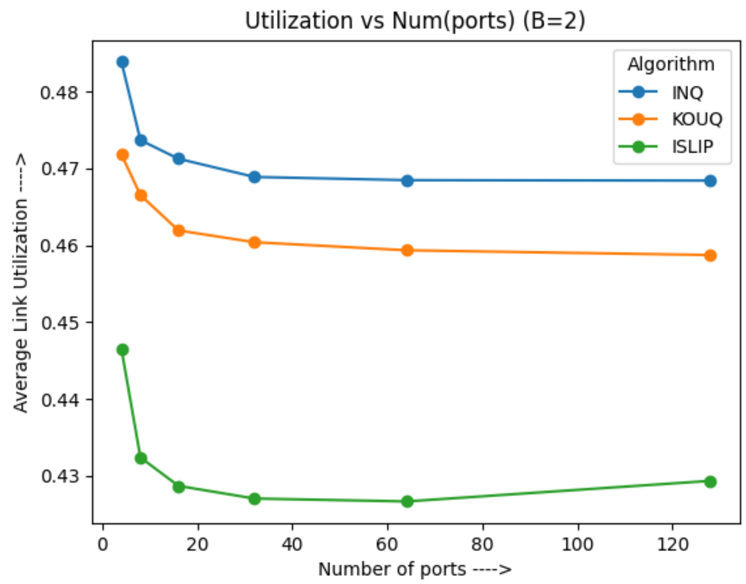

Figure 10: Graph 8 ($B = 2$)

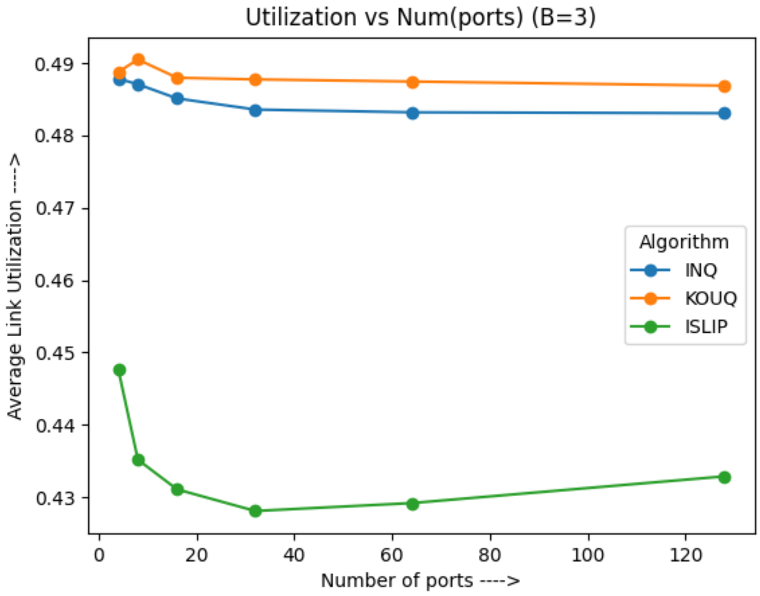

Figure 11: Graph 8 ($B = 3$)

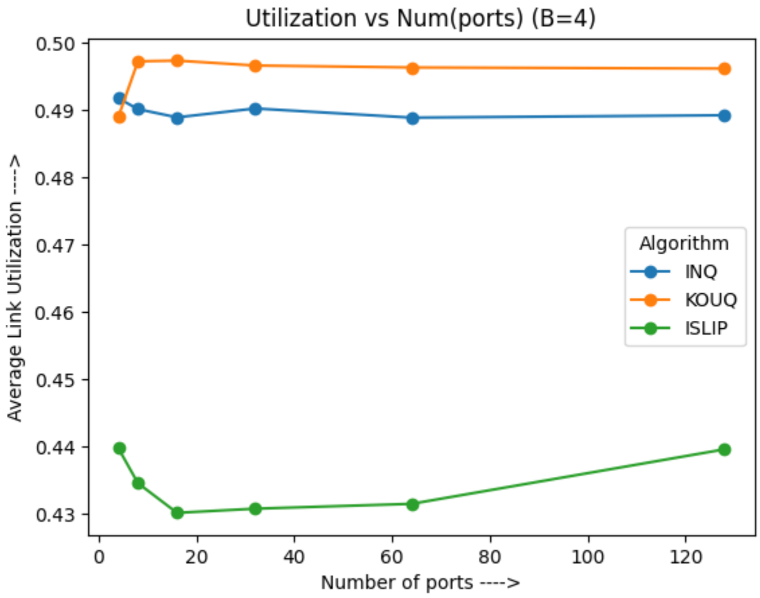

Figure 12: Graph 8 ($B = 4$)

Each of the above figures, viz. Figures (10), (11), and (12) depict the trends in average link utilization with respect to number of ports for the three algorithms implemented. Here, we combine our knowledge from the Figures (1), (3), and (7), while varying the size of the buffer $B \in \{2, 3, 4\}$. The trend stays largely intact, with ISLIP generally causing smallest average link utilization (due to its parallel approach!), while INQ and KOUQ seem to have switched places from $B = 2$ to $B = 3$.

