Assignment 2

CS:544 Topics in Networks

Vandana Mishra (194101055)

Prayas Barahate (194101013)

INTRODUCTION

In this assignment , we have studied and implemented some scheduling algorithms in packet switching. The three studied algorithms are:

1. INQ
2. KOUQ
3. ISLIP

After implementing these algorithms, we have analyzed their performance based on the following parameters:

- N: Number of input, output ports
- B: Buffer Size
- K: Knockout value
- Link Utilisation
- Average Packet delay
- Standard deviation of packets

Comparing these parameters , we have analyzed that which algorithm performs better than its competitors given a particular scenario. In order to do so, we will first present graphical representation of every algorithm. Then we will compare all three with each other and then arrive at our conclusion.

INQ

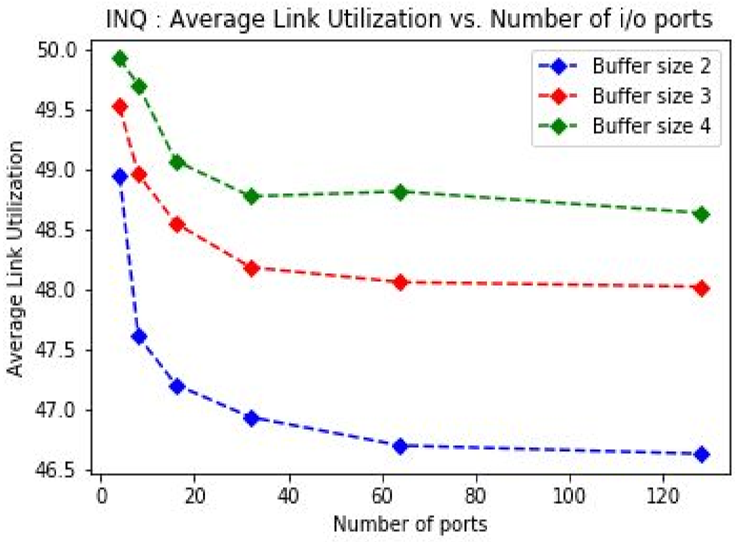
Packet Generation :

There are N queues. Each queue stores the arrival time and destination port of the packets generated at that input port. Inorder to generate a packet with some given probability, we will generate a random number between 0 & 1 . Now if this number satisfies our probability constraint we will generate a destination port and place this packet with current time and destination in its input queue if input queue is not full.

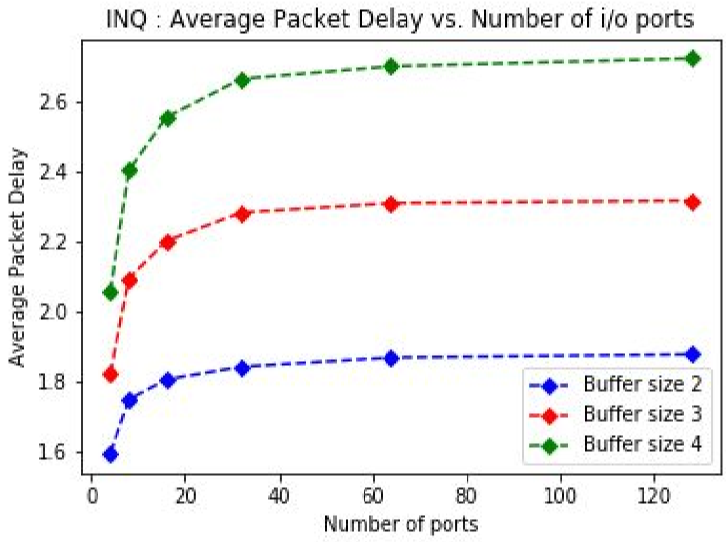
Scheduling:

If an output port has a single request, schedule it. If it has multiple requests, then we randomly select which input port's request is to be scheduled.

Link utilisation vs N and average delay vs N for buffer size 2,3 & 4



Link utilisation vs N



Average delay vs N

Analysis:

Link utilisation decreases whereas Average delay increases as the number of ports increases.

