A Report on
'The Performance of Queueing in a Packet Switch'

Submitted by -

Shivangi Garg (194101046)

Sweety Dhale (194101052)

OBJECTIVE :

The main objective of this assignment is to understand the performance of queuing in a packet switch by implementing various scheduling mechanisms as listed below:

1. INQ
2. KOUQ
3. iSLIP

SWITCH OPERATIONS :

- **Phase 1 : Traffic Generation**

This phase in switch operation corresponds to the generation of packets in each time slot and at each input port with a given value of *packetgenprob* which is 'packet generation probability'.

I. Packet is defined using structure as follows :

```
struct Packet
{
   int destination_port;
   double arrival_time;
};
```

II. A two dimensional vector of ‘Packet’ called queue is created as:

```
vector<vector<Packet>> queue(N);  // to store generated packets
```

III. Packet is generated with given probability using function if it returns true and buffer for the corresponding input port is not full.

IV. If both conditions are satisfied then a packet is generated and appended to the queue of corresponding input ports as follows.

- **Phase 2 : Packet Scheduling**

In this phase, packets generated in the first phase at each port are handled by mapping input and output ports to transmit data. We have three scheduling mechanisms to overcome this task, namely INQ, KOUQ and iSLIP.

1. INQ : If there is no contention for the required output port of a packet, then packet is transmitted. Otherwise, one of the packets is randomly chosen for transmission while others are queued as it is to their corresponding input port.
2. KOUQ : This scheduling mechanism can transmit at most K packets in a given time slot for any output port. If more than K packets arrive at any output port, then after K packets every packet gets dropped.
3. iSLIP : This mechanism schedules packets in a round-robin fashion with below three steps as follows.
    a. Request step - In this step, each input port requests to all required output ports.
    b. Grant step - Each output port keeps track of recently used input port in ‘lastUsedPort’ array. So in the time slot ‘t+1’ it gives lowest priority to the input used in the time slot ‘t’.

    c. Accept step - In the last step, which is accept step, the output port selects the input port which is free and has highest priority among all others.

- **Phase 3 : Transmission**

  In the Transmission phase, at each output port, the packet at the head of the queue is transmitted. Then delay for that packet is calculated along with link utilization.

PERFORMANCE ANALYSIS :

This part of the report compares the performance of the scheduling schemes implemented so far. It consists of performance graphs, which compares average packet delay and average link utilization for different values of N , B and K. The corresponding results obtained after varying the values of N, B and K are as shown in below graphs. :

- ➢ N varies from 4 to as high as possible
- ➢ B ∈ 2, 3, 4
- ➢ K ∈ 0.6, 0.8, 1.0*N

1. **Scheduling scheme : INQ**

- It is seen from the graph below that when *packet generation probability is 0.5*, 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.

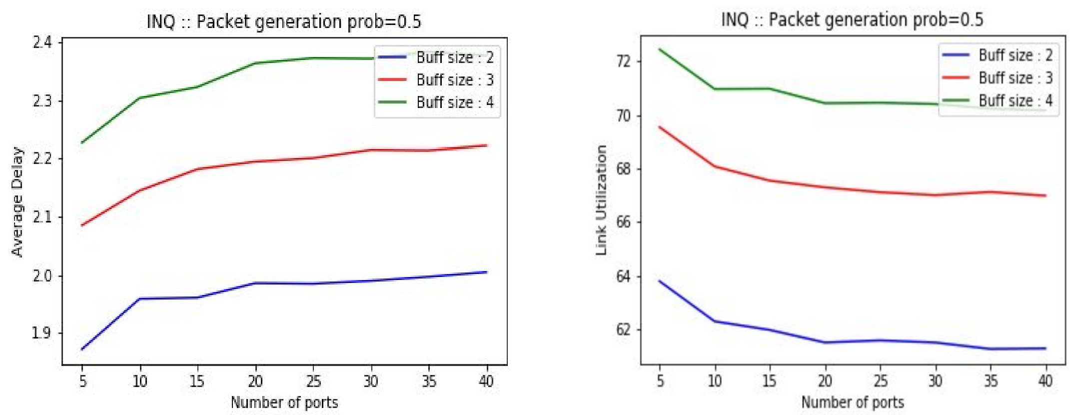

Figure 1.1 : INQ, Packet generation probability is 0.5

- When *packet generation probability is 1.0*, average delay increases slightly with increase in number of ports whereas average link utilization decreases sharply and irregularly. However, average delay increases when we increase buffer size but there is only a slight change in average link utilization with respect to buffer size.

