Concatenating Packets in Variable-Length Input-Queued Packet Switches with Cell-Based and Packet-Based Scheduling

Nabeel A. Al-Saber, Saurabh Oberoi, Roberto Rojas-Cessa, and Sotirios G. Ziavras

***Abstract*—Internet data is transmitted in variable-size packets to provide flexibility for different applications. There has been a growing interest in developing the design of Internet routers based variable-lenght packets to improve performance and reduce the amount of re-assembly memory. However, most of variable-length designs follow a time-slotted approach, which make them similar to routers that switch fixed-length packets. The use of slotted timing makes padding necessary when packet sizes are not proportional to the time-slot length. In this paper, we investigate the impact of concatenating packets to reduce the amount of padding in variable-length packet switches. This approach increases the utilization of interconnection bandwidth and overall throughput performance. Performance evaluation in an input-queued packet switch using packet concatenation is presented.**

***Index Terms*—Packet scheduling, input-queued switch, variable length packet, cell switching, packet-based scheduling, packet concatenation.**

I. Introduction

Packet switching has the goal to achieve high utilization of network resources (e.g., links) and therefore, to reduce communication costs, increase robustness for different network conditions, and improve communications reliability [1], [2]. Considering that different applications produce different characteristics on user traffic, the Internet has opted the use of variable-length packets.

High-performance packet switching based on fixed-length and variable-length packet scheduling have been both studied for providing high-performance switching. The design of many of Internet switches and routers are based architectures that process fixed-length packets (sometimes called cells) for internal switching to simplify their implementation and to make them capable of operating at very high speeds. Some other designs of Internet routers have followed similar architectures of those used in asynchronous transfer mode (ATM) switches as the the demand for higher data rates in the Internet increased to process packets rapidly. These switches use then fixed-lengh packet internally and re-assemble the original variable-lenght packets before departure. Using the fixed-size packets inside the switch simplifies the implementation of the matching scheme but the segmentation that packets undergo after arriving at the inputs and the re-assembling of packets at the outputs consume time and memory. On the other hand, using variable-lenght packets increase the implementation complexity of a switch but reassembly is not needed.

This work was partially supported by National Science Foundation under Grant Award 0435250.

The authors are with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102. Roberto Rojas-Cessa is the corresponding author. Email: rojas@njit.edu.

The design of packet switches is determined by the used interconnection fabric, queuing strategy, and packet selection policy to remove the inherent output contention. Here, we only consider switches based on crossbar switch fabrics. A crossbar-based switch may use different queueing strategy to define the forwarding mechanism. Output-queued (OQ) switches, which have buffers at the output ports, have been considered to provide the benchmark performance as this switch delivers the highest achievable throughput, lowest queuing delay, and bounded switching delay under shaped traffic that can be used for applications that require quality-of-service (QoS) guarantees. In OQ switches, packets that arrive at the inputs are forwarded to the outputs immediately. However, this architecture requires memory and an interconnection network be able to run N times faster than the line rates, or with a speedup of N, where N is the number of ports. This speedup limits the switch size to a small number of ports and makes the managing of high data rates difficult as memory continues to fall behind the increasing data rates of transmission lines.

Several other buffering architectures have been proposed to provide comparable performance to an OQ switch. Input-queued (IQ) switches have queues at the inputs and they have to resolve output contention before packets are dispatched through the interconnection network. To resolve output contention, these switches perform matching among input and output ports. Therefore, the switch performance depends on the matching scheme used. While matching is required in these switches, they have the property that the memory is required to run not faster than line speed as these memories are placed at the inputs. This condition is, however valid, as long as the matching scheme is efficient enough to provide high throughput.

IQ switch architectures have been adopted by several manufacturers of switches and routers. The introduction of virtual output queues (VOQs), where one queue per output port is placed in an input port of an IQ packet switch, is used to remove the head-of-line (HOL) blocking problem [3]. HOL blocking causes idle outputs to remain so, even in the existence of traffic for them at an idle input, thus impeding the delivery of high throughput.

