Available online at www.sciencedirect.com
SCIENCE DIRECT
Computer Communications 28 (2005) 1410–1415
computer communications
www.elsevier.com/locate/comcom

Packet-based scheduling algorithm for CIOQ switches with multiple traffic classes

Tsern-Huei Lee*, Ying-Che Kuo

Networking Technology Laboratory, Institute of Communication Engineering, National Chiao Tung University, No. 1001, Ta Hsueh Road, Hsinchu 30050, Taiwan, ROC

Received 14 May 2004; revised 29 December 2004; accepted 10 January 2005
Available online 22 January 2005

Abstract

A packet-based least cushion first/most urgent first (PB-LCF/MUF) maximal matching algorithm is presented in this paper for combined input and output queued (CIOQ) switches with multiple traffic classes. The main benefit of using a CIOQ switch is to alleviate memory bandwidth requirement while providing quality of service (QoS) guarantee. It was proved that, with a speedup factor of 2, a CIOQ switch which adopts the LCF/MUF scheduling algorithm can exactly emulate an output queued (OQ) switch for any service discipline under fixed-length packets assumption. However, in current Internet environment, packets are transported with different lengths. Therefore, it is necessary to modify the LCF/MUF scheduling algorithm for variable-length packet traffic. For ease of implementation, the proposed algorithm calculates approximate cushions and does not perform re-ordering at output ports. We found, via computer simulations, that the performance of a CIOQ switch with a speedup factor of 5 that adopts the proposed single-iteration PB-LCF/MUF algorithm is close to that of an OQ switch under the weighted round robin service discipline for offered traffic load up to 0.9. In addition, the packet departure order can be maintained under the single-iteration algorithm.

Keywords: Combined input and output queued switch; Least cushion first/most urgent first scheduling algorithm; Variable-length packet; Speedup

1. Introduction

In the past decade, many service disciplines had been proposed to provide quality of service (QoS) guarantee in an integrated services network [1–4]. Most of these algorithms were designed to be used at the output ports of an output-queued (OQ) switch. The main problem of OQ switches is that the switch fabric of an $N \times N$ switch must run N times as fast as its line rate in the worst case, where N denotes the number of input/output ports. As such, OQ switches have serious scaling problem because the advancement in memory bandwidth is much slower than the advancement in transmission speed. A parallel packet switch (PPS) architecture which is constructed with multiple identical lower speed OQ switches operating independently and in parallel was proposed to make a large-capacity switch with extremely high line-rates feasible [5–7]. Clearly, the lower speed OQ switch is still a factor to set a limit on system capacity.

To alleviate memory bandwidth requirement, input queuing is widely considered in building a large-capacity switch. However, input-queued (IQ) switches suffer from head-of-line (HOL) blocking which limits the maximum throughput to about 0.586 under uniform traffic assumption [8]. It can be improved to approach 100% throughput if traffic is delivered from input ports to output ports based on maximum matching [9]. Unfortunately, the high computational complexity of currently known algorithms prohibits maximum matching from being used in a high-speed switch. To reduce the complexity, several maximal matching algorithms (e.g. PIM [10], LPF [11], *i*SLIP [12], and FIRM [13]) have been proposed. However, none of these algorithms can provide QoS guarantee. One possible solution to achieve 100% throughput and alleviate dramatic speedup of switch fabric is to use combined input and output

* Corresponding author. Tel.: +886 3 571 2121x54527; fax: +886 3 571 0116.
E-mail address: tlee@banyan.cm.nctu.edu.tw (T.-H. Lee).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.01.002

queued (CIOQ) switch architecture. The key component of a CIOQ switch is an efficient scheduling algorithm for delivering traffic from input ports to output ports. Various scheduling algorithms have been proposed for CIOQ switches that handle fixed-length packets such as ATM cells [14–17]. In particular, it was proved in [17] that the least cushion first/most urgent first (LCF/MUF) algorithm makes a CIOQ switch with a speedup factor of 2 exactly emulate an OQ switch for any arbitrary service discipline.

