Quasi-circuit Switching and Quasi-circuit Switches

Cheng-Shang Chang and Duan-Shin Lee
Institute of Communications Engineering
National Tsing Hua University
Hsinchu 300, Taiwan, R.O.C.
Email: cschang@ee.nthu.edu.tw
lds@cs.nthu.edu.tw

Abstract

In this paper we developed the theory of quasi-circuit switching and the associated methods of building quasi-circuit switches. Quasi-circuit switching is between packet switching and circuit switching. In a quasi-circuit switched network, time is partitioned into frames. Quasi-circuit switches then ensure that for any two packets with the same input and output links, the difference of their arriving frames is the same as the difference of their departing frames. By so doing, traffic can be isolated between frames and be multiplexed within a frame. As such, quasi-circuit switching can provide quality of services at the time scale of frames while achieving statistical multiplexing gain within frames.

Quasi-circuit switches are more scalable than packet switches and circuit switches built with the same switch architecture. We show that quasi-circuit switches can be easily built based on the classical switch architectures, such as the shared memory switch architecture and the crosspoint buffered switch architecture. However, these architectures do not scale in the domain of high speed switching. In the paper we propose several *scalable* architectures, including the load balanced Birkhoff-von Neumann quasi-circuit switch, the Clos quasi-circuit switch and the Benes quasi-circuit switch. We show that finding routing paths in the Benes quasi-circuit switches is much easier than that in the Benes circuit switches.

Another nice feature of quasi-circuit switching is that statistical QoS can be easily inferred by measuring the average link utilization. There is no need for centralized control as required in circuit switching and complicated packet scheduling as required in packet switching.

Index Terms

high speed switching, Birkhoff-von Neumann switches, multi-stage switches, quality of services

I. Introduction

There are two fundamental approaches towards building core networks: circuit switching and packet switching. In circuit-switched networks, resources, including bandwidth and buffers, are reserved along a path for the duration of communication. As such, quality of service (QoS) is

This research is supported in part by the National Science Council, Taiwan, R.O.C., under Contract NSC-91-2219-E007-003, and the program for promoting academic excellence of universities 89-E-FA04-1-4.

easily guaranteed. However, as resources are reserved for dedicated use, resources are not used efficiently in circuit-switched networks. On the other hand, as there is no resource reservation in packet-switched networks, resources could be used more efficiently in packet-switched networks. However, it is much more difficult to provide QoS in packet-switched networks.

It would be nice to have the advantages from both packet switching and circuit switching. For this, we propose the concept of *quasi-circuit switching*. Quasi-circuit switching is a generalization and an abstraction of the stop-and-go queueing proposed by Golestani [8], [9]. As in [8], [9], time in a quasi-circuit switched network is partitioned into frames. Flows (such as virtual circuits in ATM [10] and traffic trunks in MPLS [12]) entering a quasi-circuit switched network are rate controlled so that the number of bits of each flow in every frame is always bounded.

The key components in a quasi-circuit switched network are *quasi-circuit switches*. As long as the total rate at every link of a quasi-circuit switch does not exceed its link capacity, a quasi-circuit switch has the following property:

(P1) For any two packets with the same input and output links, the difference of their arriving frames is the same as the difference of their departing frames. If such a difference is (resp. bounded by) d, then the switch is called a quasi-circuit switch with exact (resp. maximum) frame delay d.

A quasi-circuit switch is order-preserving if it further satisfies the following property:

(P2) Packets that arrive during the same frame and have the same input and output links depart in the First Come First Serve (FCFS) order.

These two properties can then be propagated through a quasi-circuit switched network to provide QoS guarantees. As such, one can build a quasi-circuit switch by a network of quasi-circuit switches.

In this paper, we will rehash several well-known switch architectures to quasi-circuit switches, including the shared memory switches, and the crosspoint buffered switches. Unfortunately, these known switch architectures are either too simple to scale or too complicate to build. For scalable quasi-circuit architectures, we propose several multi-stage switches. The first one is the load balanced Birkhoff-von Neumann switch in [6]. In such an architecture, the memory access speed is only required to match the link speed. Moreover, it is on-line complexity is only $O(1)$.

To reduce the maximum frame delay, we propose the Clos quasi-circuit switches (with speedup). Unlike the classical Clos three-stage network, routing paths can be easily determined in the Clos quasi-circuit switches. By recursively expanding the Clos quasi-circuit switches, we derive the Benes quasi-circuit switches in which the routing path of a packet can be easily determined by the binary representation of the output of that packet. Moreover, the frame delay in the Benes quasi-circuit switch is substantially smaller than that in the load balanced Birkhoff-von Neumann quasi-circuit switch.

To increase the link utilization, one may allow packets to be dropped inside a quasi-circuit switch once the link capacity is exceeded. By so doing, one can then apply the large deviation theory to infer the statistical QoS via measuring the average link utilization. As such, there is no need for centralized control as required in circuit switching and complicated packet scheduling as required in packet switching.

