Scheduling with Relaxed Constraint for ATM-like Input-Queued Crossbar Switching Fabric in IP Router

Lixiang Xiong
School of Electrical and Information Engineering
University of Sydney
Sydney, NSW 2006 Australia
Email: xlx@ee.usyd.edu.au

Don Platt
School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
Wollongong, NSW 2522 Australia
Email: don_platt@elec.uow.edu.au

Guoqiang Mao
School of Electrical and Information Engineering
University of Sydney
Sydney, NSW 2006 Australia
Email: guoqiang@ee.usyd.edu.au

Abstract— **Scheduling scheme for ATM-like input-queued crossbar switching fabric in IP router performs a critical role in IP router. However, there is a problem among the existing scheduling schemes: when an IP router with a large number of outputs (e.g. 64, 128 or more) is connected to only a few nodes (e.g. 2 or 4 nodes, where the node may be another IP router or a switch), the switching capability is not fully utilized. In this paper, we propose an approach to improve the existing scheduling schemes: all outputs of the switching fabric are divided into a few groups whose number is equal to the number of nodes to be connected, and outputs in the same group are multiplexed into a high-speed output link which is connected to a node. Therefore all outputs of the switching fabric can join the switching. The approach is applied to several popularly used scheduling algorithms. Simulation is carried out to demonstrate a better performance.**

I. Introduction

Research on IP Router always attracts extensive attention from researchers and vendors. One popular approach to develop IP router is to apply ATM-like input-queued crossbar switching fabric [1], [2]: variable-length IP packets arriving at the IP router are fragmented into fixed-length cells, and these cells are switched by the switching fabric, which is controlled by some scheduling scheme, then are reassembled back into IP packets and transmitted out of IP router. The scheduling scheme for the switching fabric performs a critical role in the IP router. Probabilistic Iterative Matching (PIM) [3] is one of the most popularly developed scheduling schemes. By applying a random policy during scheduling, PIM can achieve an excellent throughput close to 100% with an optimum number of iterations per cell time. The optimum number of iterations per cell time should be $log_2 N$, where N is the switch size of an NxN switch. Based on PIM, many scheduling schemes are developed. Some of them still apply a random policy, such as Weighted Probabilistic Iterative Matching (WPIM) [4]. They append weight or probability to PIM to allow more flexible bandwidth allocation. The others apply a round robin policy, such as Round Robin Matching (RRM) and Iterative Round Robin Matching with SLIP (iSLIP-RRM) [5], [6], which are two of the most popularly developed round robin scheduling schemes. ISLIP-RRM is an improved version of RRM, which solves its problem of round robin pointer synchronization at outputs under the situation that all traffic flows are backlogged. Both RRM and iSLIP-RRM can achieve an excellent throughput close to 100% with an optimum number of iterations per cell time, which should follow $O(log_2 N)$. Some other round robin scheduling schemes are developed based on RRM and iSLIP-RRM, such as Iterative Round Robin Matching with Multiple Classes (IRRM-MC) [7], Deficit Round Robin (DRR) [8], Weighted Round Robin (WRR) [9].

However, the existing scheduling schemes have some limitation: each cell is destined for one output only, and an output link can only be connected to one output. Therefore, they can not solve a problem which frequently occurs in the practical network environment: an IP router with a large number of outputs (e.g. 64, 128 or more) is connected to only a few nodes (e.g. 2 or 4). One example is a router connecting access network and core network. Considering it is uneconomical to connect multiple outputs to a node via the same number of output links, especially over a long distance, usually one node is connected to only one output via one output link. Under this situation, the switching capacity can not be fully utilized since only the outputs connected to the nodes are engaged in switching.

The aim of our research is to find a solution for this pitfall of those existing scheduling schemes so that the switching capacity can be fully utilized.

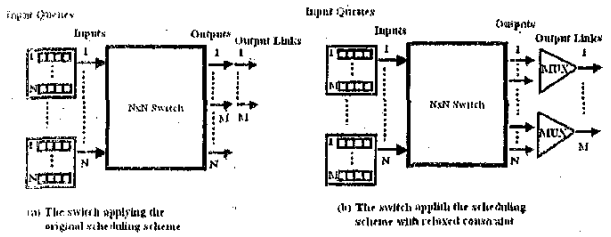
The rest of the paper is organized as follows: section II introduces our solution in detail. Section III demonstrates simulation results. Finally some conclusions and further work are given in section IV.