However, in current Internet environment, packets are transported with different lengths. Therefore, a scheduling algorithm designed for fixed-length packets has to be modified to make it useful for switching Internet packets. There are some algorithms [18,19] derived from PIM that can handle variable-length packets. However, these algorithms were developed for IQ switches and thus are difficult to provide QoS. In [20], a distributed packet fair queueing (D-PFQ) architecture was proposed to handle variable-length packets with advanced quality of service (QoS). Buffers are placed at input ports, output ports, and the crossbar switch fabric. An important advantage of the D-PFQ architecture is that no global synchronization is necessary. Numerical results showed that the D-PFQ architecture provides service that closely approximates an OQ switch employing Fair Queueing with modest speedup. However, the total number of schedulers is equal to N^2+3N for an $N\times N$ switch, meaning that a large number of states have to be maintained which may become the performance bottleneck for a large-capacity switch.

In this paper, we present the packet-based least cushion first/most urgent first (PB-LCF/MUF) algorithm for CIOQ switches and evaluate its performance via computer simulations. Since cushion calculation is complicated, to simplify the implementation of the PB-LCF/MUF algorithm, an approximate cushion is adopted. In order to provide quality of service (QoS) guarantee to different applications, our design is applicable to a network environment where packets are classified into multiple priority classes. Results show that the performance of a CIOQ switch with a speedup factor of 5 which adopts the single-iteration PB-LCF/MUF algorithm is close to that of an OQ switch under the weighted round robin (WRR) service discipline. The single-iteration version not only simplifies the matching procedure but also avoids out-of-order packet delivery from input ports to output ports.

The rest of this paper is organized as follows. In Section 2 we present the CIOQ switch model and in Section 3 we describe the PB-LCF/MUF algorithm in detail. Input traffic model together with simulation results are contained in Section 4. Finally, we draw conclusions in Section 5.

2. Switch model

Consider an $N\times N$ CIOQ switch that is as shown in Fig. 1. Assume that there are K classes of packets. At input port i, there are K queues denoted by $\mathrm{VOQ}_{i,j,k}$, $1\le k\le K$, for storing packets destined to output port j. Consequently, the total number of queues at every input port is $N\times K$. For simplicity, we assume that the link capacities of all input ports and output ports are equal. A new arrival packet is placed at the tail of the appropriate queue depending on its destination and class. A packet becomes eligible to be scheduled for delivery only after the whole packet arrives completely. Although packets are of variable lengths, we assume that every packet can be fragmented into an integral number of smaller units called cells. Time is divided into slots such that the duration of a slot equals the time between cell arrivals at input ports. A CIOQ switch with a speedup factor of S is able to make S cell transfers from each input to outputs during each time slot. In other words, a slot is further divided into S sub-slots for the switch fabric and scheduling is performed at the beginning of each sub-slot. At each output port j, there are K queues, denoted by $\mathrm{OQ}_{j,k}$, $1\le k\le K$, for storing packets. The service discipline studied in this paper is weighted round robin (WRR) because it is a scheme that can be easily implemented and is able to provide QoS guarantee.

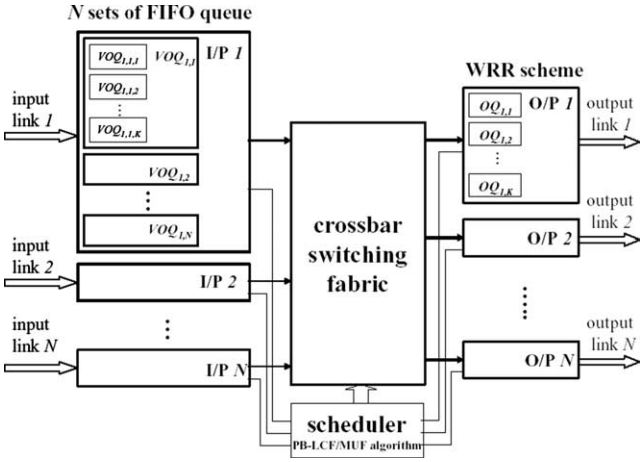

Fig. 1. A CIOQ switch which uses the proposed approximate PB-LCF/MUF matching algorithm and serves packets at output ports with the WRR scheme.

For comparison, an OQ switch is required in computer simulations. In the OQ switch, each output port maintains K queues that are served with the same WRR scheme. Identical input traffics are applied to both the CIOQ and the OQ switches. The OQ switch will be referred to as the shadow OQ switch for convenience.

