A Framed Packet Switch Without Control Loop

Le Yu, Ge Nong and Mounir Hamdi

Abstract—**In this paper, we propose a 3-stage framed packet switch using an internal speedup of 2 to avoid any control loop between any two stages of the switch. The switch segments the arriving variable-length packets at each input port into fixed-size cells and assembles the cells into frames. Then the frames are switched across the shared buffers to their destined output ports, and the cells are reassembled into packets before being transmitted to the next hop. We have designed a broad class of work-conserving scheduling algorithms for the proposed switch, and they are analyzed to be stable, i.e. achieving 100% throughput, under any admissible traffic. To gain more insights into the switch practical performance, an extensive performance evaluation study is conducted using computer simulations. Our results demonstrate that the worst-case performance can be bounded. In addition, we are able to achieve a high throughput-delay performance comparable to that of the padded frame switch which uses a much more complicated scheduling algorithm [1].**

Index Terms—**3-stage switching, load-balanced, internal speedup, 100% throughput.**

I. Introduction

Researchers on high performance packet switching architectures have achieved many impressive results in the past two decades. To avoid the high bandwidth required by an output buffer in an output queuing (OQ) switching architectures, recent research efforts have been contributed to the design and analysis of various novel architectures such us virtual-output-queuing (VOQ) switches, multi-stage switches, and parallel [2] and load-balanced frame switches [3], [4]. Of particular interest to us in this paper is the multi-stage frame switch without control loop between any two stages.

Fig. 1 depicts the architecture of an $N \times N$ frame switch, which consists of 3 kinds of buffers: the input buffers, shared buffers and output buffers. In particular, operations of the switch are synchronized over fixed-size time slots, where each time slot is defined as the time consumed by the transmission of a cell in the external line rate of each input/output buffer. At each input buffer, the arriving packets are of variable-length, and the length of each packet is assumed to be an integer number of fixed-size cells. At each input port, a packet arrives as a sequence of cell(s), and there is at most one arriving cell per time slot. Two non-blocking switching fabrics are used to provide connections between the input buffers and the shared buffers, and between the shared buffers and the output buffers. Without loss of generality, two crossbars are assumed here as the switching fabrics. Specifically, each crossbar operates as a rotator for providing round-robin permutation connections among the inputs and outputs of the crossbar, i.e., each input will connect to each output once per N permutations. Furthermore, an internal speedup of 2 is assumed for each crossbar. At each input buffer, an arriving packet is segmented into cells and the cells are assembled into frames. A scheduling algorithm is employed to control the forwarding of frames from the input buffers to the shared buffers, and from the shared buffers to the output buffers. A packet going through the frame switch will go through three stages as illustrated below:

- Packet segmentation and frame assembly: An arriving packet at an input port is segmented into cell(s), and the cells are assembled into one or multiple frames, where each frame can encapsulate up to N cells destined for the same output port. The cells of a packet could be distributed over multiple frames, and the cells in a frame could be from different packets. In a frame, the cells from a packet are arranged in such a way that they are stored consecutively in order to ease the task of packet reassembly at the destined output port.
- Frame switching: Scheduled by given algorithm, at each scheduling point, an input will select one from all the queued frames to forward to the shared buffers. Depending on the adopted strategy, the selected frame may not necessarily be fully filled by true cells. In other words, a frame may contain some *fake* cell(s) without carrying any packet data, which is called a *fake* frame to emphasize that it contains both the true and fake cells (notice that a fake frame must have at least one true cell).
- Packet reassembly: After the frame(s) containing all the cell(s) of a packet has/have been delivered to the destined

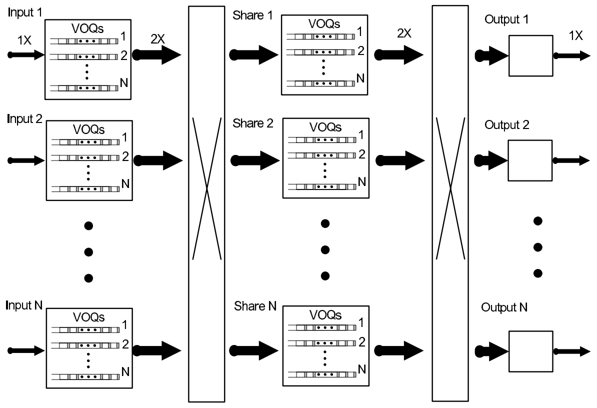


Fig. 1. The frame switch architecture.