KOUQ

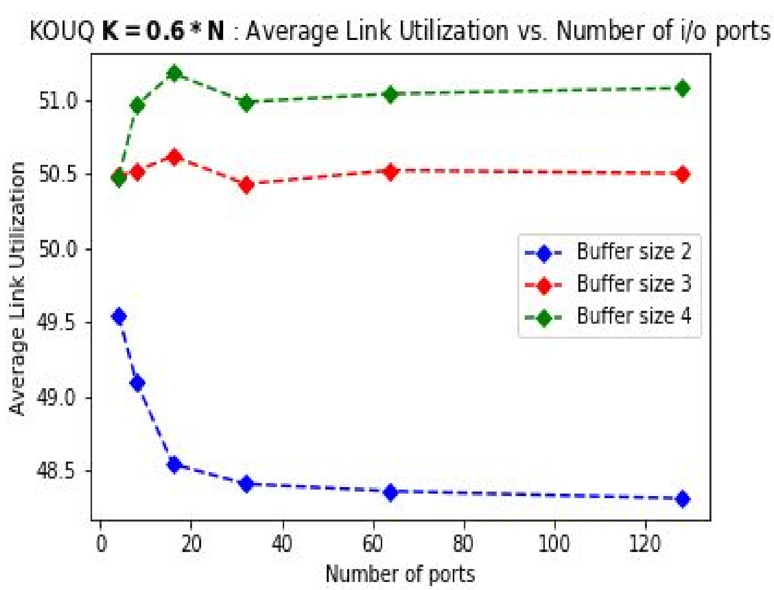
<u>Packet Generation :</u>

There are N queues, one for each output port. Each queue stores the arrival time of the packets generated for that output port. Inorder to generate a packet with some given probability, we will generate a random number between 0 & 1 . Now if this number satisfies our probability constraint, we will place this packet in its output queue if the output queue is not full or the number of packets generated for this output port is less than K.

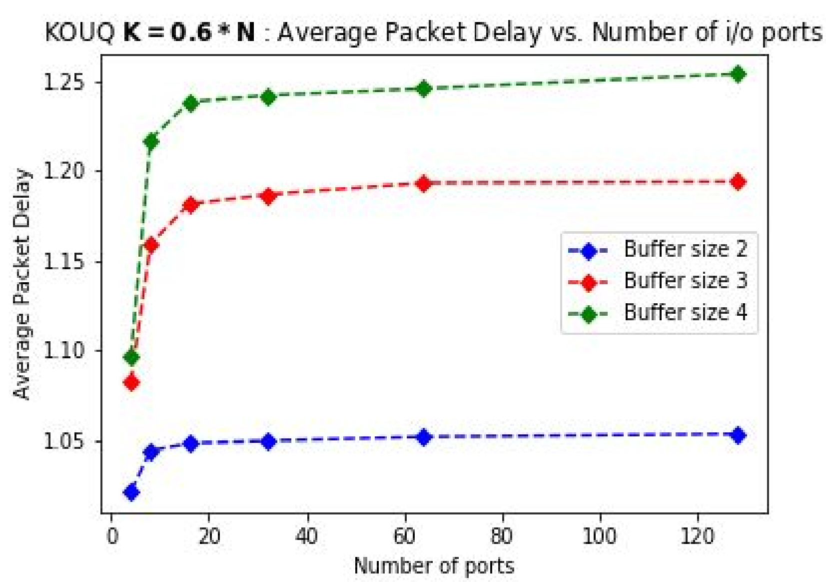
<u>Scheduling:</u>

If the output port’s queue is not empty, schedule the packet at the front of the queue.

