Quasi-output-buffered switches

Cheng-Shang Chang, Jay Cheng, Duan-Shin Lee and Chi-Feung Wu
Institute of Communications Engineering
National Tsing Hua University
Hsinchu 300, Taiwan, R.O.C.
Email: {cschang,jcheng}@ee.nthu.edu.tw
lds@cs.nthu.edu.tw
cfwu@gibbs.ee.nthu.edu.tw

***Abstract*—Output-buffered switches are known to have better performance than other switch architectures. However, output-buffered switches also suffer from the notorious scalability problem, and direct constructions of large output-buffered switches are difficult. In this paper, we study the problem of constructing scalable switches that have comparable performance to output-buffered switches. For this, we propose a new concept, called *quasi-output-buffered switch*. Like an output-buffered switch, a quasi-output-buffered switch is a *deterministic* switch that *delivers packets in the FIFO order* and *achieves 100% throughput*. Using the three-stage Clos network, we show that one can *recursively* construct a larger quasi-output-buffered switch with a set of smaller quasi-output-buffered switches. By recursively expanding the three-stage Clos network, we obtain a quasi-output-buffered switch with only 2×2 switches. Such a switch is called a packet-pair switch as it always transmits packets in pairs. By computer simulations, we show that packet-pair switches have better delay performance than most load-balanced switches with comparable construction complexity.**

***Index Terms*—output-buffered switches, load-balanced switches, packet-pair switches, delay performance.**

I. Introduction

It is known that an output-buffered switch achieves 100% throughput and has the best delay performance among all switch architectures. However, this is at the cost of N times speedup for an $N \times N$ output-buffered switch. The required speedup somehow limits us to construct a large output-buffered switch. There are several studies in the literature that achieve exact emulation of an output-buffered switch, e.g., the crosspoint-buffered switch [1], the parallel-buffered switch [13], and the combined input-output queue [15], [23]. However, all these have either non-scalable hardware complexity, or computation and communication overheads.

One of the key problems in high speed switching is whether one can construct scalable switches with comparable performance to output-buffered switches. Recent advances in load-balanced switches (see e.g., [6], [9], [14], [17]) have shed some light on that problem. A typical load-balanced switch consists of two stages: the first stage is for load-balancing that converts incoming traffic into the uniform traffic, and the second stage is for switching of the uniform traffic. Moreover, the connection patterns in the switches of both stages are *deterministic* and *periodic*. It is shown that various load-balanced switches have comparable performance to output-buffered switches. As such, they can achieve 100% throughput with $O(1)$ computation and communication overheads.

One of the main contributions of this paper is to identify the key ingredients in load-balanced switches that enable us to construct large switches with comparable performance to output-buffered switches. For this, we propose a new concept, called *quasi-output-buffered switch*. Like an output-buffered switch, a quasi-output-buffered switch is a *deterministic* switch that *delivers packets in the First-in First-out (FIFO) order* and *achieves 100% throughput*. Using the three-stage Clos network [11], we show that one can *recursively* construct a larger quasi-output-buffered switch with a set of smaller quasi-output-buffered switches. To our best knowledge, such a result on quasi-output-buffered switches seems to be the first result that allows *recursive constructions* of switches with comparable performance to output-buffered switches. Analogous to the construction of a Benes network [2], we recursively expand the three-stage Clos network to obtain a quasi-output-buffered switch with only 2×2 switches. Such a switch is called a packet-pair switch as it always transmits packets in pairs. The packet-pair switch has several nice features: 100% throughput, FIFO delivery of packets, deterministic connection patterns for 2×2 switches, self-routing of packets, and no need for communication and computation. By computer simulations, we also show that packet-pair switches have better delay performance than most load-balanced switches with comparable construction complexity.

The key theory behind our constructions of quasi-output-buffered switches is a refined *calculus* based on a traffic characterization in [8]. Such a traffic characterization allows us to describe a flow of packets by a single "rate." It is shown that the aggregated flow has the "rate" equal to the sum of the "rate" of each individual flow. Round-robin splitting of a flow yields several subflows with smaller "rates." Moreover, a departing flow has the same "rate" as that of the arriving flow provided that the system is "stable." Unlike the theory of effective bandwidth (see e.g., [16] and references therein), the refined calculus does not need the *independent* assumption on the flows.

The paper is organized as follows: in Section II, we introduce the traffic characterization and its associated calculus. Then we define the concept of a quasi-output-buffered switch. In Section III, we propose the three-stage construction of a quasi-output-buffered switch. The packet-pair switches are

introduced in Section IV. Finally, the paper is concluded in Section V, where we address possible extensions and future research problems.

II. Quasi-output-buffered switches

A. Traffic characterization

A flow is commonly known as a sequence of packets that have the same source and destination pair in a switch (or a network of switches). In most switching papers, traffic characterizations for flows in a switch (or a network of switches) are usually assumed to follow certain traffic models, e.g., Bernoulli arrival processes and Markov processes. These traffic models are too specific for our constructions of quasi-output-buffered switches. Instead, we will use a much more general traffic characterization for a flow of packets in [8]. Throughout this paper, we only consider the discrete-time setting and make the following assumptions:

(A1) Time is slotted and synchronized in every link.

(A2) Packets are of the same size and they can be transmitted in a time slot.

Definition 1: (i) A stochastic process $\{Q(t), t \geq 0\}$ is said to have a finite moment generation function if there exists a $\theta > 0$ such that

$$\sup_t \mathsf{E}[e^{\theta Q(t)}] < \infty. \tag{1}$$

(ii) For a flow A, we will use $A(t)$ to denote the cumulative number of packets that arrives by time t for that flow. Flow A is said to be λ-moment generating function bounded from above (λ-m.b.f.a.) if for every $\epsilon > 0$, the stochastic process $\{Q(t), t \geq 0\}$ defined below has a finite moment generation function:

$$Q(t) = \max_{0 \leq s \leq t} [A(t) - A(s) - (\lambda + \epsilon)(t - s)]. \tag{2}$$