Le Yu and Ge Nong (corresponding author) are with the Department of Computer Science, Sun Yat-sen University, Guangzhou, P.R.C. Their works were supported by the Fundamental Research Funds for the Central Universities of China (grant No. 11lgzd04). Email: issng@mail.sysu.edu.cn.

Mounir Hamdi is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. His work was supported by the HKUST grant RPC10EG21. Email: hamdi@cse.ust.hk.

output port, the packet must be reassembled from its cell(s) before it can depart to the next hop. Depending on how frames are delivered to the output port, the complexity for packet reassembly can be very simple or complex.

The frame switches of various architectures have been studied by many researchers, and how to schedule frames from the input buffers to the shared buffers has been identified as key for achieving high-performance. As summarized in [1], a number of scheduling algorithms have been proposed to used in a frame switch, e.g., first come first served, earliest deadline first, full frames first, application flow-based routing, uniform frame spreading (UFS), full ordered frames first and padded frames (PF). In particular, a 100% throughput with in-order packet delivery has been theoretically proven to be achieved by the PF and contention-reservation (CR) [5] switches using no internal speedup. In the existing load-balanced switches, a control loop is commonly found for scheduling frames to be forwarded from the input buffers to the shared buffers. This control loop generally requires instantaneous communications between the input buffers and shared buffers, where the on-the-fly queuing status of the shared buffers may need to be promptly broadcast to each input buffer. As a result, when the switch size grows, the presence of a control loop will likely increase the switch's design and implementation complexity. For instance, in the feedback-based scheduling algorithms proposed in [6] for load-balanced two-stage switches, the propagation delay between the linecards and switch fabrics is not negligible for a multi-cabinet implementation and must be carefully handled. In the sequel, we present the results of our attempt for designing a high-performance framed packet switch without control loop.

The rest of this paper is organized as follows. Section II describes our proposed switch model. Section III and IV detail the performance of our switch architectures using stability analysis and computer simulations, respectively. Section V concludes the paper.

II. The Switch Model

As mentioned previously, the performance of a frame switch is heavily dependent on the switch scheduling algorithm. But we first introduce the following switch model to back up our analysis. To simplify the presentation, we define 1 cycle as two phases, where one phase consists of $N/2$ time slots. That is, in the switch in Fig.1 using an internal speedup of 2 (**Notice**: a speedup of 2 is also assumed in [6] for the feedback-based switch using any work-conserving port-based scheduling algorithm to be stable under any admissible traffic), one phase is the time for forwarding a frame from an input buffer to the shared buffers, or from the share buffers to an output buffer.

Work-Conserving Frame Switch

- *Input Buffers*: At the beginning of each phase, an input buffer is work-conserving if (1) there is at least one full frame, or (2) there is any VOQ with a fake frame and there was no fake frame from this VOQ serviced during the time interval $(t - N^2, t-1]$. In other words, the 2nd case is to satisfy at most 1 fake frame per output can be forwarded from an input buffer per each N cycles. Given an internal speedup of 2, a frame can be forwarded from an input buffer to the shared buffers in one phase. The 1st cell of a frame is always forwarded to the 1st shared buffer. Each input buffer has a space of N^2 cells.
- *Shared Buffers*: Each shared buffer is work-conserving at the beginning of each phase and has an infinite space.
- *Output Buffers*: Each output buffer is work-conserving at each time slot and has an infinite space.

This switch model is quite general and allows us to design efficient scheduling algorithms, provided that the work-conserving constraints are satisfied. For example, to reduce the packet SAR (segmentation and reassembly) complexity, all the cells belonging to a packet can be forwarded to the shared buffers continuously. That is, if the cells of a packet span over multiple frames, then all these frames will be forwarded from the input buffers to the shared buffer continuously. When the first cell of a packet P arrives at the input i, and suppose that it sees a fake frame F being forwarded to the shared buffers, we can either 1) fill the remaining idle slots of F with the cells of P; or 2) store P_{ij} to another new frame. For the 1st method, the frames going to other outputs might be starved, but the average framing delay is likely to be reduced. For the 2nd method, we can expect a more fair scheduling for the frames destined for different outputs, however, the mean framing delay might be longer, especially when N is large. Our approved patent [7] disclosed the invention of a shared queuing frame switch with an improved delay performance by shortening the packet framing delay using the invented frame assembly buffer.

