International Journal of Intelligent Information Systems
2015; 4(2-1): 7-12
Published online January 28, 2015 (http://www.sciencepublishinggroup.com/j/ijiis)
doi: 10.11648/j.ijiis.s.2015040201.12
ISSN: 2328-7675 (Print); ISSN: 2328-7683 (Online)

Packet switch scheduler for increasing sending packet

Kazunori Omori, Myungryun Yoo, Takanori Yokoyama

Information Engineering, Tokyo City University, Tokyo, Japan

Email address:
g1381501@tcu.ac.jp (K. Omori), yoo@cs.tcu.ac.jp (M. Yoo), yokoyama@cs.tcu.ac.jp (T. Yokoyama)

To cite this article:
Kazunori Omori, Myungryun Yoo, Takanori Yokoyama. Packet Switch Scheduler for Increasing Sending Packet. *International Journal of Intelligent Information Systems*. Special Issue: Logistics Optimization Using Evolutionary Computation Techniques.
Vol. 4, No. 2-1, 2015, pp. 7-12. doi: 10.11648/j.ijiis.s.2015040201.12

Abstract: Recently, the need of the high speed packet switch is increased. The Re-2DRR scheduling algorithm based on 2DRR scheduling algorithm provides high throughput communication on a packet switch. However, computer network is using many cases, that huge data communication, complex, and other. This paper proposes a new method to increase choices in algorithm variation for three specific systems and more easily implementation than Re-2DRR. The effectiveness of the proposed algorithm is shown through simulation studies.

Keywords: Packet Switch Scheduling algorithm; 2DRR; High-Throughput

1. Introduction

In recent years, the number of traffic packet network is on the increase [1], [2]. And they are making many kinds of systems on a computer networks, and they use to a computer network for special use for example, multimedia communications [3], [4], and other. They are needed more the high speed packet switch scheduler for special use [5]. Richard presented the basic two Dimensional Round Robin (2DRR) scheduling algorithm [6]. The four matrixes, Request Matrix (RM), Pattern Matrix (PM), Scheduling Matrix (SM), and Allocation Matrix (AM) are used in the basic 2DRR. Each matrix size is N x N, where N means the number of inputs and outputs [7]. The basic 2DRR provides high throughput, fair access [8], [9] and simple working on a packet switch. However, few input and output node could not be permissioned in some timeslots because the basic 2DRR use only value of PM to running scheduling. It occurs decreasing throughput and some transmission delay.

The Repetitive two Dimensional Round Robin (Re-2DRR) scheduling algorithm provides higher throughput than that of the basic 2DRR scheduling algorithm [10]. The Re-2DRR is using new matrix about Sub-AM. It is solve the 2DRR delay point. Although the Re-2DRR do making Sub-AM many times, and it hindered an implementation.

In this paper, we propose two method based on Re-2DRR. And we increase choices in algorithm for special use. First method is changing to making Sub-AM operation that the basic Re-2DRR was remaking many times at it have empty packets. If this method recognize an empty packet at AM, that just made Sub-AM group once time. It is suppression a repeat operation. Second method is making priority port in Re-2DRR. It enables high throughput at specific port.

2. Related Works

2.1. The Basic Two Dimensional Round Robin (2DRR)

In an *N* x *N* switch (Fig.1), up to *N* different requests can be simultaneously served by the switch in one time slot such that no two requests are in the same row or column in the request matrix. In order to select such *N* elements of the request matrix, RM, the basic 2DDR examines elements of RM that belong to generalized diagonals.

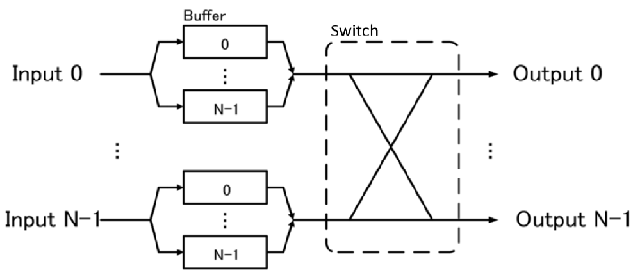

Figure 1. *N x N Switch*

Definition 1: A generalized diagonal is a set of *N* elements in an *N* x *N* matrix, such that no two elements are in the same row or column.