With $Q(0) = 0$, we note that $Q(t)$ in (2) is in fact the recursive expansion of the Lindley equation [18]

$$Q(t) = \max[0, Q(t-1) + a(t) - (\lambda + \epsilon)], \tag{3}$$

where $a(t) = A(t) - A(t-1)$ is the number of packets that arrive at time t. In view of (3), $Q(t)$ is simply the number of packets (or more precisely bits with $Q(t)$ being a real number) at time t when we feed flow A to a work conserving link with capacity $\lambda + \epsilon$. It is known from the Loynes construction [20] that $\{Q(t), t \geq 0\}$ converges in distribution to a steady state random variable $Q(\infty)$ if the sequence $\{a(t), t \geq 1\}$ is stationary and ergodic with a mean rate not greater than λ. However, traffic characterization by the mean rate of a stationary and ergodic sequence is not strong enough to have a finite moment generation function of the steady state random variable $Q(\infty)$. For this, we need a stronger condition in [3]. Let

$$a^*(\theta) = \limsup_{t \to \infty} \frac{1}{\theta t} \sup_{s \geq 0} \log \mathsf{E}[e^{\theta(A(t+s) - A(s))}] \tag{4}$$

be the minimum envelope rate (MER) with respect to $\theta > 0$ (or known as the effective bandwidth function, see e.g., [16]). When

$$Q(t) = \max_{0 \leq s \leq t} [A(t) - A(s) - (a^*(\theta) + \epsilon)(t - s)], \tag{5}$$

it was shown in Theorem 3.8 in [3] that

$$\sup_t \mathsf{E}[e^{\theta Q(t)}] < \infty, \tag{6}$$

This shows that flow A is $a^*(\theta)$-m.b.f.a. for any $\theta > 0$. One can further choose the best traffic characterization by letting $\rho = \inf_{\theta > 0} a^*(\theta)$ and thus flow A is ρ-m.b.f.a. We note that for many stochastic processes, the value ρ is simply the mean arrival rate, as illustrated in the following example for the Bernoulli arrival process.

Example 2: Consider the Bernoulli arrival process with mean ρ, i.e., with probability ρ there is an arriving packet in a time slot and this is independent of everything else. For such an arrival process,

$$a^*(\theta) = \frac{1}{\theta} \log(\rho e^{\theta} + (1 - \rho)), \tag{7}$$

and

$$\inf_{\theta > 0} a^*(\theta) = \lim_{\theta \to 0} a^*(\theta) = \rho. \tag{8}$$

Thus, the Bernoulli arrival process with mean ρ is ρ-m.b.f.a.

In view of Example 2, our traffic characterization is only slightly stronger than the traffic characterization by the mean arrival rate. The additional assumption on the bounded moment generation functions lead to the following three important properties: the superposition property in Lemma 3, the splitting property in Lemma 4 and the departure property in Lemma 5. The proofs of Lemma 3, Lemma 4 and Lemma 5 are omitted due to space limitation and they can be found in the full report [4].

In the following lemma, we derive the superposition property for two flows.

Lemma 3: **(Superposition)**

(i) If both $\{Q_1(t), t \geq 0\}$ and $\{Q_2(t), t \geq 0\}$ have finite moment generating functions, then $\{Q_1(t) + Q_2(t), t \geq 0\}$ also has a finite moment generating function.

(ii) If flow A_1 is λ_1-m.b.f.a. and flow A_2 is λ_2-m.b.f.a., then the superposition of the two flows A_1 and A_2 (defined by $A_1(t) + A_2(t)$) is $\lambda_1 + \lambda_2$-m.b.f.a.

We note that the proof of Lemma 3 is based on the Cauchy-Schwartz inequality and $Q_1(t)$ and $Q_2(t)$ in Lemma 3(i) (resp. A_1 and A_2 Lemma 3(ii)) need not be independent. As discussed before, if we can view λ_1 as the "mean" rate for flow A_1 and λ_2 as the "mean" rate for flow A_2, then the aggregated flow has the "mean" rate $\lambda_1 + \lambda_2$.

The second property is the splitting property.

Lemma 4: **(Round-robin splitting)** Consider a flow A. Suppose that we split flow A in the round robin fashion into

p sub-flows $A_1, A_2, \ldots, A_p$ with

$$A_m(t) = \lceil \frac{A(t) - m + 1}{p} \rceil, \quad m = 1, 2, \ldots, p. \tag{9}$$

If flow A is λ-m.b.f.a., then for all $m = 1, 2, \ldots, p$, flow A_m is λ/p-m.b.f.a.

The intuition of Lemma 4 is quite obvious. If we view λ as the "mean" rate for flow A, then flow A_m, obtained from round-robin splitting, has the "mean" rate λ/p.

The third property is the departure property.

Lemma 5: **(Departure)** Consider a flow A that is fed into a system (along with possible other flows). Let flow B be the departure flow of flow A, i.e., $B(t)$ is the cumulative number of flow A packets that depart from the system by time t. Also, let $Q(t)$ be the total number of packets (including packets from flow A and other flows) in the system at time t. If (i) flow A is λ-m.b.f.a., and (ii) $\{Q(t), t \geq 0\}$ has a finite moment generation function, then flow B is also λ-m.b.f.a.

The departure property shows that if flow A has the "mean" rate λ, then flow B, the departure flow of flow A, also has the "mean" rate λ provided that the system is "stable" (in the sense of bounded moment generation function). As we shall see later, the superposition property, the splitting property, and the departure property provide us a simple calculus for our traffic characterization in a network of switches.

We note that it is difficult to obtain the departure property in Lemma 5 if one uses weaker traffic characterizations, such as stationarity and ergodicity. On the other hand, it is possible to obtain such a departure property by using stronger traffic characterizations, such as the (σ, ρ)-deterministic traffic characterization in the network calculus [12]. However, such a deterministic traffic characterization cannot be used for stochastic analysis needed in our later development.

B. Output-buffered switches

A switch that has M input links and N output links is called an $M \times N$ switch. A (local) flow in an $M \times N$ switch is the sequence of packets that have the same input link and output link. As there are M inputs and N outputs, there are MN flows for an $M \times N$ switch.

Let flow $A_{i,k}$ be the flow from input i to output k, and $A_{i,k}(t)$, $i = 1, 2, \ldots, M$, $k = 1, 2, \ldots, N$, be the cumulative number of packets that arrives by time t for that flow. Also, let $B_k(t)$, $k = 1, 2, \ldots, N$, be the cumulative number of packets that depart from output k by time t, and $Q_k(t)$ be the number of packets stored at the k^{th} output at time t.

Definition 6: **(Output-buffered switch)** An $M \times N$ switch is called an output-buffered switch if it satisfies the following two properties when it is started from an empty system at time 0 (i.e., $Q(0) = 0$):

(i) packets destined for the same output depart in the First-in First-out (FIFO) order, and

(ii) for all $k = 1, 2, \ldots, N$,

$$Q_k(t) = \max[0, Q_k(t-1) + \sum_{i=1}^{M} a_{i,k}(t) - 1], \tag{10}$$

where $a_{i,k}(t) = A_{i,k}(t) - A_{,k}(t-1)$ is the number of flow $A_{i,k}$ packets that arrive at time t.