Maximum weight matching (MWM) schemes have been used to show that IQ switches with VOQs can provide 100% throughput under admissible traffic [4] while using no speedup. However, MWM schemes have intrinsically high computation complexity that is translated into long resolution time and high hardware complexity. This makes these schemes prohibitively expensive for a practical implementation with currently available technologies. An alternative is to use maximal-weight matching schemes. These schemes can provide high throughput performance under uniform and nonuniform traffic patterns with, however, a large number of iterations. The hardware and time complexity of these schemes can be considered high for the ever increasing data rates because of the large number of iterations, and because of the number of parameters needed to be compared in the selection process. Furthermore, some weight-based schemes may starve queues with little traffic to provide more service to the congested ones, therefore, presenting unfairness [5].

Maximal-size matching can be used to resolve contention in cell-based IQ switches [6] in a fast manner. Schemes based in round-robin matching, such as iSLIP [7], iDRRM [10], [11], and SRR [8] are examples of such schemes and these can deliver 100% throughput under uniform traffic with a single iteration. iSLIP showed that the desynchronization effect, where arbiters reach the point where each of them prefers to match with different input/outputs, is beneficial for switching under this traffic pattern. However, it is difficult for schemes based on round-robin selections to provide 100% throughput under nonuniform traffic patterns without speedup or load-balancing stages [9] that can be tailored for traffic with pre-known distributions. The exhaustive dual round-robin matching (EDRRM) scheme [12] has shown a throughput higher than iSLIP under nonuniform traffic patterns. Another alternative for cell-based switches is to perform matching for a batch of cells, instead of for a single cell. Matching in train-of-cells basis have been shown to improve throughput under optimal train sizes for each different traffic scenarios [13]. However, it is not clear how to select the batch size.

A class of matching algorithms based on randomized selection has been proposed to achieve high throughput [14], [15] with low computation complexity. These randomized schemes keep the matches that are likely to continue sending cells. These schemes are ALGO3, APSARA, LAURA, and SERENA. ALGO3 uses a Hamiltonian walk to achieve stability. The other three use the randomized weight augmentation, a merging procedure, and randomness and the information provided by recent arrivals to achieve 100% throughput under admissible traffic. These randomized matching algorithms require computations and comparisons of the sum of the weight for each matching VOQ-output pair in every time slot. The resulting matching schemes are a combination of weight-based and weightless approaches where randomness may play a role in their delivered performance. In [16], [17] has been shown that holding the matches of busy flows for longer time is beneficial to achieve higher switching performance than using a complex matching policy. On the other hand, the performance of cell- and packet-based switching modes have been shown with some performance differences [18]-[20].

In this paper, we consider both cell-based and packet-based scheduling for comparison. In cell-based scheduling, a packet is divided into fixed-size segments. A cell is then equivalent to a segment in cell-based switching. Each segment is scheduled for dispatching to its destined output independently of both the transmission status of the packet and of the packet length. In packet-based scheduling, the scheduling in a packet mode, the segments of a packet are transmitted consecutively after the first segment of the packet is selected by the matching scheme and transmitted to the output port. However, packets may not have a length exactly proportional to the length of a segment, then padding may be used, independently of the scheduling mode, and that decreases the effective bandwidth utilization in a packet switch.

Here, we propose to use packet concatenation to reduce the amount of padding. With this approach, variable-size packets can be concatenated to make a longer internal packets such that only the last segment of the last concatenated packet is padded. While padding is reduced, therefore increasing the effective utilization of link bandwidth, the throughput of a switch may be affected.

We study switching performance in both cell- and packet-based scheduling under variable length packet traffic to compare these two scheduling modes and to observe the effect of using packet concatenation. We propose simple modifications to an existing switching scheme, iSLIP [7] for handling of variable-length packets. We discuss the inherent bandwidth wastage due to the padding and study the efficiency obtained with the proposed approach.