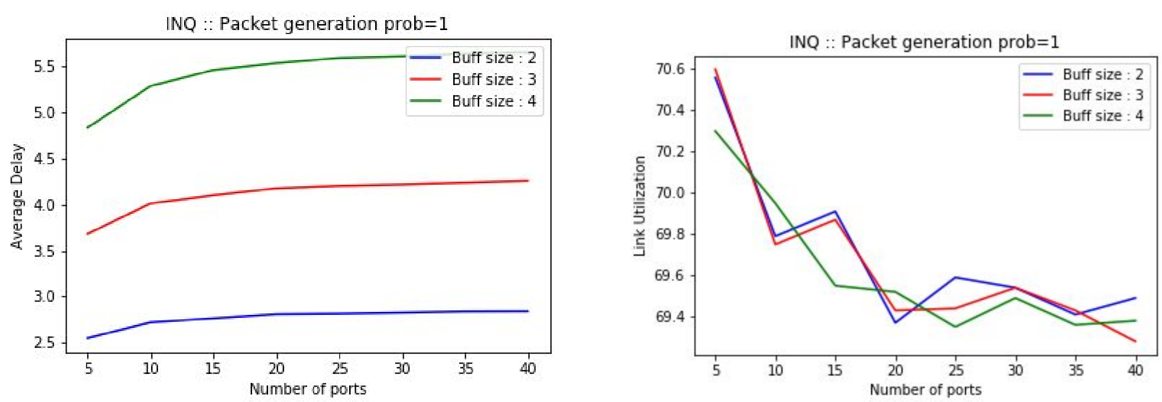

Figure 1.2 : INQ, Packet generation probability is 1.0

Scheduling scheme : <u>KOUQ</u>

- When *packet generation probability is 0.5 and K equals 0.6*N*, 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.

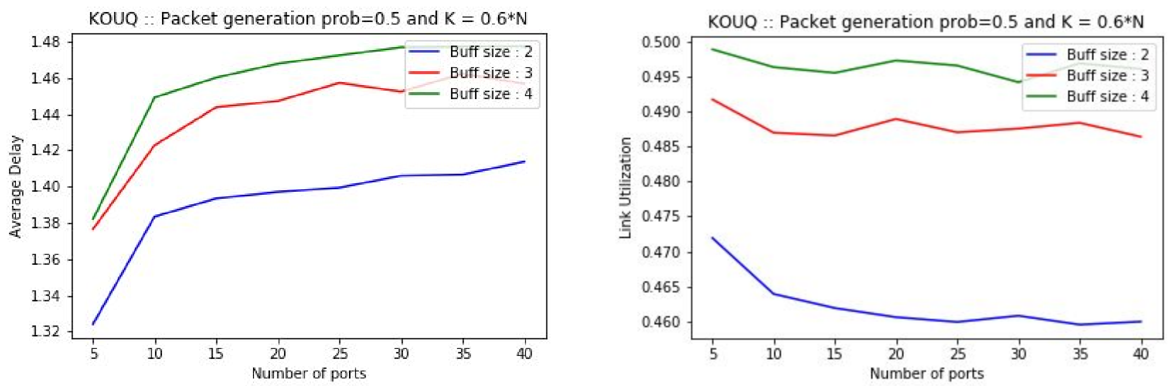

Figure 2.1 : KOUQ, Packet generation probability is 0.5 and K=0.6*N

- When *packet generation probability is 1.0 and K equals 0.6*N*, 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.