III. Stability Analysis

Given an $N \times N$ switch of infinite buffer capacity, its reference OQ switch refers to the $N \times N$ OQ switch with infinite output buffers. Let $A(t)$, $B(t)$ and $C(t)$ be the cumulative true cell arrivals destined for output j at all the input buffers, shared buffers and output buffers up to time slot t (starting from time 0), respectively. Now, we derive an upper bound for the gap between the queue lengths of the switch and its reference OQ switch.

At an input buffer, when there is only one full frame at time slot t, the number of queued cells is at most $N(N-1)+1$ (including the one arriving at t if there is any). Given an internal speedup of 2, the input buffer will complete servicing a full frame by $t+N$ (Notice: the serviced full frame may not be the one seen at t, but rather another one becoming full frame after t). That is, at least N cells will leave in $[t, t+N-1]$. However, at most $N-1$ more cells can arrive in $[t+1, t+N-1]$, for there is at most one cell arrival per time slot. Hence, there are not more than $N(N-1)+1$ queued cells in the input buffer at time $t+N$ (including the one arriving at $t+N$ if there is any), resulting in the number of queued cells in $[t, t+N]$ to be upper bounded by N^2. At time slot $t+N$, if there is at least one full frame, the input buffer will choose one to service immediately, and the queued cells will not increase when the frame is being serviced. This process repeats until there are no full frames, where the number of cells is at most $N(N-1)$. Hence, at any time, the number of queued cells in each input buffer is upper bounded by N^2. Given N input buffers, we have in total at most $K_3 = N^3$ queued cells in all the input buffers. Hence, we have an upper bound for the cumulative cell departures destined for output j from all the input buffers:

$$B(t) \geq A(t) - K_3 \tag{1}$$

Let $B_f(t)$ and $C_f(t)$ be the fake cell arrivals and departures at the shared buffers up to time slot t. Noticing that a frame in the shared buffers is always serviced from its 1st cell contained in the 1st shared buffer, a HOL frame may wait up to $(N-1)/2$ time slots for its service to be started and take $N/2$ time slots for its service to be completed. As a result, a HOL frame may wait up to $(N-1)/2 + N/2 = N - 1/2$ time slots for its service to be completed. Hence, let $K_1 = N - 1/2$, we have:

$$\begin{aligned} C(t) &= \min_{0\leq s\leq t}[B(t) + B_f(t) - 2(t-s)] - & (2)\\ &\quad C_f(t) - K_1 \\ &\geq \min_{0\leq s\leq t}[B(t) - (t-s)] + & (3)\\ &\quad \min_{0\leq s\leq t}[B_f(t) - (t-s)] - C_f(t) - K_1 & (4) \end{aligned}$$

Notice that $\min_{0\leq s\leq t}[B_f(t) - (t-s)]$ is the cumulative number of cells serviced by a work-conserving sever with the cumulative arrivals given by $B_f(t)$. Let $B_{fi}(t)$ denote the cumulative number of cells in $B_f(t)$ coming from input i, we know that $B_{fi}(t)$ is a (δ, ρ)-upper constrained traffic, where $\rho = (N-1)/N^2$ and $\delta = N - 1$, which means that $B_{fi}(t) - B_{fi}(s) \leq \rho(t-s) + \delta$ for any $s \in [0, t]$. From Lemma 1.2.1 in [8], we have that B_f is $(N\delta, N\rho)$-upper constrained, which means that $B_f(t) - B_f(s) \leq N\rho(t-s) + N\delta$. Lemma 1.3.2 in [8] says that, for a work-conserving server with service capacity c, if the offered traffic is (ρ', δ')-upper constrained and $\rho' \leq c$, then the maximum queue length is not greater than δ'. Because $N\rho = N(N-1)/N^2 = 1 - 1/N < 1$, for a work-conserving server of service capacity 1 and $B_f(t)$ as the input, we will have $Q_f(t) = B_f(t) - C_f(t) \leq N\delta = N(N-1)$. Let $K_2 = N(N-1)$, we have $C_f(t) \geq B_f(t) - K_2$. For this reference work-conserving server, we have $C_f(t) = \min_{0\leq s\leq t}[B_f(t) - (t-s)]$, hence,

$$\min_{0\leq s\leq t}[B_f(t) - (t-s)] \geq B_f(t) - K_2 - K_1 \tag{5}$$

Substituting (5) into (4), we have

$$C(t) \geq \min_{0\leq s\leq t}[B(t) + (t-s)] + B_f(t) - C_f(t) - K_2 - K_1 \tag{6}$$