3. The PB-LCF/MUF algorithm

In the LCF/MUF algorithm which handles fixed-length packets, the cushion $C(x_{i,j})$ of a packet $x_{i,j}$ holding by some input port i that is destined to output port j is defined to be the number of packets currently residing in output port j that will leave the shadow OQ switch earlier than the packet $x_{i,j}$.

The cushion between input port i and output port j, denoted by $C(i,j)$, is the minimum of $C(x_{i,j})$ of all packets buffered at input port i destined to same output port j. If there is no packet destined to output port j, then $C(i,j)$ is set to infinity. In addition, an $N\times N$ scheduling matrix whose (i,j)th entry equals $C(i,j)$ is associated with the LCF/MUF algorithm. The basic operation of the LCF/MUF consists of two steps. In step 1, the (i,j)th entry of the scheduling matrix which satisfies $C(i,j)=\min_{k,l}\{C(k,l)\}$ is selected. If the selected entry is infinity, then stop. If there is more than one entry with the least cushion residing in different columns, then choose the packet with the smallest arrival time among those input ports which correspond to the selected entries. In step 2, eliminate the ith row and the jth column (i.e. match output port j to input port i) of the scheduling matrix. If the reduced matrix becomes null, then stop. Otherwise, use the reduced matrix and go to step 1. It was proved in [17] that, with a speedup factor of 2, a CIOQ switch which adopts the LCF/MUF scheduling algorithm can exactly emulate an OQ switch.

For the PB-LCF/MUF algorithm, the cushion of a packet destined to output port j can be defined as the total transmission time of all packets currently at output port j that will leave the shadow OQ switch earlier than the packet. As mentioned before, an approximate cushion is used to simplify implementation of the PB-LCF/MUF algorithm since the calculation of real cushion is complicated. Our choice of the approximate cushion is simply the number of packets currently queued at the destination output port. For example, the approximate cushion of a packet with class k waiting for scheduling to output port j is equal to the number of packets queued at $OQ_{j,k}$. To eliminate the possibility of interleaving the cells of multiple packets to save the reassembling effort, a packet will be delivered in consecutive sub-slots until it is completely delivered.

3.1. Single-iteration matching

The single-iteration PB-LCF/MUF scheduling algorithm is described below. It consists of three phases.

Phase 1. Input port i sends a request to output port j for all i and j ($1\le i,j\le N$). The request contains the arrival times of the HOL packets of all the K queues. The arrival time is set to infinity if a queue is empty. Note that if input port i is matched to output port j in the previous sub-slot and there are more cells for the same packet, then the request sent by input port i to output port j will be the same as that sent in the previous sub-slot.

Phase 2. Output port j sends a grant message back to input port i if it was matched to input port i in the previous sub-slot and there are more cells of the same packet waiting to be delivered to output port j. Otherwise, it calculates the cushions for the HOL packets contained in the requests and selects the request with the least cushion. If there is a tie, then the packet with the earliest arrival time wins the contention. It then sends a grant message back to the input port which sent the selected request in Phase 1. The cushion is contained in the grant message.

Phase 3. Input port i selects the grant sent by output port j if it was matched to output port j in the previous sub-slot and there are more cells of the same packet waiting to be delivered to output port j. Otherwise, it selects the grant with the least cushion. If there is a tie, then the packet with the earliest arrival time is selected. Assume that the grant message sent by output port k is selected. Input port i sends a cell of the HOL packet to output port k.

It is worth noting that the proposed algorithm causes no cell interleaving of packets and therefore no re-assembly buffer at output ports is needed. Moreover, the departure order of packets can also be maintained at output ports with the single-iteration version. The reason is explained as follows. Consider, for example, the 2×2 CIOQ switch shown in Fig. 2. Assume that input port 1 is matched to output port 1 in the previous sub-slot and there are more cells of packet z waiting to be delivered. Let packets x and y be of the same class (different from the class of packet z) at different input ports. Assume that both x and y are HOL packets and destined for the same output port 2 and x arrived earlier than y. Then, in current matching sub-slot, output port 1 will grant input port 1 for packet z and output port 2 will grant input port 1 for packet x in phase 2. Consequently, packet y will be blocked until packet x has been scheduled. Therefore, packet out-of-order will not happen at any output port.

