Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 373-378

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

OQSMS: Optimal Queue Selection Based Multicast Scheduling Algorithm for Input-Queued Switches

[1]K.Navaz and [2]Dr.Kannan Balasubramanian

[1]*Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India.*
[2]*Professor, Department of CSE, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, India.*

ARTICLE INFO

Article history:
Received 23 June 2015
Accepted 25 August 2015
Available online 2 September 2015

Keywords:
Fan-out, Head-of-line, Multicast, Queue Selection.

ABSTRACT

This paper presents the design of a scheduler for an NxN input-queued multicast switch. It is well known that Head-of-Line (HOL) blocking will decrease the throughput of input-queued switch with FIFO. Therefore achieving a maximum possible throughput in input-queued multicast is a challenging task. The Existing approaches are based on fan-out splitting, request-accept-grand, request-grand, message forwarding, and prediction techniques. These approaches require more time for execution but there is no assurance for achieving maximum throughput. In this paper we propose an OQSMS algorithm which works in two stages: Queue Selection and Reservation Set. RV is the preprocessing task which stores the queue combination. Based on the queue combination optimal queue is selected. OQSMS achieves the maximum possible throughput and it outperforms the existing MDRR(Multicast Dual Round Robin). In current art of work, queues in each input port are selected by round robin method. In this paper an optimal queue is selected in each time slot and fan-out split is also made after the queue selection. By using this algorithm maximum possible throughput is achieved without HOL blocking.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: K.Navaz and Dr.Kannan Balasubramanian., OQSMS: Optimal Queue Selection Based Multicast Scheduling Algorithm for Input-Queued Switches. ***Aust. J. Basic & Appl. Sci.,*** *9(27): 373-378, 2015*

INTRODUCTION

The demand for network bandwidth is growing much faster than the increase in commercially available memory bandwidth, causing a growing interest in input-queued switches. A switch consists of three parts:1) Input queues 2) Output queues and 3) A switch fabric. The switch fabric operates under a scheduling algorithm which arbitrates among cells from different inputs destined to the same output. The integrating scheduling algorithm mainly proposed here for the input queued (IQ) crossbar-fabric-based switching architecture because of its scalability, low hardware requirements, and its intrinsic multicast capabilities. Most of these algorithms were based on input VOQ where each input maintains separate queue for each output (McKeown, 1999) unicast traffic and one FIFO queue for multicast traffic (McKeown and Prabhakar, 1996; Prabhakar and McKeown, 1997; Bianco and Scicchitano, 2009). Some existing algorithms (Gupta and Aziz, 2002) Limited number of FIFO queues are maintained at each input to reduce the HOL blocking problem. This queuing architecture is named as K-MC-VOQ and the performance of such multicast switch was analyzed in (Song and Zhu, 2004; Zhu and Song, 2010).

Although a number of different architectures and implementations have been proposed for multicast switches based on round robin, dual round robin, output messaging approach and belief propagation approach, we propose an interactive scheduling algorithm to achieve maximum throughput without HOL. Till now, it is a challenging one to achieve higher and higher throughput in the scheduling algorithm. In this paper we propose an optimal queue selection based multicast scheduling algorithm for input-queued switches. The rest of the paper is organized as follows. In Section II, related works on designing multicast scheduling algorithms are reviewed. In Section III, Assumed switch architecture and proposed algorithm design have been summarized. In Section IV, Performance evaluation and result analysis have been presented. Finally, we conclude the paper in Section V.

Related Works:

Most existing multicast switches (Prabhakar and McKeown, 1997; Zhu and Song, 2006) require multicast switch fabric and a sophisticated central

Corresponding Author: K. Navaz, Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India
E-mail: navazit@gmail.com

scheduler for maximizing switch performance. TATRA (Prabhakar and McKeown, 1997) is a single FIFO queue based multicast algorithm, where each input port has a single shared (physical) queue for both unicast and multicast traffic. Nevertheless, TATRA suffers from the severe HOL blocking due to its single queue nature. To reduce HOL blocking, multiple dedicated multicast queues are used in (Zhu and Song, 2006) and (Gupta and Aziz, 2002). In (Zhu and Song, 2006) each input port maintains a set of multicast queues.

To reduce the HOL blocking further, a multicast packet split scheme is proposed in (Gupta and Aziz, 2002). In (Gupta and Aziz, 2002) a set of output ports is divided into m non-overlapped subsets, and each input port maintains m unicast/multicast shared queues and each them is dedicated to a subset of outputs. In (Yongbo Jiang, Zhiliang Qiu, Ya Gao, and Jun Li, 2012), Dual Round Robin Based Multicast Scheduling Algorithm called MDRR is proposed to achieve maximum throughput with low-matching overhead. Here Input Schedulers are distributed at each input, and a global pointer g is collectively maintained by all the output schedulers. Each input scheduler maintains two priority pointers which guarantee high throughput: a primary pointer and a secondary pointer. MDRR needs message transfer between input and output ports. It does not guarantee minimum delay compared with MaxService(Gupta and Aziz, 2002). When number of queues and fan-out size (ef) increase, MDRR could not yield maximum throughput than MaxService scheme does.