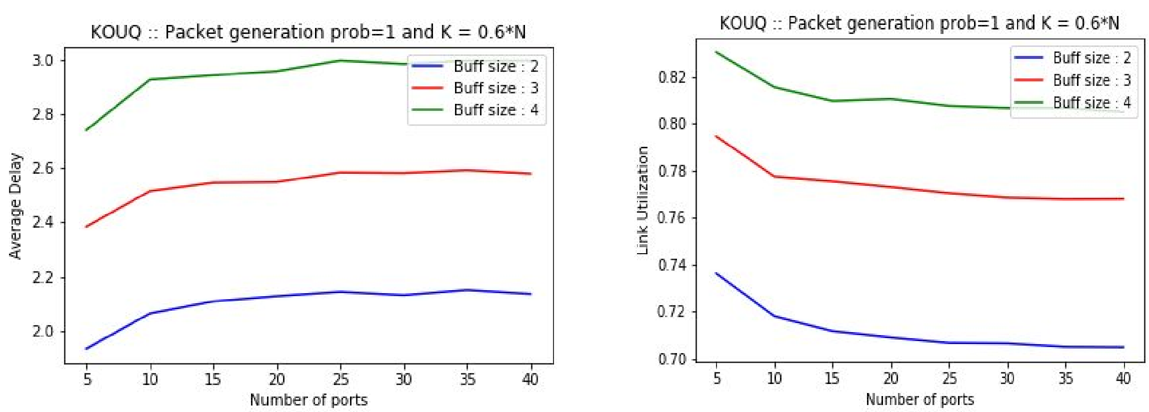

Figure 2.2 : KOUQ, Packet generation probability is 1.0 and K=0.6*N

- When *packet generation probability is 0.5 and K equals 0.8*N*, 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.

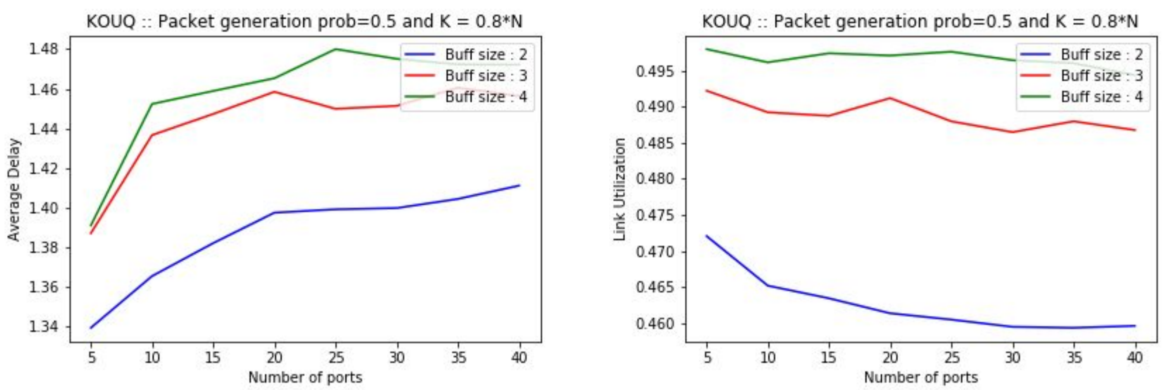

Figure 2.3 : KOUQ, Packet generation probability is 0.5 and K=0.8*N

- When *packet generation probability is 1.0 and K equals 0.8*N*, average delay increases steadily with number of ports whereas average link utilization decreases. However, both average delay and average link utilization increases with increase in buffer size.