3.2. Multiple-iteration matching

To improve the throughput performance of the switch fabric, the above matching procedure can be repeated for multiple iterations. After each iteration, an input port stops sending requests once it has sent an accept message to a certain output port. Similarly, an output port stops calculating the cushions and sending a grant message once it has received an accept message. The subsequent iterations will try to match the rest of the input and output ports that remain unmatched in previous iterations.

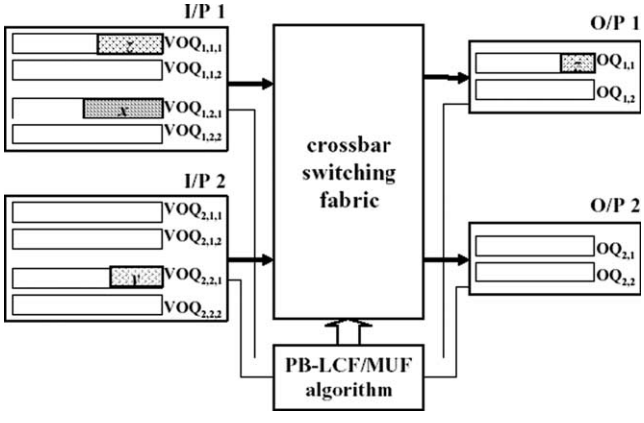

Fig. 2. Example of maintaining packet order in a 2×2 CIOQ switch.

Obviously, the throughput increases as the number of iteration increases. However, multiple iterations may result in out-of-order delivery of packets from input ports to output ports. Moreover, our simulation results show that multiple-iteration matching only yields negligible performance improvement over single-iteration matching if the switch fabric is speeded up by a factor of 2.

4. Numerical experiences

4.1. Input traffic model and performance criterion

To evaluate the performance of the CIOQ switch using the proposed matching algorithm, we measure the latency of the packets which are simultaneously fed to both the shadow OQ switch and the CIOQ switch under a practical input traffic model. The traffic model is characterized by a 2-state Markov Modulated Bernoulli Process (2-MMBP) [21,22] alternating between active and idle states. The traffic source will generate a cell every time slot when it is in the active state. A packet consists of the cells generated by consecutive active states. We assume there is no correlation between different packets and that the destination of each packet is uniformly distributed among the output ports and every packet is equally likely to be any traffic class. According to the real IP networks traffic data collected in [23] from the wide area network, the distribution of packet size has two masses traffic at 40 bytes (about a third) due to TCP acknowledgment packets and 552 bytes (about 22%) due to maximum transmission unit (MTU) limitations at many routers. Other prominent packet sizes include 44 (about 3%), 72 (about 4.1%), 185 (about 2.7%), 576 (about 3.6%), and 1500 (about 1.5%) bytes, due to Ethernet MTU. So, we approximately model the period of active state with distributions of 33% at 40 bytes, 22% at 552 bytes, 30% between 40 and 552 bytes, and 15% between 552 and 1500 bytes. The cumulative distribution function of this traffic model is shown in Fig. 3. Additionally, we convert the IP packet in bytes to the corresponding number of ATM cells using AAL5 null encapsulation technique [24]. As a result, the average packet size is about 8.295 ATM cells.

For performance evaluations, we define $d(x)=|d_{oq}(x)-d_{cioq}(x)|$ as the deviation index of packet x as a criterion. Here $d_{oq}(x)$ and $d_{cioq}(x)$ denote, respectively, the departure times of packet x for the shadow OQ switch and the CIOQ switch. Given a fixed d, we measure the percentage of packets that has a deviation index smaller than or equal to d. This percentage is represented by P_d. For example, $P_{d=0}$ equals 100% means that the CIOQ switch exactly emulates the shadow OQ switch.

