International Journal on Recent Researches In Science, Engineering & Technology
(Division of Computer Science and Engineering)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy.
It is an absolutely free (No processing charges, No publishing charges etc) Journal Indexed in JIR, DIIF and SJIF.
Research Paper
Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105

Volume 4, Issue 11, November 2016

JIR IF : 2.54
DIIF IF :1.46
SJIF IF: 1.329

COMPARATIVE ANALYSIS OF MDDR, E-MDDR AND OQSMS ALGORITHMS FOR IQ SWITCHES

NAVAZ K
Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, 627 012, Tamilnadu, India
Email: navazit@gmail.com

Dr. KANNAN BALASUBRAMANIAN
Professor, Department of CSE, Mepco Schlenk Engineering College, Sivakasi, 626 005, Tamilnadu, India
Email: kannanbala@mepcoeng.ac.in

Abstract:- The emerging applications of multicasting are becoming more and more on the Internet and it is necessary to design high-speed switches/routers. The research was done in the field of fabric design of multicast switch and algorithms. The Scheduler is the influence factor that affects the throughput and delay of the switch. Scheduling algorithms that do not concentrate on residues are not well done under non-uniform traffic conditions. In this work, performance of the switch has been analyzed by applying the pointer-less multicast scheduling algorithm OQSMS (Optimal Queue Selection Based Multicast Scheduling Algorithm), E-MDDR(Enhanced Multicast Due Date Round-Robin Scheduling Algorithm) and MDDR(Multicast Due Date Round-Robin Scheduling Algorithm). The results show that OQSMS achieves better switching performance than other algorithms under the uniform, non-uniform and bursty traffic conditions and it estimates optimal queue in each time slot based on throughput range (Rn) estimation so that it achieves maximum possible throughput.

***Index Terms*— Multicast, Switch, Delay, Throughput, Scheduling.**

1 Introduction

The Internet has seen tremendous growth in the recent years. Traditional data types such as image and text no longer dominate the World Wide Web (WWW). Compared to the past, emerging applications transmitting video and audio present much higher bandwidth and more stringent latency requirements. Moreover, large-scale applications such as content delivery and interactive worlds pose serious scalability problems for the Internet.

Today's Internet consists millions of networks with million of customers revolving around it. As a result, Internet traffic is growing by a factor of 30% per year [18] [19]. To improve performance of the Internet, the de facto Internet Protocol (IP) Unicast or client-server model has been augmented with Multicast, providing one-to-many and many-to-many delivery mechanisms. IP Multicast increases bandwidth efficiency, because packets destined to multiple receivers travel only once on common parts of the network. It also reduces server load, as a sender transmits packets only once for any number of receivers.

In ancient switches, the input output ports communicated using a single shared bus. Consequently, this bus was a limitation, as not more than one pair of ports can communicate at a time. The classical crossbar switch overcame the bottleneck imposed by this shared bus architecture that restricted the use of N input-output port pairs in parallel. The crossbar switch is an NxN matrix of 2N buses, connecting input output ports. The switches and routers basically store, route and forward these packets before they reach the

destination. One core functionality of such switches (a layer 2 switch, or a layer 3 IP router) is to transfer the packets from the input port to one of the output ports. This functionality, called switching, though appears simple, is such a challenging problem to solve at line rates that, there is a wealth of literature on this topic.

Packet switch architectures can be divided into three major categories [10]: the shared memory packet switch, the shared medium packet switch and the space division packet switch. Theoretically, each of these three architecture types can be modified to support multicast. However, in shared memory and shared medium architectures, there is a scalability problem as the need for a high-speed memory or a bus greatly limits their use when the switch size grows larger.

A crossbar switch is a switch connecting multiple inputs to multiple outputs in a matrix manner. The crossbar constraints of an IQ switch require it to schedule packets to be transferred between inputs and outputs. The throughput and delay in IQ switch are heavily dependent on this scheduling decision. In past there has been a lot of research done to design multicast scheduling algorithms for IQ switches. In addition, people are more interested in sharing knowledge and information for various purposes. Innovations in information sharing are continuously accelerated to cater user needs in such environments. These motives have encouraged the construction of sophisticated environments for effective communication to deliver information. It is important to note that the applications for IP Multicast are not solely limited to the Internet. Multicast IP can also play an important role in large distributed commercial networks. The demand for network bandwidth is very essential and many of the networking applications require high speed switching for multicast traffic at the switch/router level to preserve network bandwidth. It causes an incrementing interest in the input-queued switches. A switch consists of three components: 1) input queues for cells arriving at the input links 2) output queues for cells exit on output links 3) A switch fabric for transferring cells from the inputs to the desired outputs. LAN and Asynchronous Transfer Mode (ATM) switches are considered as a high performing internetworking protocol and uses a crossbar switch based on switched backplane. Further, these systems use the input queues for holding packets which are waiting to traverse through the switch fabric. Thus, it is known that the first in first out (FIFO) input queues can be used to maintain packets. A scheduling algorithm is utilized to configure the crossbar switch to decide the order in which packets will be accommodated.

