Analysis of iSLIP scheduling algorithm for input-queuing switches

Mitko Gospodinov, Evgeniya Gospodinova

Abstract: *The paper presents one of the most up-to-date iSLIP (iterative round robin matching with slip) algorithms for configuration and scheduling input-queuing switch fabrics. The aim of the paper is research the performance of the algorithm regarding cell delay in the queues depending on the load for input-queuing switches with 4, 8, 16 and 32 ports. The advantage of the algorithm is achieving almost 100% throughput as result of de-synchronization of output arbiters playing important role in providing low latency.*

Key words: *iSLIP algorithm, FIFO, HOL effect, throughput, cell delay.*

INTRODUCTION

The output queuing packet switch architectures are one of the most up-to-date due to providing optimal performance. They require a switch fabric speedup equal to N (N is number of switch input/output ports) and internal data transfer rate is N times higher than the external link speed. When data rates on fibre links grow very rapidly the internal speed of output queuing packet switches is difficult to meet the requirements. These requirements could be meet using input queuing switch architectures. They are designed to operate with a switching fabric running at an internal rate equal to the external links speed. If FIFO (first in first out) packet queues are used, this architecture suffers from HOL (head of the line) blocking effect that severely limits their performance to approximately 58.6% of the maximum [4]. The HOL blocking can be overcome by the use of virtual output queuing. A scheduling algorithm is used to configure the switch, deciding the order in which packets will be served. The presented iSLIP-scheduling algorithm is researched about cell delay in the switch queues and the degree of synchronization of the output arbiters. Prototype and commercial implementations of iSLIP exist in systems with aggregate bandwidths ranging from 50 to 500 Gb/s [2].

1. DESCRIPTION OF iSLIP SCHEDULING ALGORITHM

The iSLIP-scheduling algorithm uses rotating priority arbitration to schedule each active input and output in turn. The arbitration is carried out in three steps in the each iteration. The main characteristics of iSLIP are: high throughput, starvation free, fast and simple to implementation [3]. The iSLIP scheduling algorithm can be briefly described as follows:

- Step 1 (Request) – Every unmatched input sends a request to every output for which it has a queued cell;
- Step 2 (Grant) – If an unmatched output receives any requests, it chooses the one that appears next in a fixed. Round-robin schedule starting from the highest priority element. The output notifies each input whether or not its request was granted. The pointer g_i to the highest priority element of the round-robin schedule is incremented to one location beyond the granted input if the grant is accepted in step 3 of the first iteration. The pointer is not incremented in subsequent iterations.
- Step 3 (Accept) – If an input receives a grant, it accepts the one that appears next in a fixed. The pointer a_i to the highest priority element of the round-robin schedule is incremented to one location beyond the accepted output.

The main properties of the iSLIP scheduling algorithms are:

- Flows matched in the first iteration become the lowest priority in the next cell time.

- The flows aren't starved, because of the requirement that pointers aren't updated after the first iteration. The output will continue to grant to the highest priority requesting input until it is successful.
- The algorithm will converge in at most N iterations (one to every input and one every output). Each iteration will schedule zero, one or more flows.
- The high performance of iSLIP is a consequence of step 2, because the output arbiter is updated if a successful match in step 3. Moving the pointers not only prevents starvation, but it tends to desynchronise the grant arbiters. This de-synchronization of the grant arbiters achieving 100% throughput in an input-queued switch.

2. ANALYSIS OF iSLIP SCHEDULING ALGORITHM

The performance of the described algorithm is researched through simulation by conditions as follows:

- The size switch e NxN.
- Arriving packets are of fixed and equal length, called cells.
- The arrival processes is Bernoulli.
- All arrival processes have the same arrival rate and destinations are uniformly distributed over all outputs.
- Arrival process at each input is independent of arrivals at other inputs.

2. 1 Cell delay in the switch queues

The performance of iSLIP decreases differently under low and heavy loads. For a low offered load (<0.5 Erlang), the arriving cells find the arbiters in random positions and iSLIP performs in a similar manner of PIM algorithm. The cell delay for low offered load is given by [1]:

$$D = p(1-(N-1/N)^{N-1}) \quad (1)$$

If $N \rightarrow \infty$:

$$D= p(1-(1/e))$$

Where: D – mean waiting time of cells in the queues for low offered load (< 0,5 Erl.);
p - offered load;
N - switch size.