II. Quasi-circuit switching

A. *Definitions and basic properties*

In this section, we introduce the concept of quasi-switching. As discussed in Section I, quasi-switching is a generalization of the stop-and-go queueing in [8]. We will follow some terminologies used in [8]. As in [8], time is partitioned into frames. In this paper, we only consider a single frame size. Let T be the universal frame time. Everything will be normalized with respect to the frame time T.

Definition 1 *A link is said to be with capacity c if it is capable of transmitting c bits in a frame.*

Definition 2 *A flow is a sequence of packets with a common source and a common destination. A flow in a link is said to be (r, T)-smooth if the total number of bits (of the packets) of that flow in every frame in that link is bounded above by r bits. The quantity r is called the (peak) rate of that flow.*

A switch that has M input links and N output links is called an $M \times N$ switch. In this paper, we assume that all the links in a switch are *synchronized* so that all the frames at the input/output links start at the same time. In other words, there is no phase mismatch between any two links of a switch (the case with phase mismatch can be easily compensated by adding a delay constant as in [8]).

Consider an $M \times N$ switch. Let c_i^I, $i = 1, \ldots, M$, be the capacity of the i^{th} input link and c_k^O, $k = 1, \ldots, N$, be the capacity of the k^{th} output link. As there are M inputs and N outputs, there are MN local flows in an $M \times N$ switch. Let $A_{i,k}$ be the local flow for the packets that arrive at the i^{th} input and are destined to the k^{th} output of the $M \times N$ switch.

Definition 3 *The inputs of an $M \times N$ switch is said to satisfy the no overbooking condition if (A1), (A2) and (A3) listed below hold.*

(A1) *For all $i = 1, \ldots, M$, $k = 1, \ldots, N$, the local flow $A_{i,k}$ is $(r_{i,k}, T)$-smooth when it arrives at the i^{th} input of the switch.*

(A2) *The total rate coming out from an input link does not exceed its capacity, i.e.,*

$$\sum_{k=1}^{N} r_{i,k} \leq c_i^I, \quad i = 1, \ldots, M. \tag{1}$$

(A3) *The total rate going to an output link does not exceed its capacity, i.e.,*

$$\sum_{i=1}^{M} r_{i,k} \leq c_k^O, \quad k = 1, \ldots, N. \tag{2}$$

An $M \times N$ switch is called a quasi-circuit switch *with maximum frame delay d if it has the following property when its inputs satisfy the no overbooking condition:*

(P1) *For any two packets of the same local flow, the difference of their arriving frames is the same as the difference of their departing frames. Moreover, such a difference is bounded above by d. To be precise, let m_1 and m_2 (resp. $\tilde{m}_1$ and $\tilde{m}_2$) be the arriving frame and the departing frame of the first (resp. second) packet. Then*

$$m_2 - m_1 = \tilde{m}_2 - \tilde{m}_1 \leq d. \tag{3}$$

We will also call a quasi-circuit switch with exact *frame delay d if the inequality in (3) is replaced by an equality. Also, a quasi-circuit switch is order-preserving if it further satisfied the following property:*

(P2) *Packets of the same local flow that arrive during the same frame depart in the FCFS order.*

To gain more intuition of quasi-circuit switching, consider the case that the frame time T is infinity. In this case, (P1) is irrelevant and quasi-circuit switching is reduced to the usual packet switching that only guarantees the order of packets if (P2) is satisfied. To see the connection between circuit switching and quasi-circuit switching, one may further partition a frame into a group of fixed size time slots. For circuit switching, assume that every packet can be transmitted in a time slot. Then circuit switching guarantees that for any two packets of the same local flow, the difference of their arriving *time slots* is the same as the difference of their departing *time slots*. This is much stronger than (P1). To summarize, quasi-circuit switching is between packet switching and circuit switching. In circuit switching, traffic is completely isolated by slot assignment. On the other hand, traffic is completely mixed in packet switching. Quasi-circuit switching uses frames to isolate traffic. As such, it can viewed as circuit switching at the time scale of frames. Thus, it can still provide some guarantees of quality of services (QoS) at the frame level. At the same time, quasi-circuit switching allows packets to be multiplexed within a frame. Thus, it can achieve statistical multiplexing gain at the time scale of time slots. This is equivalent to the statistical line grouping in the circuit switching context (see e.g., [14]) and is known as the duration limited statistical multiplexing in [9].

There are several direct consequences from (P1) and (P2).

Proposition 4 *Consider an $M \times N$ quasi-circuit switch with maximum frame delay d. Suppose that its inputs satisfy the no overbooking condition.*

- (P3) *The maximum packet delay is bounded by $(d+1)T$, where T is the universal frame time.*
- (P4) *No packets are lost inside the switch.*
- (P5) *The local flow $A_{i,k}$ is $(r_{i,k}, T)$-smooth when it departs from the k^{th} output link of the switch.*
- (P6) *If the quasi-circuit switch is order-preserving, then packets of the same local flow depart in the FCFS order.*

Proof. The proofs for (P3) and (P4) are rather straightforward and thus omitted.