II. Scheduling with Relaxed Constraint

A. The Basic Idea

To fully utilize the switching capacity, we propose an approach to improve the existing scheduling schemes: we divide all outputs of a crossbar switch into a few groups,

0-7803-8601-9/04/$20.00 © 2004 IEEE.



(a) The switch applying the original scheduling scheme

(b) The switch applying the scheduling scheme with relaxed constraint

Fig. 1. Comparison between Scheduling with and without Relaxed Constraint

whose number is equal to the number of output links, and each link is connected to a node. The outputs in the same group are multiplexed into a high-speed output link. Cells are destined for one output link (or a group of outputs multiplexed into one output link) instead of individual output, which is the reason why the approach is referred to as scheduling with relaxed constraint. Fig. 1 (b) illustrates the idea: all N outputs of an NxN switch are divided into M groups, where M is the number of output links. In this way the unconnected outputs in the original scheme are also engaged in switching. Therefore we can expect a better utilization of the switching capacity. Assuming Virtual Output Queue (VOQ) [10] is employed, a virtual queue for each output group rather than each output is needed in the improved scheduling schemes. Thus the total number of the virtual queues in the improved scheduling schemes is decreased to NxM in comparison with the NxN input queues in the original scheduling schemes even only M outputs are engaged in switching, as shown in Fig. 1 (a).

The analysis in [11] indicates that the decreasing number of input queues can result in less computation complexity. The improvement in computation complexity becomes more significant with larger N and smaller M. When not considering multiple iteration within one cell time, the computation complexity is decreased by at least 60% for a 16x16 crossbar switch with 4 output groups, where each group contains 4 outputs. Therefore we expect that the improved scheduling schemes could achieve a much faster scheduling speed due to the significant decrease of computation complexity.

B. Scheduling with Relaxed Constraint in Approach 1

1) PIMRC 1: PIMRC 1 has three stages in one scheduling iteration, shown as follows:

- Stage 1: Request. Each unmatched input sends a request to all unmatched outputs in a group for which it has at least one queued cell.
- Stage 2: Grant. If an unmatched output receives more than one request, it grants only one request in a uniform random manner.
- Stage 3: Accept. If an unmatched input receives more than one grant, it accepts only one grant in a uniform random manner too. The matched input/output pairs established during this iteration will not join the rest iterations in the same cell time.

2) RRMRC and iSLIP-RRMRC: RRMRC contains three stages in one scheduling iteration, shown as follows:

- Stage 1: Request. The same as in PIMRC 1.
- Stage 2: Grant. The choosing policy applied in the grant stage is a round robin policy. A round robin arbiter is located at each output. The output grants the request from the input that has the highest priority based on the round robin arbiter (modulo N, where N is the number of inputs of the crossbar switch). At the end of stage 2, the pointer of the round robin arbiter is increased to one location exactly beyond the granted input (modulo N).
- Stage 3: Accept. The choosing policy in this stage is a round robin policy too. There is a round robin arbiter at each input. The input accepts the grant from the output that has the highest priority based on the round robin arbiter (modulo M, where M is the number of outputs of the crossbar switch). At the end of stage 3, the pointer of the round robin arbiter is increased to one location exactly beyond the accepted output (modulo M). The matched input/output pairs established during this iteration also will not join the rest iterations in the same cell time slot.

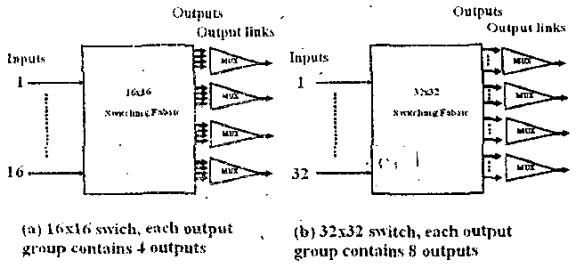
The only difference between RRMRC and iSLIP-RRMRC is that iSLIP-RRMRC will not immediately update the pointers of the round robin arbiters at outputs at the end of stage 2. They will be updated only if their grants are accepted in stage 3.

C. Scheduling with Relaxed Constraint in Approach 2

1) PIMRC 2: Similarly, PIMRC 2 also has three stages in one scheduling iteration, shown as follows:

- Stage 1: Request. Each unmatched input sends a request to the output link for which it has at least one queued cell and the corresponding group still has at least one unmatched output.
- Stage 2: Grant. There are two cases:
  - Case 1: The number of unmatched outputs in the group is greater than or equal to the number of requests received by the group. All requests can be granted.
  - Case 2: The number of unmatched outputs in the group is less than the number of requests received by the group, A limited number of requests, which is equal to the number of unmatched outputs, will be granted in a uniform random manner.
- Stage 3: Accept. It is the same as that in PIMRC 1.

