Weighted Fair Scheduling Algorithm for QoS of Input-Queued Switches

Sang-Ho Lee[1], Dong-Ryeol Shin[2], and Hee Yong Youn[2]

[1] Samsung Electronics, System LSI Division, Korea
sangho74.lee@samsung.com
http://nova.skku.ac.kr
[2] Sungkyunkwan University, Information and Communication Engineering, Korea
{drshin, youn}@ece.skku.ac.kr

Abstract. The high speed network usually deals with two main issues. The first is fast switching to get good throughput. At present, the state-of-the-art switches are employing input queued architecture to get high throughput. The second is providing QoS guarantees for a wide range of applications. This is generally considered in output queued switches. For these two requirements, there have been lots of scheduling mechanisms to support both better throughput and QoS guarantees in high speed switching networks. In this paper, we present a scheduling algorithm for providing QoS guarantees and higher throughput in an input queued switch. The proposed algorithm, called Weighted Fair Matching (WFM), which provides QoS guarantees without output queues, i.e., WFM is a flow based algorithm that achieves asymptotically 100% throughput with no speed up while providing QoS.

Keywords: scheduling algorithm, input queued switch, QoS

1 Introduction

The input queued switch overcomes the scalability problem occurring in the output queued switches. However, it is well known that the input-queued switch with a FIFO queue in each input port suffers the Head-of-Line (HOL) blocking problem which limits the throughput to 58% [1].

Lots of algorithms have been suggested to improve the throughput. In order to overcome the performance reduction due to HOL blocking, most of proposed input queued switches have separate queues called Virtual Output Queue(VOQ) for different output ports at each input port. With VOQs, Input queued switches need matching algorithm to make input-output port pairs.

Parallel Iterative Matching (PIM), which is one of Maximum Size Matching schemes, is a three-phase scheduling algorithm which uses parallelism, randomness and iteration to achieve higher throughput[2]. Some variations of PIM such as iSLIP[3] appeared. iSLIP is very efficient and its throughput can reach 100% but does not address QoS problems.

H. Jin et al. (Eds.): NPC 2004, LNCS 3222, pp. 366–373, 2004.

Another proposed algorithm is RPA which realizes Maximum Weighted Matching (MWM) scheme [4], which is based on reservation rounds where the switching input ports indicate their most urgent data transfer needs. RPA took a similar approach with different scheduling algorithm as a proposed method presented in this paper.

There has been a large amount of works on providing service guarantees in the integrated service networks. Various scheduling algorithms are proposed to provide QoS guarantees. Generalized Processor Sharing (GPS) is considered an ideal scheduling discipline[5]. The GPS is based on a fluid model where the packets are assumed to be infinitely divisible and multiple sessions may transmit traffic through the outgoing link simultaneously. Weighted Fair Queuing (WFQ) is a packetized generalized process sharing[6]. Some variations of WFQ, Self-Clocked Fair Queuing(SCFQ), Virtual-Clock(VC), Deficit Round Robin (DRR) etc. appeared in the literature to address the computational problem of WFQ.

Most of algorithms for QoS provisioning have been done in the context of output queued switch where the speed of the switching fabric and output buffer memory is required to N times the input line speed. As line speeds increase and as routers have more input ports, the required fabric speed becomes infeasible and non-scalable. For these reasons, in addition to the demand for high throughput on routers or switches with input queued architecture, there is an increasing need for supporting applications with diverse performance requirements where QoS is guaranteed.

However there has been a restriction to provide QoS guarantees in an input queued switch: input queued switch is scalable but lead to some packets not being promptly transmitted across switch fabric because enqueued packets can not be isolated, which may lead to violating QoS. Therefore the goal of providing QoS guarantees in the input queued switch is to design a scheduling algorithm which can provide QoS requirements so that queued packets are transmitted across the switch fabric promptly (i.e., throughput maximization).