Under heavy load, the algorithm serves each FIFO once every N cycles and the queues will behave similarly to M/D/1 queue with arrival rates p/N and deterministic service time N cell times. The mean waiting time of cells in the queues for low offered load is given by [5]:

$$D = p\,N/(2(1-p)) \quad (2)$$

Where: D – mean waiting time of cells in the queues for heavy offered load (>0,5 Erl.);
p - offered load;
N - switch size.

In the Table 1 are show the mean waiting time imposed by a iSLIP schedule algorithm as a function of offered load for switches with 4, 8, 16, 32 ports and M/D/1 queue. On the Fig. 1 is shown the effect of the function D = f(p). The reason for the

different characteristics under low and heavy load lies of the degree of synchronization of the arbiters.

Table 1 Waiting time of offered load

Offered load P [Erlang]	D [Time slots] N=4	D [time slots] N=8	D [Time slots] N=16	D [Time slots] N=32	D Time slots M/D/1 queue
0,1	0,06	0,06	0,06	0,06	0,06
0,2	0,12	0,12	0,12	0,13	0,12
0,3	0,17	0,18	0,19	0,19	0,21
0,4	0,23	0,24	0,25	0,25	0,33
0,5	0,29	0,30	0,31	0,32	0,50
0,6	3,00	6,00	12,00	24,00	0,75
0,7	4,67	9,33	18,67	37,33	1,17
0,8	8,00	16,00	32,00	64,00	2,00
0,9	18,00	36,00	48,00	144,00	4,50

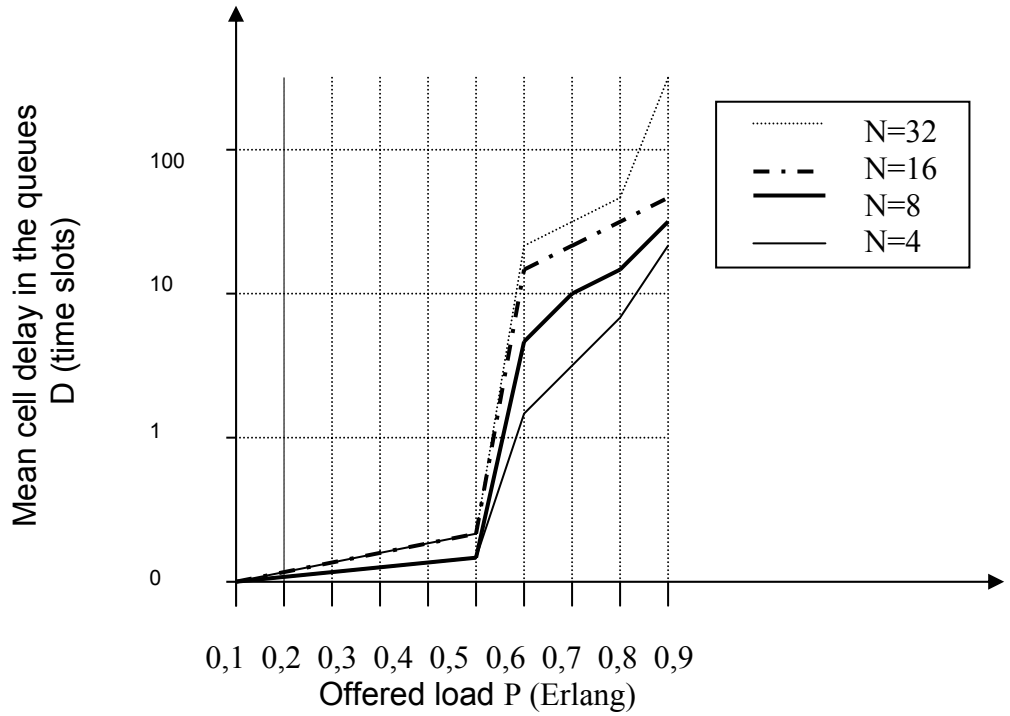

Fig. 1 Cell delay in the queues of offered load

On the Fig.2 is shown the comparison of mean waiting time of cells in the queues for iSLIP algorithm and M/D/1 queue. The iSLIP behaves very similarly to the M/D/1 queue, but with a higher delay. This is because the service policy is not constant. When a queue changes between empty and nonempty, the scheduler must adapt to the new set of queues that require service. This adaptation takes place over many cell times while the arbiters desynchronize again. During this time, the throughput will be worse than for the M/D/1 queue and queue length will increase. This will lead to increasing waiting time of the cells in the queues.