This paper presents the design of a scheduler for an input-queued multicast switch. It is assumed that: 1) Each input maintains a multiple queues for arriving multicast cells and 2) Only the cell at the head of line (HOL) can be observed and scheduled at one time slot. The scheduler is required to be: 1) Work conserving and 2) Fair. The aim of our work is to find a work-conserving, fair policy that delivers maximum throughput and minimizes input queue latency, and yet it is simple to implement in hardware. We present an optimal scheduling policy, called OQSMS, which performs extremely well and it is strict in fairness when compared to a MDRR algorithm.

System Design:

In this section we detail the switch architecture, algorithm design and its components. The proposed scheduling algorithm is targeted at synchronous input-queued (IQ) switches.

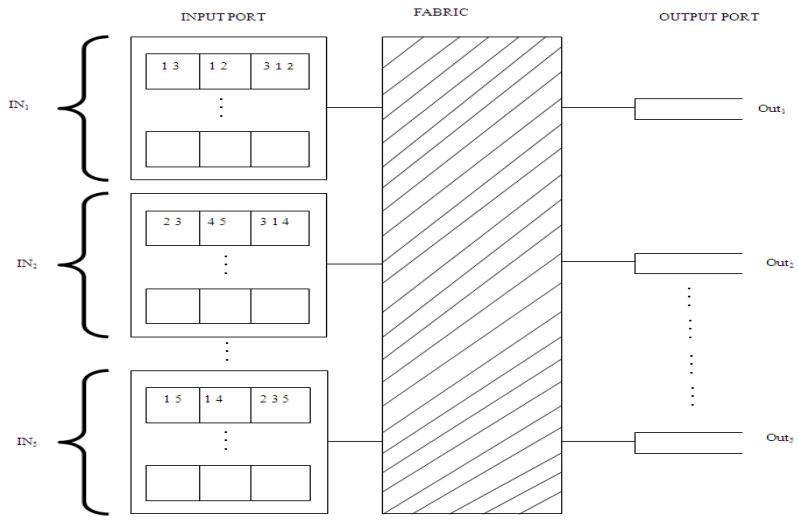

Fig. 1: 5x5 Input-Queued Switch Architecture

Here the fixed-size packet transmitted by the switch fabric is called cell. We consider only the fan-out splitting discipline where the cells may be delivered to outputs over several cell times. Any multicast cell is characterized by its fan-out set, i.e., by the set of outputs to which the cell is directed. We define the fan-out size f as the number of destinations of a multicast cell. Let us assume as NxN switch having N input ports and N output ports, and the fabric is connecting input ports and output ports at any time slot. Let us assume each input port having Q number of Non empty queues. Q_{ij} is the j^{th} queue in the i^{th} input port. Each queue contains the multicast cells with fan-out sets. The Proposed algorithm contains the following parts using the multiple cost value. Fig.1. shows an example of 5x5 switch architecture where each input port maintains five queues. For throughput Range (Rn) calculation we construct table. This table is constructed from figure 2.

Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 373-378

Table 1: Input Ports with HOL cells

Input port 1	Input Port 2	Input Port 3	Input Port 4	Input Port 5
3 1 2	3 1 4	1 3	1 5	1 3
1 2	4 5	1 5	1 3	2 3 5
1 3	2 3	1 2 5	4 5	1 4
1 3	4 5	2 3	2 3	1 5
4 5	1 5	3 4	1 2 4	3 4

3.1 Temporary Reservation Set At Each Time Slot.:

It is an iterative approach, in which each time the RV_T is estimated. It declares the queue selection and achievable throughput temporarily.

3.2 Queue Selection at each Input Port:

For each input port an optimal queue is selected such a way that the overall queue selection should result maximum throughput.

Table 2: Input Ports with Optimized Queue

Input Port	Selected Queue
1	1
2	1
3	3
4	5
5	2

Table 3: Throughput Range Calculation

Fan-out	Fan-out Split	Rn
3 1 2	3 1 2	3
3 1 4	4	1
1 2 5	5	1
1 2 4	———	0
2 3 5	———	0
Total		5

3.3 Fan-Out Split In Reservation Set:

When an queue is selected temporarily or permanently, the fan-out sets of the cells in the queue should be splitted in order to avoid HOL.