Now we prove (P5). As we assume that the local flow $A_{i,k}$ is $(r_{i,k}, T)$-smooth when it arrives at the i^{th} input of the switch, there are at most $r_{i,k}$ bits (of the packets) of that local flow in every frame. From (3), we also know that packets of the same local flow depart during the same frame if and only if they arrive during the same frame. Thus, there are at most $r_{i,k}$ bits (of the packets) of the local flow $A_{i,k}$ in every frame of the k^{th} output link. This then implies the local flow $A_{i,k}$ is also $(r_{i,k}, T)$-smooth when it departs from the k^{th} output link of the switch.

To show (P6), we consider two packets of the same local flow. Let m_1 and m_2 (resp. $\tilde{m}_1$ and $\tilde{m}_2$) be the arriving frame and the departing frame of the first (resp. second) packet. If $m_1 = \tilde{m}_1$, we know from (P2) that these two packets depart in the FCFS order. On the other hand, if $m_1 < \tilde{m}_1$, then we have from (P1) that $m_2 < \tilde{m}_2$. Thus, these two packets departs in the FCFS order. ■

In the following examples, we provide several methods for constructing quasi-circuit switches.

Example 5 (Stop-and-go queueing in shared memory switches) The simplest way to build a quasi-circuit switch is to use the *shared memory* architecture. Consider an $N \times N$ switch with an identical link capacity c. For a shared memory switch architecture (see Figure 1), packets that arrive during the same frame from all input links are read into a common shared memory. In the next frame, a central controller then writes all those packets (in the FCFS order for order-preserving) to the destined output links (according to an address lookup table). This is known as the stop-and-go queueing in [8].

During a frame time, the shared memory has to read all the packets from the current frame and write all the packets from the previous frame. Thus, its memory access speed must be at least $2Nc$. As such, the shared memory architecture is not *scalable* if the number of input/output links is large.

Note that under the no overbooking condition, packets that arrive at the same frame will all depart in the next frame. Thus, the frame delay in the shared memory switch is exactly one frame time. Also, no packets are lost inside the switch if the buffer is larger than Nc bits.

Example 6 (Crosspoint buffered switches) One way to solve the memory access speed problem is to use the crosspoint buffered switch. As shown in Figure 2, a crosspoint buffered switch consists of a crossbar switch fabric and separate buffers at the crosspoints. For an $N \times N$ switch, the number of separate buffers is N^2. Each buffer at a crosspoint stores packets from one input to one output. Call the $(i, j)^{th}$ buffer for the buffer at the crosspoint of the i^{th} input and the j^{th} output. To build an $N \times N$ quasi-circuit switch with capacity c, we allocate each buffer with buffer size larger than c. As the shared memory architecture, there are two phases of memory access. During the first phase of a frame, the $(i, j)^{th}$ buffers, $j = 1, 2, \ldots, N$, are connected to the i^{th} input link, $i = 1, 2, \ldots, N$. An arriving packet from the i^{th} input link and destined for the j^{th} output link is then placed in the $(i, j)^{th}$ buffer. During the second phase of a frame, the $(i, j)^{th}$ buffers, $i = 1, 2, \ldots, N$, are connected to the j^{th} output link, $j = 1, 2, \ldots, N$. Packets that are stored in the $(i, j)^{th}$ buffers, $i = 1, 2, \ldots, N$, are then read out sequentially to the j^{th} output link. Clearly, under the no overbooking condition, no packets are lost inside the switch and the frame delay is exactly one frame time. Moreover, the memory access speed for this architecture is only required to be twice of the link capacity. However, as pointed out in [1], the architecture based on crosspoint buffers does not scale for a large number of input/output links because of the square growth. It is only good either for small switches, or as basic architecture for the building blocks of modular multi-stage switches.

Example 7 (Load balanced Birkhoff-von Neumann quasi-circuit switches) Both switch architectures in Example 5 and Example 6 do not scale when the number of input/output ports is large. Recently, a much more scalable switch architecture, called the load balanced Birkhoff-von Neumann switch, is proposed (see e.g., [6] and references therein). In such an architecture (see Figure 3), there are two crossbar switch fabrics and buffers between these two crossbar switch fabrics. To build an $N \times N$ quasi-circuit switch with capacity c by this architecture, one first partitions time into fixed size frames. Both the $N \times N$ crossbar switches in Figure 3 have identical connection patterns for all time and change their connection patterns simultaneously at the beginning of every frame. The connection patterns are set up corresponding to a circular-shift matrix. As such, every input is connected to every output exactly once in every N frames.

There are N central buffers between these two switch fabrics. Each central buffer consists of two alternating memory blocks as the double-queue structure in [8]. The buffer size of each memory block is Nc bits, which is divided into N bins (indexed from 1 to N), each with buffer size of c bits. The j^{th} bin in every memory block is used for buffering packets that are destined for the j^{th} output, $j = 1, 2, \ldots, N$.

When a packet arrives at an input, it is immediately transferred to one of the central buffers that is connected to that input. According to the destination of the packet, the packet is placed in the bin that corresponds to the destination of that packet. When the bin is connected to the output, packets in that bin are all transferred to that output. It is shown in [6] that under the no overbook condition, the maximum packet delay is bounded by $2N - 1$ frame times. As such, the switch architecture described in [6] is a *quasi-circuit switch* with maximum frame delay $2N - 1$. The key feature of this architecture is its simplicity. Its on-line complexity is $O(1)$ and it is independent of the number of input/output links. Such complexity is much lower than input-buffered packet switches (see e.g., [16], [17], [21], [13], [2], [15]). However, its frame delay is $O(N)$. In Section II-B and Section III-A below, we will show how to reduce the frame delay by building networks of quasi-circuit switches.