Because that $B_f(t) \geq C_f(t)$, we further have

$$C(t) \geq \min_{0\leq s\leq t}[B(t) + (t-s)] - K_2 - K_1 \tag{7}$$

Substituting (1) into (7), we have

$$C(t) \geq \min_{0\leq s\leq t}[A(t) + (t-s)] - K_3 - K_2 - K_1 \tag{8}$$

Let $D_{OQ}(t)$ be the cumulative cell departures at the reference OQ switch with the cumulative arrivals given by $A(t)$, we have

$$D_{OQ}(t) = \min_{0\leq s\leq t}[A(t) + (t-s)] \tag{9}$$

Hence, substituting (9) into (8), we have the cumulative true cell arrivals at output j to be given by

$$C(t) \geq D_{OQ}(t) - K_3 - K_2 - K_1$$

Now, we come to the last step to get $D(t)$, i.e. the cumulative cell departures at output j up to time t.

$$\begin{aligned} D(t) &= \min_{0\leq s\leq t}[C(t) + (t-s)] \\ &\geq \min_{0\leq s\leq t}[D_{OQ}(t) + (t-s)] - & (10)\\ &\quad K_3 - K_2 - K_1 \end{aligned}$$

Because $\min_{0\leq s\leq t}[D_{OQ}(t) + (t-s)] = D_{OQ}(t)$, we further have from (10) the result below:

$$\begin{aligned} D(t) &\geq D_{OQ}(t) - K_3 - K_2 - K_1 \\ &\geq D_{OQ}(t) - (N^3 + N^2 - N + N - 1/2) \\ &> D_{OQ}(t) - N^3 - N^2 & (11) \end{aligned}$$

The result of (11) indicates that, given any admissible traffic, up to any time t, output j in our work-conserving frame switch will service a cumulative number of cells of at most $N^3 + N^2$ less than that done by the reference OQ switch. In other words, let $Q(t)$ be the cumulative number of queuing cells destined for output j in the reference OQ switch at time slot t, there are at most $Q(t) + N^3 + N^2$ queued cells destined for output j in our switch. Because that the reference OQ switch is stable under any admissible traffic, our switch is stable too. Hence we have the following Lemma below.

Lemma 3.1: (Stability) The work-conserving frame switch is stable under any admissible traffic, i.e. achieving 100% throughput.

IV. Performance Evaluation

We proceed to evaluate the performance of the proposed switch using computer simulations, where each simulation run lasts for 10^6 time slots. In our experiments, given the same traffic loads, three variants of scheduling algorithms satisfying the constraints given in our work-conserving frame switch model in Section II were adopted and compared with the UFS and PF algorithms. In particular, our three scheduling algorithms specified below were used to select queuing frames in each input buffer to be forwarded to the shared buffers:

- SLIP$_0$: the next non-empty VOQ first. To choose the 1st non-empty VOQ, and set its priority as the lowest for the next scheduling. The arbiter will not update anything if no queue is serviced.
- SLIP$_\mathrm{L}$: the next non-empty longest VOQ first. To choose the longest queue to serve and set its priority to the lowest

for the next scheduling, the tie between any two longest queues is resolved by their rotating priority. The arbiter will not update anything if no queue is serviced.
- RR: round-robin. The arbiter updates once per phase, i.e. twice per cycle.

Our experiments were organized into 3 sets, to investigate the mean delay as a function of offered traffic load, under different switch sizes and traffic arrivals assumptions. To simplify the presentation, we use L to denote the packet length, and $\overline{L}$ to denote the mean of L. In all the experiments, the packet arriving process at each input port is assumed to be i.i.d. Bernoulli and on/off bursty traffic modeled by the 2-state Markov process [9], and the destinations of arriving packets are *uniformly* distributed over all the N output ports. Given the same traffic loads, no internal speedup was used for UFS and PF, while an internal speedup 2 was adopted for our 3 algorithms $\mathrm{SLIP_0}$, $\mathrm{SLIP_L}$ and RR. In all the experiments, for the PF algorithm, we chose the threshold parameter $TH = 3$ for $N = 16$, while $TH = 4$ for $N = 32$ and $N = 64$.