The remainder of this paper is organized as follows. Section II describes the model of the packet switch considered in this paper. Section III describes the scheduling scheme and the padding strategy. Section IV presents the obtained results. Section V presents our conclusions.

II. Switch Model

This paper discusses cell- and packet-based scheduling in an IQ switch. We consider a single-stage $N \times N$ switch, with VOQs in the inputs. The switch is based on a crossbar fabric. A VOQ at input i stores packets for output j, and it is denoted as $VOQ(i,j)$, where $0 \leq i,j \leq N-1$. Unless otherwise stated, we consider that a VOQ can store a large number of cells. In this switch, each input can dispatch one cell each time slot and each output can receive up to one cell per time slot (i.e., no speedup is used). The switch works in a time slotted fashion. The time slot duration is the time a segment of data is transmitted from an input to an output. The segment size can be an arbitrary number of bytes long, however, equal to or larger than the smallest IP packet length (i.e., 20 bytes). The time to transmit a segment equals the duration of a time slot. A time slotted switch allows to adopt switching in cell- or packet-based switching modes. We call cell-based switching to the consideration of receiving IP packets at the inputs, where the variable-length packets are divided into segments of fixed size, called cells, which are scheduled to traverse the switch as they are selected by the scheduling scheme. A cell based scheduling scheme makes granting decisions each time slot. On the other hand, a time slotted switch using packet-based scheduling also transmits segments each time slot, however, the segments belonging to a single packet are selected as a group, such that scheduling is done in a packet-based fashion.

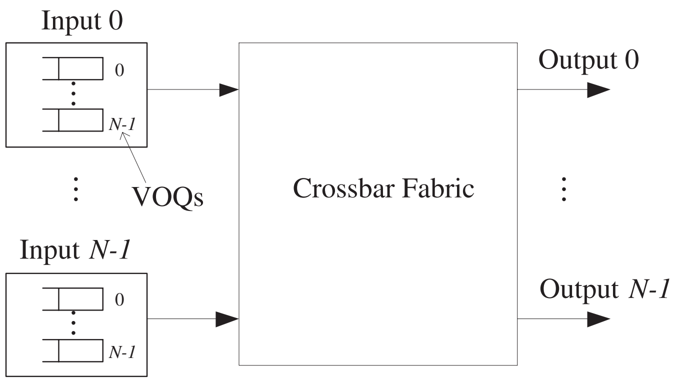

Fig. 1. Model of an IQ switch with VOQs.

The scheduling scheme resolves output contention by selecting segments in the VOQs, one per input and one per output. After the segments have been selected, the segments are transmitted in the next time slot.

The way the switch works is as follows. Variable-length packets arrive at the inputs and are stored in the VOQs according to their destinations. Packets are divided into fixed-length segments for internal switching. Those incomplete segments are padded to complete the segment length. Segments wait to be selected by the scheduling scheme. After being selected, segments are forwarded to their destined outputs. The way segments are selected is described in the following section.

III. Scheduling Schemes

To observe the effect of concatenating two or more variable-length packets, we use iSLIP [7] as a matching scheme. iSLIP does not consider the packet properties in its selection policy. The selection policy in iSLIP is round-robin based. However, this matching scheme is executed among those inputs and outputs who are not transmitting a packet. A packet is considered to be in transmission state if the first cell of this packet has been matched. A packet is considered eligible for matching if the last cell of the packet has been matched. An input port is considered eligible for matching if all of the VOQs are eligible for matching. The matching scheme is then executed on all eligible inputs and outputs. Each input has a counter and a threshold value. The threshold value indicates the number of packets that can be concatenated back-to-back. The counter keeps track of the number of packets concatenated. We limit the number of concatenated packets to avoid input starvation and unfairness, and at the same time, to observe the impact of this approach. After a number of packets equals the threshold value, packet concatenation stops, even if there are more backlogged packets. The last concatenated packet is padded as needed. If there are more packets available at the input queues, these start transmission after the next matching opportunity (and therefore, after they are granted forwarding by the scheduler executing the matching scheme).