B. *Networks of quasi-circuit switches*

In this section, we consider a network interconnected by quasi-circuit switches. We assume that *the propagation delay to traverse through a link* is compensated so that all the frames in the input links of a quasi-circuit switch start at the same time. For the ease of the presentation, we assume that the propagation delay to traverse through a link is a positive integer multiple of Δ for some $\Delta > 0$. The constant Δ has nothing to do with the frame time T. It is only used as an induction step for the proof of Theorem 8 below.

In Section II, only local flows are considered from input links to output links in a switch. In addition to local flows, we consider end-to-end flows that traverse through a series of links.

Suppose that there are I links in the network, indexed from 1 to I, and J end-to-end flows, indexed from 1 to J. Let c_i, $i = 1, \ldots, I$, be the capacity of the i^{th} link. Suppose that the j^{th} end-to-end flow is (r_j, T)-smooth when it arrives at the network. Let $D = (D_{i,j})$ be the $I \times J$

routing matrix, i.e., $D_{i,j} = 1$ if the j^{th} flow traverses through link i and $D_{i,j} = 0$ otherwise. Also, let $\mathbf{r}$ be the $J \times 1$ column vector with its j^{th} element being r_j, and $\mathbf{c}$ be the $I \times 1$ column vector with its i^{th} element being c_i. Note that the total rate at link i does not exceed its capacity if

$$\sum_{j=1}^{J} D_{i,j} r_j \le c_i. \tag{4}$$

Thus, the condition that the total rate at every link does not exceed its capacity can be written in the following matrix form:

$$D\mathbf{r} \le \mathbf{c}. \tag{5}$$

To summarize, we make the following assumptions:

(A1$'$) The j^{th} end-to-end flow is (r_j, T)-smooth when it arrives at the network (the first link of its route), $j = 1, \ldots, J$.

(A2$'$) The total rate at every link does not exceed its capacity, i.e., the inequality in (5) holds.

(A3$'$) The propagation delay to traverse a link is a positive integer multiple of Δ for some $\Delta > 0$.

Theorem 8 *Assume that (A1$'$)-(A3$'$) hold. Then the network of quasi-circuit switches has the following properties.*

(P1$'$) *For any two packets of the same end-to-end flow, the difference of their arriving frames at the network is the same as the difference of their departing frames at every link traversed by the flow.*

(P2$'$) *If every quasi-circuit switch is order-preserving, then packets of the same end-to-end flow that arrive at the network during the same frame depart in the FCFS order at every link traversed by the flow.*

(P3$'$) *The maximum delay for packets of an end-to-end flow is bounded by the sum of the propagation delay of each link and the maximum delay of each quasi-circuit switch along its route.*

(P4$'$) *No packets are lost inside the network.*

(P5$'$) *For $j = 1, \ldots, J$, the j^{th} end-to-end flow is (r_j, T)-smooth at every link traversed by the flow.*

(P6$'$) *If every quasi-circuit switch is order-preserving, then packets of the same end-to-end flow depart the network in the FCFS order.*

Proof. The key is to prove (P5$'$) by induction on time under (A1$'$)-(A3$'$). At time 0, the traffic of end-to-end flows only arrive at the network (as we assume the propagation delay of a link is positive in (A3$'$)). Thus, the induction hypothesis (P5$'$) holds trivially from (A1$'$).

Now suppose the induction hypothesis hold up to time $n\Delta$. From the induction hypothesis and (A2$'$), it follows that the no overbooking condition in (A1)-(A3) in Definition 3 are satisfied for every quasi-circuit switch in the network. Using the same argument in the proof of (P5) in Proposition 4, one can easily show that the induction hypothesis (P5$'$) holds for every frame up to time $(n+1)\Delta$ (as we assume the propagation delay of a link is positive in (A3$'$)).

That (P1$'$)-(P4$'$) and (P6$'$) hold follows trivially from (P1)-(P4) and (P6) in Definition 3 and Proposition 4. ■

Define the maximum jitter of an end-to-end flow as the difference between its maximum packet delay and its minimum delay through the network.

Corollary 9 *Under the conditions in Theorem 8, the maximum jitter of an end-to-end flow is bounded above by* $2T$.

Proof. Note from (P1$'$) that for any two packets of the same end-to-end flow, the difference of their arriving frames at the network is the same as the difference of their departing frames from the network. This then implies that the jitter is bounded by $2T$. ■

III. Recursive construction of quasi-circuit switches

A. Clos quasi-circuit switches

Theorem 8 suggests a way to construct a large quasi-circuit switch via a network of smaller quasi-circuit switches. Our first idea is to use the three-stage Clos network architecture (see e.g., [11], [20], [7]). To do this, we further assume in this section that *packets are of the same size*. Under such an assumption, each frame can be further partitioned into a group of time slots so that a packet can be transmitted within a time slot. Moreover, both the link capacity and the (peak) rate of a flow can be represented by the number of packets in a frame.