Many integrated scheduling algorithms have been proposed earlier. They have been mainly proposed for input queued crossbar switching architecture but multicast scheduling, mainly concerns how to transmit as many cells as possible from input to output. In the unicast traffic, Head-of-Line (HOL) blocking quandary occurs that is induced by first-in-first-out (FIFO) queue which gets avoided by utilizing virtual output queuing (VOQ) technique. Here, in this type of technique every single input maintains a separate queue for each output [7].

Numerous unicast scheduling algorithms have been proposed so far. iSLIP is the fast and efficient algorithm which has achieved 100% throughput in a single iteration for uniform traffic. In[10], MRR (Modified Round Robin Algorithm) proposes that it can show a performance equivalent to iSLIP, yet require less number of processing steps. Using multicast traffic [2], [5], we can avoid HOL blocking by utilizing 2N -1 queues for each input port in N×N switch. This type of queue architecture is called Multicast Virtual Output Queuing (MC-VOQ). However, in the medium/sizably voluminous switches, because of its low scalability, it is virtually not tackled. One such practical queuing scheme utilized for multicast switches is to assign a single FIFO queue at each input for all multicast traffic, however, the HOL blocking quandary limits the throughput. Whereas the other algorithms [1],[3] considered a circumscribed number of FIFO queues is maintained at each input to reduce the HOL blocking problem. Thus queuing architecture is denominated as k-MC-VOQ and performance of these multicast switches are analyzed theoretically [11], [14]. As the link speed grows dramatically, high speed switches will have less time to perform scheduling process. As a result, iterative schemes and high matching overhead would cause delay, matching overhead scales up very expeditiously, the link speed and the switch size increases, the requirement for simple and high performance switches becomes very essential.

In this paper, we analyze the average cell delay and throughput of the OQSMS, E-MDDR and MDDR algorithms for IQ switch under uniform and non-uniform traffic patterns. The rest of the paper is organized as follows. In Section 2, related works on designing multicast scheduling algorithms are reviewed. In Section 3, MDDR, E-MDDR and OQSMS algorithms have been reviewed. In Section 4, Performance evaluation and result analysis have been presented. Finally, we conclude the paper in Section 5.

2 Related Work

Most of the existing multicast switches [8], [9] require in-switch packet replication, and a sophisticated central scheduler to maximize performance of the switch. TATRA [9] is a multicast algorithm on single FIFO queue, where each input port has a single common queue for both unicast and multicast traffic. The central scheduler maintains the N virtual queues and each is destined for one output port. In each time slot, the head-of-line (HOL) packet of each input queue is scheduled to join different virtual queues according to its destination output ports. Fan-out splitting [4], which sanctions a multicast packet to be sent to a subset of its outputs, is adopted to increment the throughput of the switch. However, TATRA suffers from serious HOL blocking because of its single queue nature. TATRA avoids starvation but is additionally perplexed to implement a hardware due to heavy computations.

To minimize the HOL blocking, multiple dedicated multicast queues have been utilized in [3] and [13]. In [13], each input port has a set of multicast queues. When a multicast packet arrives, it selects one of the multicast queues to join according to its load balancing policy. In each time slot, the scheduling priority is given to either a unicast packet or a multicast packet. According to the accommodation ratio of the two types of traffic. An iterative scheduling algorithm is adopted to maximize the switch throughput.

ESLIP [6] adopts the VOQ structure to buffer unicast packets and puts all the multicast packets in a special single queue at each input port. It utilizes a variant of the iSLIP algorithm to schedule mixed unicast and multicast traffic. As can be expected, ESLIP eliminates the HOL blocking for unicast traffic, but not for multicast traffic. In an extreme situation, where all the incoming packets are multicast packets, ESLIP cannot benefit from the VOQ structure and it authentically works on the single input queued switch.

Multicast packet split scheme is proposed in [3] for further reducing the HOL blocking problem. In [3], the set of output ports is divided into m non–overlapped subsets, and each input port maintains m unicast/multicast shared queues and each is dedicated to a subset of outputs. When a multicast packet arrives, if its fan-out set wholly fit in a queue, it will join the queue, otherwise, the multicast packet is divided into smaller ones (each with a modified fan-out set) to join multiple queues. Again, an iterative scheduler is adopted to maximize throughput.

FIFOMS [8] is an efficient multicast scheduling algorithm which is proposed to avoid HOL blocking. The basic idea is to discretely store unicast/multicast packets and memory addresses. FIFOMS uses common unicast VOQ's as pointer queues. More concretely, when a multicast packet with a fan-out of f (f = 1 for unicast packet) arrives, it is time stamped and stored in shared memory, and its memory address / pointer joins f different VOQ queues according to the fan-out set. In each time slot, the scheduling priority is given to pointers which are the unicast copies of a multicast packet with the most diminutive timestamp. Indeed, In Scheduling all multicast packets are "converted" into a unicast. At this end, the HOL blocking is completely eliminated. But in order to maximize switch throughput, in-switch packet replication is still utilized for sending multiple replicas of a multicast packet in the same slot. This is achieved by an iterative scheduling algorithm, which incurs considerable amount of communication overhead.