A. *Fixed-Size Packets:* $L = 1$

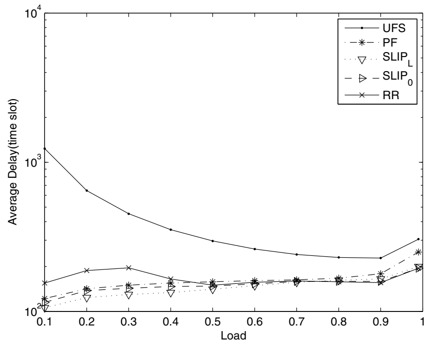
We first evaluate the switch's performance under fixed-size packet arrivals, where the size of each packet is fixed to be 1 cell and the switch sizes are varying from 16, 32 to 64. We observe that for the investigated algorithms, the delay performance curves for different switch sizes result in similar patterns as shown in Figs. 2. Specifically, the curves for PF, $\mathrm{SLIP_L}$, $\mathrm{SLIP_0}$ and RR closely cluster together, with UFS having the worst performance. This is because UFS will forward a frame only when the frame is full, however, the others will forward not only a full frame, but also a fake frame satisfying the scheduling constraints. Consequently, a packet is expected to experience a longer delay in the UFS switches.

B. *Variable-Length Packets:* $\overline{L} = 20$

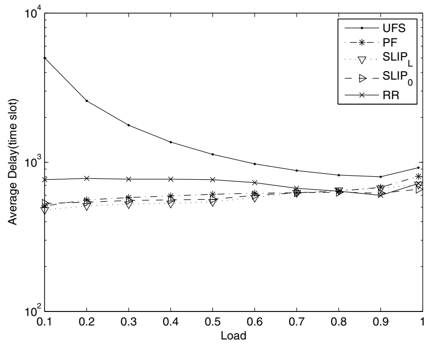
We next investigate the switch's delay performances under uniformly distributed i.i.d on-off bursty traffic, where the mean packet length is assumed to be 20 cells. From Fig. 3, we can see that for the PF switches, the average delays are the lowest among all under light traffic loads, i.e. less than 0.5. However, the curves for PF and our 3 algorithms get clustered closely together when the traffic loads become high. For different switch sizes, our algorithms have similar curve patterns with that of the PF algorithm. This well coincides with the phenomenon observed in Fig. 2.

C. *Various Mean Packet Lengths:* $\overline{L} \in \{10, 20, 30\}$

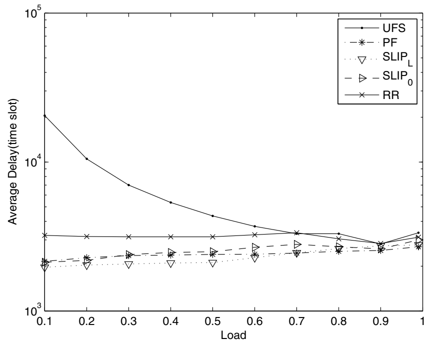
We continue to investigate a 32×32 switch under the uniformly distributed on-off bursty traffic with the mean packet lengths varying from 10, 20 to 30 cells. The results for the mean packet length of 20 are given by Fig. 3(b), and the other two are given in Fig. 4. From the figures, we can see that the curves for our switches scheduled by the $\mathrm{SLIP_0}$, $\mathrm{SLIP_L}$ and RR algorithms are very close to each other, and our switches perform slightly better than the PF switches under high traffic loads. For the 32×32 switch, the packet lengths of 10, 20 and 30 make little difference to the performances of each investigated algorithm.



(a) $N = 16$



(b) $N = 32$

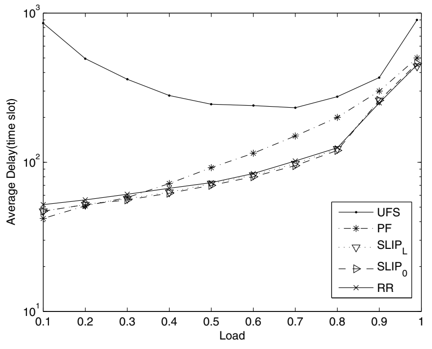


(c) $N = 64$

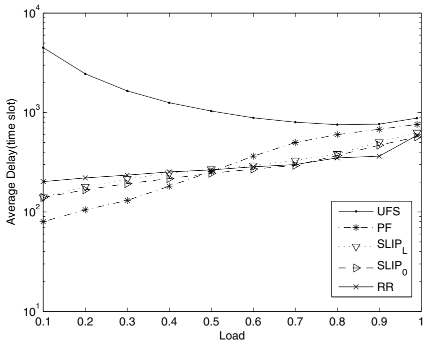
Fig. 2. The average delays for variable-size switches under the i.i.d uniform Bernoulli traffic with fixed packet length $L = 1$.