Equation (10), known as the Lindley recursion, says that all the packets that arrive at time t from flows $A_{i,k}$, $i = 1, 2, \ldots, M$, are sent to the output buffer of the k^{th} output port at the same time. If there are packets in that output buffer, then one packet will depart from the output port. We note that in the worst case there might be packets arriving from all the M flows at the same time. In that case, each output buffer is required to have the capability of receiving M packets at the same time. As such, each output buffer needs to speed up (at least) M times and that causes the notorious scalability problem for an output-buffered switch.

By recursively expanding the Lindley equation in (10) with $Q_k(0) = 0$ yields

$$Q_k(t) = \max_{0 \leq s \leq t} [\sum_{i=1}^{M} (A_{i,k}(t) - A_{i,k}(s)) - (t - s)]. \tag{11}$$

Since $Q_k(t) = \sum_{i=1}^{M} A_{i,k}(t) - B_k(t)$, it then follows that

$$B_k(t) = \min_{0 \leq s \leq t} [\sum_{i=1}^{M} A_{i,k}(s) + (t - s)]. \tag{12}$$

Note that from the FIFO property and (12) of an output-buffered switch, the departure of a packet at time t is uniquely determined by all the packets that arrive by time t. As such, if the arrival times of all the packets are delayed by a constant c, then the departure times of all the packets are also delayed by the same constant c.

To ensure the stability of an output-buffered switch, we introduce the following no overbooking condition.

Definition 7: **(No overbooking condition)** Consider an $M \times N$ switch. The input traffic is said to satisfy the no overbooking condition if

(i) for all $i = 1, 2, \ldots, M$, $k = 1, 2, \ldots, N$, flow $A_{i,k}$ is $\lambda_{i,k}$-m.b.f.a., and

(ii) for all $k = 1, 2, \ldots, N$,

$$\sum_{i=1}^{M} \lambda_{i,k} < 1. \tag{13}$$

Intuitively, the no overbooking condition in (13) indicates that the total "mean" rate to a particular output port cannot exceed 1. Under the no overbooking condition, we show that an output-buffered switch is stable in the sense of having a finite moment generation function.

Lemma 8: For an $M \times N$ output-buffered switch, if the input traffic satisfies the no overbooking condition in Definition 7, then (i) $\{Q_k(t), t \geq 0\}$ has a finite moment generation function, $k = 1, 2, \ldots, N$, and (ii) $\{Q(t), t \geq 0\}$ has a finite moment generation function, where $Q(t) = \sum_{k=1}^{N} Q_k(t)$ is the total number of packets in the switch at time t.

Such a property is called the *universal stability property* (in the sense of the existence of a finite moment generating function for the total number of packets in a switch).

Proof. (i) Using the superposition property in Lemma 3(ii), the aggregated flow to the k^{th} output is $\sum_{i=1}^{M} \lambda_{i,k}$-m.b.f.a. The result in (i) then follows directly from (13) and Definition 1.

(ii) This is a direct consequence of the superposition property in Lemma 3(i). ■

C. Definition of quasi-output-buffered switches

As discussed before, output-buffered switches do not scale due to the needed speedup. As such, it is difficult to construct a large output-buffered switch *directly*. The natural question is then whether one can construct a larger switch using a set of smaller output-buffered switches. We will show in this paper that this is possible by extracting and preserving some key properties in output-buffered switches. The switches that satisfy these key properties are called quasi-output-buffered switches (defined below), i.e., they behave like output-buffered switches but they are not exactly the same as output-buffered switches.

Definition 9: **(Quasi-output-buffered switch)** An $M \times N$ switch is called a quasi-output-buffered switch if it satisfies the following properties when it is started from an empty system at time 0:

- (P1) Deterministic mapping: the departure time of every packet is a deterministic function of the arrival times of all the packets. This implies that if the arrival times of all the packets are delayed by a constant c, then a quasi-output-buffered switch can be operated in a way (by shifting the starting time of the switch) so that the departure times of all the packets are also delayed by the same constant c.
- (P2) FIFO: packets of the same flow depart in the FIFO order.
- (P3) Universal stability: let $Q(t)$ be the total number of packets in the switch. If the input traffic of the switch satisfies the no overbooking condition in Definition 7, then $\{Q(t), t \geq 0\}$ has a finite moment generation function.

Clearly, an output-buffered switch is a quasi-output-buffered switch (from Lemma 8). These include the set of switches that achieve exact emulation of output-buffered switches (e.g., the CIOQ switch in [15]). Various versions of load-balanced Birkhoff-von Neumann switches, including the Uniform Frame Spreading (UFS) in [17], the Padded Frame in [14], and the CR switch in [24], are shown to have a constant bound when comparing to the total number of packets in the corresponding output-buffered switch. Thus, they are also quasi-output-buffered switch. However, it is not clear whether an input-buffered switch with maximum weight matching (MWM) [21] is a quasi-output-buffered switch as the universal stability property in (P3) has not been proved in the literature yet.

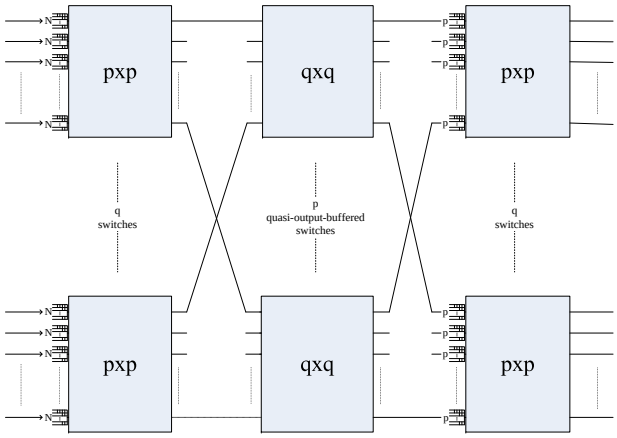

Fig. 1. A three-stage construction of a quasi-output-buffered switch

III. A THREE-STAGE CONSTRUCTION OF A QUASI-OUTPUT-BUFFERED SWITCH

A. Operation rules

In this section, we show how one can construct a larger quasi-output-buffered switch by using a set of smaller quasi-output-buffered switches. In Figure 1, we show a three-stage construction of an $N \times N$ quasi-output-buffered switch, where $N = p \times q$. In the first stage, there are q $p \times p$ input-buffered switches. Each input buffer at an input link of a switch in the first stage has N virtual output queues (VOQ). The second stage consists of p $q \times q$ quasi-output-buffered switches. Finally, in the third stage, there are also q $p \times p$ input-buffered switches. Each input buffer at an input link of a switch in the third stage has p VOQs. As in a standard Clos network [11], the switches in the first stage and those in the second stage are connected by the perfect shuffle exchange, i.e., for $m = 1, 2, \ldots, p$, $\ell = 1, 2, \ldots, q$, the m^{th} output from the ℓ^{th} switch in the first stage is connected to the ℓ^{th} input of the m^{th} switch in the second stage. Similarly, the switches in the second stage and those in the third stage are also connected by the perfect shuffle exchange, i.e., for $m = 1, 2, \ldots, p$, $\ell = 1, 2, \ldots, q$, the ℓ^{th} output from the m^{th} switch in the second stage is connected to the m^{th} input of the ℓ^{th} switch in the third stage.