In [12], Multicast Dual Round Robin Scheduling Algorithm called MDRR, it is proposed to achieve maximum throughput with low-matching overhead. Here input schedulers are distributed to each input, and a global pointer 'g' is collectively maintained by all output schedulers. Each input scheduler has two priority pointers that guarantee high throughput: a primary pointer and a secondary pointer. MDRR needs more message transfer between the input and output ports in the request phase. It does not ensure a minimum delay compared with MaxService [3]. When the number of queues and the fan-out size (ef) increase MDRR could not obtain a maximum throughput than MaxService scheme done. Dual pointer utilization in the input ports are overhead here which takes longer execution time.

Multicast Due Date Round Robin(MDDR) Scheduling algorithm achieved well under uniform traffic but not maximum under non-uniform traffic pattern. Still it has throughput degradation that residue of some of the cells are waiting for more than N time slot where N is the number of output ports or the maximum fan-out size . E-MDDR[17] scheduling algorithm has been proposed to overcome the waiting cells in the VOQ by announcing it as the emergency and N+1 time slot is allotted for these cells to transfer completely. It mainly minimizes the average cell delay and achieves the maximum throughput.

We compare the scheduling scheme called OQSMS with E-MDDR and MDDR. OQSMS achieves better switching performance than other algorithms under the admissible traffic conditions because OQSMS will estimate optimal queue selection based on more

queue combinations so that it achieves maximum possible throughput. MDDR primarily minimizes the request overhead at the output ports and eliminates the dual pointer utilization in input ports. Also E-MDDR concentrate on the residue of the cells waiting in the VOQ's for more than N cell times. No identical fan-out of the cells are consecutively scheduled. It shows that E-MDDR more preponderant than the MDDR algorithm.

3 Switch Architecture and Multicast Algorithms

In this section, we give the packet switch architecture, MDDR, E-MDDR and OQSMS multicast scheduling algorithms for input queued crossbar switch in detail.

A. Packet Switch Architecture

The scheduling algorithms are made for synchronous input-queued (IQ) switches. The fixed-size packet which is transmitted by the switch fabric is called cell. But only the fan-out splitting discipline is considered because the cells may deliver output over several cell times. Any multicast cell is characterized by its fan-out set, i.e., by using 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. The NxN switch architecture is shown in Fig. 1. Let us assume NxN switch having N input ports and N output ports, and the fabric is connecting input ports and output ports for any time slot. A small number k of FIFO queues dedicated to multicast traffic is maintained at each input port. Q_{ij} is the j^{th} queue in the i^{th} input port. Arriving multicast cells are partitioned into the k queues according to the fan-out size. Each queue contains the multicast cells with fan-out sets. A scheduling algorithm does the arbitration between the N input ports and N output ports, obtained by solving the bipartite graph-matching problem. This matching is a collection of edges, from the set of non-empty input queues to the set of output ports. Such that each input is connected to at most N outputs and each output is connected to at most one input. In each time slot, Input 'i' is connected with set of output destinations. If the fan-out of the cell is completely served, a cell is removed from the corresponding queue to output destinations by properly configuring the non-blocking multicast switch fabric otherwise a cell is retained until all its destinations are served.

B. Multicast Due Date Round-Robin (MDDR)

In [16],MDDR multicast scheduling algorithm has been proposed. Input schedulers are distributed at each input and a global pointer g is collectively maintained by all the output schedulers. Each input maintains a Due Date to be sent.This due date is generated based on the priority of cells contained in the fan-out. The highest fan-out size port gets the first priority and next fan-out size has the second priority and so on. By keeping this order, the throughput will be increased. This algorithm works in the following phases.

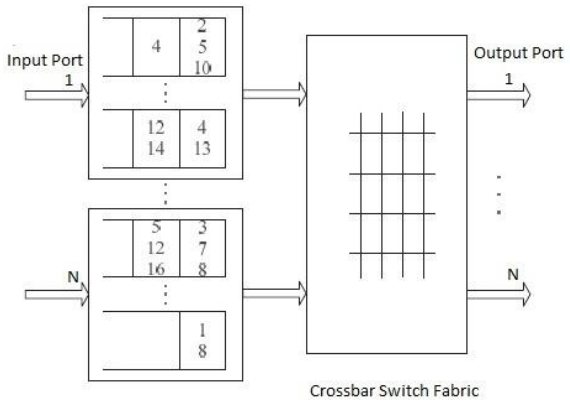

Fig. 1NxN Input Queued Switch Architecture

***Request*:** The input sends request to all the destined output ports corresponding to the first nonempty queue. At request phase, fan-out size of the current non-empty queue is measured in each input port and prioritize the input ports based on fan-out size. Next step is to assign the due dates to the cells within the fan- out. This Due dates are assigned in a priority input port which will assign the first Due Date (Due Date = 1) to the cells. On the second priority port, elements already presented in first priority are assigned to second Due Date (Due Date = 2) and remaining cells are assigned to the first Due Date (Due Date = 1) and so on. On the completion of these Due Dates, the requests will be made to output ports.