A. Concatenation of Packets

Packets are concatenated in the input queues if there is one or more packets at the input queue. To avoid very long concatenated packets that could hold the transmission between an input-output pair, the input limits the maximum number of concatenated packets to c. A packet can be scheduled and dispatched if it is not concatenated and the number of concatenated packets is smaller than c_n. Figure 2 shows an example of the concatenation process in a time-slotted switch, with a time slot using 64-byte segments. Each row shows the packets of $VOQ(1,0)$ and $VOQ(1,1)$ as they would be forwarded to the destined outputs. Figure 2.a shows the transmission of packets without using concatenation. In this case, $VOQ(1,0)$ has two packets to dispatch, one with a length of 80 bytes and the other with 150 bytes, while $VOQ(1,1)$ has also two packets, one with 135 bytes and another with 90 bytes. As shown in this figure, the packets from $VOQ(1,0)$ and $VOQ(1,1)$ would be transmitted in 5 time slots and need 90 and 95 bytes for padding, respectively. Figure 2.b shows the transmission of packets using concatenation. In this case, $VOQ(1,0)$ and $VOQ(1,1)$ would transmit those packets in four time slots and would need 26 bytes for padding for both transmissions. Therefore, this example shows that the switch would work more efficiently using packet-based scheduling.

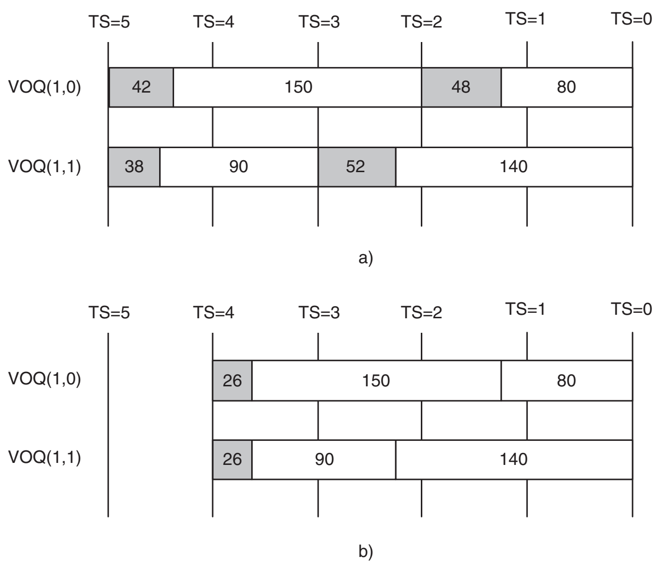

Fig. 2. Example of packet concatenation with variable-length packets.

IV. Performance Study

An event-driven simulation program, describing an IQ switch in iSLIP as a matching scheme for variable-length packets, is used to model the proposed scheme for a 32×32 IQ switch. All used traffic models are admissible (i.e., no overbooking of the link and switch capacities). Variable-length packets are created with Bernoulli arrivals and the destinations are uniformly distributed. Packets are either handled with cell- and packet-based schedulings. Packet lengths are created with a uniform distribution between 40 and 1,500 bytes per packet, unless otherwise stated.

Figure 3 shows the simulation of the IQ switch under variable-length packet. Concatenation of packets is considered for c_n values of 1 to 11 packets. Therefore, the maximum packet size is 16,500 bytes. The scheduling scheme used is 1SLIP working in two scheduling modes: 1) cell-based and 2) packet-based scheduling. The time is slotted in these two modes, where each time slot is used to transmit segments with lengths of 64 bytes. Therefore, in the cell-based scheduling, a cell is also 64 bytes long. This figure shows that under cell-based scheduling, the throughput is the highest when there is no packet concatenation or when packets have the smallest size (40 bytes). In general, it is observed that the throughput under this scheduling mode is achieved with small packet lengths and the throughput decreases as the packet length increases. In this case, the throughput goes slightly under 90% for the longest tested length of concatenated packets. This throughput degradation occurs under cell-mode scheduling as the cell size is fixed and the scheduling scheme makes decisions independently of the packet size. This results in long queuing delays for long packets as packets are segmented into cells and each cell waits for transmission independently from the transmission of the cell ahead or the transmission of the other cells of the same packet.