Link utilisation vs N and average delay vs N for buffer size 2,3 & 4



Link Utilisation vs N

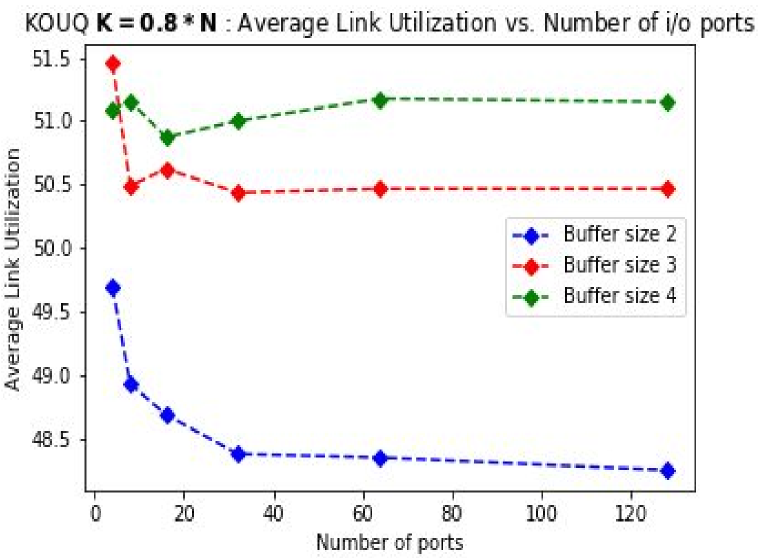


Average delay vs N

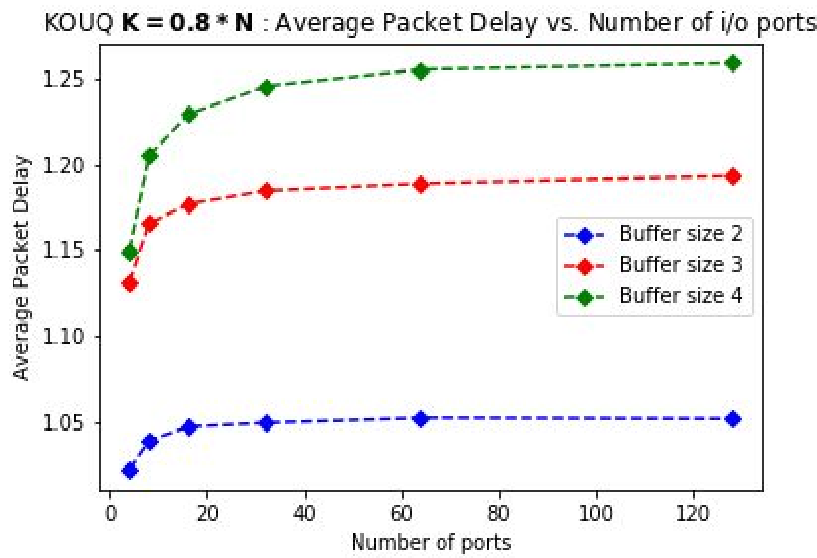
KOUQ

Link utilisation vs N and average delay vs N for buffer size 2,3 & 4

K=0.8*N



Link Utilisation vs N



Average delay vs N

<u>Analysis:</u>

When K=0.6 * N:

Link Utilisation decreases gradually when buffer size is kept 2, whereas utilisation first increases and then becomes stable when buffer size is 3 and 4. Average delay increases as the number of ports increases.

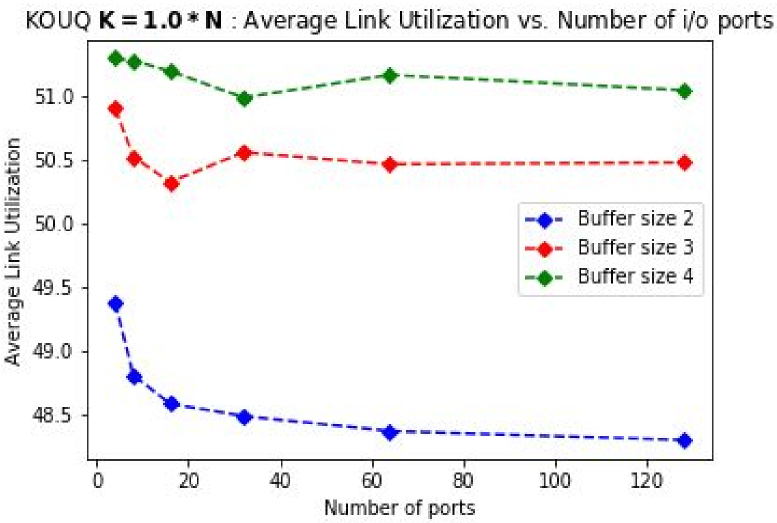
When K=0.8 * N:

Link Utilisation decreases gradually when buffer size is kept 2, whereas utilisation first decreases and then becomes stable when buffer size is 3 and 4. Average delay increases as the number of ports increases.