4.2. Simulation results

We perform simulations for a CIOQ switch with different switch sizes and speedup factors. Simulations are performed

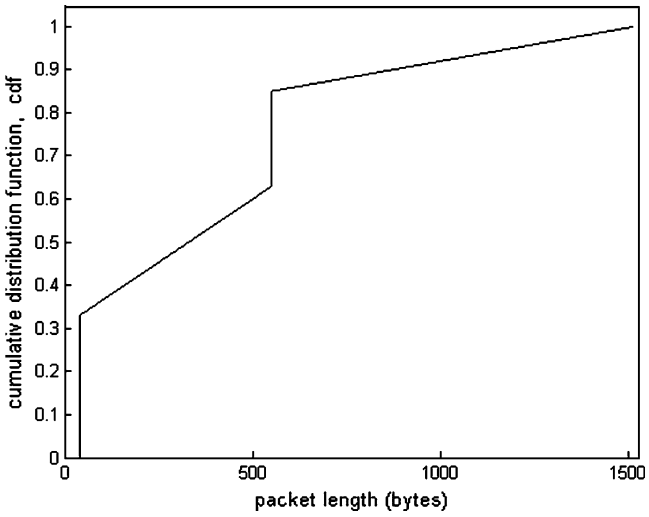

Fig. 3. Cumulative distribution function for active-period traffic of a 2-MMBP input traffic model.

for $N=16$, 32, and 64 with $K=3$ or 8. To save space, we only show the performance curves for $N=32$ and $K=3$. The weights for the three traffic classes are chosen to be $w_1=4$, $w_2=3$, and $w_3=1$. Fig. 4 shows the results for a 32×32 CIOQ switch with a speedup factor of 2 and adopts the proposed single-iteration PB-LCF/MUF algorithm. It can be seen that the performance degrades dramatically as the offered load increases for the class 1 traffic. The value of $P_{d=0}$ is about 42.77% under an offered load of 0.9 which obviously is not acceptable. To avoid drawing too many curves, which would make the figure less comprehensible, the performances of class 2 and class 3 traffics (which are similar to that for class 1 traffic) are not shown in Fig. 4. Instead, we include the performance curves of the CIOQ switch, with a speedup factor of 2, using the FIRM algorithm with packet re-ordering buffers at output ports for comparison. It can be seen that the PB-LCF/MUF

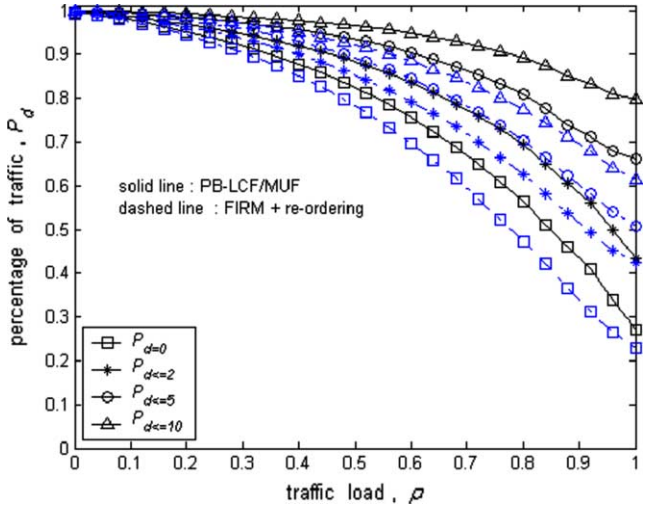

Fig. 4. Performance comparison of the proposed PB-LCF/MUF algorithm and the FIRM algorithm for a 32×32 CIOQ switch.

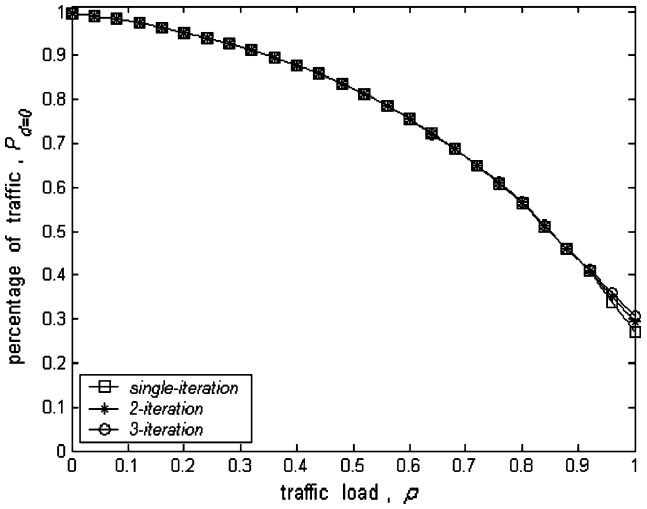

Fig. 5. Performance of the class 1 traffic under multiple-iteration matching procedure with packet re-ordering at output ports.

algorithm performs better. We also performed simulations for the CIOQ switch using the *i*SLIP algorithm. The performance of *i*SLIP algorithm is worse than that of the FIRM algorithm and is not shown in this paper.