3.4 Searching maximum throughput reservation set as final reservation set (Optimized set):

At each temporary reservation set, the temporary throughput will be estimated and compared with the previous estimation. This iteration ends when a maximum throughput range is possible.

3.5 Grant Input Ports With Selected Queue and Fan-Out Split:

The end of temporary reservation set is the final reservation set to be granted in order to send traffic.

3.6 Reservation Set:

If the reservation set RV is the array set of queue names selected to each input port. Consider the table 2.

For all the input ports we can have multiple permutations with various queues. Table 3 shows a combination of queue selection and its corresponding fan-out sets.

3.7 Queue Selection:

Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 373-378

In order to achieve the maximum throughput we have to select a queue in each input ports such that the final queue set loads to a maximum throughput. For each time slot RV will be calculated according to final RV cells are transmitted from input ports to corresponding output ports.

3.8 OQSMS Algorithm:

Queue Selection
Input **:** Time slot (t_n), Input ports In_i,
Algorithm
Rn → 0
RV Ø
Do Until Combination
RV_T (Fan-out) $_{i,j}$ = Ø
RV_T (Queue) $_{i,j}$ = 0
For each input port
q → queue number
Estimate R_n
If $R_n' > R_n$
Estimate New Fan-out
RV (Fan-out) = New Fan-out
RV (Queue) = q
$R_n = R_n'$
End If
End For
End Do

4 Performance Evaluation and Result Analysis:

The result shows that the proposed work achieves high throughput than existing work. The delay-throughput performance of MDRR and OQSMS schemes have been well demonstrated under two traffic conditions. We first apply the Bernoulli traffic to 8x8 switch. When multicast packet arrives its fan-out set consists of f output randomly chosen from all output ports, Where f is the fan-out size randomly selected between [2,8]. Fig. 2 shows the simulation results when Bernoulli traffic is applied.

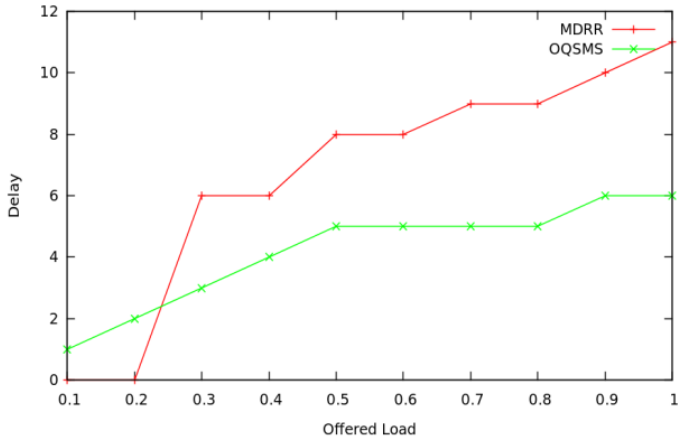

Fig. 2: Offered Load Vs Delay (Bernoulli)

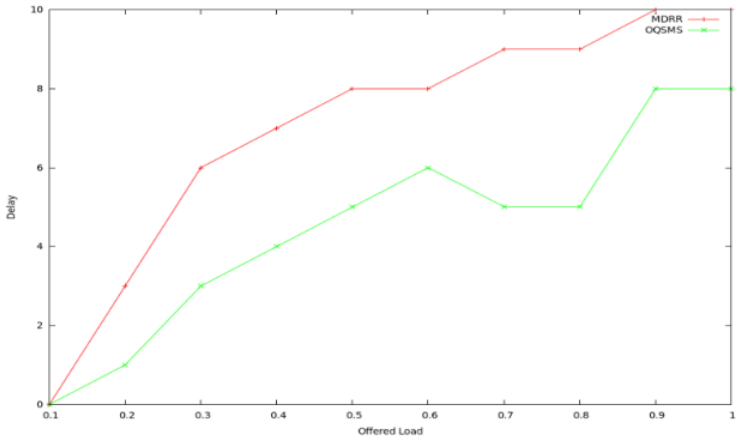

Fig. 3: Offered Load Vs Delay (Bursty)

Packet of the same burst has the same fan-out set. Fig. 3 shows the simulation results when bursty traffic is applied. Fig. 2 and 3 illustrate the average multicast delays as a function of offered load for MDRR and OQSMS under uniform Bernoulli and bursty traffic for an 8x8 switch. It's clearly show that the average delay of OQSMS is always smaller than that of MDRR for both traffic scenarios. Now fix the switch size as stable of 16x16, and we have taken the throughput of simulation using the various queue sizes. We examine the average queue size be 2, 4, 6, 8, 14 and 20. From Fig. 4 we observe that, Throughput increases only for a particular range. There will be no change when queue size increase for higher values. Fan-out sets are another factor affecting the performance of the switch, if fan-out size increases, delay also increases and if fan-out decreases throughput also decreases. From this point of view we examined a switch of 16x6 with queue size 4. We can observe the throughput improvement for the OQSMS when the mean fan-out size increases. It is shown in Fig 5.