The main idea of the three-stage construction is to accumulate packets in the first stage to form a frame. Then use the uniform frame spreading (UFS) scheme in [17] to distribute the packets in a frame *evenly* to the quasi-output-buffered switches in the second stage. Finally, packets in a frame are "re-assembled" in the last stage.

To do this, the connection patterns of the $p \times p$ switches in the first stage and the third stage are specified by the symmetric TDM switch in [7]. Recall that a $p \times p$ symmetric TDM switch implements the following periodic connection patterns: input i is connected to output j at time t if and only if

$$(i + j) \bmod p = (t + 1) \bmod p. \tag{14}$$

In other words, for any positive integer f, input i is connected

to output 1 at time $i+(f-1)p$, output 2 at time $i+1+(f-1)p$,..., and output p at time $i-1+fp$. Also, it is clear from (14) that every connection pattern in a symmetric TDM switch is *symmetric* (as input i is connected to output j if and only if input j is connected to output i). As such, output i is connected to input 1 at time $i+(f-1)p$, input 2 at time $i+1+(f-1)p$,..., and input p at time $i-1+fp$.

Now we specify the operation rules in these three stages.

(R1) Uniform frame spreading (UFS) for the switches in the first stage:

There are N VOQs at an input of a symmetric TDM switch at the first stage. When a packet destined for output j arrives, it is placed in the j^{th} VOQ, $j=1,2\ldots,N$. The switches in the first stage are operated in a frame-based manner as in the UFS scheme [17]. Every frame consists of p consecutive time slots. However, the beginning time slots of frames are different for different *inputs*. Specifically, frame f of input i of a switch in the first stage begins at the f^{th} time when input i is connected to the *first* quasi-output-buffered switch in the second stage. As such, we have from (14) that frame f of input i consists of time slots $i+(f-1)p,\ldots,i-1+fp$. If the number of packets in a VOQ is not less than p, that VOQ is called a full-framed VOQ. At the beginning of a frame, if an input of a switch in the first stage has at least one full-framed VOQ, then the switch selects one full-framed VOQ and sends p consecutive packets from that VOQ in that frame. As such, these p packets are distributed to the p $q\times q$ quasi-output-buffered switches. Otherwise, it does nothing during that frame.

(R2) Time shifted operations for the quasi-output-buffered switches in the second stage:

From the UFS scheme in (R1), we know that if there is a packet destined for output j arrives at the i^{th} input of the *first* switch in the second stage at time t, then there is also a packet destined for output j arrives at the i^{th} input of the ℓ^{th} switch in the second stage at time $t+\ell-1$, $\ell=2,\ldots,p$. In other words, the arrival process to the ℓ^{th} switch in the second stage is simply a time shifted version of that to the first switch in the second stage. Thus, they can be made to be *identical* if we run the clock in the ℓ^{th} switch by the new time $t'=t-\ell+1$. As there is a *unique* routing path to an (external) output from an input of a switch in the second stage, we know from the deterministic mapping property in (P1) that the departure process from the first switch in the second stage and that from the ℓ^{th} switch in the second stage are also *identical* with respect to the new clocks. As such, if there is a packet destined for output j arrives at the *first* input of the k^{th} switch in the *third* stage at time t, then there is also a packet destined for output j arrives at the ℓ^{th} input of the k^{th} switch in the *third* stage at time $t+\ell-1$, $\ell=2,\ldots,p$.

(R3) Inverse uniform frame spreading in the third stage:

There are p VOQs at an input of a symmetric TDM switch in the third stage. When a packet destined for output j arrives, it is placed in the $k(j)^{th}$ VOQ, where $k(j)=j-\lfloor(j-1)/p\rfloor*p$. The switches in the third stage are operated in a frame-based manner as those in the first stage. Every frame consists of p consecutive time slots. However, the beginning time slots of frames are different for different *outputs*. Specifically, frame f of output i of a switch in the third stage begins at the f^{th} time when output i is connected to the *first* input of that switch. As such, we have from (14) that frame f of output i consists of time slots $i+(f-1)p,\ldots,i-1+fp$. During a frame of output i, every input sends a packet from its i^{th} VOQ to output i (if its i^{th} VOQ is not empty).

Theorem 10: The three-stage construction described above is indeed an $N\times N$ quasi-output-buffered switch.

We note there are several early works in the literature (see e.g., [5], [10]) that also used the three-stage Clos network to construct a larger switch. To our best knowledge, it seems that Theorem 10 on quasi-output-buffered switches is the first result that allows *recursive constructions* of switches with comparable performance to output-buffered switches.

Clearly, as the switches in the first stage and the third stage are symmetric TDM switches, they are deterministic. As the quasi-output-buffered switches in the second stage satisfy the deterministic mapping property in (P1), the three-stage construction also satisfies the deterministic mapping property. Also, from the UFS scheme in (R1) and the inverse UFS in (R3), packets of the same flow depart in the FIFO order. Thus, (P2) of the three-stage construction is satisfied. It remains to show the universal stability property in (P3). This will be done in the following section.

B. Universal stability

In this section, we show the universal stability property for the three-stage construction. Denote by flow $A_{i,k}$, $i,k=1,2,\ldots,N$, the sequence of packets from input i to output k. For the proof of the universal stability property, we assume that the no-overbooking condition in Definition 7 is satisfied, i.e., for all $i,k=1,2,\ldots,N$, $A_{i,k}$ is $\lambda_{i,k}$-m.b.f.a., and for all $k=1,2,\ldots,N$,

$$\sum_{i=1}^{N}\lambda_{i,k}<1. \tag{15}$$

As the switches in the first stage are operated under the UFS scheme, it is well known (see e.g., [17], [24]) that the number of packets stored in an input buffer of a switch in the first stage is bounded above by a finite constant. This is stated in the following proposition.

