Assignment	The Performance of Queueing in a Packet Switch
Submitted by	Manideepak Gannoju (200101032) Chandrabhushan Reddy (200101027)

OBJECTIVE

The major goal of this assignment is to apply a few well-known scheduling algorithms, such as INQ (Input Queuing), KOUQ (Output Queueing), and iSLIP, in order to comprehend the functionalities and performances of packet switch.

The primary functionalities of a switch can be summarised as follows:

- ## Phase 1: Traffic Generation:

  During this stage, packets are generated at each input port and in each time slot with a packet generation probability of *"packetgenprob"*

- ## Phase 2: Packet Scheduling:

  In this stage, input and output ports are mapped to handle the packets created in the previous phase. We have 3 scheduling mechanisms to perform this task namely INQ, KOUQ and iSLIP.

  a. **INQ:** A packet is transmitted if there is no contention for the required output port. If not, a random packet is selected for transmission and the other packets are queued to the appropriate input port.
  b. **KOUQ:** This scheduling technique can send up to K packets during a particular time slot for any output port. K packets are randomly chosen for buffering if more than K packets arrive in a slot for any output port; the other packets are dropped.
  c. **iSLIP:** This mechanism schedules packets in a round-robin fashion following the below steps-
     1. **Request step -** Each input sends a request to every output for which it has a queued cell
     2. **Grant step -** In the event that an output gets requests, it selects the element with the highest priority first, proceeding in a predetermined round-robin sequence. Each input is informed by the output as to whether or not its request was approved. If and only if the grant is accepted in Step 3, the pointer to the round-robin schedule's highest priority element is increased (modulo N) by one location beyond the input that was granted.
     3. **Accept step -** In the event that an input is granted, it takes the next position in a predetermined, round-robin sequence, beginning with the element with the

highest priority. One location beyond the approved output is indicated by an increment (modulo N) of the pointer to the round-robin schedule's highest priority element.