As shown in Figure 4, there are three stages of quasi-circuit switches. The first stage consists of q $p \times p$ quasi-circuit switches with exact frame delay d_1. Every input (resp. output) link of these switches has capacity c_1 (resp. c_2). That is, the maximum number of packets that can be transmitted in a frame in an input (resp. output) link of the switches at the first stage is c_1 (resp. c_2). Every one of the p outputs from a switch at the first stage is connected to an input of the p $q \times q$ quasi-circuit switches with exact frame delay d_2 at the second stage. Every one of the q outputs from a switch at the second stage is connected to an input of the q $p \times p$ quasi-circuit switches with exact frame delay d_3 at the third stage. Every input (resp. output) link of the switches at the third stage has capacity c_2 (resp. c_1). In this three-stage switch, there are pq inputs and pq outputs.

Number the switches at the first stage from 1 to p. Do the same for the switches at the third stage. As such, we can use the $(i, \ell)^{th}$ input, $i = 1, 2, \ldots, q$, $\ell = 1, 2, \ldots, p$, to denote the ℓ^{th} input of the i^{th} switch at the first stage. Similarly, we use the $(j, \ell)^{th}$ output, $j = 1, 2, \ldots, q$, $\ell = 1, 2, \ldots, p$, to denote the ℓ^{th} output of the j^{th} switch at the third stage. The flow from the $(i, \ell_1)^{th}$ input to the $(j, \ell_2)^{th}$ output is denoted by the $(i, \ell_1) \to (j, \ell_2)$ flow. Let $r_{(i,\ell_1),(j,\ell_2)}$ be (peak) rate of the $(i, \ell_1) \to (j, \ell_2)$ flow, i.e., the maximum number of packets of the $(i, \ell_1) \to (j, \ell_2)$ flow in a frame.

The key problem of the three-stage Clos network is *routing*. Traditionally, finding the routing path in such a network requires running sophisticated algorithms that may not be scalable. For instance, in the case without speedup, i.e., $c_2 = c_1$, finding the routing path for every packet relies on the well-known Birkhoff-von Neumann decomposition (see e.g., [4] and references therein), and such a algorithm does not scale for a switch with a large number of input/output ports. The nice thing of the quasi-circuit switches is that routing can be solved *easily* at the cost of speedup, i.e., $c_2/c_1 > 1$. When $c_1 >> pq$, the cost of speedup is relatively small.

The idea is to use the round-robin routing policy for load balancing at the first stage. We consider the following two round-robin splitting policies.

Round-robin splitting:

(R1) Round-robin splitting for every flow at every input link: the k^{th} packet of the $(i, \ell_1) \to (j, \ell_2)$ flow in a frame is routed to the $(k \mod p) + 1^{th}$ output at the first stage.

(R2) Round-robin splitting for every *aggregated* flow from all the input links at a switch of the first stage: consider the aggregated $i \to j$ flow from multiplexing all the $(i, \ell_1) \to (j, \ell_2)$ flows, $\ell_1, \ell_2 = 1, \ldots, p$. The k^{th} packet of the aggregated $i \to j$ flow in a frame is routed to the $(k \mod p) + 1^{th}$ output at the first stage.

Note that there is a unique path for every packet to its destination from the second stage onward. For both round-robin routing policies, a packet destined to the $(j, \ell_2)^{th}$ output is routed to the j^{th} output at the second stage and the ℓ_2^{th} output at the third stage.

Clearly, the first scheme is much easier to do than the second one as it only needs the information for every flow at an input link. Thus, the first scheme in (R1) can be carried out *distributively at every input link*. However, for the second scheme in (R2), one needs a centralized control at every switch of the first stage as it has to gather the information of all the flows coming into that switch. A suitable architecture for the first scheme is the crosspoint buffered switch architecture in Example 6. On the other hand, the shared memory architecture in Example 5 is more suitable for the second scheme as every packet in the shared memory architecture is processed sequentially. As we shall see in the following theorem, the needed speedup for the first scheme is substantially larger than that of the second scheme.

Theorem 10 (i) *Suppose that the three-stage Clos network in Figure 4 is operated under the round-robin splitting in (R1). If*

$$c_2 \geq c_1 + p^2 q, \tag{6}$$

then the network in Figure 4 is a $pq \times pq$ quasi-circuit switch with exact frame delay $d_1 + d_2 + d_3$.

(ii) *Suppose that the three-stage Clos network in Figure 4 is operated under the round-robin splitting in (R2). If*

$$c_2 \geq c_1 + q - 1, \tag{7}$$

then the network in Figure 4 is a $pq \times pq$ quasi-circuit switch with exact frame delay $d_1 + d_2 + d_3$.

To ensure that the quasi-circuit switch is order-preserving, one may add a resequencing element after each output link from the third stage. The resequencing element only needs to reorder packets in the same frame and causes an additional frame delay.

