# Scheduling Mechanisms for Queuing in a Packet Switch

# Contents

# 1 Introduction

A packet switch plays an essential role in the data plane of any packet switched network. It accomplishes the task of forwarding an incoming packet from an input port to some output port. The packets are moved from input to output port via a switch fabric. In case the switch fabric is slower than the rate at which packets arrive, queueing is required at the input ports to avoid dropping packets. Similarly, if the switch fabric is faster than the rate at which output ports can transmit packets, queueing must be done at output ports.

Due to this queueing at the input ports, it is necessary to select packets that can be transferred to their respective output port without blocking each other. This is the task of a scheduling mechanism. A scheduling mechanism matches input ports to output ports such that no two input ports transfer packets to the same output port at the same time and vice versa.

In this report, we analyse the performance of queueing in a packet switch by implementing the following scheduling mechanisms:

1. INQ
2. KOUQ
3. iSLIP

The developed simulation, runs for a given number of timeslots. Each timeslot produces some number of packets, schedules packets waiting in the input queue and transmits packets based on the scheduling decision. It is assumed that packets are generated at each input port by independent Bernoulli Trials with probability of success equal to the given packet generation probability. The variables in the simulation are number of ports in the switch, packet generation probability at each input port, size of input/output port buffer and the number of timeslots.[1]

[1] Note: Unless otherwise specified, all parameters of the simulation are taken as their default values mentioned in the problem statement

# 2 INQ

## 2.1 Implementation

In INQ scheduling, a packet is transferred into its corresponding output port for transmission, if there is no other packet destined to the same output port. In case an output port has multiple packets contending for transmission, the scheduler randomly selects one of the packets and buffers the non-selected packets in their respective input port buffers. This form of scheduling suffers from **Head of Line Blocking** wherein an input queued packet must wait for transmission even though its destination output port is free because it is blocked by a packet at the head of the queue. This is evident from the observations below that show the low switch utilization in the case of INQ scheduling.