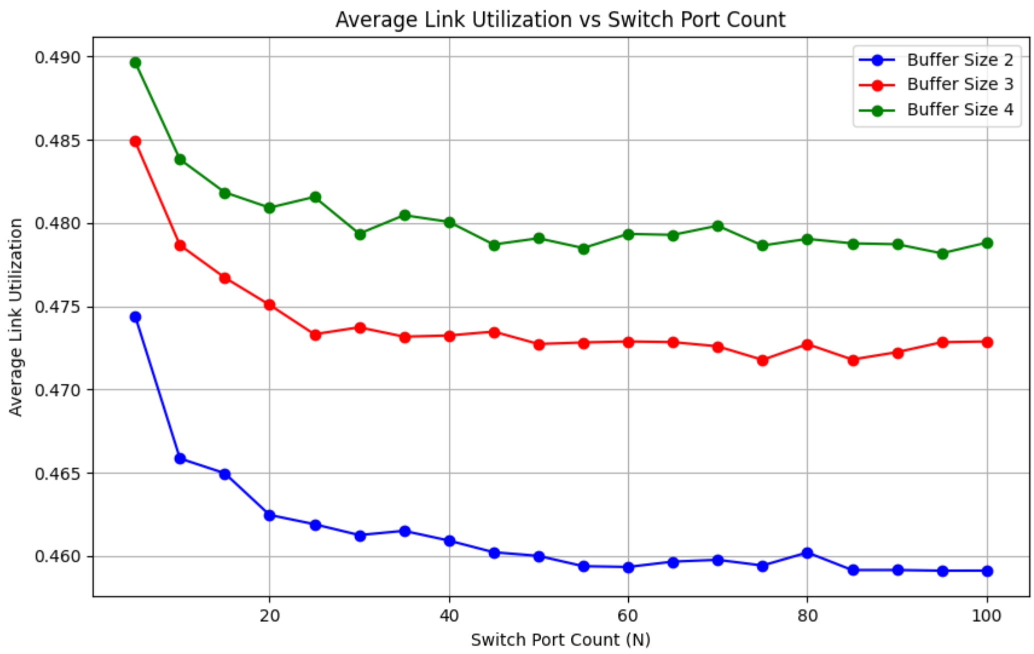
- **Phase 3: Transmission:**

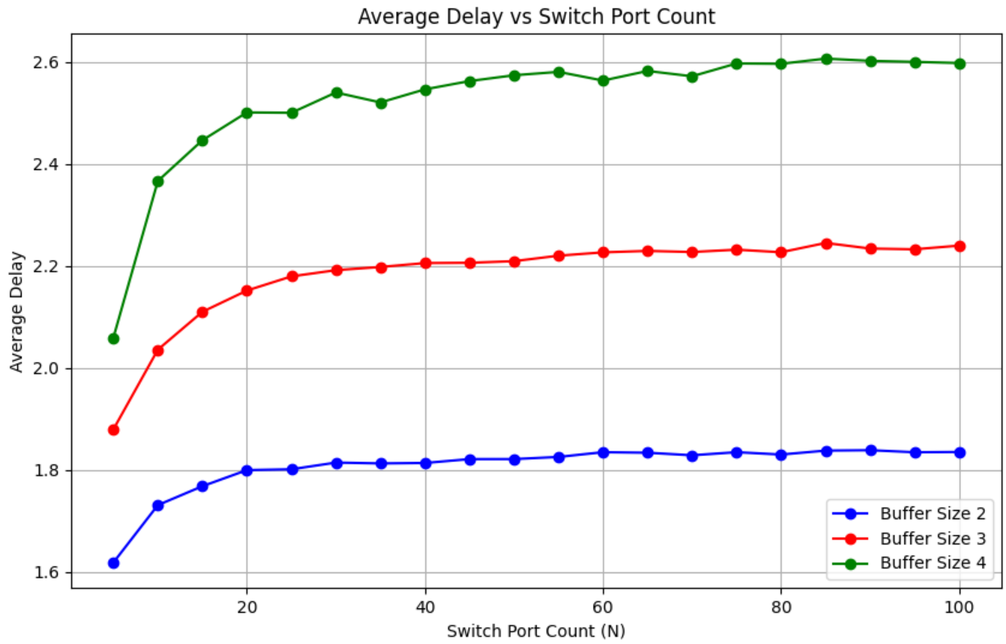
  During the Transmission phase, the packet at the front of the queue is transmitted at each output port. Next, the packet's delay and link utilization are determined.

PERFORMANCE ANALYSIS

The performance of the scheduling methods that have been put in place thus far is compared in this section. It is made up of performance graphs that show average packet delay and average link utilization for various N, B, and K values.