Different from the other scheduling schemes, it is the output group (or the output link) in PIMRC 2, instead of the individual output, that generates the grant in the grant stage. If the grant is accepted in stage 3, one unmatched output in the group will be assigned to transmit the cell. Here we apply a simple method to choose the input/output to set up the matching pair: the input/output with the least numbered identity has the highest priority to set up the matching pair. For example, a group containing unmatched outputs 1, 2, and 3, will set up two matching pairs for inputs 1 and 2. Output 1



(a) 16x16 swich, each output group contains 4 outputs

(b) 32x32 switch, each output group contains 8 outputs

Fig. 2. The switch models used in the simulation

TABLE I

THE OPTIMUM NUMBER OF ITERATIONS PER CELL TIME FOR THE 16X16 SWITCH MODEL (AVERAGE BURST LENGTH=8 CELLS)

Scheduling Algorithms	The optimum number of iterations per cell time
PIMRC 1	3
RRMRC	2
iSLIP-RRMRC	2
PIMRC 2	4
Semi-RRMRC	4
PIM	4
RRM	3
iSLIP-RRM	3

is the least numbered among all three available outputs, and so are input 1 between two inputs, therefore input 1 and output 1 will set up the first matching pair since they own the highest priority.

2) Semi-RRMRC: It is identical to PIMRC 2 except that a round robin policy rather than a random policy is applied in the accept stage of the scheduling iteration in Semi-RRMRC, which is also the reason why it is refereed to as Semi-RRMRC. Meanwhile, the round robin arbiter structure at the inputs is also changed to suit the M output links instead of N outputs, i.e., the "M" in "modulo M" should be equal to the number of output links of the crossbar switch rather than the number of the outputs.

III. Simulation and Results

A. Simulation Environment

Two-state Markov Modulated Bernoulli Process (2-MMBP) traffic model [12]–[14] is applied in the simulation, which is a popularly used traffic model for IP-based bursty traffic. A 16x16 switch with four output groups is simulated, where each group contains four outputs, as shown in Fig. 2 (a). The input queue size is assumed to be infinite. An independent 2-MMBP traffic source is run at each input. The traffic load is increased from 0.1 to 1 with a step of 0.1. Three values are chosen for the average burst length of 2-MMBP traffic source: 8, 16 and 32 cells. For comparison purpose, the simulation for the original PIM, RRM, and iSLIP-RRM is also performed in the same simulation environment. Each simulation case lasts 100,000 cell times, and the Matlab simulation program is run on a $PIII500$ pc with Windows 2000.

To investigate the impact of the larger switch size, the simulation for all PIM-based scheduling schemes (i.e. PIM, PIMRC1 and PIMRC2) is also performed on a 32x32 switch model under a similar simulation environment. The outputs of the 32x32 switch are divided into four groups and each group contains eight output, as shown in Fig. 2 (b).

B. Simulation Results for the 16x16 Switch Model

Since the results for the all three cases are similar, we only demonstrate the simulation results for the case that average bursty length is 8 cells. First, it is necessary to find the optimum number of iterations per cell time for all these scheduling schemes. We consider that when further increase in iteration number per cell time can only results in a negligible increase in throughput, the optimum number of iteration per cell time for the scheduling scheme has been reached. Table I shows the result.

Secondly the throughput performance is investigated, shown as in Table II, where the corresponding maximum throughput with the optimum number of iterations per cell time for each scheduling scheme is presented. It shows that all of them can achieve an excellent throughput (>97%).

Finally we investigate the computation complexity. Due to the statistical nature of the traffic, it is quite difficult to accurately measure the computation complexity. Here we try to measure the computation complexity using the so-called average cell time, which is obtained via dividing the total computation time during each simulation scenario by the total number of simulated cell time (i.e. 100,000 cell times). Its value is an indicator of the scheduling speed, on which the computation complexity has a critical impact. Table III illustrates the average cell time for all simulated scheduling schemes.

It is shown that all the original scheduling schemes, including 4-iteration PIM, 3-iteration RRM and 3-iteration iSLIP-RRM, run much slower than the improved scheduling schemes. When the traffic load increases, the difference appears more obvious. Under a full traffic load, the average cell time of 4-iteration PIM is more than 10 times the average cell time of 2-iteration RRMRC.

C. Simulation Results for the 32x32 Switch Model

Also because the results for all simulation cases are similar, we only demonstrate the simulation results for the case that average burst length is 8 cells .

