Volume: 03; Issue: 03; Pages:1194-1198 (2011)

ISA-Independent Scheduling Algorithm for Buffered Crossbar Switch

Dr. Kannan Balasubramanian
Department of Computer Science, Mepco Schlenk Engineering College, Sivakasi-05.
Email: kannanbala@mepcoeng.ac.in
C.Sindhu
Department of Computer Science, Mepco Schlenk Engineering College, Sivakasi-05.
Email: shristi.cse@gmail.com

----------ABSTRACT----------

Crossbars are an emerging technology in high speed router. In buffered crossbar switch a buffer is associated with each crossbar. Due to the introduction of crossbar buffers the input and output contentions are highly eliminated. Both the input and output port work independently and so the scheduling process is highly simplified. With a speed up of 2 it has been proved that 100% throughput can be achieved. In this paper we propose a 100% throughput scheduling algorithm without speedup called ISA. In ISA the scheduling decision is purely based on the state information of the local crosspoint buffer. So it is suited for a distributed implementation and it is highly scalable. The main advantage is the crosspoint use a buffer size of 1 which minimizes the hardware cost. Simulation results show that we achieve 100% throughput without any speedup.

Keywords – Buffered Crossbar switch, Crosspoint buffer, Input queued, Output queued, Speedup, Virtual Output queue,100%throughput.

Date of Submission: May 29, 2011 Date Revised: June 16, 2011 Date of Acceptance: July 30, 2011

1. INTRODUCTION

Buffered Crossbar switches are considered as a robust solution in facing the challenging design of today's routers. For the design of high performance switching fabric crosspoint buffered switches are a good solution. They reduce the complexity of the scheduling algorithm and provide a good delay performance. With the increasing bandwidth it is challenging to design efficient scheduling algorithm and high performance crossbar switches.

The packets can be buffered at the input ports, output ports, or the crosspoint of the crossbar. For an input queued (IQ) switch buffers are present only in the input ports. Several iterative maximal matching algorithms were proposed for IQ switches. Due to the efficient scheduling algorithm and economical hardware architecture these switches became popular. But they could provide 100% throughput when working with maximum weight matching algorithm.

The output queued (OQ) switch have buffers in the output port. They could provide 100 percent throughput by running various fair scheduling algorithms at the output port. However, for an N*N switch to achieve 100 percent throughput speedup of N is required. There is no buffer space in the input port, so all the N packets have to be simultaneously transmitted to the output port by the crossbar. Hence the crossbar should possess N times bandwidth as that of an input or output port. This makes the OQ switches difficult to scale. So the advantages of both the input and output queued switches where combined and the Combined Input and Output queued switches (CIOQ) were proposed. They have a buffer in the input and the output ports and need a speedup of two. With the speed up of two they can achieve 100 percent throughput with any maximal scheduling algorithms.

The traditional FIFO suffers from the head of line (HOL) blocking. So the Virtual Output Queue (VOQ) buffering technique is used in the input side of the switch. Even if the packets behind the HOL packet may be free, the packets cannot be scheduled to the output port because the HOL packet is blocked. The throughput is reduced to 58.6 percent due to HOL blocking and it has been proved.

The buffered crossbar switch is a special type of CIOQ switch, where the crosspoint has a small buffer. The output contention is entirely eliminated and the scheduling is greatly simplified because of the introduction of the crosspoint buffers. The input port sends the packets to the crosspoint buffer and it sends the packets to the corresponding output ports. So there is no possibility for two input ports to send the packets to the same output port. Since the output contention is eliminated there is no need for the input ports to cooperate with each other and so the scheduling can be conducted independently. Hence the complexity of the scheduling algorithm is highly reduced.

Volume: 03; Issue: 03; Pages:1194-1198 (2011)

The objective of the paper is to design a simple scheduling algorithm for the buffered crossbar switch to reduce the delay and time complexity. So, we present a packet scheduling algorithm called Independent Scheduling Algorithm (ISA). ISA conducts scheduling in a distributed mode and the scheduling is independent. Both the input and the output ports make scheduling decisions based on the state information of the crosspoint buffers.

To achieve 100 percent throughput the size of the crosspoint buffer should be infinite. Without speedup and with a finite size buffer there is no scheme to achieve 100 percent throughput. By introducing speedup the switch can work at the line speed and it is easy to obtain 100 percent throughput. But this technology is expensive and complex. We assume the switch without any speedup and the crosspoint buffers size to be one.