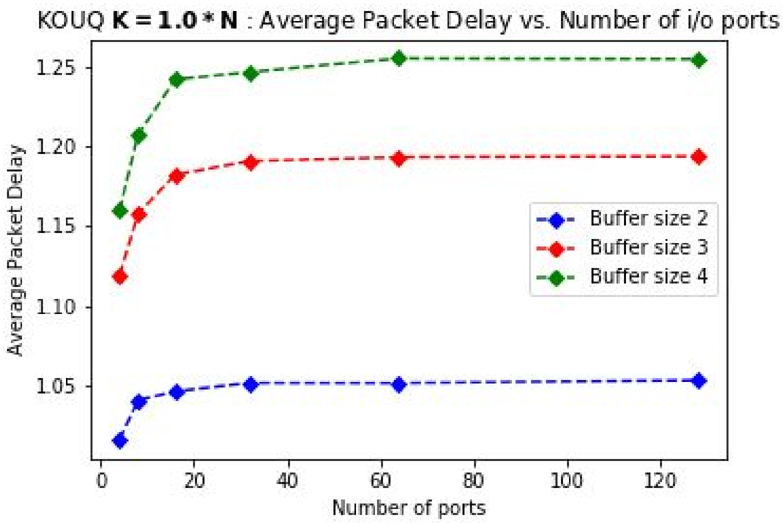
KOUQ

Link utilisation vs N and average delay vs N for buffer size 2,3 & 4

K= 1 * N



Link Utilisation vs N



Average delay vs N

Analysis:

When K=1 * N:

Link Utilisation decreases gradually when buffer size is kept 2, whereas utilisation first decreases and then becomes stable when buffer size is 3 and 4. Average delay increases as the number of ports increases.

It can be clearly seen that link utilisation is more when buffer size is higher. This can also be said about average delay. Average delay is higher when buffer size is more.

ISLIP

Packet Generation :

There are N queues, one for each input port. Each queue stores packets, which in turn contain the arrival time and the destination port. In order to generate a packet with some given probability, we will generate a random number between 0 & 1 . Now if this number satisfies our probability constraint, we will place this packet address in its input queue if the input queue is not full.

Scheduling:

In each round of iSLIP scheduling, there can be more than 1 iterations. In each iteration, following phases take place:

1)**Request phase**: It considers all input port packet's (new or old) requests, i.e. all the destination ports with respect to input ports.

2)**Grant Phase** : Each output port (if requested by at least 1 input port) grants the transmission request to the smallest number input port.

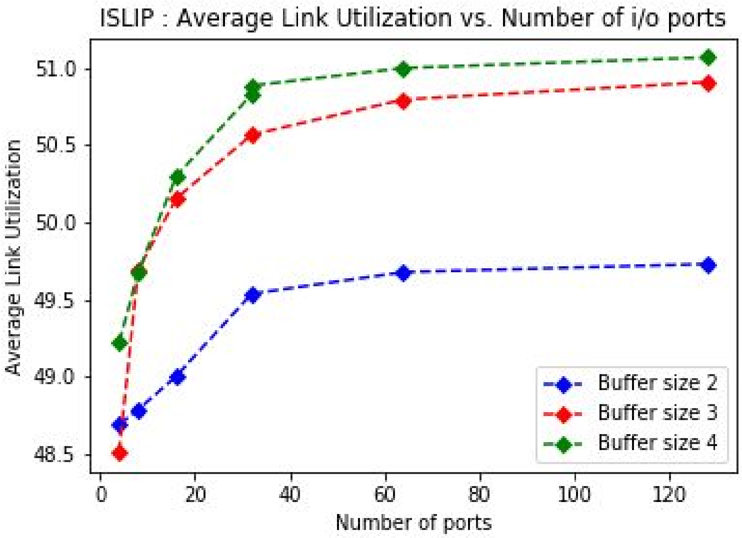
3)**Accept Phase**: All the input ports requests, which are granted in Grant Phase, are accepted.

Hence these input and output ports (accepted ones) will not be part of the next iteration. Round will be continued till all the input output ports are engaged, or none of the requested ports from the remaining input ports are available.