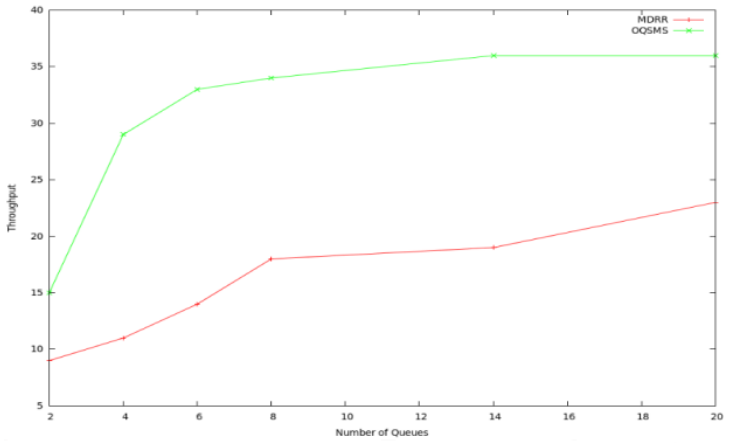

Fig. 4: Number of Queues Vs Throughput

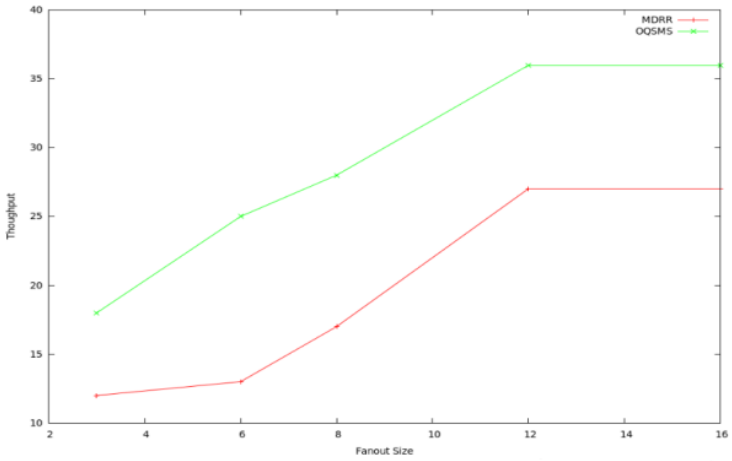

Fig. 5: Fan-out Size Vs Throughput

Conclusion:

We have applied the maximum throughput calculation to achieve the best throughput using reservation sets. This scheduling is easy to implement in the hardware. The increase in demand for network bandwidth creates a need for high-speed input-queued multicast switches. To help the designer of such switches, we have designed new algorithm. We imposed a Reservation sets on our scheduling policies, multicast and the queue selection models which is used to study the general scheduling problem. This led to the development of a fair and efficient scheduling policy.

REFERENCES

Bianco, A. and A. Scicchitano, 2009. "Multicast support in multi-chip centralized schedulers in input queued switches," Computer Networks, 53(7): 1040-1049.

Gupta, S. and A. Aziz, 2002. "Multicast scheduling for switches with multiple input-queues," in Proc. Symposium on High Performance Interconnects, pp: 28-33.

McKeown, N., 1999. "The iSLIP scheduling algorithm for input-queued switches," IEEE/ACM Trans. Netw., 7(2): 188-201.

McKeown, N. and B. Prabhakar, 1996. "Scheduling multicast cells in an input queued switch," in Proc. IEEE INFOCOM, 1: 271-278.

Prabhakar, B. and N.M cKeown, 1997. and R. Ahuja, "Multicast scheduling for input-queued switches," IEEE J. Sel. Areas Commun., 15(5): 855-866.

Song, M. and W. Zhu, 2004. "Throughput analysis for multicast switches with multiple input queues," IEEE Commun. Lett., 8(7): 479-481.

Yongbo Jiang, Zhiliang Qiu, Ya Gao, and Jun Li, 2012. "Multicast Support in Input Queued Switches with Low Matching Overhead" in IEEE Communications Letters, 16: 12.

Zhu, W. and M.S ong, 2010. "Performance analysis of large multicast packet switches with multiple input queues and gathered traffic," Computer Commun., 33(7): 803-815.

Zhu, W. and M. Song, 2006. "Integration of unicast and multicast scheduling in input-queued packet switches," Computer Networks, 50: 667-687.