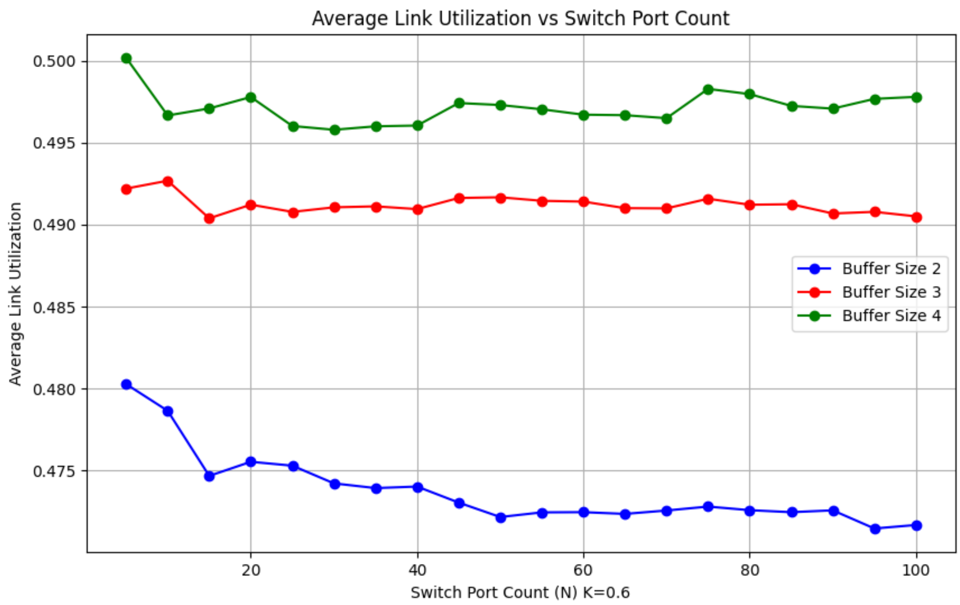
- **INQ:**

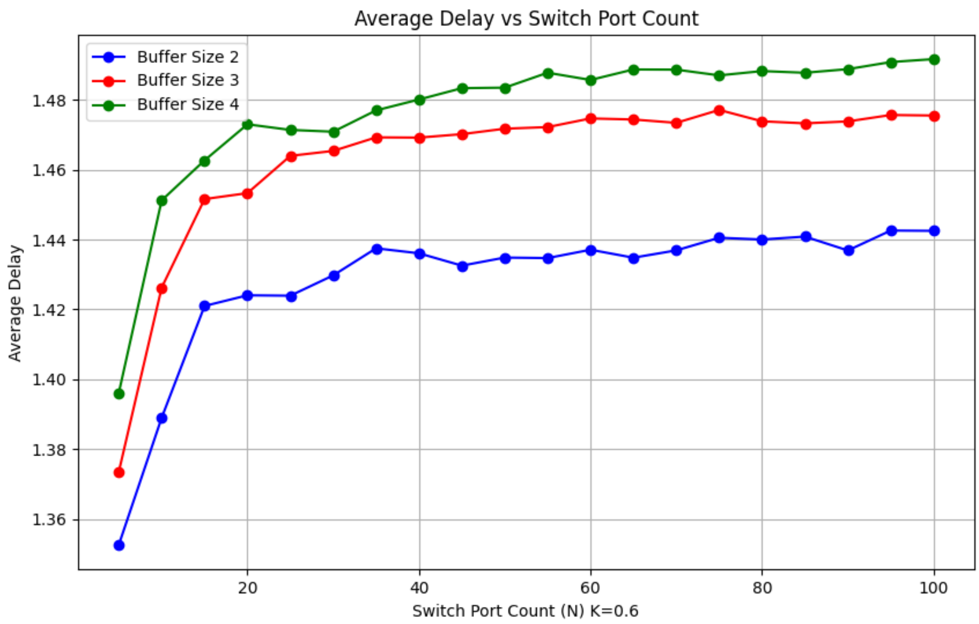




We can deduce from the above graphs that while average link utilization falls with an increase in ports, average delay increases. However, increasing the buffer size results in an increase in average delay as well as average link use.