There are $N!$ different generalized diagonals in an N x N matrix. The basic 2DRR algorithm uses only N of these diagonals by selecting one basic diagonal and then generating the remaining $N-1$ ones by shifting the basic diagonal across the matrix (so that each matrix element is covered by one of the N diagonals). That is, by sweeping a generalized diagonal pattern of length N through the request matrix, all N^2 input output pairs in the request matrix can be satisfied in N time slots. This property is used to guarantee a minimum amount of service to each input/output queue.

The basic 2DRR scheduling algorithm operates in repeating cycles of N time slots in which the time slots of each cycle are indexed by the variable L, which takes on values from 0 through $N–1$. The following 4 matrices are assumed.

1) Request Matrix (RM)

Each entry RM[R, C] is binary with the semantics:

RM[R, C] = 1, if there is at least one request for a connection from output R to output C
0, otherwise

2) Scheduling Matrix (SM)

Each entry SM[R, C] contains an integer between 0 and N–1 inclusive where

$$SM[R,C] = (C - R) \bmod N$$

If SM[R, C] = K, then RM[R, C] is covered by diagonal pattern K.

3) Pattern Sequence Matrix (PM)

Each entry PM[I, J] is an integer between 0 and N–1 inclusive with the semantics:

PM[I, J] = K implies that when the timeslot index L of a cycle is equal to J, then the I-th diagonal pattern in the sequence applied by the algorithm is the one numbered K in the diagonal pattern matrix. The ordering index I varies from 0 to N–1.

4) Allocation Matrix (AM)

It is binary entries and the semantics:

$$AM[R,C] = \begin{cases} 1, & \text{if a connection is allocated from input } R \text{ to output } C \\ 0, & \text{otherwise} \end{cases}$$

At the beginning of time slot L in a cycle, all entries of the allocation matrix are set to zero. Then a sequence of N diagonal patterns is applied to the request matrix in the order specified by the pattern sequence matrix PM. That is, the diagonal pattern with index $PM[0, L]$ is applied first followed by diagonal pattern $PM[1, L]$ . . . $PM[N-1, L]$. As these diagonal patterns are overlaid on the request matrix, the entry $AM[R, C]$ is set to 1 at the I-th point in the sequence if the following conditions are true.

- $RM[R, C] = 1$
- Input R and output C are still available for allocation (i.e., they have not been allocated to a different connection by a previously applied diagonal in the current time slot)
- $SM[R, C] = K$. where $PM[I, L] = K$

The above scheduling procedure is repeated for each cycle of N successive time slots. That is after a cycle has been completed with the use of column $N–1$ of the pattern sequence matrix.

The basic 2DRR algorithm provides a fairness guarantee that each of the N^2 input/output queues will receive at least one opportunity for service during every cycle of N time slots. However, few input and output node could not be permissioned in some timeslots because the basic 2DRR use only value of PM to running scheduling. It occur scheduler tightly and throughput delay.

2.2. The Repetitive two Dimensional Round Robin (Re-2DRR)

Re-2DRR based on the basic 2DRR. The Re-2DRR creates Sub-AM if AM is not usable because of empty sending packet. If Sub-AM is not better than AM, scheduling algorithm retry creating Sub-AM by another SM, but that retrying have a limit. All created Sub-AM is not better than AM, the Re-2DRR chooses either AM or Sub-AM.

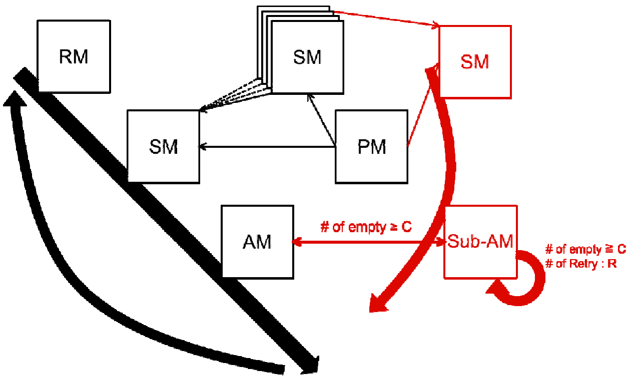

Figure 2. *Re-2DRR Scheduling Algorithm*

Fig.2 is Re-2DRR Scheduling Algorithms figure. The black color part shown the basic 2DRR scheduling work, and the red color part is the Re-2DRR scheduling works.