Proposition 11: The total number of packets in an input buffer of a switch in the first stage is bounded above by Np.

Now we show the universal stability property for the switches in the second stage. As the switches in the second stage are quasi-output-buffered switches, the key step is then to verify that the no-overbooking condition is satisfied for every quasi-output-buffered switch in the second stage.

Let flow $B^1_{i,k}$ be the departing flow of flow $A_{i,k}$ from the first stage. Then it follows from Proposition 11 and the departure property in Lemma 5 that $B^1_{i,k}$ is also $\lambda_{i,k}$-m.b.f.a. Consider the m^{th} switch in the second stage. Let $A^2_{i,k,m}(t)$ be the cumulative number of packets from flow $A_{i,k}$ that arrive at that switch by time t. As the switches in the first stage are operated under the uniform frame spreading scheme, the

packets from the same flow are distributed in a round-robin fashion to the switches at the second stage. Thus,

$$A^2_{i,k,m}(t) = \lceil \frac{B^1_{i,k}(t) - m + 1}{p} \rceil. \quad (16)$$

It then follows from the splitting property in Lemma 4 that flow $A^2_{i,k,m}$ is $\lambda_{i,k}/p$-m.b.f.a.

Let flow $A^2_{j,\ell}$ be the local flow of packets that traverse from the j^{th} input link of the m^{th} switch in the second stage to the ℓ^{th} output link of that switch. Clearly, flow $A^2_{j,\ell}$ is the aggregated flow of the set of flows $A^2_{i,k,m}$, $i = (j-1)p + 1, (j-1)p+2, \ldots, jp$, $k = (\ell-1)p+1, (\ell-1)p+2, \ldots, \ell p$. As flow $A^2_{i,k,m}$ is $\lambda_{i,k}/p$-m.b.f.a., we have from the superposition property in Lemma 3(ii) that the local flow $A^2_{j,\ell}$ is then $\sum_{i=(j-1)p+1}^{jp} \sum_{k=(\ell-1)p+1}^{\ell p} \lambda_{i,k}/p$-m.b.f.a.

Note that

$$\begin{aligned} &\sum_{j=1}^{q} \sum_{i=(j-1)p+1}^{jp} \sum_{k=(\ell-1)p+1}^{\ell p} \lambda_{i,k}/p \\ &= \sum_{i=1}^{N} \sum_{k=(\ell-1)p+1}^{\ell p} \lambda_{i,k}/p \\ &= \frac{1}{p} \sum_{k=(\ell-1)p+1}^{\ell p} \sum_{i=1}^{N} \lambda_{i,k} < 1, \end{aligned} \quad (17)$$

where we use (15) in the last inequality. As such, the no-overbooking condition for the m^{th} switch in the second stage is satisfied. In view of the definition of a quasi-output-buffered switch and the superposition property in Lemma 3(i), we then have the following proposition.

Proposition 12: (i) Let $Q^2_m(t)$ be the total number of packets in the m^{th} switch in the second stage at time t. Then $\{Q^2_m(t), t \geq 0\}$ has a finite moment generating function.

(ii) Let

$$Q^2(t) = \sum_{m=1}^{p} Q^2_m(t) \quad (18)$$

be the total number of packets in the second stage at time t. Then $\{Q^2(t), t \geq 0\}$ also has a finite moment generating function.

Now we show the universal stability property for the switches in the third stage. Consider the switch in the third stage that contains the k^{th} output. Let $A^3_{i,k,m}(t)$ be the cumulative number of flow $A_{i,k}$ packets that arrive at that m^{th} input buffer of that switch by time t. Since flow $A^3_{i,k,m}$ is simply the departure process of flow $A^2_{i,k,m}$, we have from Proposition 12(ii) and Lemma 5 that flow $A^3_{i,k,m}$ is also $\lambda_{i,k}/p$-m.b.f.a.

Let flow $A^3_{k,m}$ be the aggregated flow of the set of flows $A^3_{i,k,m}$, $i = 1, 2, \ldots, N$. Then we have from the superposition property in Lemma 3(ii) that the aggregated flow $A^3_{k,m}$ is $\sum_{i=1}^{N} \lambda_{i,k}/p$-m.b.f.a.

Let $Q^3_{k,m}(t)$ be the total number of packets destined for the k^{th} output that are stored in the m^{th} input buffer of the switch in the third stage that contains the k^{th} output. Also, let $C^3_{k,m}(t)$ be the cumulative number of time slots that the m^{th} input of that switch is connected to output k by time t. As the connection pattern of that switch is periodic with period p, we have

$$C^3_{k,m}(t) - C^3_{k,m}(s) \geq \lfloor (t-s)/p \rfloor \geq (t-s)/p - 1. \quad (19)$$

Moreover, we have from the Lindley equation (with $Q^3_{k,m}(0) = 0$) that

$$\begin{aligned} Q^3_{k,m}(t) &= \max[0, Q^3_{k,m}(t-1) + A^3_{k,m}(t) - \\ &\quad A^3_{k,m}(t-1) - (C^3_{k,m}(t) - C^3_{k,m}(t-1))] \\ &= \max_{0 \leq s \leq t} [A^3_{k,m}(t) - A^3_{k,m}(s) - (C^3_{k,m}(t) - C^3_{k,m}(s))]. \end{aligned} \quad (20)$$

Since the aggregated flow $A^3_{k,m}$ is $\sum_{i=1}^{N} \lambda_{i,k}/p$-m.b.f.a., it then follows from the no overbooking condition in (15) that $\{Q^3_{k,m}(t), t \geq 0\}$ has a finite moment generating function. Using the superposition property in Lemma 3(i), we then derive the following result.

Proposition 13: Let

$$Q^3(t) = \sum_{k=1}^{N} \sum_{m=1}^{p} Q^3_{k,m}(t) \quad (21)$$

be the total number of packets in the third stage at time t. Then $\{Q^3(t), t \geq 0\}$ also has a finite moment generating function.

Let $Q(t)$ be the total number of packets inside the three-stage construction at time t. From Proposition 11, Proposition 12, Proposition 13 and the superposition property in Lemma 3(i), we then conclude that $\{Q(t), t \geq 0\}$ also has a finite moment generation function.

