CS6040: Router Architecture and Algorithms

Semester: Jul-Nov 2024

Assignment 3 Report

Packet Switch Queueing

Name: Snehadeep Gayen
Roll: CS21B078

Submission Date: 6th Oct 2024

1 Introduction: Packet Scheduling in Switches

Packet switches and routers use a switched backplane based on simple 2x2 crossbar switches. Most often, these systems use input queues to hold incoming packets waiting to get switched through the switching fabric to their destined output ports. The packet scheduling & queueing algorithm used for this can greatly affect the performance of the switch. This assignment explores the popular iSLIP algorithm and compares them to classical No-Queuing (NOQ) and Input Queueing with Head-of-Line Blocking cases.

2 Challenges

Input Queueing with a simple First-In First-Out scheduling is known to suffer from Head-of-Line blocking effect in which, a blockage at the HOL make all the subsequent waiting packets in the queue suffer. Further, while guaranteeing high utilisation and low drop probability, the scheduling algorithm must also respect flow-wise packet order, guarantee fairness and must adapt quickly to changes in the traffic or bursts in the network. All these challenges, make it difficult to design scheduling algorithms for packet switches.

3 Simulation Experiment

This experiment designs a switch simulator to test out different scheduling algorithms, and compares their drop probabilities, delays and link utilisation. Further the simulator is flexible to handle many different parameters and is easily extensible to other scheduling algorithms.

4 Variables of the Experiment

4.1 Number of Switch Ports N

The switch is assumed to have N input ports and N output ports, numbered from 0 to $N-1$. The default value is 8.

4.2 Buffer Size B

Each input and output port can hold up to B fixed-length packets. The default value is 10.

4.3 Packet Generation Probability p

Each input port generates a packet in a time slot according to a Bernoulli(p) distribution, where p denotes the probability that a packet is generated. The default value is 0.5.

4.4 Queue Type q

This argument specifies the type of queue used for packet handling. Possible values include NOQ (No Queuing), INQ (Input Queuing), and CIOQ (Combined Input and Output Queuing). The default queue type is INQ.

4.5 Knockout K

This parameter is specific to the CIOQ queue type and determines the backplane speedup compared to the output link transmission time. $K = 2$ means that in one transmission time unit of the output port, the backplane can switch two rounds of packets. The default value is 1.

4.6 Input Queue Length Lookahead L

This parameter is also specific to the CIOQ queue type and defines the length of the input queue to be considered for packet scheduling. $L = B$ means that all the packets in the input queue will be considered for scheduling. The default value is 1.

4.7 Maximum Slots T

This argument specifies the total number of time slots for the simulation. Each time slot corresponds to the transmission time of one packet. The default value is $10,000$.

5 iSLIP Algorithm [1]

5.1 Algorithm

5.2 Time Complexity

Using various simualations with a $N \times N$ switch, the authors suggest that the iSLIP algorithm converges in $\log_2 N$ steps [**1**]. However, they have not been able to prove the bound $\mathbf{E}(I) \leq log_2 N$. Note that this is a much faster algorithm that classical Maximum Weight Matching scheduling algorithm which requires $\mathcal{O}(N^3)$ time to converge, and has time complexity comparable to Parallel Iterative Matching algorithm which is proved to converge within $\log_2 N + \frac{4}{3}$ iterations on average [**1**] .

Algorithm 1 iSLIP Algorithm

```
Initialize: All inputs and outputs are unmatched
repeat
    Step 1: Request
    for each unmatched input do
        for each output with a queued cell do
            Send a request to the output
        end for
    end for
    Step 2: Grant
    for each unmatched output do
        if output received requests then
            Grant to the next input in round robin order
        end if
    end for
    Step 3: Accept
    for each unmatched input do
        if input receives a grant then
            Accept the grant in round robin order
            Update round robin pointers for both input and output
        end if
    end for
until no change in schedule
```