Grant: In the Grant phase, if more than one request is made for the same port, the global pointer pointing one is granted and the others are rejected then the global pointer is incremented to the next position.

C. Enhanced Multicast Due Date Round-Robin (E-MDDR)

In [9], Multicast Due Date Round Robin (MDDR) scheduling algorithm has been proposed which achieved well under both uniform and non uniform traffic pattern. Still it has throughput degradation that the residue of some of the cells are waiting for more than N time slot, where N is the number of output ports or the maximum fan-out size. In this paper, the authors have proposed the E-MDDR scheduling algorithm which overcome the waiting cells in the VOQ by announcing it as the emergency and N+1 time slot is allotted for these

cells to transfer completely. It mainly minimizes the average cell delay and achieves the maximum throughput.

In this section, the authors have proposed the Enhanced Multicast Due Date Round-Robin Scheduling Algorithm (E-MDDR). It works under the request and grant phases. This algorithm concentrate on the residue of the cells waiting for more than N cell times.

E-MDDR Algorithm Steps

For any non-uniform traffic,
I. Input ports are prioritized based on the cells which is having the highest fan-out size.
ii. Priority order of the input ports are iterated one-by-one.
iii. Every input port selects the queue based on the highest fan-out.

- If two or more queues in the input port have the same fan-out size, the first non-empty queue is selected.
- Fan-out splitting scheme is applied on HOL cells of every selected queue.

iv. Due dates are assigned to the fan-out cells in the current queue of the current input port.
v. For each due date the unassigned cells set due date 1 to priori non-exist cell, and 2 to priori exist cell.
vi. Make request for the 1 due date cells to the output ports.
vii. If output port receives more than one request, choose one from the global pointer points and grant it.

- No identical fan-out of the cells are consecutively scheduled. It will be postponed to the next time slot. So that the output port utilization is improved.
- If the residue of the cells are waiting in the queue for more than N time slot, then that cells are declared as the emergency cells.
- Every N+1 time slot are allotted for arbitrating the emergency cells.

viii If output port receives only one request, it directly grants it. Repeat the iteration with the non-granted cell requests and remaining fan-outs further.

D.Optimal Queue Selection Based Multicast Scheduling Algorithm (OQSMS)

OQSMS[15]is an iterative based algorithm which works in two stages: Queue Selection and Reservation Set. RV is the pre-processing 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 E-MDDR and MDDR scheduling algorithms. In current art of work, queues in each input port are selected by round robin method. In OQSMS an optimal queue is selected in each time slot and fan-out splitting is also made after the queue selection. By using this algorithm maximum possible throughput is achieved without HOL blocking. In this section we detail the OQSMS algorithm steps.

OQSMS Algorithm Steps

For any uniform or non-uniform traffic,

i. Every input port selects the optimal queue based on the throughput range calculation (Rn)
ii. RV is the array set in which queue names of all the input ports are stored.
iii. RV_T is the temporary array set which stores all the queue combination in each time slot.
iv. Every queue combinations are iterated in RV_T to calculate the maximum possible throughput.
v. For all the queue combinations throughput range Rn is calculated.
   - Fan-out splitting scheme is applied to the HOL of the selected queues.
vi. Throughput range is estimated for the queue set in RV_T and which is compared with the previous estimation. This iteration ends until maximum throughput range Rn is possible.
   - If the current estimation value is smaller than previous estimation next port possible fan-out set is taken for Rn calculation.
   - Otherwise when the Rn is equal to the total number of output ports.
vii. Based on the selected fan-out through Rn calculation, the input port transferred the cells to the output ports directly.
viii. The residue of the fan-out cells are participated in the next time slot. The iteration is repeated until transferred all cells.

4 Performance Evaluation and Result Analysis

To study the performance of OSQMS, E-MDDR and MDDR the simulation has been conducted in NS2 that models the input queued crossbar switch of size N×N. In general, most of the experiments, we used an 8 × 8 and 16 × 16 VOQ switch. In some

experiments the value of N is specific to the experiments. The VOQ's are supplied with uniform and non-uniform traffic patterns.

This performance analysis concentrates on two Performance metrics which are Average Cell Delay and Throughput. The graphs drawn in Figs. 2-9 show that the overall performance of Delay and Throughput comparison of MDDR, E-MDDR and OQSMS algorithms.

