IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-9, Sep 2018 E-ISSN: 2347-2693

Throughput Analysis of Multicast Scheduling Algorithms by Varying NxN IQ Switch

Shaik Jumlesha [*]**, K. Navaz** [2]**, S. Athinarayanan** [3]

[1, 2& 3]Dept. Of CSE, Annamacharya Institute of Technology and Sciences, Tirupati, India

[*]*Corresponding Author:* navazit@gmail.com,

Available online at: www.ijcseonline.org

Accepted: 26/Sept/2018, Published: 30/Sept./2018

***Abstract*—** Incredible measures of exertion have gone into investigate on multicast switch texture outline and calculations. Switch measure is one of the primary factor which impacts the execution of throughput and deferral. In this work, execution of switch has been investigated by applying the progressed multicast planning calculation OQSMS (Optimal Queue Selection Based Multicast Scheduling Algorithm), due date based round-robin booking calculation MDDR(Multicast Due Date Round Robin) and double round-robin based multicast planning calculation MDRR(Multicast Dual Round Robin). Recreation results demonstrate that OQSMS accomplishes preferred exchanging execution over different calculations under the allowable movement conditions on the grounds that if the switch measure builds, OQSMS will gauge ideal line determination in view of more line mixes so it accomplishes greatest conceivable throughput.

***Keywords*—** Multicast, Throughout, MDRR, MDDR, OQSMS

I. Introduction

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.

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

Multicasting is the ability to provide point-to-multipoint connections. Driven by the Internet and its applications, such as video on demand (VOD), music on demand (MOD), teleconferencing, videoconferencing and distributed data processing, more and more communication services and applications will require that information from a source be delivered to multiple destinations. Multicasting will become an important feature for any switching network designed to support broadband integrated service digital networks (B-ISDN). Generally speaking, packet switch architectures can be divided into three major categories [11]: 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 bus greatly limits their use when the switch size grows large. A crossbar switch is a switch connecting multiple inputs to multiple outputs in a matrix manner. The crossbar constraints of an IQ switch requires 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.

The fixed-size packet transmitted by the switch fabric is also called cell. We consider only the fan-out splitting discipline that 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. The N×N switch architecture is shown in Fig.1. Let our assumptions as NxN switch having N input ports and N output ports an fabric is connecting input ports and output ports at any time slot. Let 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 fanout sets.

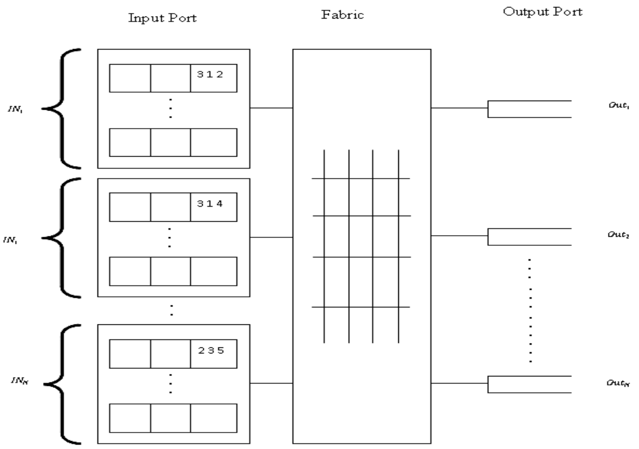

Fig. 1 NxN IQ Switch Architecture

In this paper, we analyze the throughput performance of the IQ switch by varying its size. The rest of the paper is organized as follows. In Section II, related works on designing multicast scheduling algorithms are reviewed. In Section III, MDDR and OQSMS algorithms have been reviewed. In Section IV, Performance evaluation and result analysis have been presented. Finally, we conclude the paper in Section V.

II. Related Work

Most existing multicast switches [8],[9] require multicast switch fabric and a sophisticated central scheduler for maximizing switch performance. TATRA [8] 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 [9] and [5]. In [9], each input port maintains a set of multicast queues.

To reduce the HOL blocking further, a multicast packet split scheme is proposed in [5]. In [5], 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 [10], 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[5]. When number of queues and fan-out size (ef) increase, MDRR could not yield maximum throughput than MaxService scheme does.

III. MUTICASTION SCHEDULING ALGORITHMS

A. Multicast Due Date Round-Robin (MDDR)

In [13] MDDR, 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 was generated based on the priority of cells contained in the fan-out. The highest fan-out size port having 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.

1. Request Phase

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 are measured in each input port and the priorities based on higher size is allotted. 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 obviously. 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.

2. Grant Phase

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

B. OQSMS Algorithm

In this section we detail the OQSMS[12] algorithm design and its components.

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

1. *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.
2. *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 splited in order to avoid HOL.

3. *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.
4. *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.

C. Reservation Set

If the reservation set RV is the array set of queue names selected to each input port. For all the input ports we can have multiple permutations with various queues.

D. Queue Selection

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.

IV. Results and Discussion

In this section, we compare the performance of the OQSMS algorithm with other pointer based algorithm, designed to support multicast traffic under different traffic conditions. The input traffic is generated according to a Bernoulli arrival process, in which p is the average input load i.e., the probability that a packet arrives at an input port during a timeslot and uniform bursty traffic. Bursty arrivals are based on the ON/OFF model. Specifically, in the ON state a packet arrival is generated in every time slot, in the OFF state, no packet arrives. Packet of the same burst have the same fanout set.

We consider only the multicast traffic in the simulation. When multicast packet arrives its fanout set consists of f output randomly chosen from all output ports. Where f is the fanout size randomly selected between[2,N].