Crossbar switches have long been the preferred structures for high-speed switches and routers. They provide nonblocking capability and overcome the bandwidth limitation of bus-based switches. With the ever-increasing demand for more bandwidth and higher throughput, it has become a more and more challenging task to design high performance crossbar switches and efficient scheduling algorithms. With ISA each input port makes a decision only based on the state information of the crosspoint buffers. Hamiltonian walk is applied and the queue with highest weight is scheduled. The crossbar sends the packets to the corresponding output ports. So the scheduling at both the input and output ports are independent.

2. PREVIOUS WORK

Scheduling algorithms have an objective to achieve 100 percent throughput and to reduce the delay. So the algorithm should emulate the output queue to achieve 100 percent throughput. Also the algorithm should be simple and have less time complexity.

A buffered crossbar switch architecture known as SQUISH and SQUID were proposed in [1] They achieve 100 percent throughput with the minimum required crossbar buffer size. They have a less time complexity of O(log N), where N is the switch size. In [2] they have proved that a CIOQ switch with a speedup of just two can behave identically to an OQ switch. They can employ a wide variety of packet scheduling algorithms such as strict priority, WFQ, etc.

Lotfi Mhamdi and Mounir Hamdi proposed a scheme Most Critical Buffer First (MCBF) in [3] which requires much less hardware than the existing schemes. It is a stateless algorithm but it shows optimal stability performance. Xiao et al. proposed Shortest Crosspoint Buffer First (SCBF) in [4] that achieves 100 percent throughput for any admissible traffic. They proved that a CIOQ switch operating under SCBF with any work-conserving output arbiters can achieve 100 percent throughput for any admissible traffic. It has a time complexity of O(log N) and it is feasible for high performance switches.

In [5] the authors explored how a buffered crossbar switch can provide guaranteed performance, with practical scheduling algorithms and less time complexity. With scheduling algorithms which are distributed, and with a speedup of two, buffered crossbars provide throughput, rate and delay guarantees. In [6] a simple scheduling scheme named Modified Current Arrival First – Lowest TTL First (MCAF-LTF) was presented. They do not need any costly time stamping mechanism. With a speedup of two and a one-cell-internally buffered crossbar switch they can exactly emulate an OQ switch.

In [7] a simple scheduling algorithm LIPS was proposed which achieves 100 percent throughput. They use fixed size crossbar buffers and they can schedule variable length packets without the need of segmentation and reassembly. They schedule the packets with a higher throughput, shorter packet delay and cheaper hardware cost. In [8] the authors propose Deficit Round Robin that achieves perfect fairness in terms of throughput, and require only O (1) work to process a packet, and is simple enough to implement in hardware. Where services cannot be broken up into smaller units in case of load balancing Deficit Round Robin can be used.

3. INDEPENDENT SCHEDULING ALGORITHM

In this session we present our new ISA scheme.

3.1 Switch Design

The switch model considered in this paper is illustrated in Fig. 1, where N input ports and N output ports are connected by a crossbar switching fabric, which has no speedup. An input port has N virtual output queues to store packets destined for different output ports. Each crosspoint has an exclusive buffer size 1. The output port has a single queue. Since it has no speed up the bandwidth of the input port, output port and crossbar is R. All input ports and output ports work independently and asynchronously. When a packet arrives at the switch, it is first stored in the input queue. Then the packet is sent from the input queue to the crosspoint buffer, and then from the crosspoint buffer to the output queue and finally delivered to the output line.

Since the packets at different locations are to be scheduled, there are three types of scheduling involved in a buffered crossbar switch. which we call input port scheduling, switch scheduling and output port scheduling, respectively. In input port scheduling, an input port selects one of the virtual queues and sends the packet to the crosspoint buffer. In crossbar scheduling, an output port selects one of the crosspoint buffers and the packet is send from the crosspoint to the output queue. In output scheduling, an output port selects a output queue and sends the packet to the output line.