Delay: A multicast cell is stored in the queue until all the destinations in its fan-out set are reached. The multicast delay of a cell is calculated as the cell times that the cell stays in the queue until it is removed. Delay increases when they become unstable as the offered load increases.

Throughput: Throughput is the another performance measurement used in this investigation which is defined as the ratio between the total number of cells forwarded to output interfaces, and the total number of cells arrived at input interfaces. It is essentially a measure of the cell loss probability at input queues.

Fig. 2 shows the average cell delay against the offered load for MDDR, E-MDDR and OQSMS under uniform traffic. We can observe that as output load increases, OQSMS is very effective in reducing and constantly maintaining the average delay, and performs reasonably well. It is shown that the average delay of OQSMS is always smaller than that of MDDR and E-MDDR, especially under heavy load. The reason is that MDDR does not concentrate on residue of cells which are waiting for more than M time slots.

Fig. 3 shows the simulation result that the delay occurred according to the load offered by non-uniform traffic. E-MDDR and OQSMS have no higher level significance of difference in delay. MDDR could not achieve the minimum delay comparing with E-MDDR and OQSMS because the identical fan-out of the cells are consecutively scheduled.

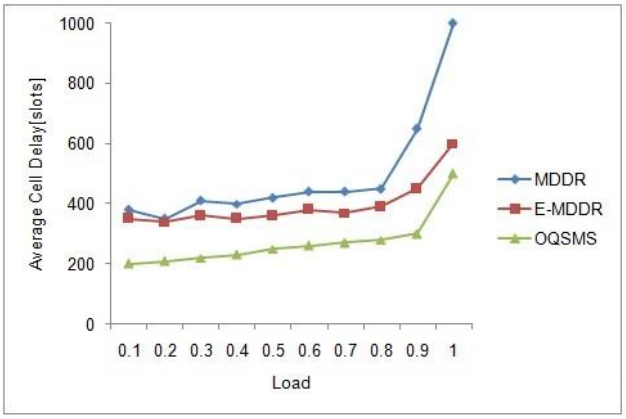

Fig. 2 Average Cell Delay as a function with respect to Load for uniform traffic

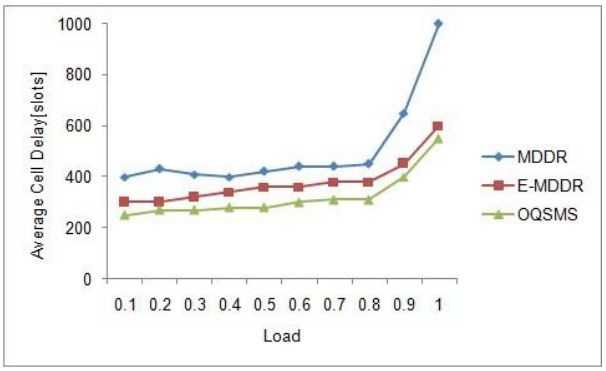

Fig. 3 Average Cell Delay as a function with respect to Load for non-uniform traffic

Fig. 4 illustrates the improvement of maximum achievable throughput performance introduced by increasing the number of queues under uniform traffic. It shows that OQSMS achieves nearly above 95% throughput than MDDR and E-MDDR for the number of queues above 5. We can observe that when number of queues are increased OQSMS obtains a high switching performance while pointer based algorithms obtain low throughput when number of queues are high.We can conclude that small number of queues are enough to maintain the constant level throughput for pointer based algorithms.

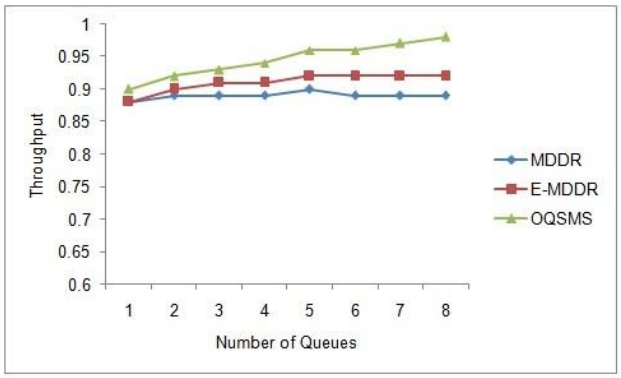

Fig. 4Number of Queues Vs Throughput under uniform traffic

Fig. 5 shows the throughput analysis of OQSMS, MDDR and E-MDDR under uniform traffic pattern with respect to load. Above 95% throughput has been achieved by OQSMS. It's shown that when the load is above 60% the throughput of MDDR is decreased.