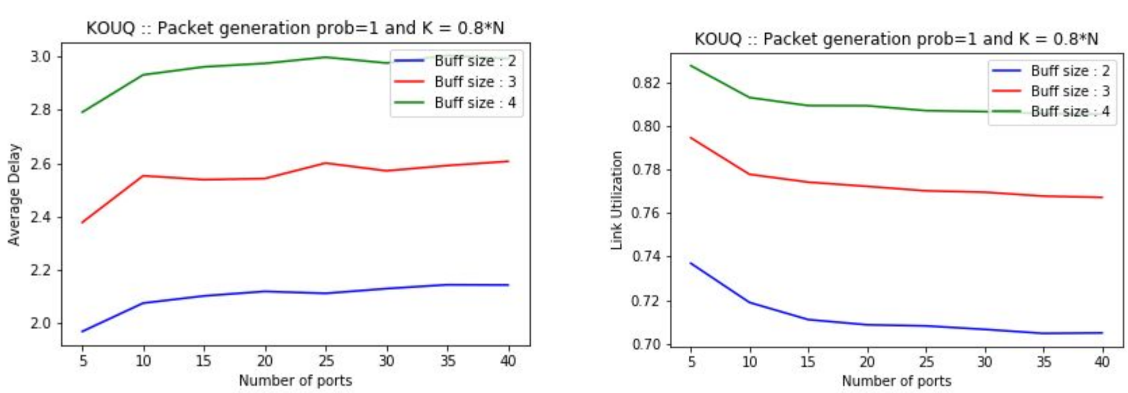

Figure 2.4 : KOUQ, Packet generation probability is 1.0 and K=0.8*N

- When *packet generation probability is 0.5 and K equals N*, average delay increases rapidly with number of ports whereas average link utilization decreases. However, both average delay and average link utilization increases with increase in buffer size.

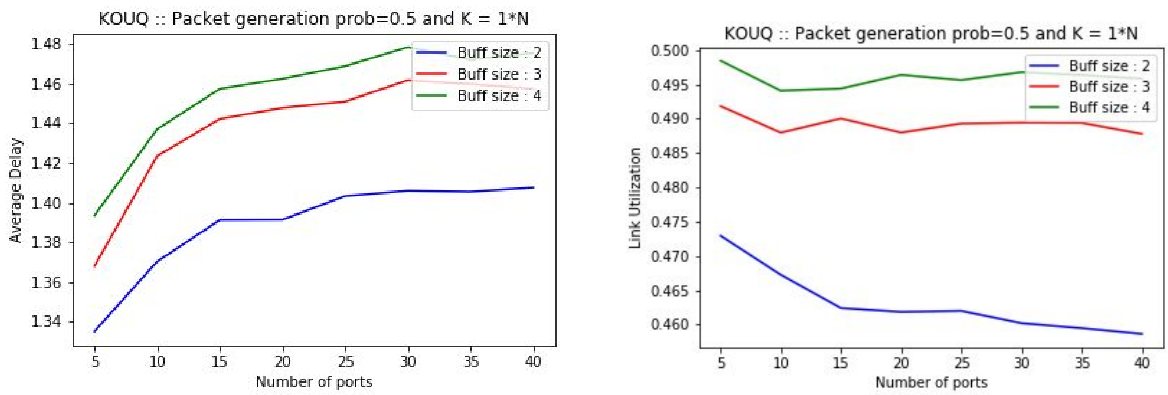

Figure 2.5 : KOUQ, Packet generation probability is 0.5 and K=N

- When *packet generation probability is 1.0 and K equals 0.8*N*, average delay increases steadily with number of ports whereas average link utilization decreases. However, both average delay and average link utilization increases with increase in buffer size.

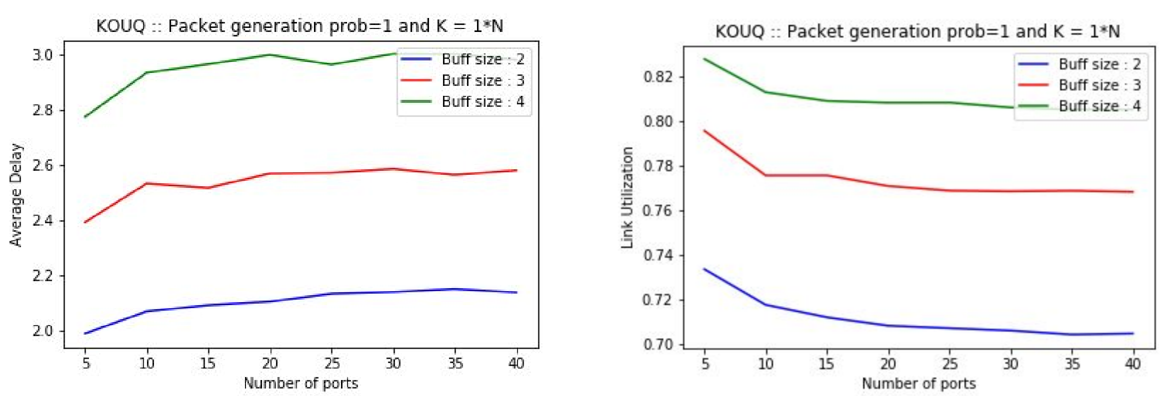

Figure 2.6 : KOUQ, Packet generation probability is 1.0 and K=N

Scheduling scheme : iSLIP

- As seen from the results below, when *packet generation probability is 0.5*, there is just a slight change in average packet delay with an increase in the number of ports but increases with increase in buffer size. However, average link utilization decreases sharply with increase in number of ports.