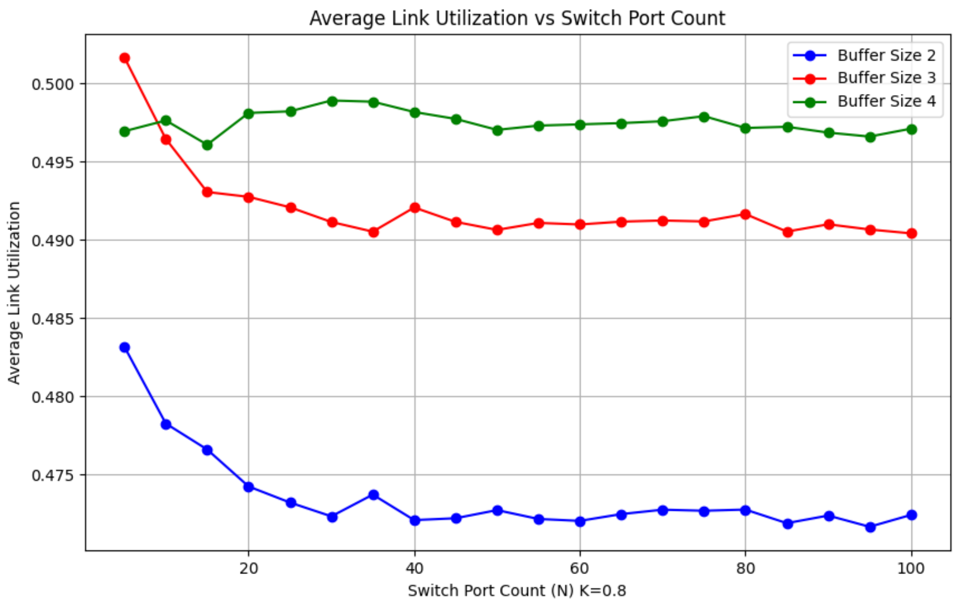
- **KOUQ:**
  - **K = 0.6*N:**

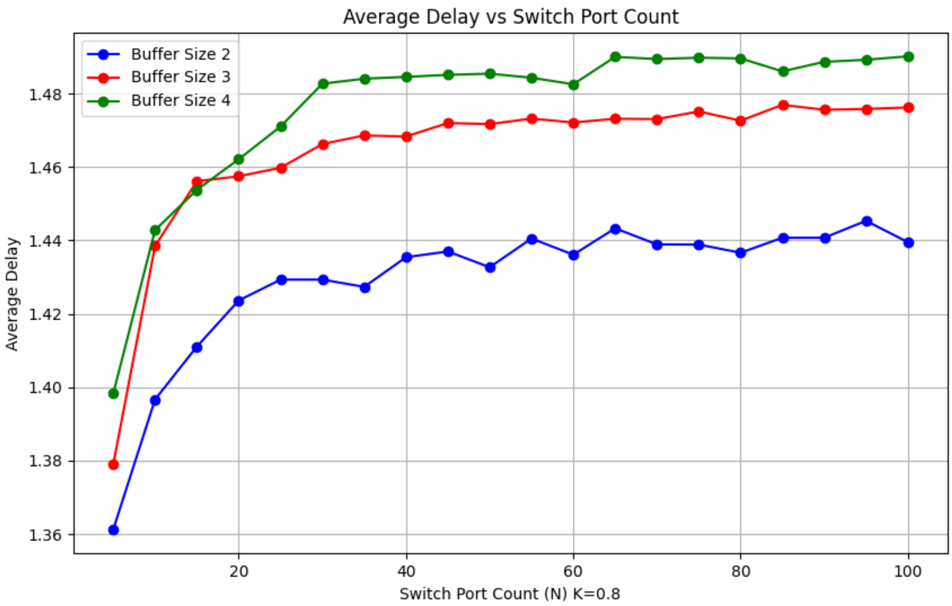




From above graphs we can infer that average delay increases with increase in number of ports whereas average link utilization decreases. However, both average delay and average link utilization increases when we increase buffer size