In the Re-2DRR scheduling algorithm, two thresholds are used. The first threshold is the number of empty sending packet for decision of available AM (threshold “C”) and the second threshold is the upper limit of retrying to create Sub-AM (threshold “R”). The scheduler works more effectively by setting two thresholds optionally.

The Re-2DRR can operation flexibly by changing that two threshold. In threshold C, it changing probability of operating Re-2DRR. And threshold R is changing probability of algorithm chose Sub-AM and number of making Sub-AMs. In table 1 show a variation of the Re-2DRR operation results by N=4.

We recognize the operation of making Sub-AM is burdened with implementation. And the Re-2DRR is making for high throughput at all port in packet switch communications.

Table 1. *Result of Re-2DRR in simulation (N=4)*

C	R	Making Sub-AM	Sending by Sub-AM	End Time	Ave Packet Wait Time
1	1	18833	4315	277192	138556
	2	33922	5494	277004	138433
	3	49911	6227	277325	138635
	4	61212	6655	276782	138430
2	1	8465	1159	277082	138483
	2	15763	1472	277196	138536
	3	23608	1527	277161	138560
	4	29519	1593	277045	138485
3	1	4042	257	277077	138554
	2	8284	334	277278	138651
	3	12703	363	277351	138692
	4	16608	386	277230	138545

3. System to Assume

In this paper we define three specific systems. It is make more packet loading.

3.1. Less than N Port Have More Packet Load

It is less than *N* port have more packet loading. It like Fig.3 (a). In Fig.3 (a) mean a communication of Input 1 to Output 1 (point of red) use more packet traffics.

This traffic is estimated to be about some node use big data communication temporary.

3.2. Static N Port Have More Packet Load

This specific system is have more packet load at number of *N* communications. Example in Fig.3 (b) mean static number of 4 input to number of 4 output (place of red) have more packet loading.

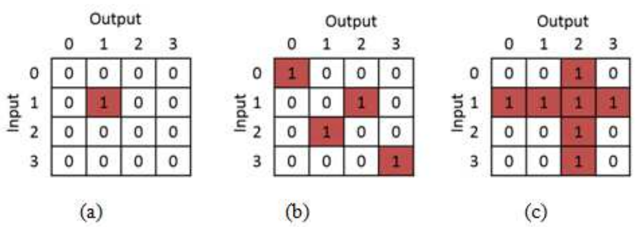

Figure 3. *A place of more packet load*

This traffic is estimated to be about number of *N* nodes and number of *N* nodes make static communication and use a big data.

3.3. One to All and All to One Port Have More Packet Load

This specific system is like a broadcast communication. In Fig.3 (c) mean input 1 to all output, and all input to output 2 have more packet load.

4. Improvement

In this paper, we improve portion of Re-2DRR. And we regarded two methods.

4.1. Selectable Sub-AM Group

In Re-2DRR, if Sub-AM worse than AM about the number of empty packets, scheduler make more another one Sub-AM for R times. This method improve about portion of making Sub-AM. If AM have many empty packet, this algorithm making the number of G Sub-AMs. After that, scheduler checking the better in Sub-AM group and select the best Sub-AM. If AM is better than Sub-AM groups Sub-AMs after made Sub-AM group, that algorithm send by AM.

In this method suppression a repeat making Sub-AM operation.

In Fig.4, the black color is same of Re-2DRR. The red color is this method point.