## 2.2 Observations and Graphs

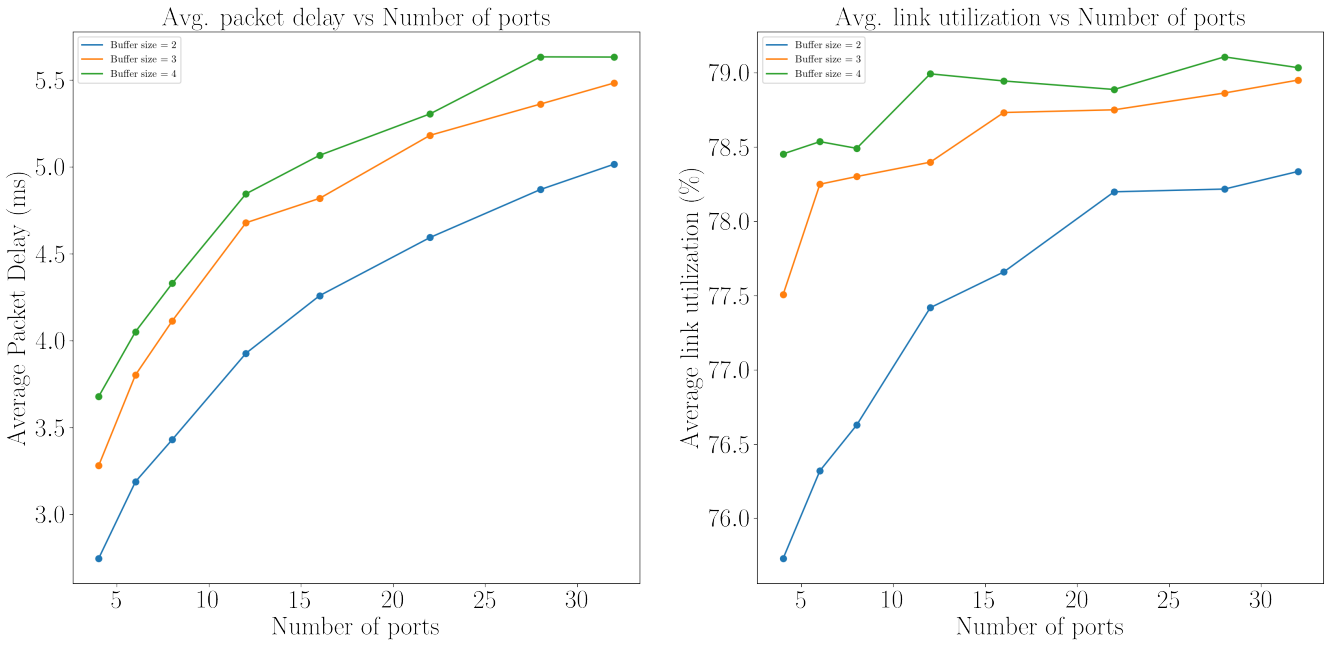


Figure 2.1: Effect of buffer size on INQ scheduling

- **Average packet delay increases with more number of ports.** Since at each input port, packets are generated with certain probability (p) independently, the total number of packets generated in each time slot is a binomial distribution. The average number of packets per slot is $np$ which increases with n (number of ports). Hence, due to greater traffic, the packet delay increases.
- **Increasing the buffer size increases the packet delay**. This is because a greater buffer size implies more packets are queued at the input port. Since INQ suffers from HOL blocking, the packets that are queued behind the head will have greater delay.
- **Increasing the buffer size increases link utilization**. This is because a greater buffer size allows more packets to be queued for transmission. Ultimately, throughout the simulation period, a larger number of packets will get transmitted after being queued at the input port. A smaller buffer drops more packets and hence has lower utilization.

# 3 KOUQ

## 3.1 Implementation

In KOUQ Scheduling, a knockout value (K) is chosen as a fraction of the total number of ports in the switch. If atmost K packets are destined to a particular output port, all of them are queued in the corresponding output buffer in FIFO order. If more than K packets contend for an output port, K random packets are chosen and the remaining are dropped. Each output buffer has packets arranged in according to their arrival times. As a result, at each output port **packet with lowest arrival time**(at the head of queue) is scheduled for transmission leading to a lower packet delay.

## 3.2 Observations and Graphs

### 3.2.1 Variation with knockout (K)

The graphs below have been obtained by running the simulation with packet generation probability $\mathbf{p = 0.5}$.