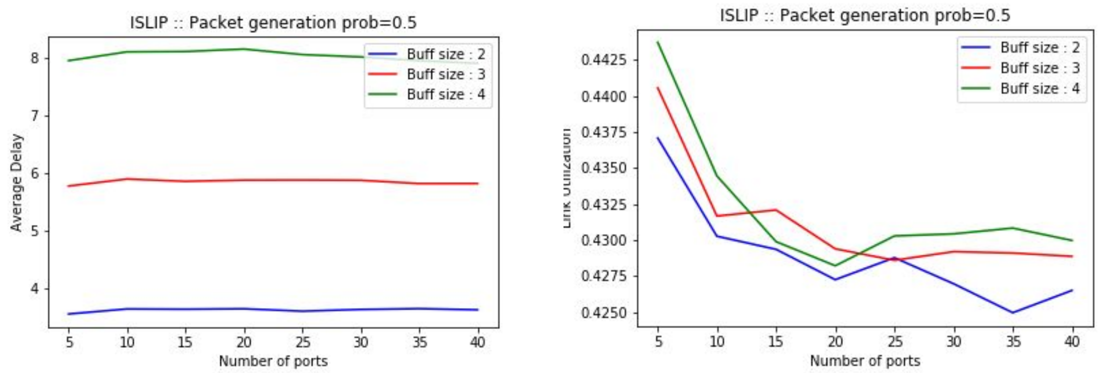

Figure 3.1 : iSLIP, Packet generation probability is 0.5

- When *packet generation probability is 1.0*, there is just a slight change in average packet delay with an increase in the number of ports but increases with increase in buffer size. However, average link utilization decreases sharply with increase in number of ports and no change when increases buffer size.

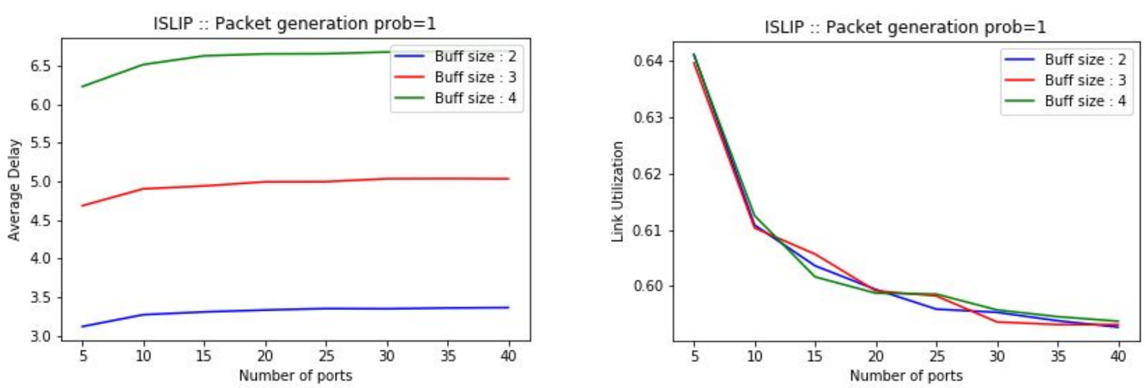

Figure 3.2 : iSLIP, Packet generation probability is 1.0

CONCLUSION :

From the results obtained so far, in case of INQ scheduling mechanism, average delay increases with increase in number of ports and increase in buffer size. Also, link utilization decreases with increase in both, number of ports and buffer size.

In the case of KOUQ scheduling mechanism, with decrease in buffer size there is significant decrease in average delay. KOUQ has a slight increase in packet delay as compared to iSLIP. In terms of link utilization iSLIP outperforms KOUQ.

When there is a case of iSLIP scheduling scheme, average delay is almost constant though we increase the number of ports. Also, average delay increases with increase in buffer size. But link utilization is very poor.

So, if one wants to reduce the average delay, iSLIP is a better choice over INQ and KOUQ because iSLIP gives a fair chance to every port, hense no starvation and gives better performance.