IV. Packet-pair switches

A. Architecture

In the case that N is a power of 2, we can recursively construct an $N \times N$ quasi-output-buffered switch by the three-stage construction in Section III (as in the construction of a Benes network [2]). To do this, we first note that for $N = 2$ we can simply choose $p = 2$ and $q = 1$ in the three-stage construction in Section III. Since a 1×1 switch can be simply replaced to a single link, the three-stage construction for this is equivalent to the (two-stage) load-balanced Birkhoff-von Neumann switch with the uniform frame spreading scheme. For such a switch, the frame size is 2 and packets are transmitted in pairs under the uniform frame spreading scheme. Now we can define packet-pair switches recursively as follows:

Definition 14: **(Packet-pair switches)**

(i) A 2×2 packet-pair switch is the 2×2 load-balanced Birkhoff-von Neumann switch with the uniform frame spreading scheme.

(ii) An $N \times N$ packet-pair switch is constructed by the three-stage construction in Section III with $p = 2$ and $q = N/2$, i.e., there are $N/2$ 2×2 input-buffered switches in the first stage, two $\frac{N}{2} \times \frac{N}{2}$ packet-pair

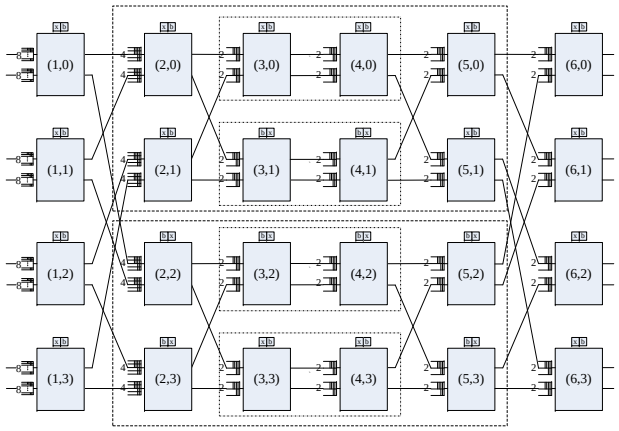

Fig. 2. An 8×8 packet-pair switch

switches in the second stage, and $N/2$ 2×2 input-buffered switches in the third stage.

By recursively expanding the $N \times N$ packet-pair switch, we have a network of $2 \log_2 N$ stages with each stage consisting of $N/2$ 2×2 switches. In Figure 2, we show an 8×8 packet-pair switch.

The operations of a packet-pair switch can also be specified in details by recursively expanding the operations in (R1) and (R3). In the following, we describe the detailed operations of an $N \times N$ packet-pair switch with $N = 2^n$. For the ease of the presentation, we index the inputs/outputs from $0, 1, 2, \ldots, 2^n - 1$. Also, the $N/2$ switches at each stage are indexed from $0, 1, 2, \ldots, 2^{n-1} - 1$.

(R4) Uniform frame spreading for the first n stages:

For $j = 1, 2, \ldots, n$, the m^{th} 2×2 switch in the j^{th} stage consists of 2^{n-j+1} VOQs at each input. These 2^{n-j+1} VOQs are indexed from $0, 1, 2, \ldots, 2^{n-j+1} - 1$. The connection patterns of the switch are periodic with period 2. It is set to the "bar" state when

$$t + \sum_{\ell=2}^{j} \lfloor \frac{m \mod 2^{n-\ell+2}}{2^{n-\ell+1}} \rfloor$$

is an odd number and to the "cross" state otherwise. Suppose a packet destined for output k arrives at a switch in the j^{th} stage. Let $b_n b_{n-1} \ldots b_1$ be the binary presentation for k, i.e., $k = \sum_{\ell=1}^{n} b_\ell 2^{\ell-1}$. The packet is routed to the $k_1(j)^{th}$ VOQ, where $k_1(j) = \sum_{\ell=1}^{n-j+1} b_{\ell+j-1} 2^{\ell-1}$. A VOQ is called a full-framed VOQ if the number of packets in that VOQ is not less than 2. When an input is connected to the first output at time t, it selects a full-framed VOQ and sends two consecutive packets (packet-pair) from that VOQ at time t and $t + 1$.

(R5) Self-routing for the last n stages:

For $j = n + 1, 2, \ldots, 2n$, the m^{th} 2×2 switch in the j^{th} stage consists of two VOQs at each input, indexed by 0 and 1. Its connection patterns are the same as those of the m^{th} switch in the $2n + 1 - j^{th}$ stage. Suppose a packet destined for output k arrives at the switch. Let $b_n b_{n-1} \ldots b_1$ be the binary presentation for k, i.e., $k = \sum_{\ell=1}^{n} b_\ell 2^{\ell-1}$. The packet is routed to the $k_2(j)^{th}$ VOQ, where $k_2(j) = b_{2n-j+1}$. When the switch is in the "bar" state at time t, VOQ 0 is selected and its head-of-line packet is transmitted at time t. Otherwise, VOQ 1 is selected and its head-of-line packet is transmitted at time t.

Note that the 2×2 switches in the first n stages of the $N \times N$ packet-pair switch is operated under the UFS scheme with frame size 2. From Proposition 11, it follows that the total number of packets in an input buffer of a switch in the j^{th} stage, $j = 1, 2, \ldots, n$, is bounded above by $2^{n-j+1} \times 2$. Moreover, we have from the deterministic mapping property that the arrival process to *any* input buffer of a 2×2 switch in the $n + 1^{th}$ stage is simply a time shifted version of the arrival process to the *first* input buffer of the *first* switch in the $n + 1^{th}$ stage. In view of this, the first n stages in fact perform load balancing for the incoming traffic at the $N \times N$ packet-pair switch.

Now we consider the Bernoulli arrival traffic in Example 2. With probability $0 \leq \rho < 1$, there is a packet that arrives at an input of the $N \times N$ packet-pair switch. This is independent of everything else. With probability $r_{i,k}$, an arriving packet at input i is destined for output k. This is also independent of everything else. Note that from the law of total probability, we must have

$$\sum_{k=1}^{N} r_{i,k} = 1, \tag{22}$$

for all $i = 1, 2, \ldots, N$. For such a model, flow $A_{i,k}$ (the sequence of packets from input i to output k) is a Bernoulli arrival process with mean $\rho r_{i,k}$. From Example 2, flow $A_{i,k}$ is $\lambda_{i,k}$-m.b.f.a., where

$$\lambda_{i,k} = \rho r_{i,k}. \tag{23}$$

In view of (22), we have

$$\sum_{k=1}^{N} \lambda_{i,k} < 1, \tag{24}$$

for all $i = 1, 2, \ldots, N$. As the $N \times N$ packet-pair switch is a quasi-output-buffered switch, we then have the following universal stability result.

Theorem 15: For the Bernoulli arrival traffic described above, there exists a $\theta > 0$ such that

$$\sup_t \mathsf{E} e^{\theta Q(t)} < \infty, \tag{25}$$

where $Q(t)$ is the total number of packets in the $N \times N$ packet-pair switch.