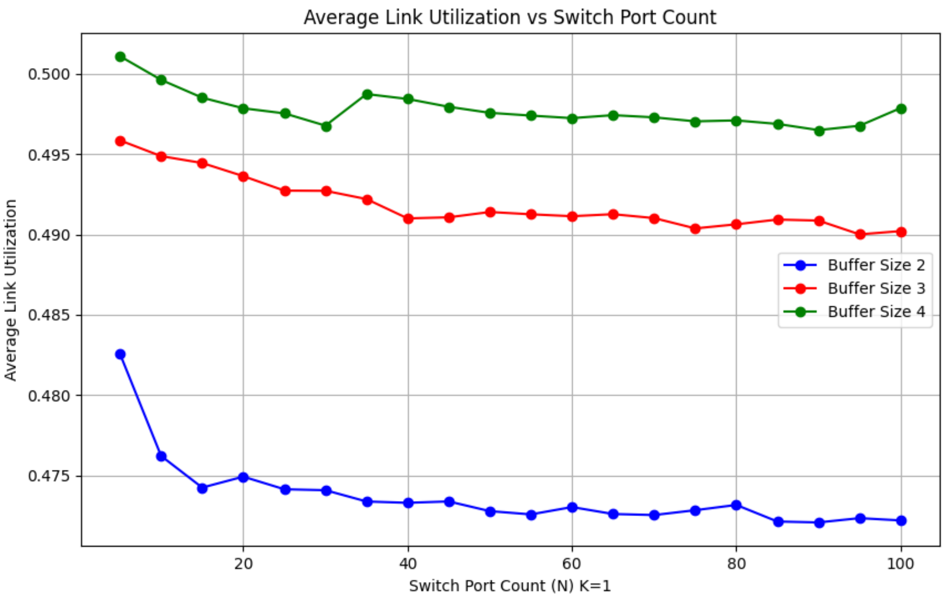
- **K = 0.8*N:**

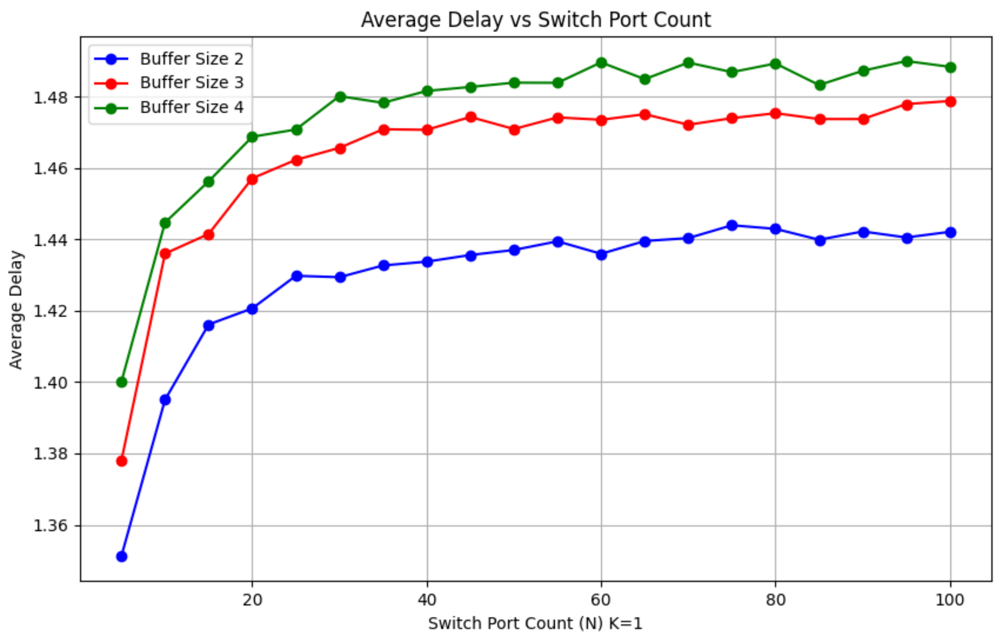




From the above graphs we can tell that average delay increases rapidly with number of ports whereas average link utilization decreases steadily. However, both average delay and average link utilization increases with increase in buffer size