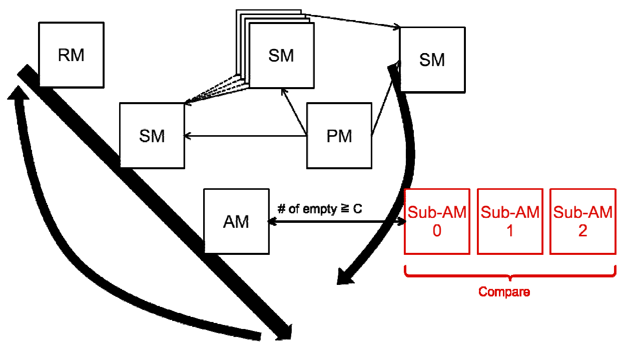

Figure 4. *Selectable Sub-AM group*

4.2. Make Priority Port

The basic Re-2DRR is choose AM or Sub-AM about the number of empty packet. This method is making priority port in operation of checking and choosing SM or Sub-SM of the basic Re-2DRR.

This method algorithm is choosing AM or Sub-AM preferentially about a set priority. For example in Fig.5, it is example of AM or Sub-AM. And place of yellow are priority ports in this time. This method algorithm choosing (b), because place of yellow is priority ports have permission of sending packet. So this method algorithm don't need the threshold of C, because this method just judge by priority port.

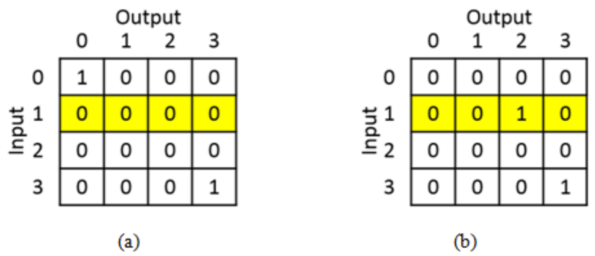

Figure 5. *Example of priority port*

5. Validation

To validate proposed two methods on Re-2DRR, several numerical tests are performed. We compared proposed selectable Sub-AM group on Re-2DRR, setting priority port on Re-2DRR and basic Re-2DRR. Numerical tests are performed with *N* x *N* matrix (*N*= 4). The throughput and the finish time of all sending packets are compared. The throughput is average waiting time of all packets in packet switch buffer. *C* and *R* is the threshold in Re-2DRR. *C* is the number of empty sending packet for decision of available AM and *R* is the upper limit of retrying to create Sub-AM. *G* is the threshold in the method of selectable Sub-AM group about the number of Sub-AM in Sub-AM group. The method of making priority port isn't need *C*.

Table 2, 3, 4, 5, 6, 7 and 8 show the comparison of results by basic Re-2DRR and two method on Re-2DRR. Table 2 is the results on more treble packet load to (1, 1) port. Table 3, 4 and 5 are the results on the number of *N* port (in this time N is 4: (0, 0), (1, 1), (2, 3) and (3, 2) ports) have more treble packet load. Table 4 is the results about each four ports average of packet waiting time. And table 5 is the results about four ports maximum of packet waiting time. Table 6, 7 and 8 are the results on input 1 to all output and all input to output 2 have more treble packet load. Table 7 is the results about each seven ports average of packet waiting time. And table 4 is the results about seven ports maximum of packet waiting time.

In table 2, Sub-AM group (*C*:1 *G*:4) and Sub-AM group (*C*:1 *G*:3) results close to Re-2DRR (*C*:1 *R*:4) and Re-2DRR(*C*:1 *R*:3) results except for Make Sub-AM group. The basic Re-2DRR don't making Sub-AM groups, so need extravagant making Sub-AM operation. However, method of Sub-AM group on Re-2DRR made Sub-AM group in advance. It mean the method of Sub-AM group in Re-2DRR can working by less making Sub-AM operation. The (1, 1) ports throughput of method of making priority port on Re-2DRR is more desirable than those of the others. But another values are increasing. But Priority (*R*:3) keep values increasing and keep high (1, 1) ports throughput.

In table 3, the method of making Sub-AM group on Re-2DRRs results are close to the basic Re-2DRR. And it shows the method of making Sub-AM groups making Sub-AM operation are less than the basic Re-2DRRs making Sub-AM operation. However the method of making priority port on Re-2DRR is not have enough effects. This method can't cause empty packet sending, because it give very high priority to setting four priority ports at (0, 0), (1, 1), (2, 3) and (3, 2). There are get big priority area, so that method increasing chose AM. In table 6, 7 and 8, this method setting seven priority port at same of more packet load ports, which is get more big priority area. It is block to making Sub-AM. Priority (*R*:4) limit in table 6, 7 and 8 is setting priority at input 1 about (1, 0), (1, 1), (1, 2) and (1, 3). It mean setting priority area narrowing. In conclusion, the method of making priority port on Re-2DR R have a just effect on one priority port like table 2.

