

# A Performance Evaluation of Multiple Input Queued (MIQ) Switch with Iterative Weighted Slip Algorithm

S N Kore, Sayali Kore, Ajinkya Biradar  
Electronics dept.  
Walchand college of Engineering,  
Sangli, India

Dr. P J Kulkarni  
Computer Science dept.  
Walchand college of Engineering,  
Sangli, India

**Abstract**— Many researchers had evaluated the throughput and delay performance of virtual output queued (VOQ) packet switches using iterative weighted/un-weighted scheduling algorithms. Prof. Nick Mckeown from Stanford University had evolved with excellent iterative maximal matching (i-slip) scheme which provides throughput near to 100%. Prof. Kim had suggested multiple input queued architecture which also provide more than 90 % throughput for less number of input queues per port. (In VOQ N queues per port are used). Our attempt is to use MIQ architecture and evaluate delay, throughput performance with i-slip algorithm for scheduling. While evaluating performance we had used Bernoulli's and Bursty (ON-OFF) traffic models.

**Keywords**- Network communications; Packet-switching networks;routing protocols; Sequencing and scheduling.

## I. INTRODUCTION

A High speed switches mainly classified as input queued (IQ) switch, output-queued (OQ) switch, and combined input-and output-queued (CIOQ) switch. An OQ switch buffers cells at the output ports. OQ switches guarantee 100% throughput since the outputs never idle as long as there are packets to send. An  $N \times N$  OQ switch must operate N times faster than the line rate. Memory technology cannot meet that kind of high-speed requirement [1]. Therefore, IQ and CIOQ switches have gained widespread attention. The input queue switch has limitation of throughput equal to 58.6% [1] [2]. The most common architecture is the CIOQ switch in which buffering occurs both at the input and at the output. But CIOQ always need speedup high speed-up factor of two to provide 100% throughput. Both IQ and CIOQ switches use virtual output queuing in which each input maintains a separate queue for cells destined for each output [2][3].

Matching algorithms for Virtual output queuing removes head-of-line (HOL) blocking and overcomes limit on the throughput single FIFO queue [1]. In virtual output queued switches scheduling or selection of packets at HOL is critical issue. Many algorithms have been proposed for scheduling an IQ switch to obtain high throughput. All the algorithms find a matching between the inputs and outputs, but they were derived with different weighing techniques. Under the matching paradigm, the scheduler matches an input with an output and finds the maximal number of those pairs in a given time slot. This usually takes a few iterations in one time slot. Numerous algorithms work in iterative way and most of them are variants of i-slip algorithms [4] [7] [8]. The i-slip

algorithm innovated by Prof. Nick-Mckeown had played vital role in development of switching architecture [4] [5] [6] [7] [8]. In multiple input queued (MIQ) architecture there are M queues per input port. Total NM queues are used in MIQ whereas  $N^2$  queues are used in VOQ switches. Even with  $M=8$  and  $N=64$  throughput achieved is greater than 92%. In VOQ we need to handle 4096 queues and in case of MIQ only 512 queues need to be handled. It's quite interesting to analyze the performance of MIQ with i-slip. The i-slip algorithm have not been evaluated for multiple input queued switch (MIQ) where number of queues per input port is less than N if size of switch is  $N \times N$ . We are reporting the performance of i-slip in MIQ under Bernoulli's arrival and bursty arrival.

## II. SWITCH AND TRAFFIC MODEL

### A. Switch Model

This section describes the switch model. Here number queues per port (M) used are less that size of switch ( $N \times N$ ) where  $M \leq N$ . In VOQ  $N^2$  queues needs to be taken care where as in MIQ only NM Queues needs to be taken care. Our aim is to obtain throughput to be 100%, that restrict condition that every cell slot time we need to select non-conflicting N input-output matches among NM matches ( $N^2$  in case of VOQ). Suppose  $M=2$ , indicate that there are two queues per port.