Here the influence of switch size parameter N was investigated. Queue size parameter k was fixed according to the output ports. The results are presented in table 1 and table 2. Throughput is the performance measure 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 throughput as a function of switch size for OQSMS, MDDR and MDRR under Bernoulli traffic condition. While increasing the switch size, OQSMS achieves the higher throughput than other two pointer based algorithms since switch size increases OQSMS will have more queue combination this will lead to achieve the maximum possible throughput. MDDR achieves low throughput in larger size switch because queued is selected for a port based on higher fanout size.

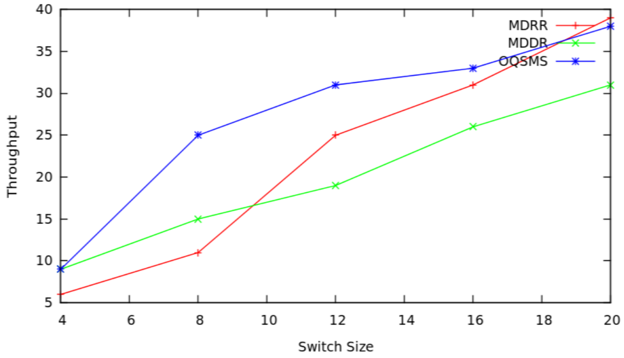

Fig.2 Switch Size Vs Throughput (Bernoulli)

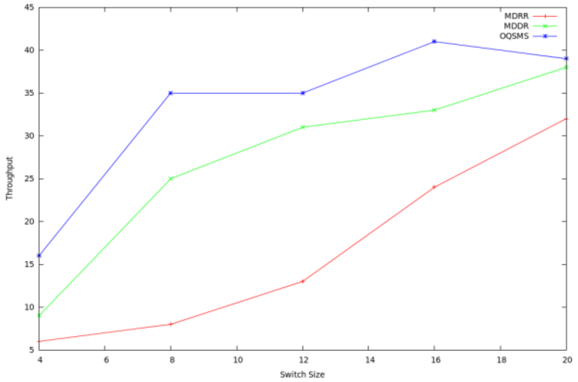

Fig.3 Switch Size Vs Throughput (Bursty)

Fig.3 shows the throughput as a function of switch size for OQSMS, MDDR and MDRR under Bursty traffic condition. While increasing the switch size, OQSMS achieves the higher throughput as Bernoulli arrivals. Here MDRR achieves low throughput when switch size increases because bursty arrivals group of cells carry the same fanout and queue selection is based on the primary and secondary pointer.

It’s clearly show that the OQSMS is always having higher throughput than the other algorithms for both traffic scenarios.

V. Conclusion

The advanced scheduling algorithms OQSMS and pointer based algorithms to compare the throughput while switch size is increasing. From the investigation OQSMS achieves higher throughput when switch size is large, since this algorithm calculates throughput range (Rn) in every timeslot. This lead to achieve the higher throughput in both traffic patterns.

REFERENCES

[1] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," IEEE/ACM Trans. Netw., vol. 7, no. 2, pp. 188–201, Apr. 1999.

[2] N. McKeown and B. Prabhakar, "Scheduling multicast cells in an input queued switch," in Proc. 1996 IEEE INFOCOM, vol. 1, pp. 271–278.

[3] B. Prabhakar, N. McKeown, and R. Ahuja, "Multicast scheduling for input-queued switches," IEEE J. Sel. Areas Commun., vol. 15, no. 5, pp. 855–866, June 1997.

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

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

[6] M. Song and W. Zhu, "Throughput analysis for multicast switches with multiple input queues," IEEE Commun. Lett., vol. 8, no. 7, pp. 479–481, July 2004.

[7] W. Zhu and M. Song, "Performance analysis of large multicast packet switches with multiple input queues and gathered traffic," Computer Commun., vol. 33, no. 7, pp. 803–815, May 2010.

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

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

[10] Yongbo Jiang, Zhiliang Qiu, Ya Gao, and Jun Li, "Multicast Support in Input Queued Switches with Low Matching Overhead" in IEEE COMMUNICATIONS LETTERS, VOL. 16, NO. 12, DECEMBER 2012.

[11] F.A. Tobagi, "Fast Packet Switch Architectures For Broadband Integrated Service Digital Networks", Proceedings of the IEEE, Vol.78, No.1, p.p. 90-167, January, 1990

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

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

Authors Profile

Dr S Jumlesha pursued Bachelor of Science from S V University and Master of Engineering from Satyabama University of Tamilnadu, India in year 2003 and 2008 respectively. Completed Ph.D from JNTUH Hyderabad. and currently working as Professor in Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences, Tirupathi, India. He is a member of LMISTE,IAENG & ICSES computer society. He has published more than 32 research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences. His main research work focuses on Image processing, Data Mining, Machine learning and artificial Intellegence. He has15 years of teaching experience and 5 years of Research Experience.

Dr Navaz K pursued Bachelor of Technology and Master of Engineering from Anna University of Tamilnadu, India in year 2006 and 2009 respectively. He is also completed Ph.D from MSU Tirunelveli and currently working as Associate Professor in Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences, Tirupathi, India. He is a member of IAENG & ICSES computer society. He has published more than seven research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences. His main research work focuses on Computer Networks, Network Security, Cloud Security and Privacy, IoT and Computational Intelligence based education. He has 9 years of teaching experience and 5 years of Research Experience.

Dr Athinarayanan S pursued Bachelor of Science and Master of Technology from from MS Univer sity of Tamilnadu, India in the year 2003 and 2008 respectively. Completed Ph.D from MSU Tirunelveli and curre ntly working as Professor in Department of Computer Science and Engineering, Annamacharya Institute of tute of Technology and Sciences, Tirupathi, India. He is a member of LMISTE,IAENG & ICSES computer society. He has published more than 22 research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences. His main research work focuses on Image processing, Computer Networking. He has 11 years of teaching experience and 7 years of Research Experience.