In this paper, we propose a scheduling algorithm for providing QoS guarantees and high throughput in an input queued switch. The proposed algorithm, called Weighted Fair Matching (WFM), which is a flow based algorithm that provides bandwidth allocation. Like other matching algorithms, it can achieve asymptotically 100% throughput under uniform traffic.

The WFM in input queued switches is unique in a sense that the selection right and corresponding matching mechanism based on virtual finishing time of WFQ is done at the output port where the number of connections to the output ports and the virtual finishing time stamps already computed and transferred by input ports are involved.

This paper is organized as follows. Section 2 gives a basic principle of the proposed scheduling method. Section 3 shows the performance based on simulation. The conclusion is drawn in section 4.

2 Weighted Fair Matching Algorithm

We now propose an algorithm, WFM, which applies Weighted Fair Queueing (WFQ) [6] at the input port switch. This algorithm operates as a scheduler to avoid HOL-blocking and to provide QoS guarantees simultaneously. Like other scheduling algorithms in input queued switch, WFM uses multiple virtual queues at the input port for each output port. In this section, we first describe how to derive a WFQ in the input queued switch and then present a WFM.

2.1 Applying Weighted Fair Queueing

The GPS is an ideal scheme for fluid traffics which are assumed to be infinitely divisible and multiple connections may transmit traffic through the output port simultaneously at different rates. Its packetized version, WFQ scheduling algorithm can be thought of as a way to emulate the hypothetical GPS discipline by a practical packet-by-packet transmission scheme.

For an $N \times N$ output queued switch, the bandwidth of each output port is shared by N flows. In this case, each output port has a WFQ system which is composed of a WFQ server and N queues for N flows. In every slot, each output port's WFQ server selects one among its own queues.

Figure 1 depicts the overall block diagram of WFM in a 2×2 input queued switch. We shall denote the kth input and output ports by I_k and O_k, respectively. Let $F(i,j)$ is the flow which is switched from I_i to O_j.

Applying WFQ to the input queued switch is not much different from the case of the output queued switch. In the input queued switch, the flow, $F(i,j)$, is a backlogged Q_i^j which denotes a virtual output queue for O_j in I_i. Like output queued switches, there are N WFQ systems. Let S_j be a WFQ server in O_j. This server includes a virtual-time for tracking normalized fair service amount. As shown in Fig.1, a WFQ system for O_j is composed of S_j and N VOQs located in the input ports and destined for O_j.

In input ports, all arriving packets are tagged with virtual finishing times computed according to WFQ based on allocated bandwidth. The time-stamp,i $TS_{i,j}^k$, associated with k'th packet of the $F(i,k)$ is calculated as follows:

$$TS_{i,k}^k = max\{v_j(t), TS_{i,j}^{k-1}\} + \frac{L_{i,j}^k}{\omega_{i,j}} \tag{1}$$

where $L_{i,j}^k$ denotes a packet length and $v_j(t)$ the virtual-time of S_j.

2.2 Description of WFM

In the previous subsection, we described WFQ to share each output port in input queued switch. For input queued switches, the main problem is how to match input-output ports to get high throughput. In [14], WFQ is used to make input-output port matching with simple sequential scheduling. But this approach did not show to provide QoS in an input queued switch.

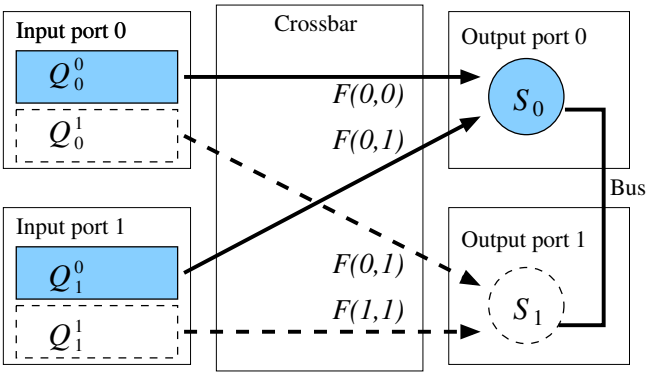

Fig. 1. Weighted fair queueing in an 2×2 input queued switch

We propose a scheduling scheme which operates as not only matching input-output ports but also providing QoS. As shown in figure 1, the switch model for WFM has non-buffered crossbar and its all output ports are connected to a shared medium, called a bus whose main role broadcasts information on input-output port matching to all output ports concerned. Three steps are used to resolve the conflict among input ports using. It is described as follows:

Step 1: Request. Each input port sends a request to every output port for which it has a queued cell. Each request corresponds to nonempty VOQ and includes the time-stamp (i.e., virtual finishing time), $TS_{i,j}^k$, of the cell at the head of the VOQ. All received requests along time-stamps are stored at a request array of each output. The request array of O_j is denoted by R_j and it consists of N elements, denoted by $R_j[i]$ with $1 \leq i \leq N$, which contains corresponding virtual finishing time. In addition, if an output port receives one or more requests, it counts the number of connections destined for itself which is denoted by C_j.

Step 2: Sort and Output Port Selection. Every output ports are sorted based on the number of received requests (i.e., C_j) in an increasing order. It determines turns of which output port is granted the right to select an input port ahead of other output ports. The reason why a sorting operation is performed with an ascending order is that an output port with the smallest number of backlogged flow has less input ports to match so that it is granted to make its selection earlier than the others, which brings a higher probability of matching pairs. For example, if two output ports have a relation with $v'_m(t) > v'_n(t)$, The input-output pair for O_m should be determined earlier than O_n.

Step 3: Input Port Selection and Matching. Once granted to choose input ports, the output port picks up an input port with smallest virtual finishing time expressed by via time-stamps. Furthermore, at most one input port among unmatched input ports is chosen. On selecting an input port, the information about "matched ports" is transferred (or broadcasted) by a common bus to all output ports so that the same input is not permitted to be selected by another

output. This process is repeated until all matchings are done sequentially at the output port.

In short, the selection priority is granted to the output port with smallest number of connections denoted by C_j (via step 2), on the other hand, the matching mechanism by the selected output is done based on the time-stamps of input ports connected (via step 3), which is different from other approaches taken in RPA where a matching is done by input ports.

3 Simulation

We perform simulations to illustrate the capability of fast switching and QoS provisioning of the proposed method. With simulation experiments, we show about switching performance, delay control capability, bandwidth allocation capability, and fairness. Each subsection describes those results.

3.1 Switching Performance

Simulation is performed on a 16×16 switch, where each input has 16 flows for each output port, totally 256 flows. Each flow reserves same bandwidth as each weight. To evaluate switching capability, we measured average delay time and compared it with iSLIP and RPA algorithms. In this simulation iSLIP operate as 4-iteration, because the minimum required number of iteration iSLIP is $log_2 N$ [3]. Concerning the input traffic, we consider two types of models.

1. Uniform traffic : cells arrive with Bernoulli arrival process, the cell output ports are selected with random independently
2. Bursty traffic : cells arrive with on-off arrival process modulated by a two-state Markov chain with destinations uniformly distributed over all output ports

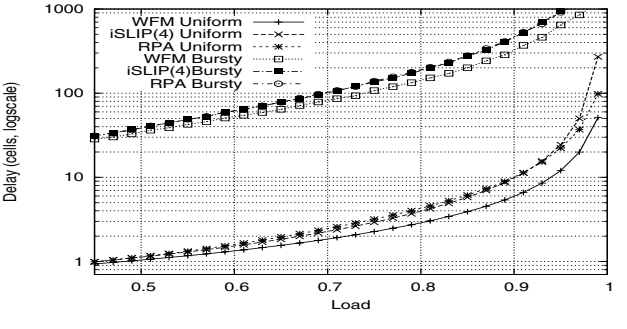

Fig. 2. Average delay under uniform traffic

Fig. 2 shows the curves of the average cell delay normalized with respect to slot time. For the uniform traffic, WFM provides improvement over iSLIP in average delay. At light loads below 60%, All have similar delay time, while for high load above 60%, delay time with WFM are less than half of iSLIP and RPA. For the loads less than 90%, RPA have longest delay time. As the result WFM provide improved delay performance over iSLIP and RPA, which is due to transient delay at a start and priority assignment. Hence, iSLIP is capable of achieving 100% denotes that WFM achieve 100% throughput for uniform traffic.