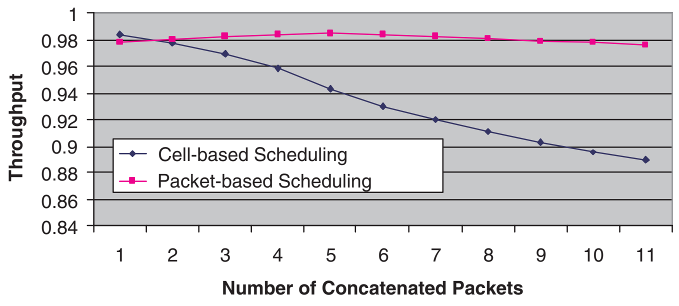

Fig. 3. Throughput of an IQ switch using packet concatenation in both cell- and packet-based scheduling.

In packet-based scheduling, the throughput remains high and close to 98% for small packet lengths. We observe that the throughput increases slightly when there are four concatenated packets, and it decreases to slightly below 98% for large packet lengths. From this graph, we observe that packet-based and cell-based scheduling provide comparable performance for short packets, and the packet-based scheduling outperforms cell-based scheduling for long packets. When the packets have small size (as the size approaches to the length of the the internal segments), cell-based scheduling seems to be optimum. This observation is consistent with those reported in the literature of cell-based switching. However, as the packet-lenght increases, the performance is below acceptable levels.

In this experiment, we also counted the number of padding bytes used in packet concatenation. Here, we compare cell-based scheduling without packet concatenation and packet-based scheduling with packet concatenation. Figure 4 shows the number of padding bytes for both cell- and packet-based scheduling. The number of padding bytes in cell-based scheduling is large for any packet length (and number of concatenated packets). However, the number of padding bytes decreases as the packet length increases with packet-based scheduling.

Following the experiment above, Figure 5 shows the band-

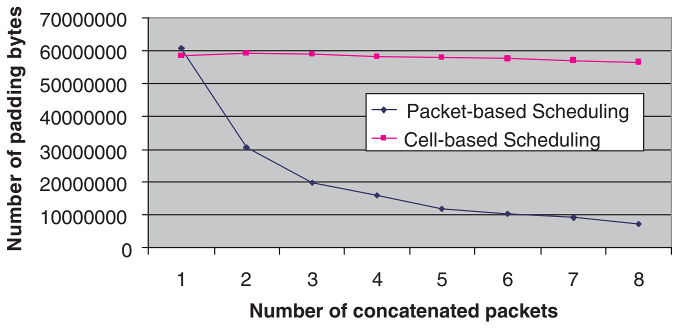

Fig. 4. Padding of packets for cell- and packet-based scheduling.

width wastage ratio, defined as the number of cumulative number of padding bytes divided by the total number of forwarded bytes. The bandwidth wastage ratio is proportional to the number of padding bytes. This ratio can be used to calculate the speedup needed in a switch for transmitting packets combined with padding.

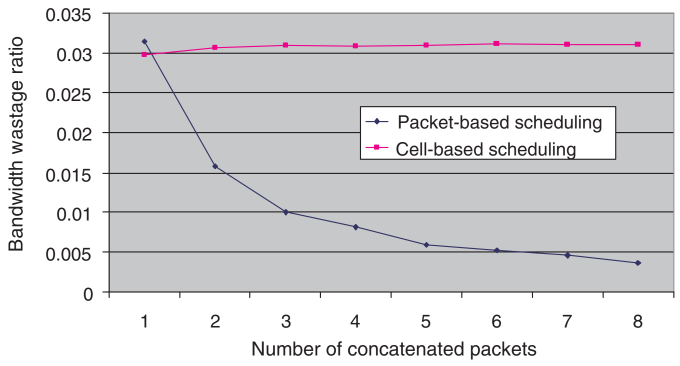

Fig. 5. Wastage of an IQ switch using packet concatenation.