V. Conclusion

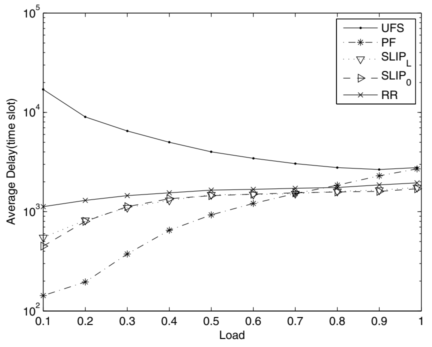
The load-balanced frame switching architecture has been proposed as a candidate for building scalable Internet packet switches. In order to achieve high performance, some kind of control loop commonly exists between the input buffers and shared buffers for a scheduling algorithm to automatically adapt to dynamic changes of queuing status in the switch. In this paper, we have presented a frame switch with an internal speedup of 2 (the same speedup is also assumed for the



(a) $N = 16$



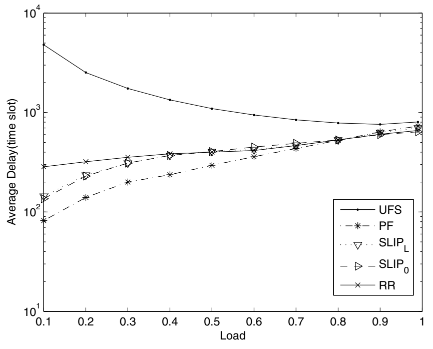
(b) $N = 32$



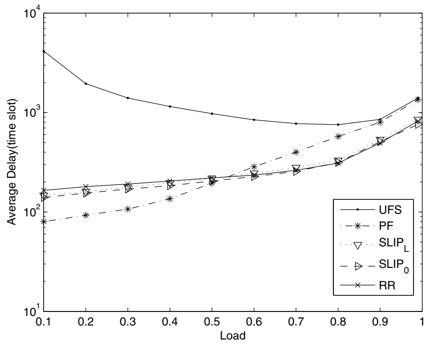
(c) $N = 64$

Fig. 3. The average delays for variable-size switches under the i.i.d uniform bursty traffic with mean packet length $\overline{L} = 20$.

switch using feedback-based scheduling to be stable under any admissible traffic in [6]), which we call the work-conserving frame switch, as well as three sample algorithms in an attempt to remove any control loop from the switch. The theoretical analysis shows that, our work-conserving frame switch is stable under any admissible traffic. Without any control loop, the scheduling at each stage of our proposed switch can perform independently, which helps reduce the switch's design and implementation complexity.



(a) $\overline{L} = 10$



(b) $\overline{L} = 30$

Fig. 4. The average delays for a 32×32 switch under the i.i.d uniform bursty traffic with variable mean packet lengths.

REFERENCES

[1] J. J. Jaramillo, F. Milan, and R. Srikant, "Padded frames: A novel algorithm for stable scheduling in load-balanced switches," *IEEE/ACM Trans. on Networking*, vol. 16, no. 5, pp. 1212–25, Oct. 2008.

[2] S. Iyer and N. McKeown, "Analysis of the parallel packet switch architecture," *IEEE/ACM Trans. on Net.*, vol. 11, no. 2, pp. 314–24, Apr. 2003.

[3] C. S. Chang, D. S. Lee, and Y. S. Jou, "Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering," *Computer Communications*, vol. 25, no. 6, pp. 611 – 622, 2002.

[4] C. S. Chang, D. S. Lee, and C. M. Lien, "Load balanced Birkhoff-von Neumann switches, part II: multi-stage buffering," *Computer Communications*, vol. 25, no. 6, pp. 623 – 634, 2002.

[5] C. L. Yu, C. S. Chang, and D. S. Lee, "CR switch: A load-balanced switch with contention and reservation," *IEEE/ACM Trans. on Net.*, vol. 17, no. 5, pp. 1659 – 71, Oct. 2009.

[6] B. Hu and K. L. Yeung, "Feedback-based scheduling for load-balanced two-stage switches," *IEEE/ACM Trans. on Net.*, vol. 18, no. 4, pp. 1077 – 90, Aug. 2010.

[7] G. Nong, "Frame assembly circuit for use in a scalable shared queuing switch and method of operation," *US Patent and Trademark Office*, no. 7,206,325, Apr. 2007.

[8] C. S. Chang, *Performance Guarantees in Communication Networks*. London, UK: Springer-Verlag, 2000.

[9] G. Nong, M. Hamdi, and J. Muppala, "Performance evaluation of multiple input-queued ATM switches with PIM scheduling under bursty traffic," *IEEE Trans. on Comm.*, vol. 49, no. 8, pp. 1329–33, Aug. 2001.