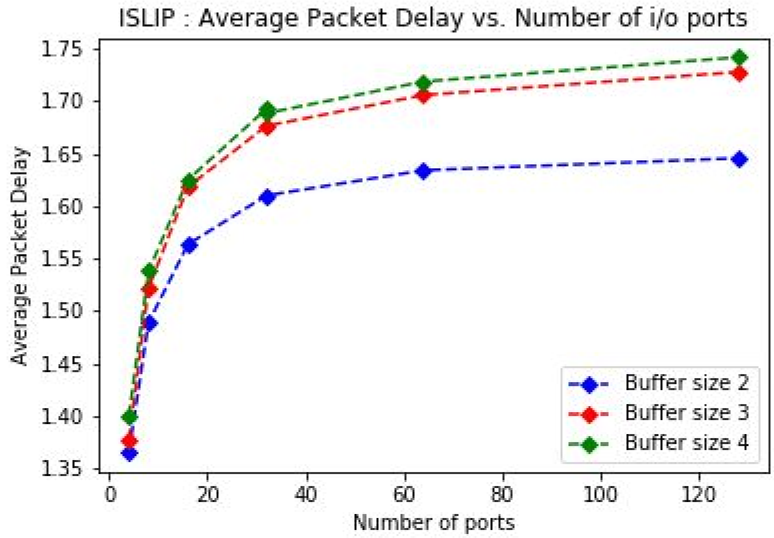
ISLIP

Link utilisation vs N and average delay vs N for buffer size 2,3 & 4

Probability = 0.5



Link Utilisation vs N



Average delay vs N

Analysis:

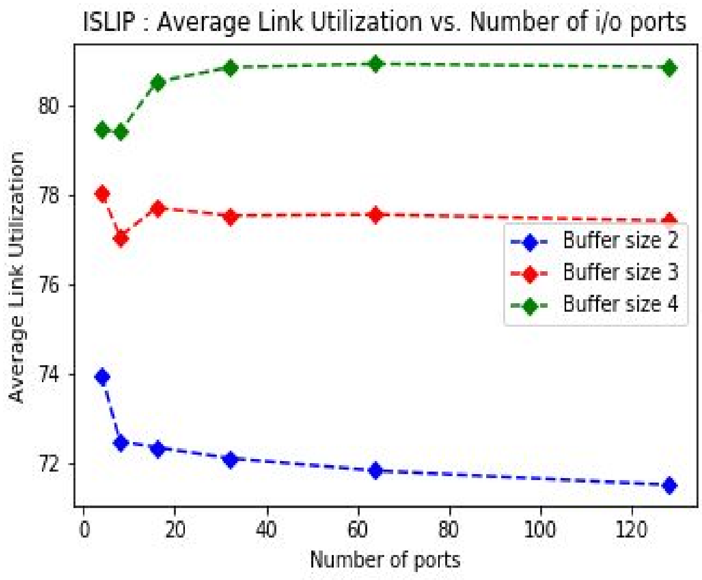
Link utilisation is more when the size of buffer is higher. As the number of ports increases, link utilisation also increases.

Similarly, average delay is higher when buffer size is higher. As the number of ports increases , average delay also increases.

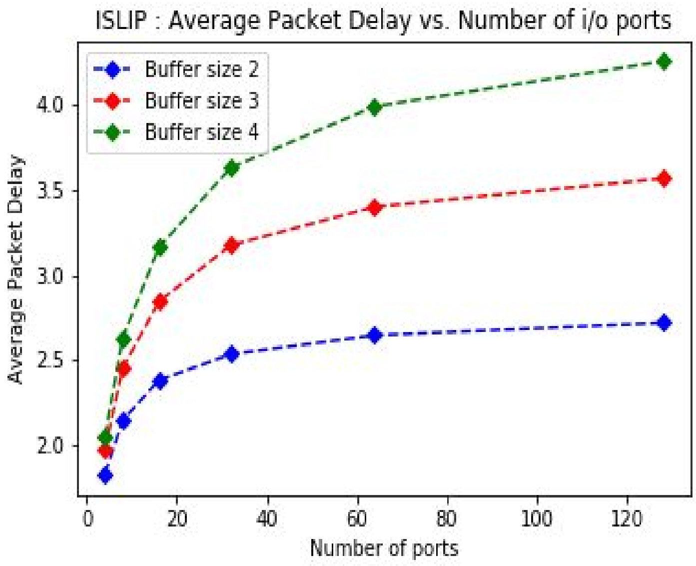
ISLIP

Link utilisation vs N and average delay vs N for buffer size 2,3 & 4

Probability = 1



Link Utilisation vs N



Average delay vs N

Analysis:

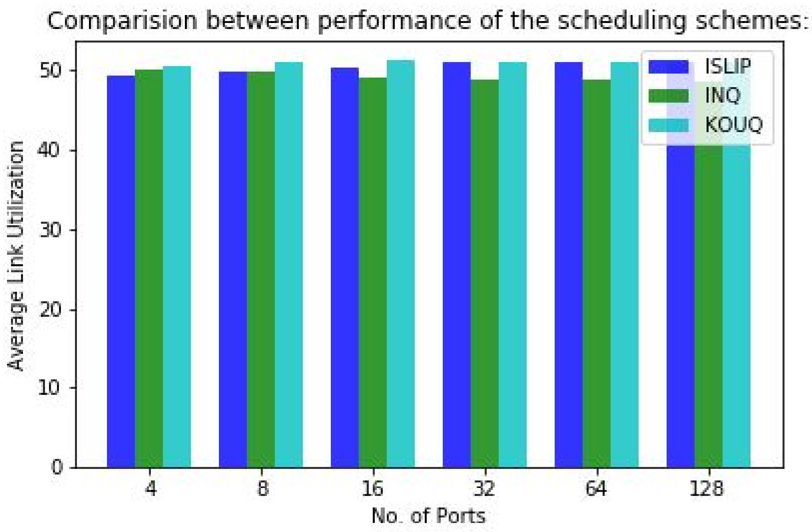
Link utilisation is more when the size of buffer is higher. As the number of ports increases, link utilisation first decreases and then again increases and then becomes stable.

Similarly, average delay is higher when buffer size is higher. As the number of ports increases , average delay also increases.

CONCLUSION

We have analyzed performance of all the three queueing algorithms above . Now we will compare link utilisation and average delay of the three techniques with one another.

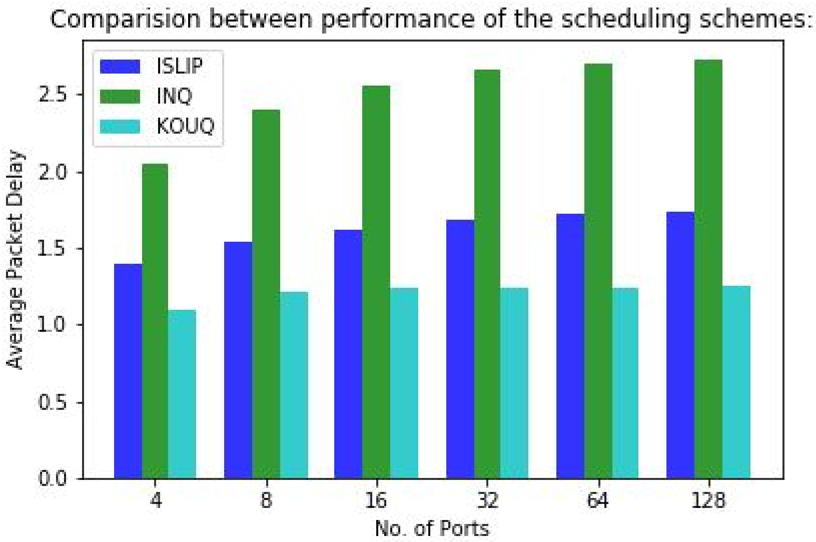
Link utilisation



1. KOUQ and ISLIP performs better than INQ in terms of link utilisation.
2. ISLIP performs slightly better than KOUQ.

   So, Link utilisation wise **ISLIP** > **KOUQ** > **INQ**

Average delay



1. Delay of INQ is highest.
2. Delay of ISLIP is lowest

   Average delay wise **INQ** > **KOUQ** > **ISLIP**

CONCLUSION:

As algorithm with good link utilisation and less average delay is better, we conclude ISLIP is better than KOUQ which is better than INQ.