Proof. (i) For this network of quasi-circuit switches, the $(i,\ell_1) \to (j,\ell_2)$ flow is now split into p sub-flows as there are p routing paths from the $(i,\ell_1)^{th}$ input to the $(j,\ell_2)^{th}$ output (with each of them traversing through a central switch). As the routing policy is round-robin, the (peak) rate of each sub-flow is then bounded above by $\lceil \frac{r_{(i,\ell_1),(j,\ell_2)}}{p} \rceil$. Since all the quasi-circuit switches at the same stage have the same frame delay, it follows from (P1) that packets of these p sub-flows in the m^{th} frame from the $(i,\ell_1)^{th}$ input link will arrive at the $(j,\ell_2)^{th}$ output link in the $m+d_1+d_2+{d_3}^{th}$ frame if every link inside the network is not overbooked. To see that the no overbooking condition is satisfied, we consider a particular output link at the i^{th} switch of the first stage. Let M be the total number of packets that can be routed to that link in a frame. Note that there are p^2q sub-flows traversing through that link and thus

$$M \le \sum_{\ell_1=1}^{p}\sum_{j=1}^{q}\sum_{\ell_2=1}^{p}\lceil \frac{r_{(i,\ell_1),(j,\ell_2)}}{p} \rceil \le p^2q + \sum_{\ell_1=1}^{p}\sum_{j=1}^{q}\sum_{\ell_2=1}^{p}\frac{r_{(i,\ell_1),(j,\ell_2)}}{p}. \tag{8}$$

From the nooverbooking condition for every input link of the first stage switches, we have

$$\sum_{j=1}^{q}\sum_{\ell_2=1}^{p} r_{(i,\ell_1),(j,\ell_2)} \le c_1. \tag{9}$$

Using (9) and the assumption in (6), we then have

$$M \le p^2q + c_1 \le c_2. \tag{10}$$

The argument for the output links at the second switch is similar.

(ii) The proof is quite similar to that in (i). It suffices to show that the no overbooking condition is satisfied for every link inside the network. Let $r_{i,j} = \sum_{\ell_1=1}^{p}\sum_{\ell_2=1}^{p} r_{(i,\ell_1),(j,\ell_2)}$ be the rate of the $i \to j$ aggregated flow. Consider a particular output link at the i^{th} switch of the first stage. Let M be the total number of packets that can be routed to that link in a frame. As the routing policy is round-robin for the aggregated flows, we then have

$$M \le \sum_{j=1}^{q}\lceil \frac{r_{i,j}}{p} \rceil \le q-1+\lceil \sum_{j=1}^{q}\frac{r_{i,j}}{p} \rceil. \tag{11}$$

From the nooverbooking condition in (9), we have

$$\sum_{j=1}^{q} r_{i,j} \le pc_1. \tag{12}$$

Using (12) and the assumption in (7), we then have

$$M \le q - 1 + c_1 \le c_2. \tag{13}$$

The argument for the output links at the second switch is similar.

■

In particular, if we choose $p = q = \sqrt{N}$ and $c_1 = N^{\frac{3}{2}}$ for the round-robin splitting scheme in (R1), then $c_2 = 2N^{\frac{3}{2}} = 2c_1$. Thus, for this case, a speedup factor of 2 is enough to build an $N \times N$ quasi-circuit switch by the three-stage architecture. By choosing a larger frame time T, one can further increase c_1 so that the speedup factor can be further reduced. For instance, if $c_1 = kN^{\frac{3}{2}}$, then $c_2/c_1 = 1 + \frac{1}{k}$. However, reducing the needed speedup is at the cost of delay as now the frame time is increased by k times.

B. Benes quasi-circuit switches

Another interesting case of the Clos networks is when the total number of input/output links N is a power of 2. In this case, one can construct a Benes network by using 2×2 quasi-circuit switches. In Figure 5, one first constructs a three-stage Clos network with 2×2 quasi-circuit switches at the first stage and the last stage. Every input link at the first stage and every output link at the third stage has capacity c_1. The second stage consists of two $\frac{N}{2} \times \frac{N}{2}$ quasi-circuit switches with link capacity c_2. As N is a power of 2, each of the two $\frac{N}{2} \times \frac{N}{2}$ quasi-circuit switches in Figure 5 can be further implemented by a three-stage Clos network with 2×2 quasi-circuit switches at the first stage and the last stage, and two $\frac{N}{4} \times \frac{N}{4}$ quasi-circuit switches with link capacity c_3 at the second stage. One can expand the $\frac{N}{4} \times \frac{N}{4}$ switches recursively and we obtain a multi-stage switch with all 2×2 quasi-circuit switches. In Figure 6, we show an 8×8 Benes quasi-circuit switch. For an $N \times N$ Benes network, it is well-known that the number of stages of is $2\log_2 N - 1$ and the number of 2×2 quasi-circuit switches is $N \log_2 N - \frac{N}{2}$.