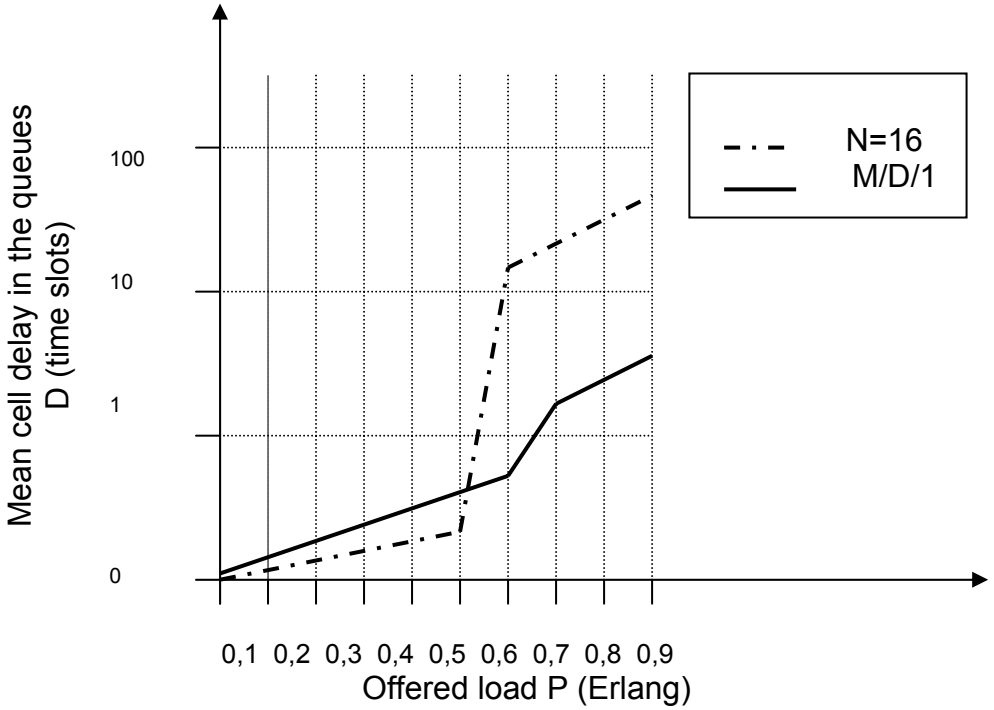

Fig. 2 Waiting time in the queues for iSLIP algorithm and M/D/1 queue of offered load

2. 2 De-synchronization of the iSLIP algorithm

Performance of iSLIP algorithm is defined by the degree of synchronization of the output schedulers. The expected number of synchronized output schedulers at time *t* is given by [5]:

$$S(t) \approx N-\lambda N[(N-1)/\lambda N]^{\lambda\lambda 1N} - \lambda^2{}_1 N[(\lambda_1 N-1)/\lambda_1 N]^{\lambda N-1} \quad (3)$$

Where N - number of ports;

λ - arrival rate averaged over all inputs;

$\lambda_1 = 1-\lambda$.

In the Table 2 are shown analytical approximation for the average number of synchronized output schedulers for a 16x16 switch with Bernoulli arrivals. The graphical results are shown on the Fig. 4.

Table 2 The number of synchronized output schedulers

Offered load P [Erlang]	Number of synchronized output schedulers
0,1	10
0,2	9
0,3	9
0,4	9
0,5	8
0,6	8
0,7	7
0,8	5
0,9	3
1,0	0