In summary, the packet-pair switch has the following nice features:

1) It achieves 100% throughput.
2) It delivers packets in the FIFO order.
3) It only contains 2×2 switches and the connection patterns of these 2×2 switches are deterministic and periodic with period 2.
4) Packets are self-routed through the network of 2×2 switches.
5) No communication and computation is needed.

B. Delay analysis

To gain some intuition on the delay performance of the packet-pair switch, let us consider the *uniform* Bernoulli traffic, i.e., $r_{i,k} = 1/N$ for all i and k in the Bernoulli traffic.

For a 2×2 switch in the *first* stage, there are N VOQs at each input. Recall that the operation of a 2×2 switch at an input is to transmit a full-framed VOQ when it is connected to the first output of the 2×2 switch. A full-framed VOQ in this case is simply a VOQ that contains at least two packets. As such, we can implement the N VOQs by two parts: the first part for storing packets that have not been"paired," and the second part for storing packets that have been "paired." For this, there are N queues with buffer size 1 in the first part, indexed from 1 to N, and two VOQs (for the two outputs of the 2×2 switch) in the second part. Suppose a packet of flow k arrives at the switch. If the k^{th} queue in the first part is empty, the arriving packet is placed in the k^{th} queue. On the other hand, if the k^{th} queue is *not* empty, the arriving packet and the packet stored in the k^{th} queue are "paired" and they can be moved to the two VOQs in the second part (at the beginning of the next frame).

In view of the two-part implementation of the N VOQs, the delay at a switch in the first stage consists of two parts: (i) the delay for "pairing" and (ii) the queueing delay for transmitting through the 2×2 switch. To compute the "pairing" delay, note that only the odd numbered packets in a flow need to wait for "pairing," and the "pairing" delay for an odd numbered packet is simply the interarrival time of the next packet. Under the uniform Bernoulli traffic, the expected interarrival time of a flow is N/ρ. Thus, the expected "pairing" delay is $N/2\rho$. For the queueing delay, we approximate the arrival process to the two VOQs in the second part by the Bernoulli arrival traffic with arrival rate ρ. As the connection pattern is periodic with period 2, this model is a special case of the uniform Bernoulli traffic model in [6] (with $N = 2$). Thus, the expected queueing delay can be approximated by $1/2(1-\rho)$. Adding these two parts of delay, the expected delay through a switch in the first stage can be approximated by

$$\frac{N}{2\rho} + \frac{1}{2(1-\rho)}. \tag{26}$$

If we approximate the arrival process to every input of a 2×2 switch in the packet-pair switch by the uniform Bernoulli traffic with arrival rate ρ, then using the same argument as that in the first stage yields the following approximation for the expected delay through a switch in the j^{th} stage:

$$\frac{N}{2^j\rho} + \frac{1}{2(1-\rho)}, \quad j = 1, 2, \ldots, n, \tag{27}$$

$$\frac{1}{2(1-\rho)}, \quad j = n+1, \ldots, 2n, \tag{28}$$

as there is no "pairing" delay for the last n stages.

Summing up the delay in (27) and (28), we can approximate the expected delay through the $N \times N$ packet-pair switch by

$$\frac{N-1}{\rho} + \frac{\log_2 N}{(1-\rho)}. \tag{29}$$

In Figure 3, we compare our approximation in (29) with computer simulation. As shown in Figure 3, our approximation (APPR) is a conservative estimate of the delay of the packet-pair switch (PP). The reason for that is the arrival process to every input of a 2×2 switch in the packet-pair switch is *not* the uniform Bernoulli traffic. In fact, it is much more regular (less random) than the uniform Bernoulli traffic. This is because "pairing" takes time and it is less likely to have two consecutive pairs with the same destination.

To reduce the "pairing" delay of the packet-pair switch in light traffic, we can also use the idea proposed in the padded frame scheme [14]. At the beginning of a frame, if there is no full-framed VOQ in an input-buffer of a switch in the first n stages, we can pad a fake packet to a VOQ with only one packet to form a padded frame (with frame size 2). Then the padded frame is transmitted inside the packet-pair switch. Clearly, it is most beneficial to generate padded frames in the first stage. The gain starts to diminish as the number of stages is increased. For this, we define a parameter n^+ as the number of stages that allow padded frames to be generated. To ensure stability, the number of padded frames inside the packet-pair switch has to be restrained. For this, we only allow padded frames to be generated when the total number of packets in the *first* input-buffer of the first switch in the $n+1^{th}$ stage does not exceed a threshold TH. Such an enhancement is called a packet-pair-plus (PP^+) switch in this paper.

C. Simulations

In this section, we perform various simulations for packet-pair switches. In all our simulations, the switch size N is chosen to be 32. Each simulation run contains 10^6 time slots. In Figure 3, we consider the uniform Bernoulli traffic model and plot the delay of the packet-pair switch (PP), the packet-pair-plus switch (PP^+), the ideal output-buffered switch (OB), the uniform frame spreading scheme (UFS) in [17], the padded frame scheme (PF) in [14], and the Contention and Reservation switch (CR) in [24]. Certainly, the output-buffered switch has the best delay performance (at the cost of N times speedup). The packet-pair switch outperforms both the UFS scheme and the padded frame scheme. It also beats the CR switch in heavy traffic. However, its delay is higher than that in the CR switch in light traffic. This is because the CR switch uses the contention mode in light traffic, while the packet-pair switch wastes a lot of time to form a frame of two packets in light traffic. In this simulation, the packet-pair-plus switch is run with $n^+ = 3$ and $TH = 2$, i.e., only the first 3 stages are allowed to generate padded frames when the total number of packets in the first input-buffer of the first switch in the 6^{th} stage does not exceed 2. The delay of the PP^+ switch is much better than that of the PP switch in light traffic and is comparable to that of the PP switch in heavy traffic. Similar results are also shown in Figure 4 under the uniform Pareto traffic model in [6].