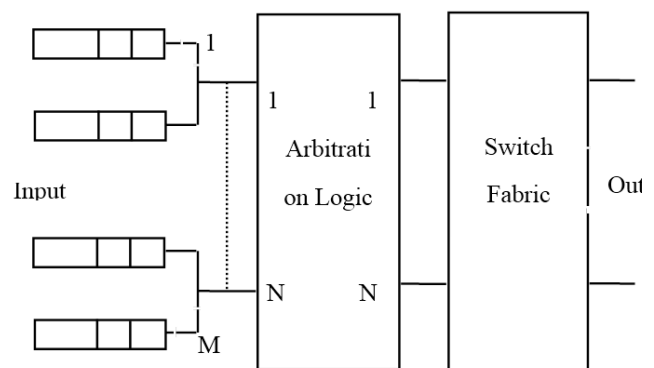


Figure 1. MIQ Switch

Arrivals, destined for output ports with even number are saved in one queue at input ports and others are saved at another queue. In general arrival to an output port N is saved in  $k^{\text{th}}$  queue at input port where  $k = N \bmod M$  where  $k=1,2,\dots,M$ . This approach introduces a new problem as there

are now (a maximum of)  $N^2$  packets at HOL in case of VOQ and NM packets in MIQ for selection.

The problem of selecting, N packets among NM packets to transmit becomes much more complex scheduling problem. The performance of such architecture is determined by the arbitration algorithm. This is illustrated in section 3.

### B. Traffic Model

**Bernoulli's arrival:** In this arrival process the cell arrived in each time slot is identical and independent of other time slot. Assume that probability that cell arrives is p. Each arrived cell chooses output equally likely. Hence traffic is said to be uniformly distributed over output port Please do not revise any of the current designations.

**Bursty arrival:** Basically this type of modeling of traffic source is called as ON-OFF type. Here in ON-period (active) source sends packets & in OFF-period (silent) no packets are sent.

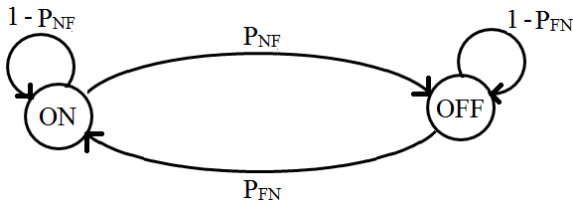


Figure 2. ON-OFF Traffic Model

Time is slotted and packets are generated in slot hence it is called as Markov Modulated Bernoulli's process (MMBP) with two states.

It is further classified as MMDP i.e. as Markov Modulated Deterministic process.

$$\text{State transition matrix } T = \begin{bmatrix} 1 - P_{NF} & P_{NF} \\ P_{FN} & 1 - P_{FN} \end{bmatrix}$$

$$\text{Prob[ON state]} = \frac{P_{FN}}{P_{FN} + P_{NF}}$$

$$\text{Prob[OFF state]} = \frac{P_{NF}}{P_{FN} + P_{NF}}$$

$P_n$  = Prob that ON state has length 'n' slot i.e being ON state it will remain for another (n-1) times in ON state and then goes to OFF state.

$P_n = (1 - P_{NF})^{n-1} \cdot P_{NF}$ . Its geometric distribution with Mean burst length  $L_b$ .

$$L_b = \sum_{i=1}^{\infty} i P_{NF} (1 - P_{NF})^{i-1} L_b = \frac{1}{P_{NF}}$$

$$\text{Offered Load} = \rho = \frac{P_{FN}}{P_{NF} + P_{FN}}$$

Burst length chosen is 16 and offered load  $\rho$  is 0.8 then  $P_{NF} = 0.0625$  and  $P_{FN} = 0.25$  which are used to change the state of the system. If system is in ON state it always generate packet uniformly distributed to any output port till system changes the state.

## III. PREPARE YOUR PAPER BEFORE STYLING

### A. Round-Robin Matching (RRM) algorithm

Before RRM is very similar to Prof. Anderson's Parallel Iterative Matching (PIM) [3] [4], where packet selection is done at random, it uses modulo N round robin arbiters, one for each input and one for each output. Each arbiter maintains a pointer, indicating the element that currently has highest priority. RRM operates as follows:

**1. Request:** Each unmatched input sends a request to each output if it has at least one packet at HOL.

**2. Grant:** Each output that has received at least one request selects one request to grant by means of its round-robin arbiter. It chooses the input that appears next in the round robin, starting from the input currently being pointed to. The pointer which is advanced (modulo N) to one beyond the input just granted.