For the bursty traffic, WFM has also best performance. In this work, the bursty on length is 32.

3.2 Delay Control Capability

Fig. 3 shows the capability of delay control. The simulation is performed under the same situation as in section 3.1, but all flows to each output port have different weights. Each output port has 16 flows, the flows' weights are configured as 1 to 16. We took samples for $F(1,1)$, $F(4,1)$, $F(8,1)$, $F(12,1)$ and $F(16,1)$ at O_1.

The delay control ability of WFM is compared with output queued switches. The result of the output queued switch is shown in Fig. 4. At light loads below 60%, each flow's delay is almost identical that of output queued switch, while for high load above 60%, delay time of all flows are more than those of output queued switch. However, WFM can control each flow's delay.

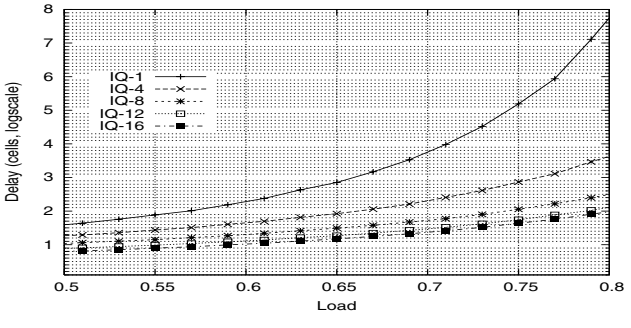

Fig. 3. Delay per flow using WFM

3.3 Throughput with Weighted Fair Bandwidth Allocation

In this subsection, we demonstrate WFM's ability of allocating bandwidth among input ports in proportion to their reservations. The simulation is performed on a 8×8 switch where each input port has one flow, totally 8 flows, destined to O_1. Each flow is assigned the weight as 1 to 8. As shown Fig. 5, the bandwidth

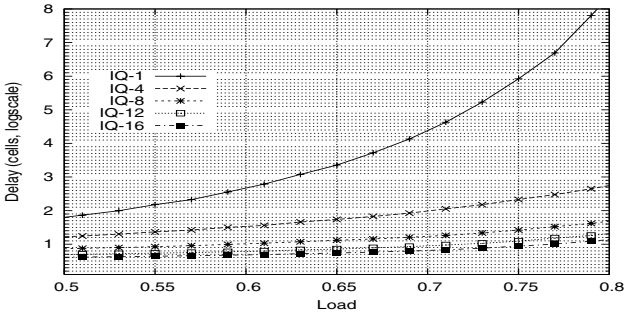

Fig. 4. Delay per flow using WFQ in output queued switch

is distributed in proportion to each flow's weight under uniform traffic. WFM can allocate the switch bandwidth.

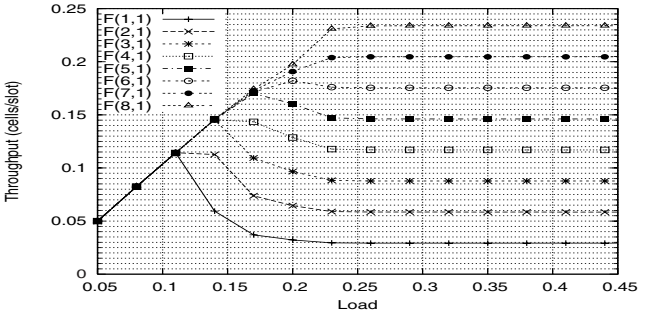

Fig. 5. Throughput per flow under uniform traffic

Fig. 6 presents the result under bursty traffic with busrty length 32. The bandwidth of each flow is also allocated in proportion to it's weight.