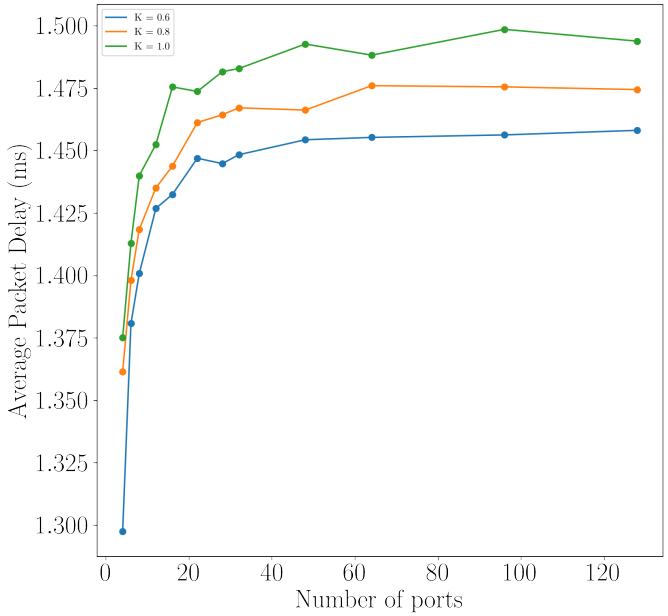


Figure 3.1: Avg. packet delay vs Number of ports

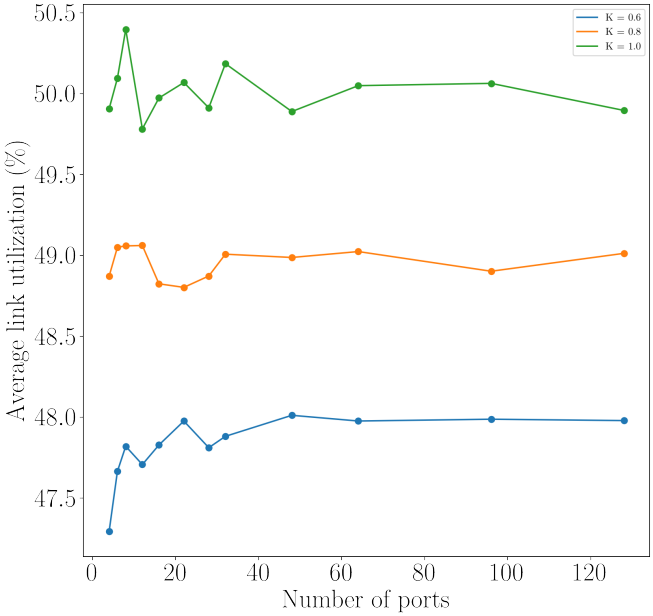


Figure 3.2: Avg. link utilization vs Number of ports

- **Average packet delay increases with K**. This is because a larger value of K implies more packets are buffered at the output port. Therefore, a KOUQ switch with larger K tends to have more packets queued for transmission at the output port which increases the delay for packets behind the head of queue.
- **Average link utilization increases with K**. As explained above, larger values of K tend to buffer more packets at the output port and thereby drop fewer packets. This is evident from the simulation output files since **smaller values of K have a greater KOUQ drop probability** while for larger K the drop probability tends to 0. Hence, throughout the simulation duration, a KOUQ switch with larger K manages to transmit a greater number of packets and hence achieves better utilization.