**3. Accept:** Similarly, each input that has received at least one grant will select one grant to accept by means of its round-robin arbiter. It chooses the output that appears just next in the round robin, starting from the output currently being pointed to. The pointer is advanced (modulo N) to one beyond the output just accepted. Unfortunately, RRM does not perform very well even under uniform i.i.d. Bernoulli arrivals; saturation throughput is merely 63%, which is close to that of PIM. The reason for reduction in throughput is because output arbiters tend to synchronize, causing multiple arbiters to grant to the same input, which leads to a waste of grants and thus poor throughput.

### B. i-SLIP in VOQ with Bernoulli's arrival

i-slip is an improvement on RRM, aimed at preventing synchronization of arbiters. Its operation is very similar to RRM, with only a modification in step 2 of how the pointers are updated:

**1. Request:** Same as RRM.

**2. Grant:** Each output that has received at least one request will select one request to grant by means of its round-robin arbiter. It chooses the input that appears next in the round robin, starting from the input currently being pointed. The pointer is advanced to one beyond the input just granted if and only if the grant is accepted in step 3.

**3. Accept:** Same as RRM. Note that this is almost identical to non-iterative SLIP, with the exception of the added condition in steps 2 and 3: the pointers are only updated in the first iteration for reasons of fairness. Compared to SLIP, iterative SLIP improves performance further when the numbers of iterations are increased. On an average i-SLIP appears to converge in about  $O(\log(N))$  iterations, a result similar to PIM..

Fig.3 indicates that saturation throughput under VOQ (i.e for 16x16 switch with 16 queues per port) can be achieved to be 100% under 1, 2 or 4 slip. Increasing Number of iterations improves the delay performance.

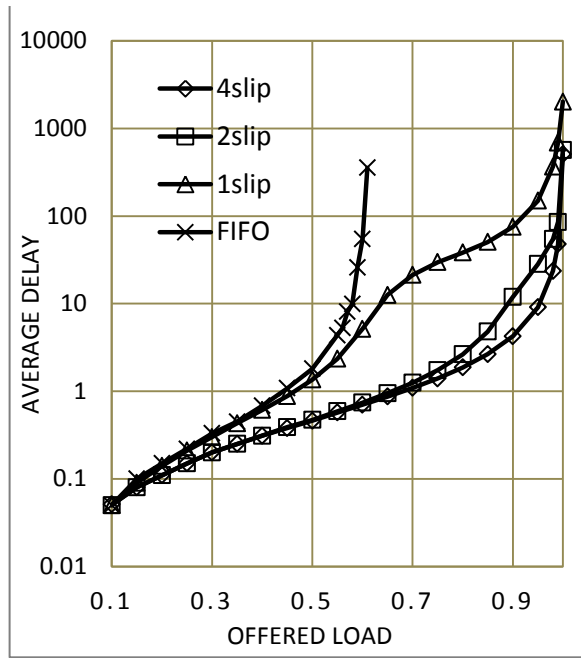


Figure 3. Delay performance of switch size 16x16 with VOQ with Bernoulli's arrival and i-slip of 1,2,4.

C. *i-SLIP in VOQ with Bursty arrival*

Fig.3 shows the performance, evaluated for switch size of 16x16 with 16 queues per port and bursty traffic (ON-OFF) with different burst size with multiple number of iterations. Burst size selected is 16 and 64.

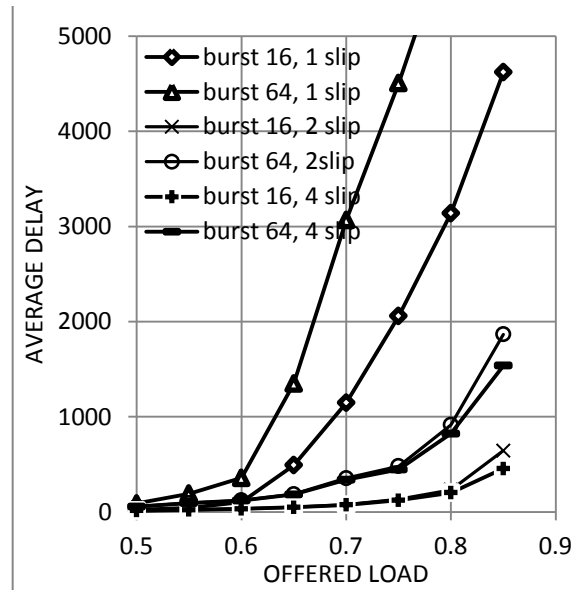


Figure 4. Delay performance of switch size 16x16 with VOQ with Bursty arrival and i-slip of 1,2,4.