To study the reasons for the throughput increase observed in the packet-based scheduling, we simulated a switch using different segment sizes. The difference in the segment sizes defines different time slot durations. Figure 6 shows the throughput of packet-based scheduling using different slot sizes. This figure shows that as the segment size increases, the throughput increases slightly for large (concatenated) packet sizes. However, when the segment size is large, the throughput of small packets decreases as a large number of padding bytes is needed. However, the minimum throughput with long segment sizes is higher than 94%.

Figure 7 shows the throughput of the IQ switch using cell-based scheduling with different cell (i.e., segment) sizes. The figure shows that the throughput is low when the size of the concatenated packet increases. The throughput of this switch can be increased by increasing the segment size. In this experiment, the segment size is increased at lower rate than the concatenated packet size, therefore the throughput is not kept high. However, it is clear that if the cell size is increased, the throughput can be kept high as the concatenated packet size increases.

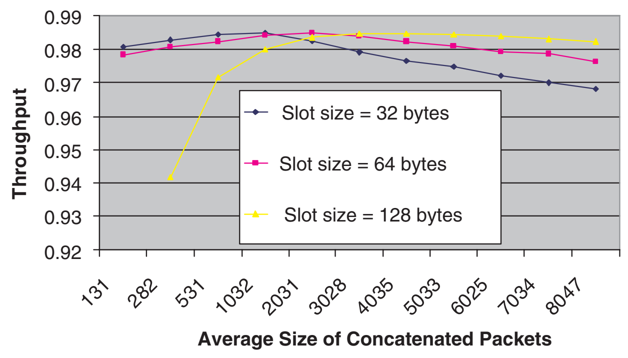

Fig. 6. Throughput of packet-mode scheduling for different slot sizes.

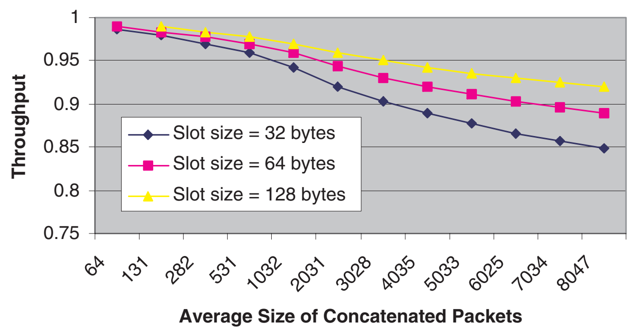

Fig. 7. Throughput of cell-mode scheduling for different slot sizes.

V. Conclusions

We proposed the use of concatenation of two or more packets for internal switching to increase the utilization of link bandwidth. This approach effectively increases the bandwidth utilization and also increases the throughput performance of a switch, as shown in the experimentation presented in an IQ switch using a pre-existing matching scheme. The input-queued switch under study used iSLIP for cell-based scheduling and a small variation of the iSLIP scheduling scheme for packet-based scheduling in the transmission of variable length packets. We compared the throughput performance of cell-based and packet-based scheduling schemes, and observed that packet-based scheduling outperforms cell-based scheduling for an average packet length larger than a segment size, where a segment is the number of bytes processed internally in a time slot. However, for small packet lengths, with a size close to the segment size, cell-based and packet-based scheduling modes have equivalent performance. These results indicate that the performance of cell-based schemes may need to consider the packet length if the actual traffic is to conform the specifications of Internet packets. As the segment size is increased, the throughput of both scheduling scheme increases. The advantage to packet-based scheduling is that the switch performance is more insensitive to the segment size as switching is done by considering the segments that belong to a packet. We observed that the segment size needs to be considered in the study of switch performance under variable-length packets to achieve the best performance.

References

[1] Y. Ganjali, A. Kesharvazian, and D. Shah, "Cell Switching versus Packet Switching in Input-Queued Switches," *IEEE/ACM Trans. on Networking*, Vol. 13, Issue 4, pp. 782-789, August 2005.