### 3.2.2 Variation with buffer size (B)

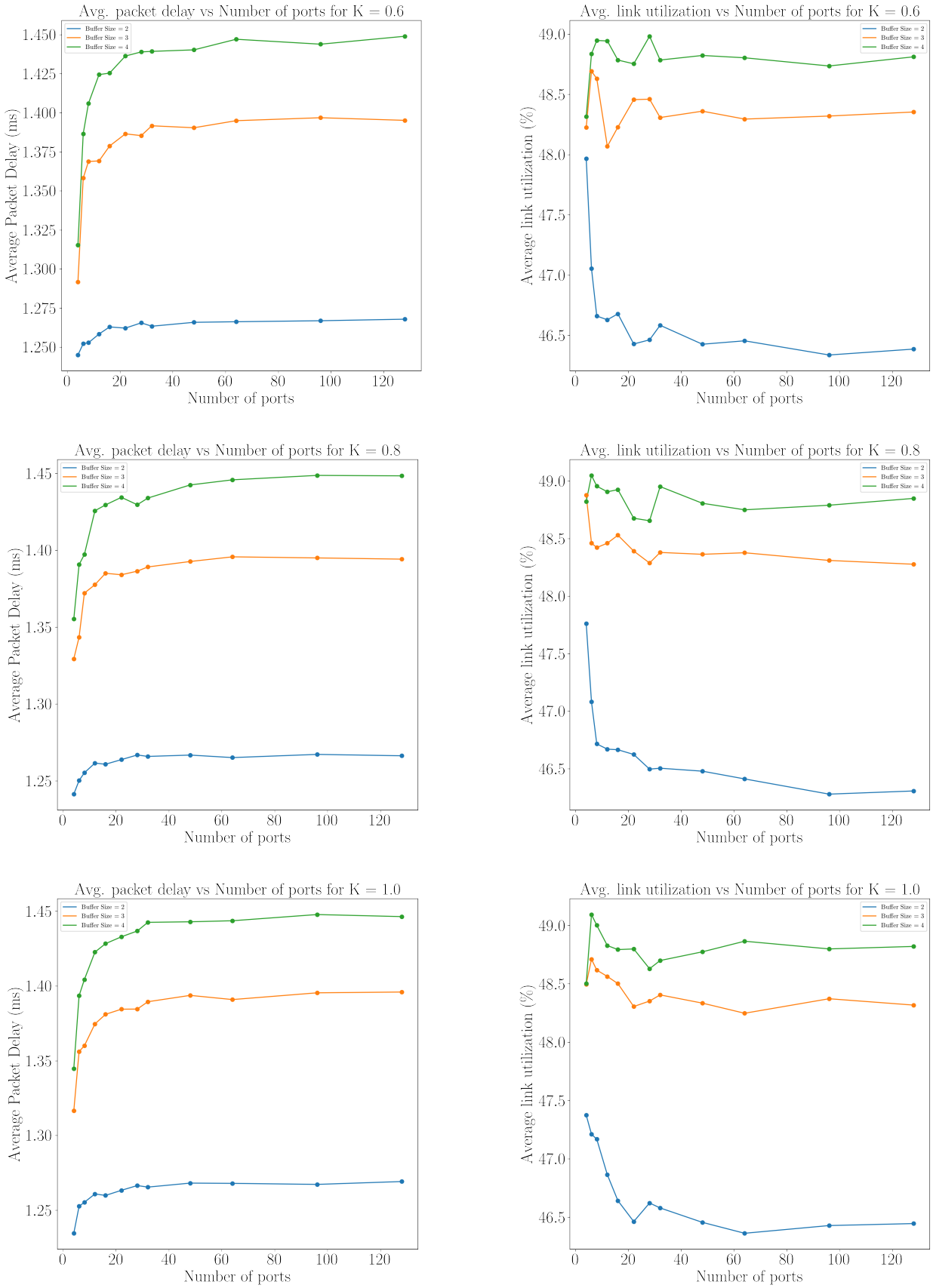


Figure 3.3: Variation of packet delay and utilization with buffer size for different values of knockout K

- Different values of K demonstrate a similar trend with varying buffer size.
- In general, **average packet delay increases with buffer size**. In KOUQ scheduling, packets are queued at the output port. With a greater buffer size, more packets can be accommodated at each output port but transmission suffers a bottleneck due to the fact that only one packet can be transmitted per output port at each time slot.
  - As a result, with greater buffer size, packets that are queued at the output port suffer greater delay than compared to smaller buffer size.
  - In the case of a smaller buffer size, packets that don't fit within the buffer are just dropped.
- After a certain number of ports, around 40, the average packet delay **saturates**. This can be explained by the fact that by further increasing the number of ports, all buffers at the output ports continue to be utilized to their complete capacity.
- **Greater buffer size has greater average link utilization.** This is because a larger number of packets can be queued and transmitted rather than dropped. Effectively, the link succeeds in transmitted more packets throughout the simulation duration if the buffer size is greater.
- Similar to average packet delay, the values of average link utilization also saturate after around 40 ports since all buffers are used to their maximum capacity. No further improvement is possible.
- Packet utilization decreases with number of ports for K = 0.6. This is because increasing the number of ports increases the traffic at the switch but due to a smaller knockout value, a larger number of packets are dropped rather than being transmitted.

# 4 iSLIP

## 4.1 Implementation

iSLIP scheduling is based on finding a maximal size matching between the input and output ports. It combines principles from both Parallel Iterative Matching and Basic Round Robin Matching and finds a maximal matching by sending requests from input ports to output ports, output ports granting requests using a round robin arbiter and input ports accepting grants using another round robin arbiter. It eliminates head of line blocking by using virtual output queueing.