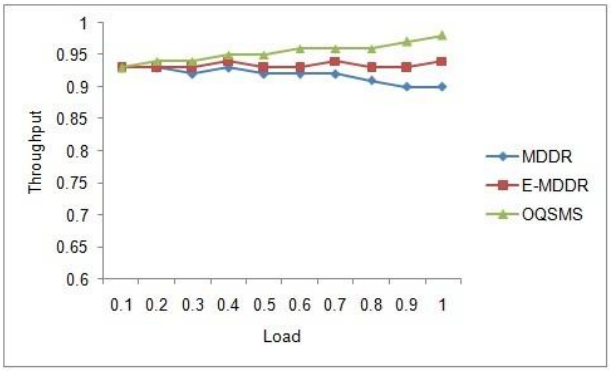

Fig. 5 Throughput as function with offered load for uniform traffic

Fig. 6 illustrates the throughput as a function of multicast fraction and compare the maximum achievable throughput of OQSMS with MDDR and E-MDDR under non-uniform burst traffic pattern. It's shown that irrespective of the arrival traffic pattern OQSMS always achieves higher throughput performance than MDDR and MDRR since when the load increases OQSMS has more queue combinations so that it achieves more throughput.

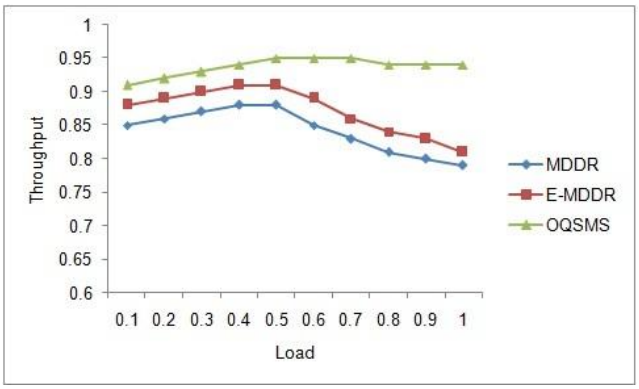

Fig. 6 Throughput as function with offered load for non-uniform traffic

Fan-out size points to the number of outputs that are destined. This fan-out size is also a performance attribute at output port load. Fig. 7 shows the maximum achievable throughput as a function of mean fan-out size under non-uniform traffic for a 20x20 switch. We can observe the throughput improvement of OQSMS when fan-out size is above 8. Since more transmission possibilities could be provided to the output ports and also when fan-out increases iteration steps to calculate throughput range in OQSMS are effectively reduced which yields good switching performance.

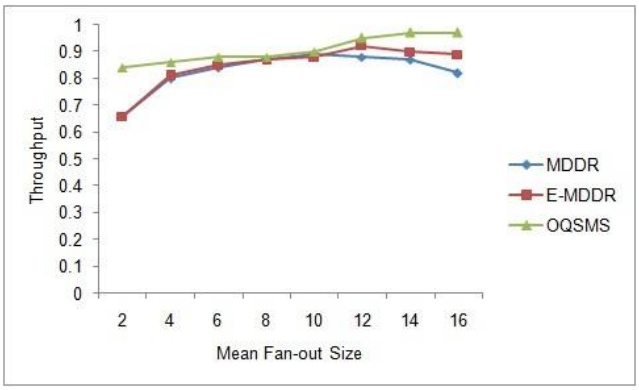

Fig. 7 Throughput as function of mean fan-out size for non-uniform traffic

It is shown that the MDDR and E-MDDR obtain minimum throughput when fan-out increase. In MDDR when fan-out increases, most of the due date assignment value 1 is postponed to the next time slot which is slightly reduces the throughput performance. But in E-MDDR N^{th} time slot is allotted for arbitrate the emergency cells until completely scheduled. This is reason E-MDDR has slight improved throughput comparing with MDDR.

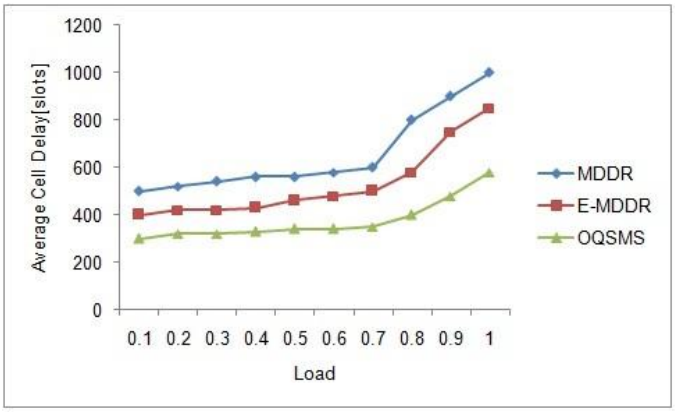

Fig. 8 Average Cell Delay as a function with respect to Load for non-uniform bursty traffic

Fig. 8 shows the simulation result that the delay occurred according to the load offered by non-uniform burst traffic. OQSMS achieved minimum delay comparing with E-MDDR and MDDR. MDDR takes more time slot to arbitrate the cells than E-MDDR. It's clearly shown that MDDR could not handle the burst traffic efficiently.