Delay performance is degraded as burst size is increased. Such model is analytically analyzed by Prof. Kelinrock and Prof. Kim with restricted rule. Here Iterative slip is un-weighted i.e. matching of input output does not consider any bias such as length of queues or longest port first etc. With slip

more than 2 does not improve the performance under lighter input load (less than 0.7) but it observed that under higher input load (more than 0.85) through and delay performance is improved.

Fig. 4 indicate the delay performance of i-slip for switch size 16x16 with number of queues per port are 4,8 with number of iterations are 1,2,4 slip. In 8 queues per port with 2 slip and 4 Queues per port with 4 slip has same performance. It's obvious that increase in number of queues per port and increasing number iteration in i-slip will give performance nearer to output queuing.

IV. I-SLIP IN MIQ

Our attempt is to evaluate the performance of i-slip algorithm if number of queues per port is reduced to M where  $M < N$  the model is identified as MIQ. Here 16x16 with number queues per port reduced to 8 with 1-slip, then this model is equivalent and approximated to even and odd queues where throughput is saturated to 76.4 % [9]. Iterative slip in VOQ suggested by Prof. McKeown is implemented in Cisco router 1200 and giving best performance [5]. Our attempt is that the McKeown's i-slip implementation can be extended to MIQ where management of  $N^2$  queues reduces to NM queues only. Even through the saturation throughput is limited in MIQ can be overcome by implementing iterative i-slip. Number of iteration of  $O(\log(N))$  are sufficient for achieving throughput of 100%. In fig 2 it clearly shows that in 16x16 switch with 8 number of queues per port and slip 1,2 4 has increased saturation throughput from 76%, 91% , 98% respectively. As the number of iteration is increased delay performance is also improved.

A. *Weighted i-SLIP in MIQ with Bernoulli's arrival:*

Here in Fig.5 simulation graphs are drawn for number of queues per port 4 and 8 along with variation of slip. Arrival Traffic is Bernoulli's arrival with uniform distribution. Each input port will send requests to output port depending on HOL packet destination address along with queue length in that queue. Each input port can send maximum M requests to the output arbiter. In case of VOQ, there might be maximum N requests from each input port. Total Number of requests sends to output arbiters can be NM, which is reduced in MIQ (In VOQ it is  $N^2$ ).

Arbiter at the output port will receives number of requests. Arbiter at the output will grant one request among the received from various input ports which have highest queues length. Grants received at input port i from different output ports j are evaluated. If multiple grants are received then one is chosen which has highest queues length. Once the input arbiter accepts the jth port request then queue number,  $j \text{ mod } M$  is evaluated to select queue from corresponding input port to remove cell from its HOL.

Let system be queues/port be  $M=8$  and number of ports be  $N =16$ . In VOQ there is  $M=N$ , hence each input port can maximum send 16 requests to 16 arbiters at output ports if there is cell at HOL. In case of MIQ there maximum 8 requests will be sends from each input port to different 16 output arbiter. At input port 1 there are 8 queues and queue no.1 at input port 1 can store cells destine to output port 1 or 9.