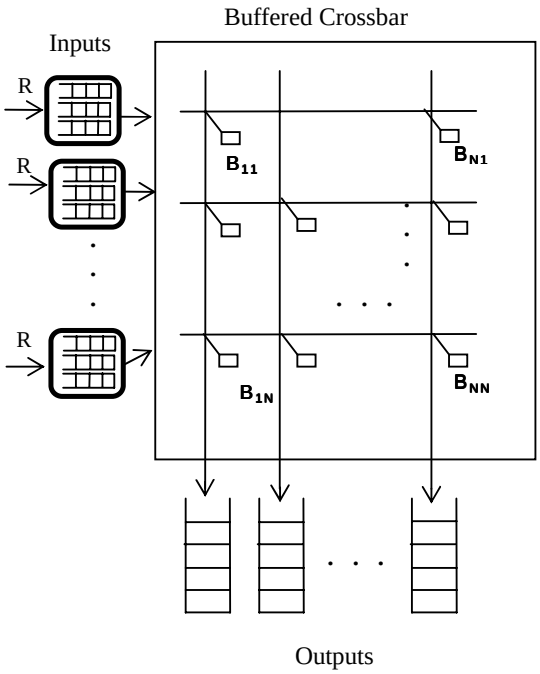

Fig.1 Structure of a crosspoint buffered switch

3.2 Algorithm

Input port scheduling and switch scheduling of ISA are conducted in an independent and distributed manner, and they only rely on the state information of the crosspoint buffers.

In input port scheduling, when the transmission channel of an input port to the crosspoint buffers is idle, the input port applies Hamiltonian Walk selects one of its VOQ's which has the highest weight and whose corresponding crosspoint buffer is empty. The Hamiltonian weight is calculated by :

$$\mathbf{H(n) = H(n\text{-}1) \bmod N! + 1} \quad (1)$$

The Hamiltonian walk is generated by Johnson–Trotter algorithm [9]. The weight W(n) is calculated by the product of Hamiltonian weight (H(n)) and the queue length of the VOQ (Q.L(n)) which is represented as :

$$\mathbf{W(n) = H(n) \times Q.L(n)} \quad (2)$$

The VOQ with the highest weight is send to the crosspoint buffer. The crosspoint buffer is selected in round robin fashion. All the input ports are scheduled independently. Switch scheduling is similar to input port scheduling. When the transmission channel of an output port from the crosspoint buffers is idle, the output port selects a crosspoint buffered packet and saves it in its output queue.

TABLE 1 - ISA

Input Port Scheduling:

```
Packets in the input port are independently scheduled as{
while true do{
        select a VOQ from the input port whose corresponding crossbar buffer is empty;
        apply Hamiltonian walk and find the queue with highest weight;
        transfer packets in the VOQ to the crossbar buffer in Round robin fashion;
        if the channel to the outpur port is idle{
            transfer the pacekt to the corresponding output queue;
        }
    }
}
```

Switch Scheduling :

```
Packets in the output port are independently scheduled as{
    while true do {
        select a crossbar buffered packet in Round robin fashion;
        transfer the packet to the corresponding output queue;
    if the channel to the output line is idle{
            transfer the pacekt to the output queue;
        }
    }
```

For easy understanding, the pseudo code description for the input port scheduling and switch scheduling of ISA is presented in Table 1. Note that, in input port scheduling, the scheduling candidates of an input port are only the VOQ's whose crosspoint buffers are empty. Similarly, in switch scheduling, an output port only needs to test whether a crosspoint buffer is occupied or empty.

3.3 Comparison with existing schemes

The ISA depends only on the state information of the local crosspoint buffers. So they work independent of each other. Hence this algorithm is purely distributed because the arbiters can be distributed at different locations. In the existing schemes like maximum weight matching the arbiters need to exchange information when making scheduling decisions.

The time complexity of ISA is less. It has a complexity of O(log N). This is because all the input ports and the output ports make the scheduling decision independently. But in the existing schemes such as maximum weight matching and maximum size matching the time complexity is $O(N^3)$ [10] and $O(N^{2.5})$ [11] , respectively.

4. Simulation Results

In this section, we compare the delays with the existing algorithms like SQUISH, SQUID [1] and LIPS [7] via simulations. The traffic patterns studied in this section are uniform Bernoulli traffic model.