Table 2. *Comparison on (1, 1) port have more packet load*

	Make Sub-AM	**Send by Sub-AM**	**End Time**	**Ave Wait Time**	**Make Sub-AM group**	**(1, 1) Ave Wait**	**(1, 1) MAX Wait Time**
Re-2DRR (*C*:1 *R*:4)	543976	6908	398402	1561.236	-	2667	4456
Re-2DRR (*C*:1 *R*:3)	414737	6042	399563	1564.182	-	2677	4516
Sub-AM group (*C*:1 *G*:4)	546784	6077	394756	1553.451	136696	2643	4735
Sub-AM group (*C*:1 *G*:3)	418731	5916	397935	1560.104	139577	2662	4541
Priority (*R*:4)	1134177	136500	396138	1670.850	-	1106	2876
Priority (*R*:3)	583186	115229	397203	1659.550	-	1560	4093

Table 3. *Comparison on the number of N port have more packet load*

	Make Sub-AM	**Send by Sub-AM**	**End Time**	**Ave Wait Time**	**Make Sub-AM group**
Re-2DRR (*C*:1 *R*:4)	332910	67320	423527	1993.703	-
Re-2DRR (*C*:1 *R*:3)	257648	55580	424641	1995.891	-
Sub-AM group (*C*:1 *G*:4)	386220	39035	423077	1989.144	96555
Sub-AM group (*C*:1 *G*:3)	285531	36749	425153	1998.776	139577
Priority (*R*:4)	90322	73489	440118	2047.618	-
Priority (*R*:3)	90400	73511	445007	2065.108	-

Table 4. *Comparison on the number of N ports have more packet loads average of packet waiting time*

	(0, 0)	**(1, 1)**	**(2, 3)**	**(3, 2)**
Re-2DRR (*C*:1 *R*:4)	2649	2659	2668	2661
Re-2DRR (*C*:1 *R*:3)	2656	2676	2647	2654
Sub-AM group (*C*:1 *G*:4)	2653	2659	2670	2653
Sub-AM group (*C*:1 *G*:3)	2661	2692	2687	2647
Priority (*R*:4)	2403	2589	2767	2582
Priority (*R*:3)	2466	2593	2780	2612

Table 5. *Comparison on the number of N ports have more packet loads maximum of packet waiting time*

	(0, 0)	**(1, 1)**	**(2, 3)**	**(3, 2)**
Re-2DRR (*C*:1 *R*:4)	4794	4646	4628	4620
Re-2DRR (*C*:1 *R*:3)	4480	4739	4519	4660
Sub-AM group (*C*:1 *G*:4)	4663	4651	4597	4468
Sub-AM group (*C*:1 *G*:3)	4940	4773	4547	4574
Priority (*R*:4)	4813	4954	4713	4938
Priority (*R*:3)	4783	4636	4877	4811

Table 6. *Comparison on input 1 to all output and all input to output 2 have more packet load*

	Make Sub-AM	**Send by Sub-AM**	**End Time**	**Ave Wait Time**	**Make Sub-AM group**
Re-2DRR (*C*:1 *R*:4)	1857306	99209	796028	2991.344	-
Re-2DRR (*C*:1 *R*:3)	1430890	95902	799003	2998.931	-
Sub-AM group (*C*:1 *G*:4)	2162800	96774	802024	3007.809	540700
Sub-AM group (*C*:1 *G*:3)	1640709	95880	807698	3024.896	546903
Priority (*R*:4)	0	0	801634	3122.378	-
Priority (*R*:3)	0	0	806794	3133.948	-
Priority (*R*:4) limit	73350	3689	799421	3103.385	-

Table 7. *Comparison on input 1 to all output and all input to output 2 have more packet load s average of packet waiting time*