TABLE I. DELAY PERFORMANCE UNDER BERNOULLI’S ARRIVAL FOR SWITCH 16x16 WITH 8 QUEUES PER PORT AND SLIP OF 1, 2, 4

16x16 switch with 8 queues per port					
4 slip		2 slip		1 slip	
Load	Delay	Load	Delay	Load	Delay
0.1	0.05	0.1	0.05	0.1	0.05
0.15	0.08	0.15	0.08	0.15	0.09
0.2	0.12	0.2	0.11	0.2	0.14
0.25	0.16	0.25	0.16	0.25	0.21
0.3	0.2	0.3	0.2	0.3	0.3
0.35	0.26	0.35	0.26	0.35	0.43
0.4	0.32	0.4	0.32	0.4	0.61
0.45	0.4	0.45	0.4	0.45	0.9
0.5	0.49	0.5	0.5	0.5	1.4
0.55	0.61	0.55	0.63	0.55	2.48
0.6	0.76	0.6	0.8	0.6	5.81
0.65	0.95	0.65	1.02	0.65	18.87
0.7	1.23	0.7	1.81	0.66	25
0.75	1.63	0.75	2.84	0.67	33.11
0.8	2.28	0.8	5.65	0.68	44.63
0.85	3.49	0.85	7.03	0.69	64.63
0.9	6.64	0.9	29.05	0.7	114.19
0.95	27.8	0.91	48.52	0.75	2329
0.96	67	0.92	100		
0.97	276	0.93	301		
0.98	652				
0.99	1115				
1	1537				

Hence input arbiter at port number 1 can send request for HOL at M=1 to outputs port 1 or output port 9 depending on current address in HOL cell. While sending queue length it is number of cells waiting in its queue which contains cells destined to output port 1 and 9.

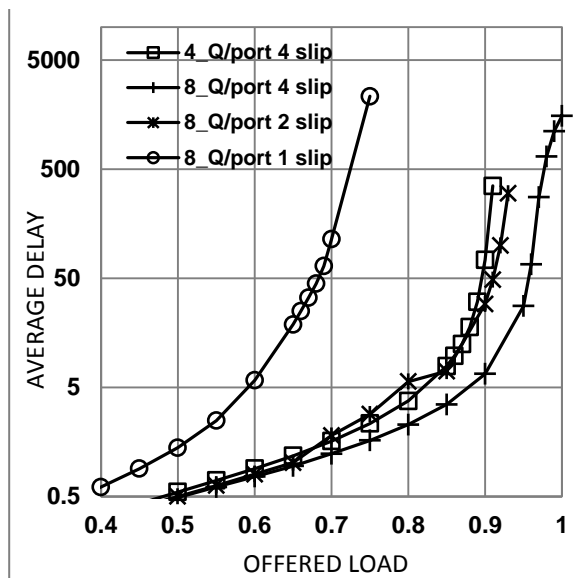


Figure 5. Delay Performance under Bernoulli’s arrival for Switch 16x16 with 8 queues per port and slip of 1,2,4

The input output ports for which matching is obtained will not take any part in further iterations. It is observed that 4

iteration are sufficient to find maximal match and throughput to be 100%.

Fig.5 shows the graph of delay performance for 16x16 switch with 8 queues per port. Here total input queues are 128 instead of 256. In case of 8 queues per port with 1-slip limits maximum maximum throughput approximated to 76%. As the number of slips are increased then throughput increases to 84% & 98% with slip of 2& 4. Delay is also bounded under heavy traffic load conditions.

Fig. 5 indicate the delay performance of i-slip for switch size 16x16 with number of queues per port are 8 with number of iterations are 1,2,4 slip. In 8 queues per port with 2 slip and 4 Queues per port with 4 slip has same performance. It’s obvious that increase in number of queues per port and increasing number iteration in i-slip will give performance nearer to output queuing. i-slip in MIQ with bursty arrival

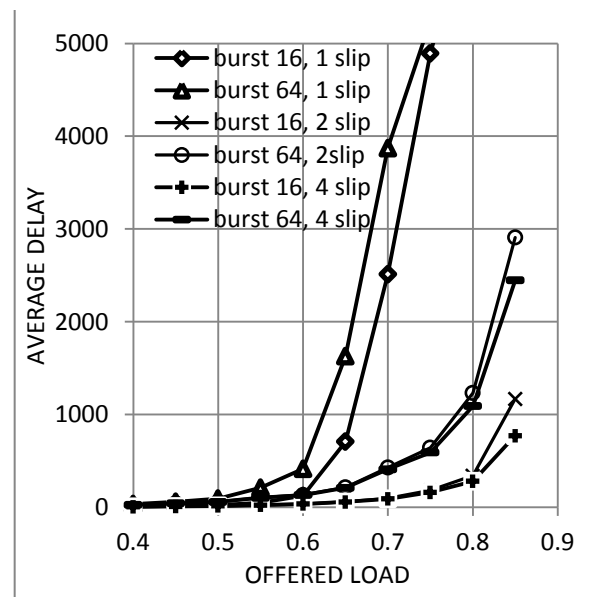


Figure 6. Fig. 6 Delay Performance under Bursty arrival for Switch 16x16 with 8 queues per port and slip of 1,2,4

In Fig.6 performance evaluation of MIQ switch with switch size of 16 and number of queues per port = 8 are taken with different burst size and slip is varied as 1, 2, 4. It is observed that as slip is increased throughput delay performance approaches output queuing. It’s always recommended if traffic is bursty then increase slip for better performance.