Let $n = \log_2 N$. For the round-robin splitting scheme in (R1), one can easily verify from the recursive equation that one needs

$$c_{j+1} = c_j + \frac{4N}{2^j}, \quad j = 1, 2, \ldots, n-1. \tag{14}$$

As such, the link capacity of the most inner 2×2 quasi-circuit switch is

$$c_n = c_1 + 4N - 8. \tag{15}$$

On the other hand, for the round-robin splitting scheme in (R2), one needs that

$$c_{j+1} = c_j + \frac{N}{2^j} - 1, \quad j = 1, 2, \ldots, n-1. \tag{16}$$

As such, the link capacity of the most inner 2×2 quasi-circuit switch is

$$c_n = c_1 + N - \log_2 N - 1. \tag{17}$$

Unlike the classical Benes network, finding the routing path in the Benes quasi-circuit switch is very easy. Index the output links at the last stage from $0, 1, 2, \ldots, N-1$. Consider a packet that is destined for output j (at the last stage). Let $b_1 b_2 \ldots b_n$ be the n-bit binary representation of j, e.g., 100 for $j = 4$ in Figure 6. This binary representation is added in the packet header. As in the three-stage Clos quasi-circuit switch, the routing policy used in the Benes quasi-circuit switch consists of two parts: self-routing and load-balancing.

(i) Self-routing: when the packet arrives at stage $n - 1 + j$, $j = 1, 2, \ldots, n$, the packet is routed to the upper link if $b_j = 0$ and the lower link if $b_j = 1$.

(ii) Load-balancing: when the packet arrives at stage j, $j = 1, 2, \ldots, n-1$, and it is the k^{th} packet in that frame that has the same $b_1 b_2 \ldots b_{n-j+1}$'s in its packet header, then the packet is routed to the upper link if k is even, and the lower link if k is odd. This can be implemented by keeping a 2^{n-j+1}-bit vector at each input link of stage j. Let $\ell = b_1 2^0 + b_2 2^1 + \ldots + b_{n-j+1} 2^{n-j}$. Then the packet is routed to the upper link if ℓ^{th} bit of that bit vector is 0 and the lower link otherwise. Moreover, the ℓ^{th} bit of that bit vector is toggled after the packet is routed.

Note that for the round-robin splitting scheme in (R1), load-balancing is performed (and implemented) for the flows at each input link. On the other hand, for the round-robin splitting scheme in (R2), load-balancing is performed for the aggregated flows of all the input links.

Example 11 (Benes quasi-circuit switch) In this example, we argue that one can build a 256×256 Benes quasi-circuit switch with OC-192 input/output links using today's technology for memory. Suppose that we choose the universal frame time $T = 125\mu s$. Furthermore, the frame is partitioned into time slots, each is capable of transmitting 64 bytes. For an OC-192 link, its link capacity is 64 times of that of an OC-3 link (149.760 Mbits/sec [10]). Each OC-192 link is roughly 10 Gbits/sec and the overall capacity of this switch is then roughly 2.56 Tbits/sec.

One can easily calculate that the capacity of an OC-192 link in this setting, denoted by c_1, is 2340 time slots/frame. Suppose that we use the shared memory architecture in Example 5 to build each of the 2×2 quasi-circuit switch. As such, we may use the round robin splitting scheme in (R2). From (17), the needed speedup is $1 + \frac{N - \log_2 N - 1}{c_1} \approx 1.11$ (with $N = 256$). The needed memory access speed is around $10 + 10 + 10 * 1.11 + 10 * 1.11 = 42.2$ Gbits/sec. This is within the limit of the memory access speed in today's technology.

Note that there are $2\log_2 N - 1$ stages in the Benes network and each stage causes exactly one frame delay. Adding the additional one frame delay for resequencing after the last stage, the maximum delay is $2(\log_2 N)T = 2ms$ (if the input/output links satisfy the no overbooking condition). Note that the maximum delay in the load balanced Birkhoff-von Neumann switch architecture in Example 7 is $(2N-1)T = 63.875ms$, which is substantially larger than that in the corresponding Benes quasi-circuit switch.

IV. Statistical QoS

In this section, we show how to operate a network of quasi-circuit switches with statistical QoS. For this, we consider a network of quasi-circuit switches as described in Section II-B and assume that these quasi-circuit switches are *lossy*, i.e., packets are dropped in a quasi-circuit switch only when the total number of bits going to an output link of that switch exceeds its capacity. In addition to this, we assume that flows come and go. To be specific, assume that class j flows arrive according to a Poisson process with rate λ_j. The duration of a class j flow is a random variable with mean $1/\mu_j$ and this is independent of everything else. Then it follows from the classical theory for networks of $M/G/\infty$ queues (see e.g., [19], [18]) that the number of class j flows in any frame is a Poisson random variable with mean λ_j/μ_j. Moreover, these random variables are independent of each other.

Suppose that every class j flow is (r_j, T)-smooth and that the number of bits in a frame of a class j flow is a random variable with $\bar{r}_j$ (when the flow arrives at the network). Such random variables are also independent of everything else. Let

$$r_{\max} = \max_{1 \le j \le J} r_j \tag{18}$$

be the maximum peak rate of the flows entering the network and

$$\rho_i = \frac{\sum_{j=1}^{J} D_{i,j} \bar{r}_j \frac{\lambda_j}{\mu_j}}{c_i} \tag{19}$$

be the average utilization of link i. In the following theorem, we show how one infers the quality of service via the average link utilization.