## 4.2 Observations and Graphs

### 4.2.1 Variation with Number of iSLIP iterations (L)

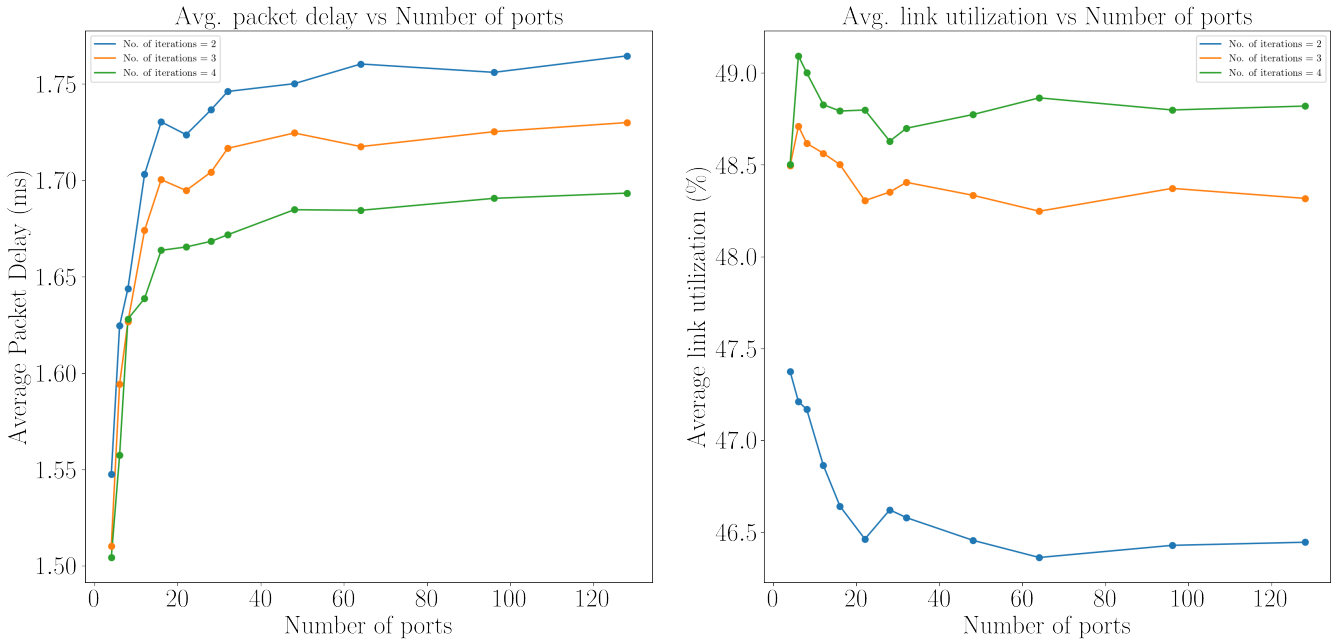


Figure 4.1: Effect of number of iterations on iSLIP scheduling

- **Average packet delay decreases with number of iSLIP iterations.** This is because greater the number of iterations, greater the number of matched input and output ports. As a result, packets have to spend shorter duration being queued at the input port.
- **Average link utilization increases with number of iSLIP iterations.** This is because a better matching is found with more number of iterations. Hence, more packets are forwarded per time slot which makes better use of the switch.

### 4.2.2 Variation with buffer size (B)

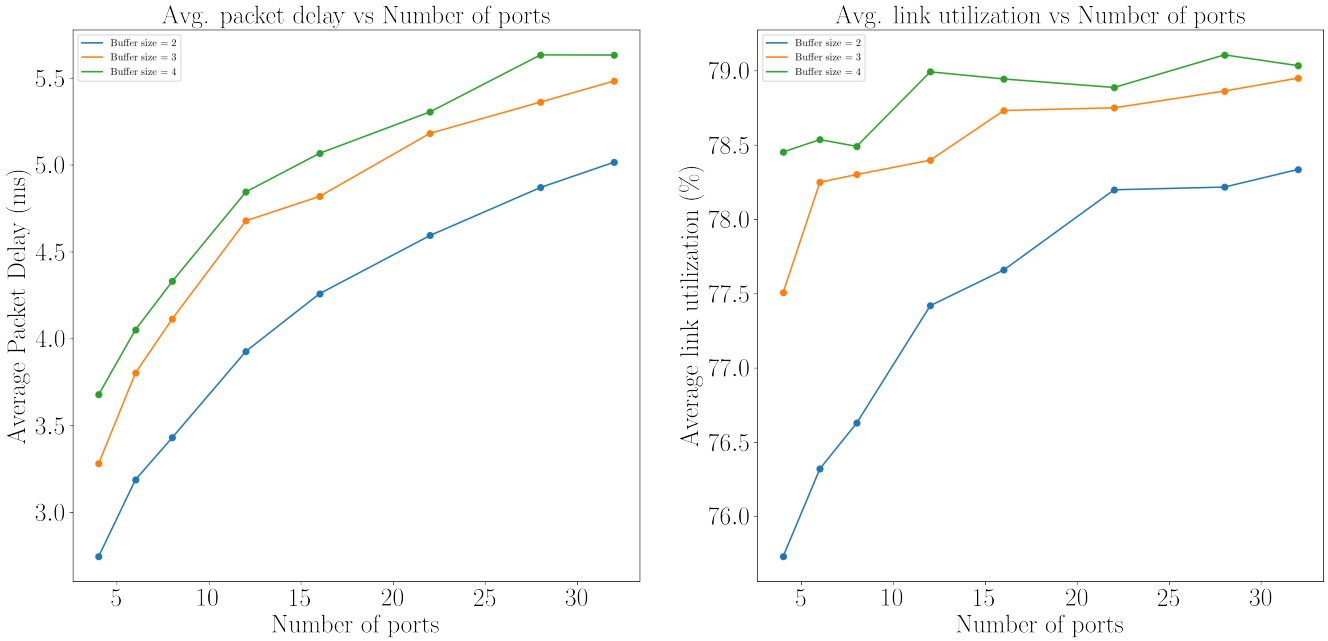


Figure 4.2: Effect of buffer size on iSLIP scheduling

- **Average packet delay and average link utilization** increase with increasing buffer size. The reasons are similar to those explained in the previous sections.