In almost all cases, there is a net initial decrease in utilization For ISLIP, this becomes a gradual increase as the number of ports increases; for KOUQ, the case of $N = 4$ seems to cause a seemingly abnormal low link utilization as the buffer size increases, but this may be due to the buffer being mostly empty throughout the simulation for these specific cases. INQ, however, seems relatively stable, apart from an initial drop in the $B = 2$ case.

3.9 Graph 9: Comparing all algorithms

In graph (13), we see an interesting set of trends. INQ and KOUQ have more delays with an increase in packet generation probability, with INQ outpacing KOUQ. Meanwhile, ISLIP experiences a sharp and sudden increase in delay between $p = 0.1$ and $p = 0.2$. However, this sharp increase is most likely due to the randomness as a result of grant conflicts. It must be noted that, in some trials, the value of average packet delay at $p = 0.1$ can be as low as 1.87 or as high as 8.84. As the probability rises and packets are more 'surely' generated, the delay that arises decreases due to the general efficiency of ISLIP. Note that the value set for B, N, K may be responsible for the displayed trend.

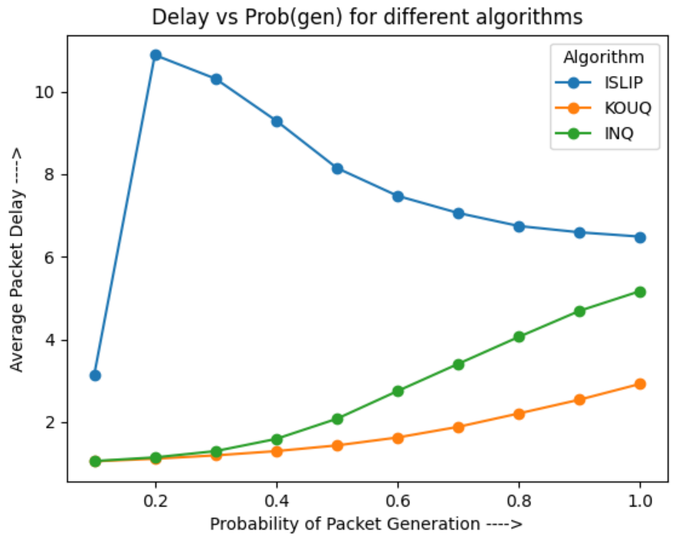

Figure 13: Graph 9 ($B = 4, N = 8, K = 0.6$)

3.10 Graph 10: Comparing all algorithms

Finally, we arrive at Figure (14). Here, we observe that, as packet generation probability increases, the link utilization increases for all three queueing algorithms. ISLIP leads to lowest average link utilization, due to its inherent parallelism, while KOUQ has the highest average link utilization, perhaps due to packets retransmissions as a result of drops. Overall, INQ's values for link utilization seem to be in between that of the other two algorithms.

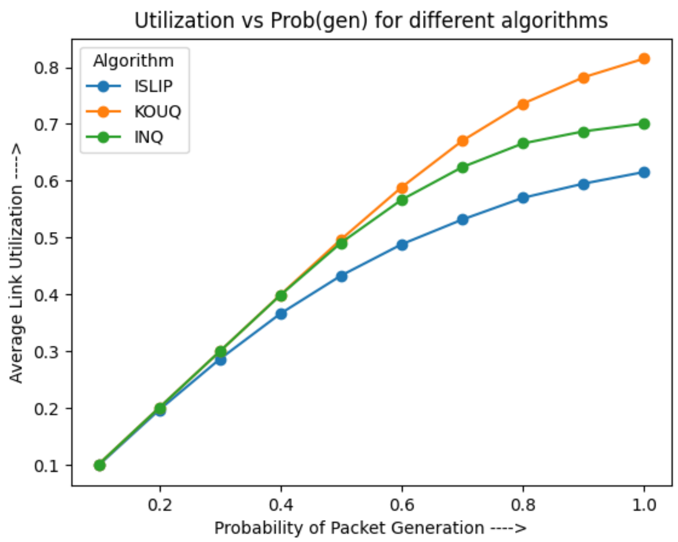

Figure 14: Graph 10 ($B = 4, N = 8, K = 0.6$)

4 Observations and Conclusions

The following are the conclusions that can be drawn from this assignment, from the graphs generated above, in brief:

- As the number of ports increases, the average packet delay seems to increase (INQ/KOUQ). For ISLIP, there is an initial bump followed by a gradual decrease.
- As the number of ports increases, the link utilization generally decreases, with a possible initial increase (INQ/KOUQ), although ISLIP provides a more complex relation between the two.
- Link utilization is poorest for ISLIP scheduling algorithm across all inspected values of buffer size.
- For higher probability of packet generation, KOUQ provides the highest link utilization.
- ISLIP is less likely to cause starvation of a port due to the embedded round-robin logic within its granting phase.