4 Conclusion

In this paper we proposed a scheme, called weighted fair matching (WFM) for providing QoS in an input queued switch. We described how to apply a weighted fair queueing of the output queued switch to the input queued switch and proposed a simple matching method. The WFM is a flow based fair scheduling algorithm and operate sequentially. Its main feature is to provide good throughput and to allocate the output bandwidth in a simple manner. We showed that the proposed scheme achieved 100% throughput with low latency and provided QoS guarantees.

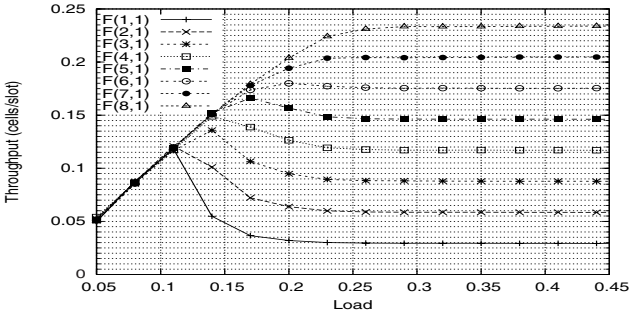

Fig. 6. Throughput per flow under bursty traffic

References

1. M. Karol, M. Hluchyj, S. Morgan : Input versus output queueing on a space division switch, IEEE Transactions on Commumnication, vol.35 (1987) 1347–1356
2. T. Anderson, S. Owicki, I. Saxe, C. Thacker : High speed switch scheduling for local area networks, ACM Transactions on Computer Systems, vol.11 (1993) 1871–1894
3. Mckeon N., Mekkittikul A.: A practcal scheduling algorithm to achieve 100% throughput in input-queued switches, Proceedings of IEEE INFOCOM'98, vol. 2 (1998) 792-799
4. Ajmone Marsan M., Bianco A., Leonardi E. : RPA: a simple efficient and flexible policy for input buffered ATM switches, IEEE Communication Letters, vol.1 (1997) 83-86
5. Parekh A.K. Gallager R.G: A generalized processor sharing approach to flow control in integrated services network, ACM Transactions on Computer Systems, Vol. 11 (1993), 319-352
6. A.Demers, S. Keshav, S. Shenker: Analysis and Simulation of a Fair Queueing Algorithm, Proceedings of SIGCOMM89, (1989) 3-12
7. S. Golestani: A self-clocked fair queueing scheme for broadband applications, Proceedings of IEEE INFOCOM'94, (1994) 636-645.
8. L. Zhang: VirtualClock: a new traffic control algorithm for packet switching networks, ACM Transactions on Computer Systems, Vol. 9, (MAY 1991) 101-124
9. M. Shreedhar, G. Varghese: Efficient fair queueing using deficit round robin, Proceedings of SIGCOMM, (1995).
10. Ge Nong, Mounir Hamdi: On the provision of Quality-of-Service Guarantees for Input Queued Switches, IEEE Communications Magazine, (December 2000) 62-69
11. D. Stiliadis, A.Varma: Providing bandwidth guarantees in an input-buffered crossbar switch, Proceedings of IEEE INFOCOM'95, (1995) 960-968
12. N. Ni, L. N. Bhuyan: Fair scheduling and buffer management in internet routers, Proceedings of IEEE INFOCOM'02, (2002) 1141-1150
13. Xiao Zhang, L. N. Bhuyan: Deficit Round-Robin Scheduling for Input-Queued Switches, IEEE Journal on selected areas in communications, Vol. 21, (MAY 2003) 584-594
14. Sang Ho Lee and Dong Ryeol Shin: "A simple pipelined scheduling for input queued switch", ISCIS 2003 , November 2003, 844-851.