# 5 Comparison and Conclusion

The following graphs are produced for packet generation probability $\mathbf{p = 0.8}$ and number of iSLIP iterations $\mathbf{L = 4}$

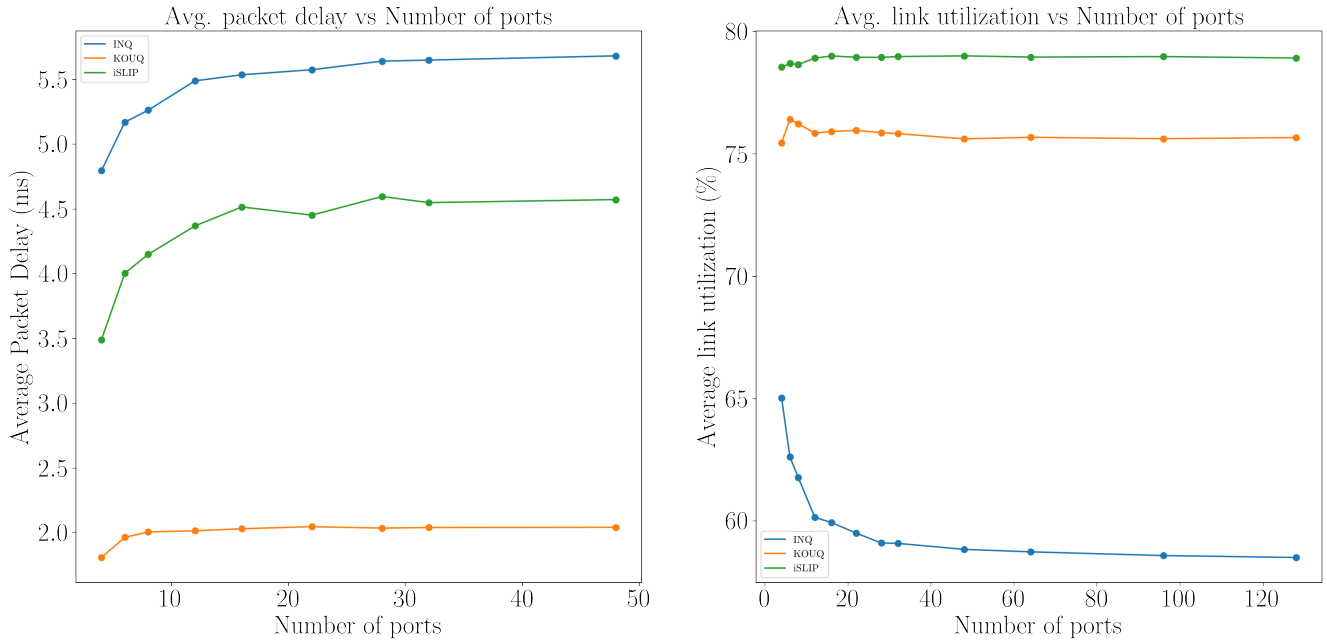


Figure 5.1: Comparison of all three scheduling mechanisms

- **INQ has the highest average packet delay followed by iSLIP and KOUQ**. INQ suffers from head of line blocking which causes it to have a greater delay. iSLIP eliminates this with **virtual output queueing** and **maximal bipartite matching**. KOUQ has the lowest average packet delay but this comes at a **cost**. A lot of packets are dropped by the KOUQ switch making it less robust.
- **iSLIP has maximum link utilization followed by KOUQ and INQ.** iSLIP with 4 iterations tries to find the best possible maximal matching between input and output ports and makes best use of the link and hence has the highest utilization. KOUQ avoid queueing at input ports by transferring packets in batches of K per time slot while INQ queues several packets at the input port. This causes KOUQ to have better utilization than INQ, which has the least utilization.
- **INQ shows decreasing trend of average link utilization**. This is because with more number of ports, greater traffic is generated which can lead to greater probability of internal blocking.