As mentioned before, to increase the throughput of the switch fabric, one can perform the matching procedure iteratively to approach a maximal matching. But multiple-iteration will cause out-of-order packet delivery and increase the complexity of the matching procedure. In our experiments, the matching algorithm was performed with one, two, or three iterations and, if needed, packet re-ordering buffers are provided at output ports. The results are shown in Fig. 5 (again, for a speedup factor of 2). As can be seen in Fig. 5, multiple-iteration matching does not result in noticeable performance improvement.

One method to both increase throughput and maintain packet order is to speed up the switch fabric. This method is feasible for moderate speedup factors. We plot in Fig. 6

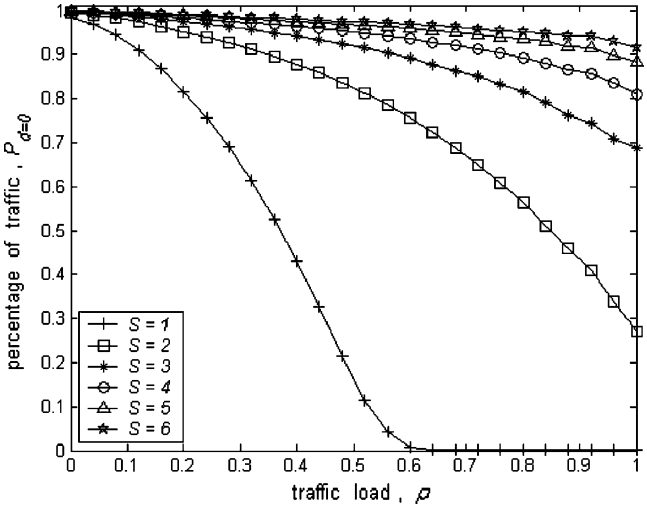

Fig. 6. Performance of the class 1 traffic as a function of offered load for various speed-up factors.

the performance curves for class 1 traffic with speedup factor $S=1$–6. It can be seen that, for $S=1$ which means no speedup, the system performance degrades seriously as the traffic load increases. The reason is that the single-iteration approximate PB-LCF/MUF algorithm is not a maximum matching. In fact, it is not even a maximal matching. However, the total number of matches achieved in a time slot increases (and hence system performance improves) dramatically with speedup. The value of $P_{d=0}$ is larger than 90% under traffic load up to 0.9 when the speedup factor is increased to 5. As mentioned before, we also performed simulations for $N=16$ and 64 and the results show that a speedup factor of 5 still yields satisfactory performance. Similar results were obtained for eight traffic classes with various weights. Therefore, we conclude that the CIOQ switch with a speedup factor of 5 which adopts the proposed single-iteration approximate PB-LCF/MUF algorithm can closely emulate an OQ switch.

5. Conclusions

Since the memory bandwidth sets a limit on switch capacity, the CIOQ switch is likely to be more suitable than the OQ switch in building a large-capacity switch while providing QoS guarantee. In this paper, we have proposed a packet-based least cushion first/most urgent first scheduling algorithm for CIOQ switches that handle variable-length packets in current Internet environment. Since the proposed algorithm requires only one iteration, the packet re-ordering problem is eliminated. Numerical results obtained from computer simulations show that our proposed algorithm yields satisfactory performance when the speedup factor is 5 for various switch sizes and traffic classes. We believe that, with a modest speedup, a CIOQ switch can exactly emulate an OQ switch even for variable-length packets. However, we were not able to determine a necessary and sufficient speedup number for exact emulation. Rigorous mathematical proof of exact emulation obviously is an interesting further research topic. Possible realization of the proposed PB-LCF/MUF algorithm is another one that can be further studied.