Table IV illustrates the optimum number of iterations per cell time for PIM, PIMRC 1 and PIMRC 2 on both the 16x16 and the 32x32 switch models. We can see that the optimum number of iterations for both PIM and PIMRC 2 follow $O(log_2 N)$ when the switch size increases, while PIMRC 1 has the same optimum number of iterations per cell time on both switch models.

Table V illustrates the throughput results. It can be observed that the larger switch size doesn't affect their throughput

TABLE II

THROUGHPUT FOR THE 16X16 SWITCH MODEL (AVERAGE BURST LENGTH=8 CELLS)

Traffic load	3-iteration PIMRC 1	2-iteration RRMRC	2-iteration iSLIP-RRMRC	4-iteration PIMRC 2	4-iteration Semi-RRMRC	4-iteration PIM	3-iteration RRM	3-iteration iSLIP-RRM
0.1	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
0.2	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
0.3	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
0.4	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
0.5	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
0.6	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
0.7	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
0.8	100.0%	100.0%	100.0%	100.0%	100.0%	99.9%	100.0%	100.0%
0.9	100.0%	99.4%	99.7%	100.0%	100.0%	99.9%	99.9%	99.8%
1	99.7%	98.0%	98.3%	99.3%	99.3%	98.5%	98.1%	98.1%
	Throughput							

TABLE III

AVERAGE CELL TIME RESULT FOR THE 16X16 SWITCH MODEL (AVERAGE BURST LENGTH=8 CELLS)

Traffic load	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
3-iteration PIMRC 1	1.50	1.60	1.69	1.80	1.91	2.04	2.16	2.31	2.46	2.76
2-iteration RRMRC	1.17	1.28	1.36	1.42	1.50	1.56	1.62	1.68	1.79	1.82
2-iteration iSLIP-RRMRC	1.17	1.25	1.35	1.41	1.49	1.54	1.62	1.71	1.77	1.90
4-iteration PIMRC 2	1.96	2.05	2.15	2.26	2.34	2.46	2.58	2.79	3.01	3.17
4-iteration Semi-RRMRC	1.90	1.94	1.97	1.99	2.02	2.09	2.16	2.28	2.44	2.65
4-iteration PIM	8.41	8.66	8.98	9.36	9.84	10.40	11.17	12.40	14.09	18.68
3-iteration RRM	6.36	6.49	6.63	6.86	7.06	7.47	7.92	8.67	9.89	13.00
3-iteration iSLIP-RRM	6.34	6.45	6.64	6.79	7.07	7.41	8.05	8.67	9.74	13.41
	Average cell time (ms)									

TABLE IV

THE OPTIMUM NUMBER OF ITERATIONS PER CELL TIME FOR THE 16X16 AND 32X32 SWITCH MODELS (AVERAGE BURST LENGTH=8 CELLS)

16x16 switch		32x32 switch	
Scheduling Algorithms	The optimum number of iterations per cell time	Scheduling Algorithms	The optimum number of iterations per cell time
PIM	4	PIM	5
PIMRC 1	3	PIMRC 1	3
PIMRC 2	4	PIMRC 2	5

performance significantly since they all remain at an excellent level (>98%).

Fig. 3 illustrates the average cell time results. It shows that the average cell time is increased significantly while the switch size increases. This is reasonable since the larger switch size can result in a higher computation complexity. Meanwhile, the faster speed of the improved scheduling schemes appears more significant with the increase of the switch size.

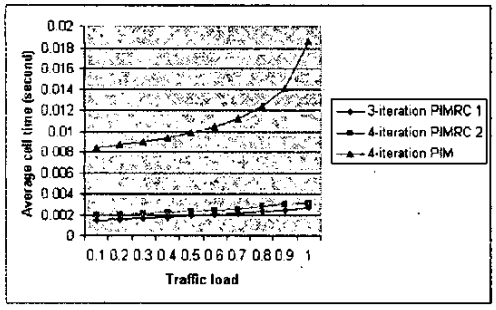
IV. CONCLUSION AND FURTHER WORK

In this paper, an approach referred to as scheduling with relaxed constraint was proposed. Several improved scheduling schemes by applying this approach were also demonstrated, which are based on PIM, RRM, and iSLIP-RRM.