	(1,0)	**(1,1)**	**(1,2)**	**(1,3)**	**(0,3)**	**(2,2)**	**(3,2)**
Re-2DRR (*C*:1 *R*:4)	3851	3946	3894	3797	3836	3900	3880
Re-2DRR (*C*:1 *R*:3)	3552	3653	5250	3171	3333	3339	3740
Sub-AM group (*C*:1 *G*:4)	3352	3550	5264	3557	3706	3618	3169
Sub-AM group (*C*:1 *G*:3)	3519	3691	5320	3280	3376	3433	3744
Priority (*R*:4)	3910	3860	3905	3892	3908	3944	3954
Priority (*R*:3)	3859	3861	3951	3961	3973	3968	3901
Priority (*R*:4) limit	3803	3899	3892	3909	3910	3955	3844

Table 8. *Comparison on input 1 to all output and all input to output 2 have more packet loads maximum of packet waiting time*

	(1,0)	**(1,1)**	**(1,2)**	**(1,3)**	**(0,3)**	**(2,2)**	**(3,2)**
Re-2DRR (*C*:1 *R*:4)	9169	9423	9351	9460	8808	8458	8395
Re-2DRR (*C*:1 *R*:3)	8111	8143	8892	7378	7935	8032	8462
Sub-AM group (*C*:1 *G*:4)	8185	7961	9095	8018	7906	8011	7502
Sub-AM group (*C*:1 *G*:3)	8018	8424	9499	7938	8008	8400	8808
Priority (*R*:4)	8451	8753	9262	8659	9320	9339	9033
Priority (*R*:3)	8549	8936	9376	9054	8881	9117	8881
Priority (*R*:4) limit	8520	8728	8915	8648	8712	8934	8678

6. Conclusion

A new two method on Re-2DRR is proposed in this paper. The proposed method of making Sub-AM group on Re-2DRR is designed for less making Sub-AM operation. It mean more easer working and implementation. And the method of making priority port on Re-2DRR is designed for the high throughput in one specific priority port communications.

Acknowledgements

This work is supported in part by JSPS KAKENHI Grant Number 24500046.

References

[1] M. Takajo, S. Kimura and Y. Ebihara, "A Proposal of Grouped Two-Dimensional Round Robin Schedulers for Multiaccess Communications," The Transactions of the Institute of Electronics, Information and Communication Engineers B, Vol. J82-B, No. 4, 1999, pp. 560-568.

[2] Craig Partridge et al., "A 50-Gb/s IP Router," IEEE/ACM Transactions of Networking, Vol. 6, No. 3, 1998, pp. 237-248.

[3] T. Tsuda, "Multimedia Communication System," The Institute of Image Information and Television Engineers, Vol. 45, No. 1, 1991, pp. 31-34.

[4] K. Nishimura, T. Mori and Y. Ishibashi, "Video-on-Demand with Multiple Readouts," The Institute of Image Information and Television Engineers, Vol.48, No. 3, 1994, pp. 287-294.

[5] ITU-T: ITU-T Recommendation I.371, 1993, pp.2-4.

[6] R. O. LaMaire, "Two-Dimensional Round-Robin Schedulers for Packet Switches with Multiple Input Queues," IEEE/ACM Transactions on Networking, Vol. 2, No. 5, 1994, pp. 471-482.

[7] M. J .Karol, M. G. Hluchyj and S. P. Morgan, "Input versus output queueing on a space-division packet switch," IEEE Trans. Commun., Vol. COM-35, 1987, pp. 1347-1356.

[8] J. Wong, J.p. Sauve, and J.S. Field, "A Study of Fairness in Packet Switching Networks," IEEE Trans. Commun., col.COM-30, No.2, 1982, pp. 346-353.

[9] E. L. Hahne, "Round-Robin Scheduling for Mac-Min Fairness in Data Networks," IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, 1991, pp. 1024-1039.

[10] K. Omori, M. Yoo , T. Yokoyama, "A Packet Switch Scheduler based on 2DRR for High-Throughput," The 75th National Convention of IPSJ, 1L-4, 2013, pp. 205-206.