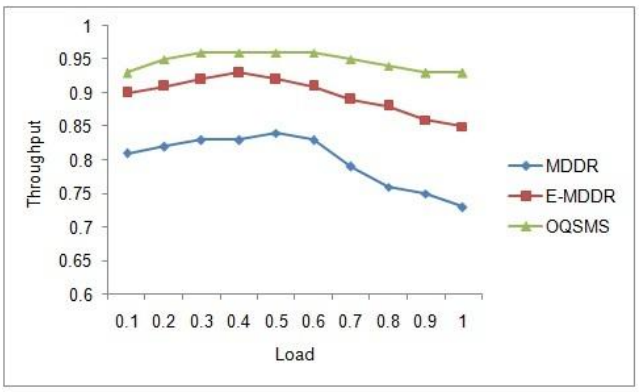

Fig. 9 Throughput as function with offered load for non-uniform bursty traffic

Fig. 9 illustrates the throughput as a function of multicast fraction and compare the maximum achievable throughput of OQSMS with MDDR and E-MDDR under non-uniform burst traffic pattern. It's shown that irrespective of the arrival traffic pattern OQSMS always achieves higher throughput performance than E-MDDR and MDDR. When the load is above 50% MDDR throughput is decreased. We can conclude that the algorithm which does not concrete the residue could not handle the burst traffic efficiently.

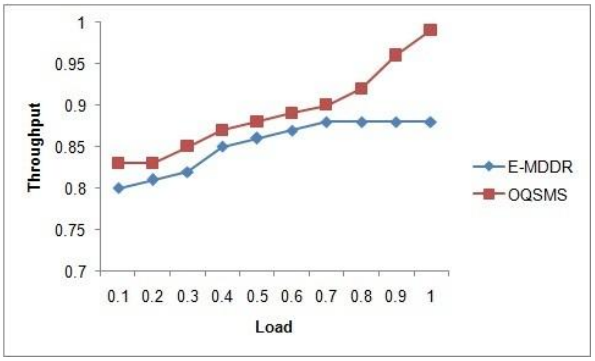

Fig. 10 Throughput as function of 64x64 switch for uniform traffic

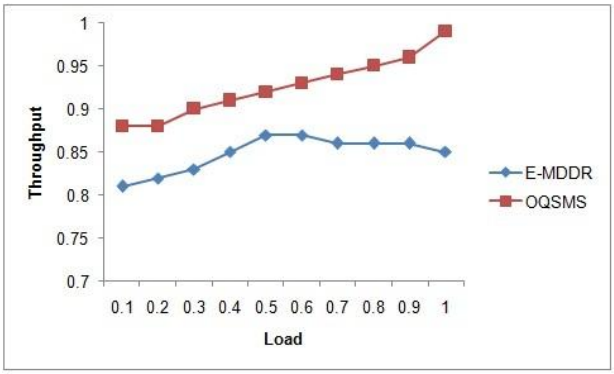

Fig. 11 Throughput as function of 128x128 switch for uniform traffic

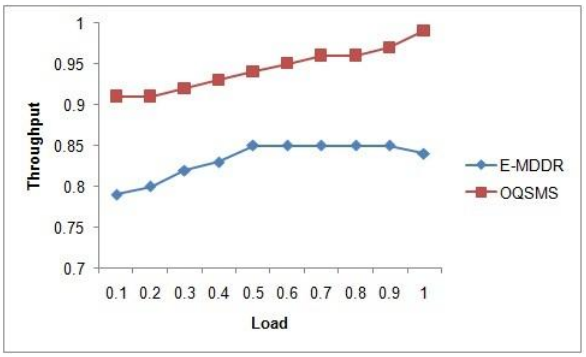

Fig. 12 Throughput as function of 256x256 switch for uniform traffic

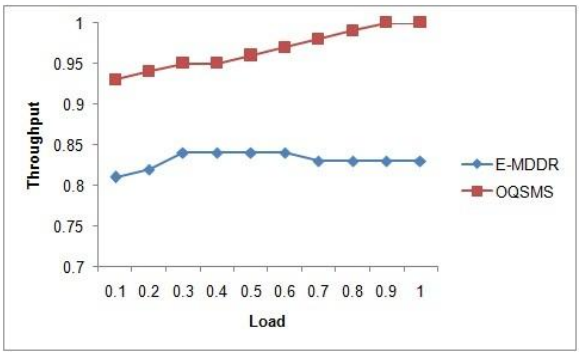

Fig. 13 Throughput as function of 512x512 switch for uniform traffic

Lastly, we investigate how a switch's size, i.e., the number of inputs and outputs, impact our scheduler's performance. Figure 10-13 show the throughput performance of 64x64, 128x128, 256x256 and 512x512 under uniform traffic scenario. We see that E-MDDR scheduler's performance degrades with increasing switch sizes. When the input load reaches 50% the throughput of E-MDDR is degraded. This is due to multicast cell's large fan-out. But we notice that when the increase of switch size and load, OQSMS achieves above 95% of throughput because of high fan-out and switch size OQSMS reduces the computation steps for scheduling the cells.