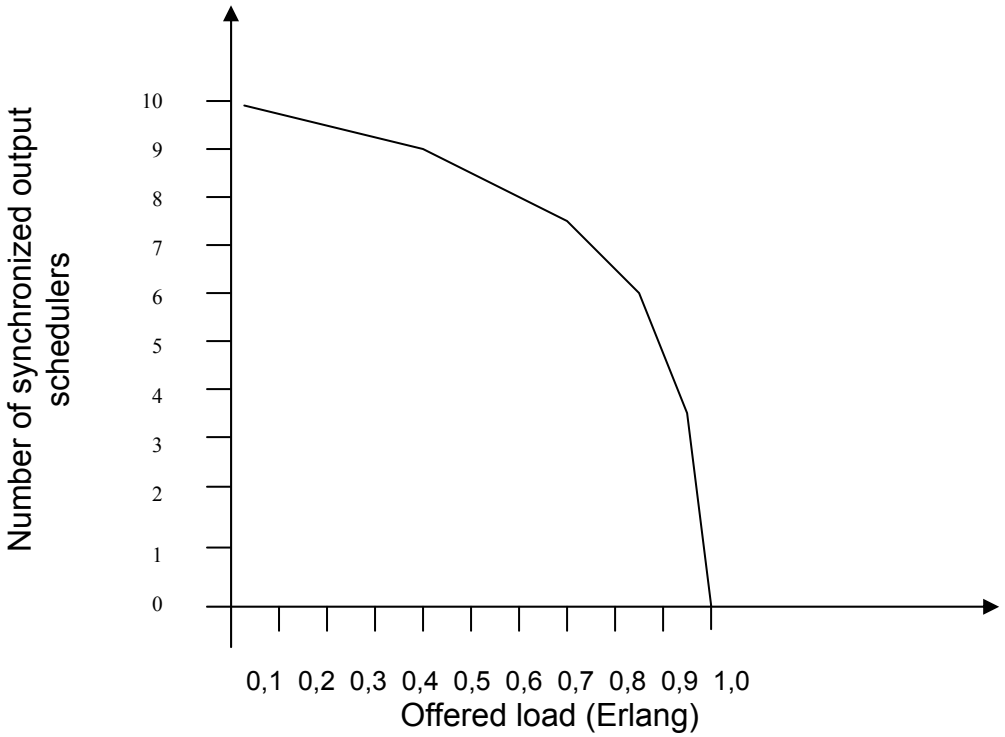

Fig. 3 The number of synchronized output arbiters of offered load

The synchronization of the grant pointers limits performance by random arrival process. The Fig. 3 shows the number of synchronized output arbiters as a function of offered load for 16x16 switch. The degree of synchronization decreases when the offered load is increased.

3. CONCLUSIONS

From analysis of iSLIP schedule algorithm can be done the following conclusions:

- iSLIP can achieve 100% throughput, because it uses rotating priority arbitration to schedule each active input and output in turn.
- This algorithm is fast, simple and can be implemented in a single chip.
- The iSLIP should not allow a nonempty virtual output queuing to remain unserved indefinitely.
- To achieve the highest bandwidth switch is needed the schedule algorithm does not become the performance bottleneck.
- The reason for the different characteristics under low and heavy load lies of the degree of synchronization of the arbiters.
- The iSLIP-scheduling algorithm behaves very similarly to the M/D/1 queue, but with a higher delay, because the service policy is not constant.

The iSLIP algorithm presented in this paper is implemented in 16 – port commercial IP router with an aggregate bandwidth of 60 Gb/s in the recent line of Cisco Gigabit routers (Cisco 12000 series) and 32-port prototype Tiny Tera switch with an aggregate bandwidth of 0.5 Tb/s.

REFERENCES

[1] Iversen, V., "Teletraffic engineering", TU Denmark, 1999.

[2] McKeown, N. "Achieving 100% throughput in an input-queued switch". IEEE INFOCOM '96, San Francisco.

[3] McKeown, N. "The tiny tera: A small high-bandwidth packet switch core", IEEE Micro, vol.17, pp. 26-33, 1997.

[4] Mekkittikul, A., N. McKeown, "A practical scheduling algorithm for achieving 100% throughput in input-queuing switches", INFOCOM'98, San Francisco.

[5] Mekkittikul, A., "A Starvation-free algorithm for achieving 100% troughput in an input-queued switch", IEEE ICCCN, 1996.

ABOUT THE AUTHORS

Assoc. Prof. Mitko Gospodinov, PhD, Central Laboratory of Mechatronics and Instrumentation at Bulgarian Academy of Sciences, Fhone: +359 887 426 490, E-mail: mitgo@abv.bg

Assistant Prof. Evgeniya Peneva Gospodinova, Central Laboratory of Mechatronics and Instrumentation at Bulgarian Academy of Sciences, Phone: +359 886 626 911, E-mail: jenigospodinova@abv.bg