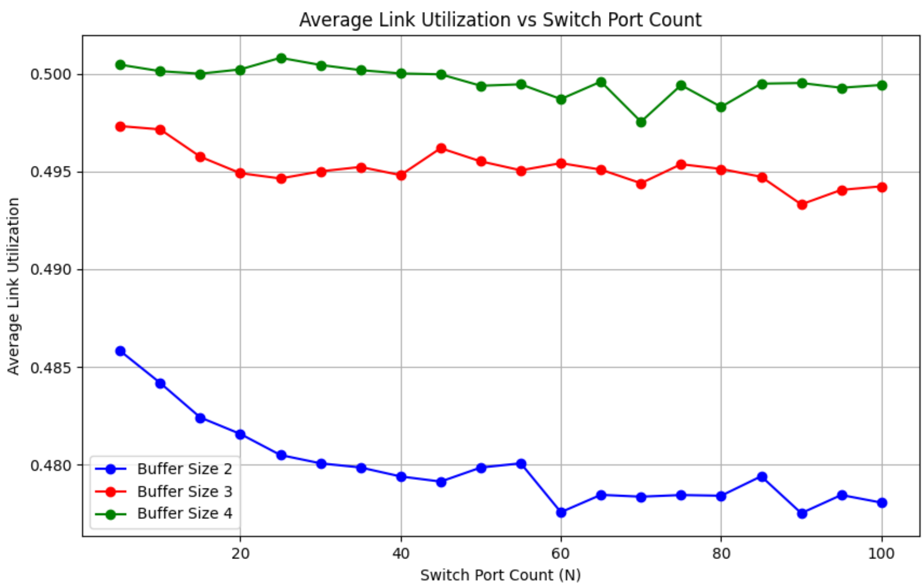
- **K = 1*N:**

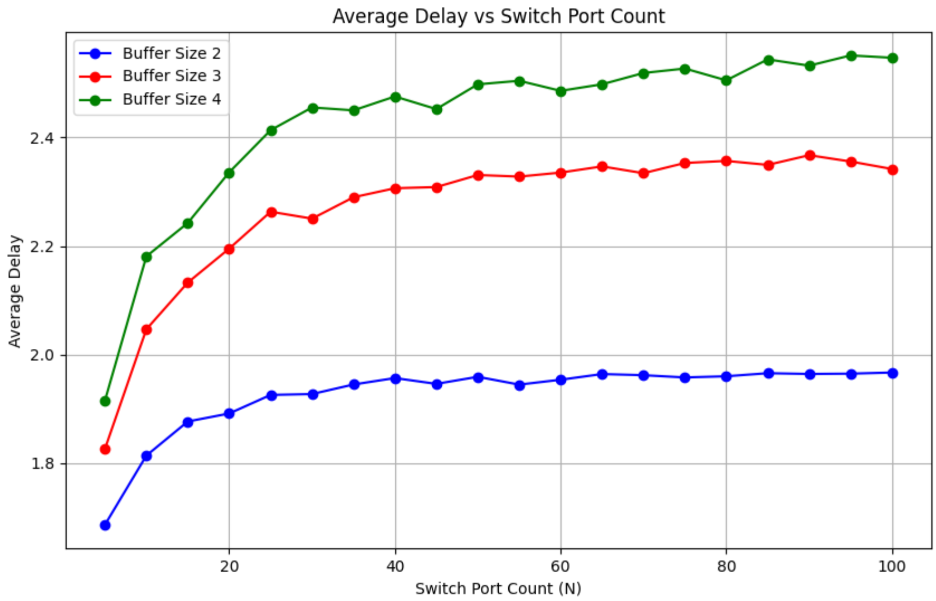




The above graphs show that while average link utilization declines with port count, average delay rises quickly. However, as buffer capacity increases, so do average delay and average link utilization.

- **iSLIP:**





The graphs above suggest that adding more ports has a marginal effect on average latency, but that buffer size increases average delay and link usage drops as port count grows

CONCLUSION

According to the data collected thus far, the average delay for the INQ scheduling mechanism rises as the number of ports and buffer size grow. Also, when the number of ports and buffer size rise, link usage falls.

When using the KOUQ scheduling mechanism, the average delay significantly decreases as the buffer size decreases. When it comes to packet latency, KOUQ is marginally slower than iSLIP. When it comes to link usage, iSLIP performs better than KOUQ.

Even when we increase the number of ports in an iSLIP scheduling scheme, the average delay remains nearly constant. Additionally, as buffer size increases, average latency rises as well. However, connection utilization is quite low.

Therefore, iSLIP is a better option than INQ and KOUQ if one wants to lower the average delay because it offers each port an equal chance, preventing hunger and improving performance.