Our future work includes, implementing the E-MDDR and OQSMS algorithms with feedback based two-stage switch architecture and analyze the performances also extend the OQSMS algorithm to the integrated scheduler which supports both unicast and multicast scheduling simultaneously.

5 Conclusion

In this paper we have analyzed the performance of the scheduling algorithms OQSMS, E-MDDR and MDDR. From the simulation results that OQSMS achieves more that 95% of the throughput under both uniform and non-uniform traffic pattern. OQSMS estimates optimal queue selection based on more queue combinations so that it achieves minimum delay and maximum in throughput. E-MDDR performance is better than MDDR since E-MDDR concentrate on the residue of the cells waiting more than N cell times also identical fan-out of the cells are not consecutively scheduled which achieved the better output port utilization. We can also conclude that when the switch size is increased OQSMS achieves maximum throughput compared with E-MDDR.

References

[1] Bianco A, Giaccone P, Leonardi E, Neri F, and Piglione C., "On the number of input queues to efficiently support multicast traffic in input queued switches," *In Proceedings of Workshop on High Performance Switching and Routing*, pp. 111–116, 2003.

[2] Bianco A, Scicchitano A., "Multicast support in multi-chip centralized schedulers in input queued switches," *Computer Networks*, vol. 53, no. 7, pp. 1040–1049, 2009

[3] Gupta S, and Aziz A., "Multicast scheduling for switches with multiple input-queues," *In Proceedings of High Performance Interconnects Symposium*, pp. 28–33, 2002.

[4] Marsan M.A, Bianco A, Giaccone P, Leonardi E, and Neri F, "Multicast traffic in input-queued switches: optimal scheduling and maximum throughput," *IEEE/ACM Transactions on Networking*, vol. 11, no. 3, pp. 465–477, 2003.

[5] McKeown N, and Prabhakar B., "Scheduling multicast cells in an input queued switch," *In Proceedings of IEEE INFOCOM*, vol. 1, pp. 271–278, 1996.

[6] McKeown N., "A Fast Switched Backplane for a Gigabit Switched Router," *Business Communication Review*, vol. 27, no. 12, 1997.

[7] McKeown N, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Transactions on Networking*, vol. 7, no. 2, pp. 188–201, 1999.

[8] Pan D. and Yang Y., "FIFO-based multicast scheduling algorithm forvirtual output queued packet switches," *IEEE Transactions on Computers*, vol. 54, no. 10, pp. 1283–1297, 2005.

[9] Prabhakar B, McKeown N, and Ahuja R., "Multicast scheduling for input-queued switches," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 5, pp. 855–866, 1997.

[10] ShanmugamArumugam, ShanthiGovindaswamy., "Performance of the Modified Round Robin Scheduling Algorithm for Input-Queued Switches Under Self-Similar Traffic," *In Proceedings of the International Arab Journal of Information Technology*, vol.3, no.2, 1996.

[11] Song M, and Zhu W., "Throughput analysis for multicast switches with multiple input queues," *IEEE Communications Letters*, vol. 8, no. 7, pp. 479–481, 2004.

[12] Yongbo Jiang, ZhiliangQiu, Ya Gao, and Jun Li, "Multicast Support in Input Queued Switches with Low Matching Overhead", *IEEE Communications Letters*, vol. 16, no. 12, 2012.

[13] Zhu W, and Song M., "Integration of unicast and multicast scheduling in input-queued packet switches," *Computer Networks*, vol. 50, pp. 667– 687, 2006.

[14] Zhu W, and Song M., "Performance analysis of large multicast packet switches with

multiple input queues and gathered traffic," *Computer Communications*, vol. 33, no. 7, pp. 803–815, 2010.

[15] Navaz K, and Kannan Balasubramanian., "OQSMS:Optimal Queue Selection Based Multicast Scheduling Algorithm for Input-Queued Switches," Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 373-378

[16] Navaz K, Dr.Kannan Balasubramanian., "Multicast Due Date Round-Robin Scheduling Algorithm for Input-Queued Switches" International Journal of Computer Network and Information Security, 2016, 2, 56-63

[17] Navaz K, and Kannan Balasubramanian., "E-MDDR SCHEDULING ALGORITHM FOR INPUT-QUEUED SWITCHES," i-manager's Journal on Computer Science, Vol. 4 l No. 1 l March - May 2016

[18] W.Bux, W.Denzel, T.Engbersen, A.Herkersdorf, and R.Luijten, "Technologies and Building Blocks for Fast Packet Forwarding", IEEE Communications Magazine, vol. 39, no. 01, pp. 70–77, 2001.

[19] L.Roberts, "Beyond Moore's Law: Internet Growth Trends", IEEE Computer Magazine, vol. 33, no. 1, pp. 117–119, 2000.