5.3 Salient Features

5.4 Example Run

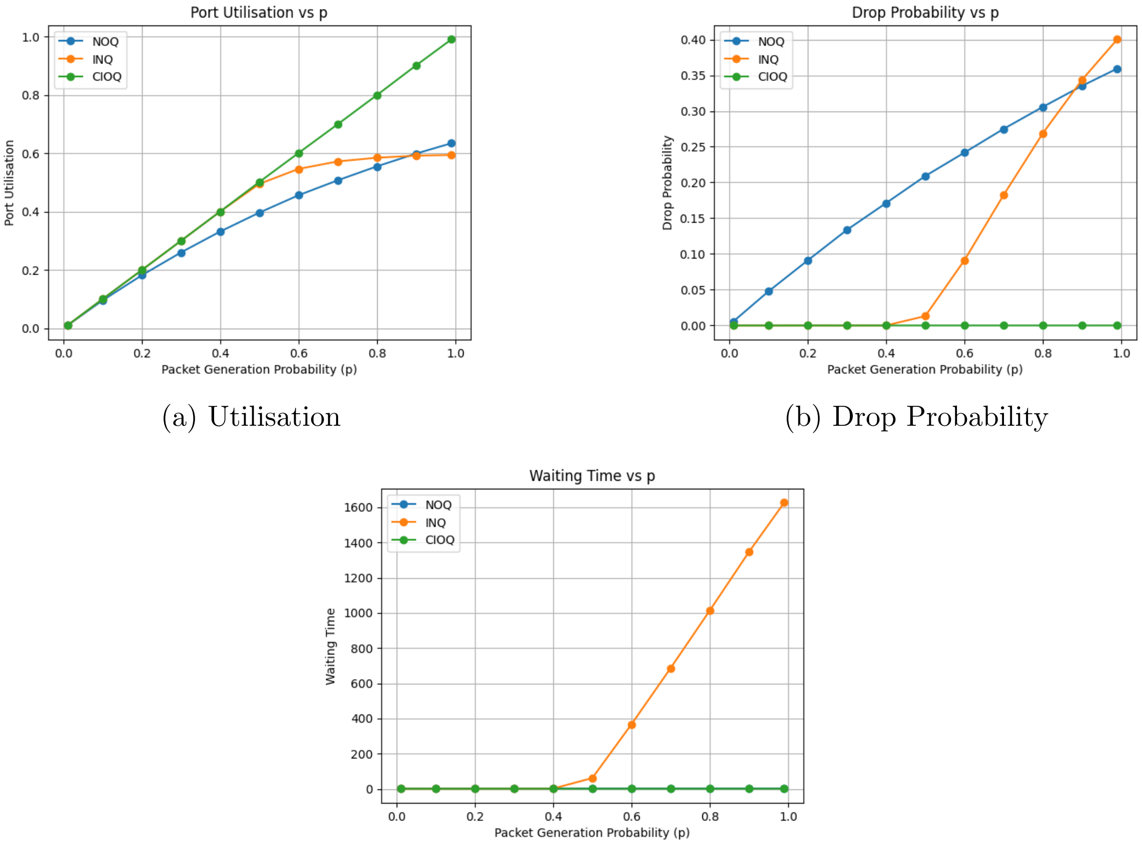
6 Results

6.1 Comparison with different probabilities

The above figure compares the link utilisation, drop probability and waiting time for the different algorithms with default parameters. For CIOQ, $K = 4$ and $L = 6$ was chosen.

6.1.1 Utilisation

From the figure we see that iSLIP algorithm has the best utilisation and increasing linearly with the increase in probability. As derived in class, we see that NOQ has a saturation utilisation of 64% while INQ can only have a maximum utilisation of 58.6%. One important insight from this is that although NOQ outperforms INQ, at lower loads, the INQ gives a much better utilisation than NOQ.



(a) Utilisation

(b) Drop Probability

(c) Wait Time

Figure 1: Comparing the different Queueing algorithms

6.1.2 Drop Probability

CIOQ with iSLIP algorithm has as a consistent nearly 0 drop probability, whereas NOQ and IOQ have drop probabilities reaching to upto 40% for $p = 1$. Again, here INQ outperforms NOQ under low loads, but becomes worse under higher loads.

6.1.3 Waiting Time

The waiting time for NOQ and CIOQ is consistently 1 time slot. But for INQ, the wait time increase approximately linearly after $p = 0.5$, and reached a maximum of 1600 time slots. Thus, considering waiting time as the metric, NOQ and CIOQ are preferred.

]

7 Conclusion

The iSLIP performs much better than No Queue (NOQ) and Input Queueing with FIFO scheduling (INQ) on link utilisation and drop probability metrics. Further it also achieves packet delays of nearly 1 time slot not unlike NOQ. Among INQ and NOQ, INQ suffers from much higher packet delays. At lesser loads, INQ outperforms NOQ in drop probability and link utilisation, but NOQ surpasses INQ at higher loads, as is expected from the theoretical results.

References

[1] N. McKeown. "The iSLIP scheduling algorithm for input-queued switches". In: *IEEE/ACM Transactions on Networking* 7.2 (1999), pp. 188–201. DOI: `10.1109/90.769767`.

[2] M. Karol, M. Hluchyj, and S. Morgan. "Input Versus Output Queueing on a Space-Division Packet Switch". In: *IEEE Transactions on Communications* 35.12 (1987), pp. 1347–1356. DOI: `10.1109/TCOM.1987.1096719`.

[3] Long Gong et al. "SERENADE: A Parallel Iterative Algorithm for Crossbar Scheduling in Input-Queued Switches". In: *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*. 2020, pp. 1–6. DOI: `10.1109/HPSR48589.2020.9098995`.

List of Figures

1 Comparing the different Queueing algorithms 4