4.1 Average Delay

Fig. 2. shows the delay of ISA for different switch sizes. We simulated uniform Bernoulli traffic model for different switch sizes. As the switch size increases there is an increase in the delay also. This is because as the switch size increases the delay at the input port decreases. Each output port is associated with a single crosspoint buffer. So as the switch size increases there is an increase in the delay in the output port. This causes an increase in the delay. Fig.3. shows the comparison of delay with the existing schemes for uniform Bernoulli traffic. ISA has lesser delay than the existing algorithms.

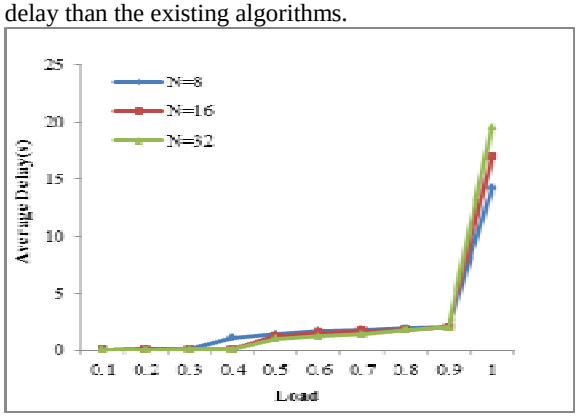

Fig. 2. Average delay of ISA for different switch sizes

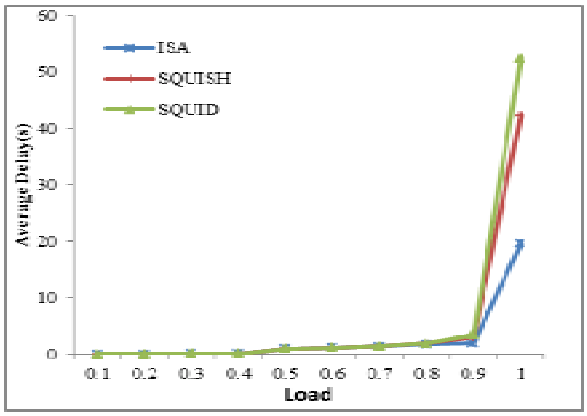

Fig. 3. Averege delay for ISA,SQUISH and SQUID for N=32

4.2 Maximum Queue Length

In order to achieve 100 percent throughput the input port and output port should have sufficient buffer space to avoid packet loss. Here the maximum queue length is the total number of bytes buffered in the virtual queues in the input port and in the output port for the entire simulation time. Fig. 4. shows the maximum queue length of different algorithms under uniform traffic model. The maximum queue length of the input port is lesser than the maximum queue length of the output port.

4.3 Throughput

The relationship between throughput and the effective load is depicted in Fig. 5. All have similar curves and achieve 100 percent throughput. This is because the input and the output ports can handle the packets without any drop. It is noted that there is no difference on the throughput performance without speedup of two.

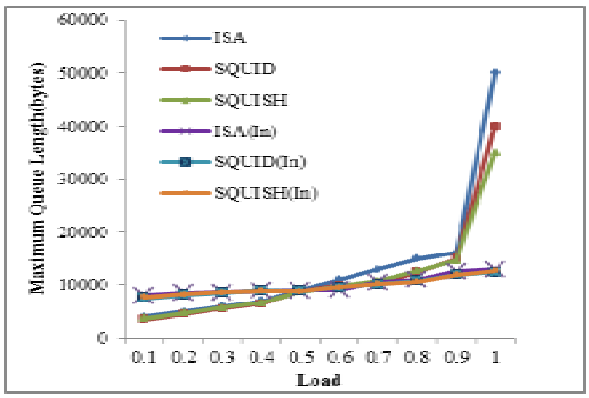

Fig . 4. Maximum queue length for different algorithms

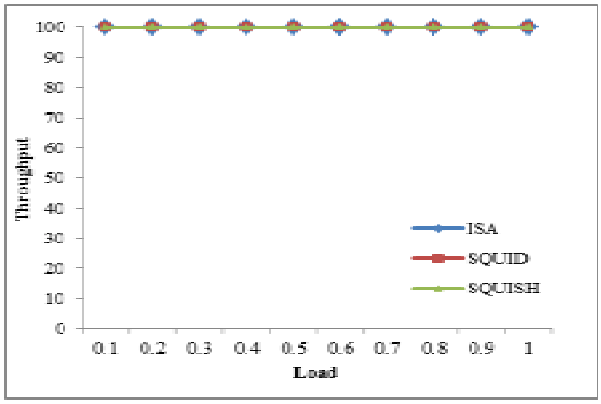

Fig. 5. Throughput of different algorithms

5. Conclusion and future work

In this paper we proposed the Independent Scheduling Algorithm (ISA) for buffered crossbar switch with the crosspoint buffer size as one and no speedup. Here both the input and the output ports work independent of each other. So the input and the output contention are greatly

reduced. Also the scheduling is based only on the state information of the crosspoint buffer. Hence the implementation is distributed. The complexity of ISA is O(log N). It can achieve 100 percent throughput at Bernoulli uniform traffic model.