### V. CONCLUSION

Here performance of i-slip under MIQ structure with uniform Bernoulli’s arrival and bursty (on-off) arrivals. Increasing number of iterations is more flexible than increasing number of queues port and is the key for obtaining good delay throughput performance. Increase in the burst size degrades the performance of switch even under virtual output queuing. Maximum Weight matching algorithm can be the better solution to provide good delay throughput performance. Such algorithms are computationally complex and have to be implemented on parallel architectures for real time application.

There are different variants of i-slip are available and our work can be extended to these algorithms to obtain better performance.

#### REFERENCES

- [1] M.G.Hluchyj and M.J.Karol, "Queuing in high performance packet switching" IEEE J.Selected Area of Communication, Vol6, pp.1587-1597, December 1988
- [2] S. N. Kore, V. B. Dharmadhikari, H. S. Jamadagni, "Cell Selection Policies in Multiple Input Queued ATM Switch", Centre for Electronics Design and Technology, IISC Bangalore-12. Published in The Canadian Conference on Broadband Research (International conference) CCBR - 99
- [3] McKeown, N., "The iSLIP Scheduling Algorithm for Input-Queued Switches," IEEE/ACM Trans. Networking, vol. 7, no. 2, Apr. 1999, pp. 188-201.
- [4] C. Koliás, I.kleinrock "On odd even ATM Switch", IEICE Trans. On commun. vol.e81b, no. 2, Feb. 1998, pp. 244-250
- [5] Saad Mneimneh, "Matching From the First Iteration: An Iterative Switching Algorithm for an Input Queued Switch" IEEE/ACM Trans. Networking, vol. 16, No. 1, Feb. 2008, pp. 206-217.
- [6] Alessandra Scicchitano, Andera Binaco, Palol Giaccone, Emilio Leonardi, Enrico Schiattarella, "Disrtibuted scheduling in input queued switches" IEEE Comm. Society ICC, 2007
- [7] Mohsen Bayati, Balaji Prabhakar, Devavrat Shah, Mayank Sharma, "Iterative Scheduling Algorithms" IEEE Comm. Society IEEE INFOCOM, 2007
- [8] Kevin F. CHEN, Edwin H.-M. SHA, S.Q. ZHENG "Fast and Noniterative Scheduling in Input-Queued Switches" Int. J. Communications, Network and System Sciences, 2009, 3, pp. 169-247

#### AUTHORS PROFILE

**Mr. S N kore:** received his B.E degree in Electronics and Telecommunication from University of Pune at Pune and M.E. degree in electronics from Shivaji University at Kolhapur in 1983 and 1991respectively.He has worked as scientific officer at Tata Institute of Fundamental Research (TIFR) at Pune from 2nd Nov'83 to 1st Oct'84. Currently he is an Associate Professor of electronics engineering at Walchand College of engineering, Sangli, Maharashtra.

**Miss Sayali S.Kore:** she had done B.tech in Electronics Engineering from Walchand College of Engineering Sangli in 2012. She is recipient of gold medalist in 2012 at WCE Sangli. Her area of interest is Signal processing, Image processing, VLSI Engineering and Computer Networking.

**Mr. Ajinkya Biradar:** received his B.E. degree in Electronics and Telecommunication from University of Pune at Pune in 2010 and is pursuing M.Tech in electronics from Walchand College of engineering, Sangli

**Dr.P.J. Kulkarni:** He is currently working as Deputy Director of Walchand College of Engineering (WCE) Sangli. He is Prof. in Department of Comp. Sci. and Engineering at WCE Sangli. He is recipient of state level Ideal Teacher award winner in 2011-12. He has published more than 24 papers in international Journal and national journals. Also completed 9 research projects. He has guided more than 10 Phd in Shivaji University.