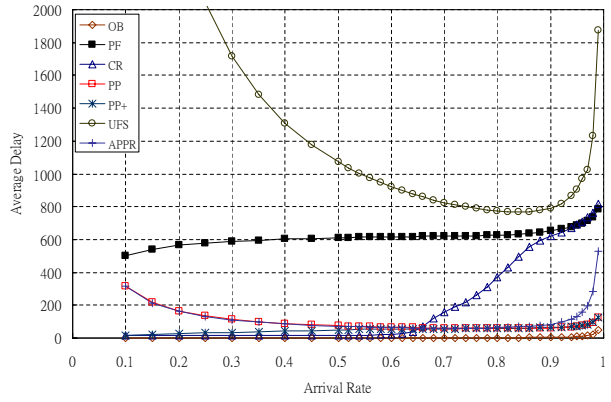

Fig. 3. Delay comparison for the uniform Bernoulli traffic model

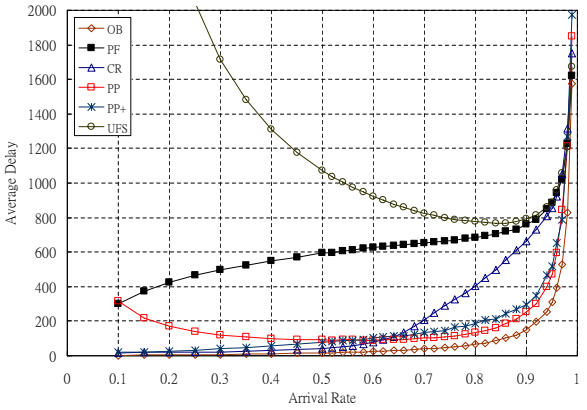

Fig. 4. Delay comparison for the uniform Pareto traffic model

V. Conclusions

In this paper, we proposed a new concept, called *quasi-output-buffered switch*. Like an output-buffered switch, a quasi-output-buffered switch is a deterministic switch that delivers packets in the FIFO order, and achieves 100% throughput. Using the three-stage Clos network, we showed that one can *recursively* construct a larger quasi-output-buffered switch with a set of smaller quasi-output-buffered switches. By recursively expanding the three-stage network, we obtained a packet-pair switch with only 2×2 switches. By computer simulations, we showed that packet-pair switches have better delay performance than most load-balanced switches with comparable construction complexity.

There are several problems that require further study:
(i) As argued in (29), the $N \times N$ packet-pair switch has $O(N)$ delay. It is shown in [22] that it is possible to obtain $O(\log N)$ delay in an $N \times N$ input-buffered crossbar switch (though this is at the cost of non-scalable communication and computation overheads by using Birkhoff-von Neumann decomposition). It would be of interest to find a scalable switch architecture that achieves $O(\log N)$ delay without any computation and communication.

(ii) We note that it is possible to replace the deterministic 2×2 switches in a packet-pair switch by fixed interconnecting networks. As such, one might be able to embed a packet-pair switch inside a fixed interconnecting network, e.g., a DWDM network. The problem is then how to do this efficiently.

References

[1] H. Ahmadi and W. E. Denzel, "A survey of modern high-performance switching techniques," *IEEE Journal of Selected Areas in Communications*, Vol 7. pp. 1091-1103, 1989.

[2] V. E. Benes. *Mathematical Theory of Connecting Networks and Telephone Traffic*. New York: Academic Press, 1965.

[3] C.-S. Chang, "Stability, queue length and delay of deterministic and stochastic queueing networks," *IEEE Trans. on Automatic Control*, Vol. 39, pp. 913-931, 1994.

[4] C.-S. Chang, J. Cheng, D.-S. Lee and C.-F. Wu, "Quasi-output-buffered switches," *Techincal Report*, 2007. Available from http://www.ee.nthu.edu.tw/~cschang/quasioutput.pdf.

[5] C.-S. Chang and D.-S. Lee, "Quasi-circuit switching and quasi-circuit Switches," *Proceedings of IEEE ITRE 2005*.

[6] 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, pp. 611-622, 2002.

[7] C.-S. Chang, D.-S. Lee, and Y.-J. Shih, "Mailbox switch: a scalable two-stage switch architecture for conflict resolution of ordered packets," *Proceedings of IEEE INFOCOM*, Vol. 3, pp. 1995-2006, Hong Kong, 2004.

[8] C.-S. Chang, J. A. Thomas, and S.-H. Kiang, "On the stability of open networks: a unified approach by stochastic dominance," *Queueing Systems*, Vol. 15, pp. 239-260, 1994.

[9] H. J. Chao, J. Song, N. S. Artan, G. Hu, and S. Jiang, "Byte-focal: a practical load-balanced switch," *IEEE High Performance Switching and Routing*, 2005.

[10] N. Chrysos and M. Katevenis, "Scheduling in non-blocking buffered three-stage switching fabrics," *Proceedings of IEEE INFOCOM*, 2006.

[11] C. Clos, "A study of nonblocking switching networks," *BSTJ*, Vol. 32, pp. 406-424, 1953.

[12] R. L. Cruz, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Tran. Inform. Theory*, Vol. 37, pp. 114-131, 1991.

[13] S. Iyer and N. McKeown, "Making parallel packet switch practical," *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, U.S.A.

[14] J.-J. Jaramillo, F. Milan, R. Srikant, "Padded frames: a novel algorithm for stable scheduling in load-balanced switches." Technical report from University of Illinois at Urbana-Champaign.

[15] S.-T. Chuang, A. Goel, N. McKeown and B. Prabhkar, "Matching output queueing with a combined input output queued switch," *Proceedings of IEEE INFOCOM*, pp. 1169-1178, New York, 1999. 1997.

[16] F. P. Kelly, "Notes on effective bandwidths," *Stochastic networks: Theory and Applications,* pp. 141-168, Oxford University Press, 1995.

[17] I. Keslassy, S.-T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, "Scaling internet routers using optics," *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[18] D. V. Lindley, "The theory of queues with a single server," *Proc. Camb. Phil. Soc.*, Vol. 48, pp. 277-289, 1952.

[19] J. D. C. Little, "A proof for the queueing formula $L = \lambda W$," *Operations Research*, Vol. 16, pp. 651-665, 1961.

[20] R. M. Loynes, "The stability of a queue with non-independent inter-arrival and service times," *Proc. Camb. Phil. Soc.*, Vol. 58, pp. 497-520, 1962.

[21] N. McKeown, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch," *Proceedings of IEEE INFOCOM*, pp. 296-302, 1996.

[22] M. J. Neely, E. Modiano, and Y.-S. Cheng, "Logarithmic delay for $N \times N$ packet switches under the crossbar constraint," *IEEE Transactions on Networking*, 2007.

[23] I. Stoica and H. Zhang, "Exact emulation of an output queueing switch by a combined input output queueing switch," *IEEE IWQoS'98*, pp. 218-224, Napa, California, 1998.

[24] C.-L. Yu, C.-S. Chang, and D.-S. Lee, "CR switch: a load-balanced switch with contention and reservation," *Proceedings of IEEE INFOCOM*, 2007.