Regarding future work there are several ways to extend this paper. The interaction between the input and the output ports can be loosen. Before making the scheduling decision the crosspoint buffer needs to be checked, if it is occupied or not. This introduces delay in the switch scheduling. Different techniques may be explored to remove the delay.

We have considered only fixed sized packets. We showed that the buffers in the crosspoint need to store only one packet to achieve 100 percent throughput. However it might be possible to schedule variable size packet.

ACKNOWLEDGEMENT
The authors would like to thank several people for listening patiently and providing valuable ideas. They thank Dr.Muneeswaran and Dr.K.Mala for valuable discussions.

REFERENCES

[1]. M.Katevenis, G.Passas, D.Simos, I.Papaefstathiou, N.Chrysos, "Variable Packet Size Buffered Crossbar (CICQ) Switches," *ICS-FORTH*,Sep 2003.

[2]. S-T.Chuang, A.Goel, N.McKeown, B.Prabhakar, "Matching Output Queueing with a Combined Input Output Queued Switch," *IEEE Infocom*, June 1999.

[3]. L.Mhamdi, M.Hamdi, "MCBF: A High Performance Scheduling Algorithm for Buffered Crossbar Switch," *IEEE Comm. Letters*, vol. 7,no. 9, pp.451-453, Sep. 2003.

[4]. X.Zhang, L.N.Bhuyan, "An Efficient Scheduling Algorithm for Combined Input-Crosspoint-Queued (CICQ) Switches," *IEEE Global Telecomm. Conf. (GLOBECOM '04)*, Nov. 2004.

[5]. S-T. Chuang, S.Iyer, N. McKeown, "Practical Algorithms for Performance Guarantees in Buffered Crossbars," *IEEE INFOCOM '05*, March 2005.

[6]. L.Mhamdi and M.Hamdi, "Output Queued Switch Emulation By A One-Cell-Internally Buffered Crossbar Switch," *IEEE Global Telecomm. Conf. (GLOBECOM '03)*, pp. 3688-3693, Dec. 2003.

[7]. D.Pan and Y.Yang, "Localized Independent Packet Scheduling for Buffered Crossbar Switches," *IEEE Transaction on Computers*,vol. 58, no. 2, Feb. 2009.

[8]. M.Shreedhar and G. Varghese, "Efficient Fair Queuing Using Deficit Round-Robin,"*IEEE/ACM Transactions on networking*, vol. 4, no. 3, June. 1996.

[9]. [Online] : http://en.wikipedia.org/wiki/Steinhaus-Johnson-Trotter_algorithm

[10]. R. Tarjan, "Data Structures and Network Algorithms," *Proc. CBMS-NSF Regional Conference Series in Applied Math*., Dec. 1983.

[11]. J. Hopcroft and R. Karp, "An $N^{5/2}$ Algorithm for Maximum Matching in Bipartite Graphs," *SIAM J. Computing*, vol. 2, no. 4, pp. 225-231, Dec. 1973.

Authors Biography

Dr. Kannan Balasubramanian received the Ph.D degree in Computer Sceince from UCLA, and the M.Tech degree in Computer Sceince and Engineering from IIT Bombay, India and his Msc(Tech) degree in Computer Sceince from BITS., Pilani, India. He is a Professor Mepco Schlenk Engineering College, Sivakasi, India. His research interest includes Network architecture, Protocols, Security and performance.

C.Sindhu received her M.E in Mepco Schlenk Engineering College, Sivakasi. She completed her B.E from Sun College of Engineering and Technology, Nagercoil in 2009. She is a lecturer in Dr.Sivanthi Aditanar College Of Engineering. Her research interest is to design efficient scheduling algorithms for switches.