[2] M. A. Marsan, A. Bianco, P. Giaccone, E. Leonardi, F. Neri, "Packet Scheduling in Input-Queued Cell-Based Switches," Proc. INFOCOM 2001, Vol. 2, pp. 1085-1094, 2001.

[3] M. Karol, M. Hluchyj, "Queuing in High-performance Packet-switching," *IEEE J. Select. Area Commun.*, vol. 6, pp. 1587-1597, December 1988.

[4] N. McKeown, A. Mekkittikul, V. Anantharam, J. Walrand, "Achieving 100% Throughput in an Input-queued Switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260-1267, August 1999.

[5] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California at Berkeley, Berkeley, CA, 1995.

[6] T.E. Anderson, S.S. Owicki, J.B. Saxe, and C.P. Tacker, "High-speed Switch Scheduling for Local Area Networks," ACM Trans. on Computer Systems, vol. 11, no. 4, pp. 319-352, November 1993.

[7] N. McKeown, "The iSLIP scheduling algorithm for Input-queued Switches," IEEE/ACM Trans. Networking, vol. 7, no. 4, pp. 188-201, April 1999.

[8] Y. Jiang and M. Hamdi, "A fully Desynchronized Round-robin Matching Scheduler for a VOQ Packet Switch Architecture," IEEE HPSR 2001, pp. 407-411, May 2001.

[9] C-S. Chang, D-S. Lee, and Y-S. Jou, "Load Balanced Birkhoff-von Newman Switches," IEEE HPSR 2001, pp.276-280, April 2001.

[10] H.J. Chao, J-S. Park, "Centralized Contention Resolution Schemes for a large-capacity Optical ATM Switch," IEEE ATM Workshop 1998, pp. 11-16, May 1998.

[11] E. Oki, R. Rojas-Cessa, and H. J. Chao, "PMM: A Pipelined Maximal-Sized Matching Scheduling Approach for Input-Buffered Switches," IEEE Globecom 2001, pp. 35-39, Nov. 2001.

[12] Y. Li, S. Panwar, H.J. Chao, "The Dual Round-robin Matching Switch with Exhaustive Service," IEEE HPSR 2002, pp. 58-63, 2002.

[13] A. Bianco, M. Franceschinis, S. Ghisolfi, A.M. Hill, E. Leonardi, F. Neri, R. Webb, "Frame-based Matching Algorithms for Input-queued Switches," IEEE HPSR 2002, pp. 69-76, 2002.

[14] L. Tassiulas, Linear complexity algorithms for maximum throughput in radio networks and input queued switches, IEEE INFOCOM 2 (1998) pp. 533-539, 1998.

[15] P. Giaccone, B. Prabhakar, D. Shah, Randomized scheduling algorithms for high-aggregate bandwidth switches, IEEE J. Sel. Area Commun. 21 (2003), pp. 546-559, 2003.

[16] R. Rojas-Cessa and C-B. Lin, "Captured-Frame Eligibility and Round-robin Matching for Input-queue Packet Switches," *IEEE Commun. Letters*, vol.8, issue 9, pp. 585-587, Sep. 2004.

[17] R. Rojas-Cessa and C-B. Lin, "Captured-frame matching schemes for scalable input-queued packet switches," *Computer Communications*, 13 pages, May 2007.

[18] H. Kim, J. Son, K. Kim, "A packet-based scheduling algorithm for high-speed switches," Proc. IEEE TENCON, Vol. 1, pp. 117-121, August 2001.

[19] C. Hu, X. Chen, W. Li, and B. Liu, "Fixed-length switching vs. variable-length switching in input-queued IP switches," Proc. IEEE Workshop on IP Operations and Management, pp. 117-122, October 2004.

[20] Y. Ganjali, A. Keshavarzian, D. Shah, "Cell Switching Versus Packet Switching in Input-Queued Switches," IEEE/ACM Trans. on Network., Vol. 13, Issue 4, pp. 782-789, August 2005.