References

[1] H. Zhang, Service disciplines for guaranteed performance service in packet-switching networks, Proceedings of IEEE 83 (10) (1995) 1374–1396.

[2] D. Ferrari, D.C. Verma, A scheme for real-time channel establishment in wide-area networks, IEEE Journal on Selected Areas in Communications 8 (1990) 368–379.

[3] A.K. Parekh, R.G. Gallager, A generalized processing sharing approach to flow control in integrated services networks: the single node case, IEEE Transactions on Communications 1 (1993) 344–357.

[4] J.C.R. Bennett, H. Zhang, WF^2Q: worst-case fair weighted fair queueing, Proceedings of IEEE INFOCOM 1996; 120–128.

[5] S. Iyer, A. Awadallah, N. McKeown, Analysis of a packet switch with memories running slower than the line-rate, Proceedings of IEEE INFOCOM 2000; 529–537.
[6] S. Iyer, N. McKeown, Making parallel switches practical, Proceedings of IEEE INFOCOM 2001; 1680–1687.
[7] D. Khotimsky, S. Krishnan, Stability analysis of a parallel packet switch with bufferless input demultiplexors, Proceedings of ICC 2001; 100–111.
[8] M. Karol, M. Hluchyj, S. Morgan, Input verse output queueing on a space division switch, IEEE Transactions on Communications 35 (1987) 1347–1763.
[9] N. McKeown, V. Anantharam, J. Walrand, Achieving 100% throughput in an input-queued switch, Proceedings of IEEE INFOCOM 1996; 296–302.
[10] T.E. Anderson, S.S. Owicki, J.B. Saxe, C.P. Thacker, High speed switch scheduling for local area networks, IEEE/ACM Transactions on Computer Systems 11 (1993) 319–352.
[11] A. Mekkittikul, N. McKeown, A practical scheduling algorithm to achieve 100% throughput in input-queued switches, Proceedings of IEEE INFOCOM 1998; 792–799.
[12] N. McKeown, The *i*SLIP scheduling algorithms for input-queued switches, IEEE/ACM Transactions on Networking 7 (1999) 188–201.
[13] D.N. Serpanos, P.I. Antoniadis, FIRM: a class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues, Proceedings of INFOCOM 2000; 548–555.
[14] P. Krishan, N.S. Patel, A. Charny, R.J. Simcoe, On the speed-up required for work-conserving crossbar switches, Proceedings of IWQoS, Napa, California, USA 1998; 225–234.
[15] I. Stoica, H. Zhang, Exact emulation of an output queueing switch by a combined input output queueing switch, Proceedings of IWQoS, Napa, California, USA 1998; 218–224.
[16] S.T. Chuang, A. Goel, N. McKeown, B. Prabhakar, Matching output qeueueing with a combined input/output-queued switch, IEEE Journal on Selected Areas in Communications 17 (1999) 1030–1039.
[17] T.H. Lee, Y.W. Kuo, J.C. Huang, Quality of service guarantee in a combined input output queued switch, IEICE Transactions on Communications E83-B (2000) 190–195.
[18] G. Nong, M. Hamdi, Burst-based scheduling algorithms for non-blocking ATM switches with multiple input queues, IEEE Communications Letters 4 (6) (2000) 202–204.
[19] S.H. Moon, D.K. Sung, High-performance variable-length packet scheduling algorithm for IP traffic, Proceedings of IEEE GLOBECOM 2001; 2666–2670.
[20] C.S. Donpaul, H. Zhang, Implementing distributed packet fair queueing in a scalable switch architecture, Proceedings of IEEE INFOCOM 1998; 282–290.
[21] S.Q. Li, Performance of a non-blocking space-division packet switch with correlated input traffic, Proceedings of IEEE GLOBECOM 1989; 1754–1763.
[22] S.C. Liew, Performance of various input-buffered and output-buffered ATM switch design principles under bursty traffic: simulation study, IEEE Transactions on Communications 42 (1994) 1371–1379.
[23] http://www.nlnar.net/NA/Learn/packetsizes.html
[24] J. Heinanen, Multiprotocol encapsulation over ATM adaptation layer 5, RFC 1483, July 1993.