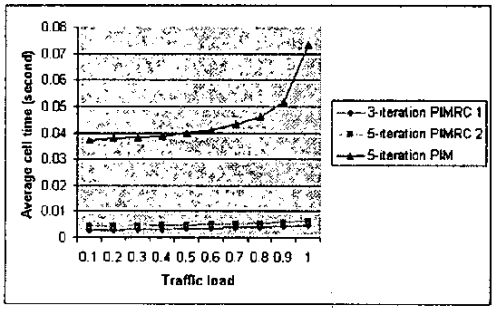
The simulation results on a 16x16 switch model indicated that the improved scheduling schemes achieve an excellent throughput (>97%), while having a much faster scheduling speed than the original. The simulation results for the PIM-based scheduling schemes on a 32x32 switch model indicated the advantage of the faster scheduling speed of the improved scheduling schemes appears more significant.

TABLE V
THROUGHPUT COMPARISON RESULT FOR THE 16X16 AND 32X32 SWITCH MODELS (AVERAGE BURST LENGTH =8)

16x16 switch				
	Traffic load			
Throughput	0.1	0.4	0.7	1
4-iteration PIM	100.00%	100.00%	99.98%	98.47%
3-iteration PIMRC 1	100.00%	100.00%	100.00%	99.69%
4-iteration PIMRC 2	100.00%	100.00%	100.00%	99.25%
32x32 switch				
	Traffic load			
Throughput	0.1	0.4	0.7	1
4-iteration PIM	100.00%	99.99%	99.98%	98.45%
3-iteration PIMRC 1	100.00%	100.00%	100.00%	99.87%
4-iteration PIMRC 2	100.00%	100.00%	100.00%	99.40%



(a) 16x16 switch



(b) 32x32 switch

Fig. 3. Average cell time comparison for the 16x16 and 32x32 switch models (average burst length=8)

However, we have not obtained a clear relationship between the switch size and the optimum number of iterations per cell time for our proposed scheduling schemes. Also the situation that an unequal number of outputs is assigned to each group has not been considered, which is very helpful for flexible bandwidth allocation. Finally research remains to be done on investigating the fairness of our new scheduling schemes. Further research is being carried out in these areas.

REFERENCES

[1] N. Yamanaka, "Next generation internet backbone router," in *ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, 2001. Joint 4th IEEE International Conference on*, 2001, pp. 316–319.

[2] S. Keshav and R. Sharma, "Issues and trends in router design," *Communications Magazine, IEEE*, vol. 36, no. 5, pp. 144–151, 1998.

[3] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High-speed switch scheduling for local-area networks," *ACM Transactions on Computer Systems*, vol. 11, no. 4, pp. 319–352, 1993.

[4] D. Stiliadis and A. Varma, "Providing bandwidth guarantees in an input-buffered crossbar switch," in *INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE*, 1995, pp. 960–968 vol.3.

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

[6] N. McKeown, P. Varaiya, and J. Walrand, "Scheduling cells in an input-queued switch," *Electronics Letters*, vol. 29, no. 25, pp. 2174–2175, 1993.

[7] S. Motoyama, "Cell delay modelling and comparision of iterative scheduling algorithms for atm input-queued switches," in *Communications, IEE Proceedings. Feb. 2003.*, pp. 1407–1411 vol.2 Issues: 1.

[8] G. Nong, M. Hamdi, and K. Letaief, "Efficient scheduling of variable-length ip packets on high-speed switches," in *Global Telecommunications Conference, 1999. GLOBECOM '99*, vol. 2, 1999, pp. 1407–1411 vol.2.

[9] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose atm switch chip," *Selected Areas in Communications, IEEE Journal on*, vol. 9, no. 8, pp. 1265–1279, 1991.

[10] Y. Tamir and G. Frazier, "High-performance multiqueue buffers for vlsi communication switches," in *Computer Architecture, 1988. Conference Proceedings. 15th Annual International Symposium on*, 1988, pp. 343–354.

[11] L. Xiong, "Scheduling in packet switches with relaxed constraint," New South Wales, Australia, Master thesis, to be published within 2004.

[12] C. Ng, L. Bai, and B. Soong, "Modelling multimedia traffic over atm using mmbp," *Communications, IEE Proceedings-*, vol. 144, no. 5, pp. 307–310, 1997.

[13] A. Adas, "Traffic models in broadband networks," *Communications Magazine, IEEE*, vol. 35, no. 7, pp. 82–89, 1997.

[14] W.-C. Miao and J.-F. Chang, "Individual sojourn delay analysis of an atm switch receiving heterogeneous markov-modulated bernoulli processes under fifo and priority service disciplines," *IEICE Transactions on Communications*, vol. E80-B, no. 5, pp. 712–725, 1997.