Theorem 12 *For the model described in this section, if $\rho_i < 1$, then the probability that packets are dropped at the i^{th} link in a frame is bounded above by $\exp(-\Gamma(c_i, \rho_i, r_{\max}))$, where*

$$\Gamma(c_i, \rho_i, r_{\max}) = \frac{c_i}{r_{\max}} (\rho_i - 1 - \log \rho_i), \tag{20}$$

Note that $\log x < x - 1$ for all $x \neq 1$. Thus, $\Gamma(c_i, \rho_i, r_{\max}) > 0$ for $\rho_i < 1$. Under the condition $\rho_i < 1$, this probability that packets are dropped at the i^{th} link in a frame can be made very small if the link capacity c_i is much larger than $r_{\max}$.

Proof. Consider a particular frame at link i. Let N_j be the number of class j flows in that frame, and $R_{j,\ell}$ be the random variable that represents the number of bits of the ℓ^{th} class j flow in that frame, $\ell = 1, \ldots, N_j$, $j = 1, \ldots, J$. The event that packets are dropped at the i^{th} link in that frame is the same as the event that the i^{th} link exceeds its capacity in that frame, i.e., $\{\sum_{j=1}^{J} D_{i,j} \sum_{\ell=1}^{N_j} R_{j,\ell} > c_i\}$. Using the well-known Chernoff bound in the large deviation theory (see e.g., [3], [5] for a similar argument) and the fact that N_j's are independent Poisson random variables with mean λ_j / μ_j, it follows that

$$\log \mathsf{P}\Big(\sum_{j=1}^{J} D_{i,j} \sum_{\ell=1}^{N_j} R_{j,\ell} > c_i \Big) \le -\sup_{\theta \ge 0} [\theta c_i - \sum_{j=1}^{J} D_{i,j} \frac{\lambda_j}{\mu_j} (e^{\Lambda_j(\theta)} - 1)], \tag{21}$$

where

$$\Lambda_j(\theta) = \log \Big(\frac{\bar{r}_j}{r_j} e^{r_j \theta} + (1 - \frac{\bar{r}_j}{r_j}) \Big), \quad j = 1, \ldots, J.$$

Since flow j is (r_j, T)-smooth, it is also $(r_{\max}, T)$-smooth. Replacing $\Lambda_j(\theta)$ in (21) with

$$\log\Big(\frac{\bar{r}_j}{r_{\max}} e^{r_{\max}\theta} + (1 - \frac{\bar{r}_j}{r_{\max}})\Big)$$

yields

$$\log \mathsf{P}\Big(\sum_{j=1}^{J} D_{i,j} \sum_{\ell=1}^{N_j} R_{j,\ell} > c_i\Big) \le -\frac{c_i}{r_{\max}}(\rho_i - 1 - \log \rho_i) = -\Gamma(c_i, \rho_i, r_{\max}). \tag{22}$$

■

In view of Theorem 12, one only needs to know three things to infer the quality of a link: the average utilization ρ_i, the link capacity c_i and the maximum rate of a flow $r_{\max}$. There is no need to know the routing matrix D and the detailed statistics of each flow (the flow arrival rate λ_j, the average duration of a flow $1/\mu_j$, and the average rate $\bar{r}_j$, $j = 1, \ldots, J$). Note that the information for the average utilization of a link can be collected *locally* by measuring the average number of bits used in a frame. As such, there is no need for centralized control as required in circuit switching and complicated packet scheduling as required in packet switching.

In the following example, we illustrate how one infers the statistical QoS via the average link utilization.

Example 13 (Statistical QoS) Consider the single link case with the frame time $T = 125\mu s$. Suppose that every flow is a flow of Ethernet packets and there is at most one packet in a frame for each flow. Such an assumption seems reasonable as the rate of a flow is limited by the rate of an interface card in a host, which is in general much slower than the rate of a high speed link in a core network. Since the maximum size of an Ethernet packet is 1500 bytes, each flow is (r, T)-smooth with $r = 12000$ bits. Note that the capacity of an OC-48 link is 292320 bits/frame. The capacity of an OC-192 link is four times larger than the capacity of an OC-48 link. Similarly, the capacity of an OC-768 link is 16 times larger than the capacity of an OC-48 link. Let P_o be the bound in Theorem 12 for the probability that packets are dropped in a frame in the link. In Figure 7, we plot $\log_{10} P_o$ as a function of the average link utilization for the OC-48, OC-192 and OC-768 links. Suppose that we would like to operate the link so that

$\mathsf{P}_o \leq 10^{-9}$. Then for an OC-48 link, the average utilization cannot exceed 19%. For an OC-192 link, the average utilization can be raised to 48%. For an OC-768 link, the average utilization can be even raised to 70%. Now suppose we lower the requirement to $\mathsf{P}_o \leq 10^{-6}$. Then the average utilization for an OC-48 (resp. OC-192,OC-768) link is roughly 27% (resp. 59%,75%).

V. Conclusion

In this paper we developed the theory of quasi-circuit switching and the associated methods of building quasi-circuit switches. In comparison with packet switching and circuit switching, quasi-circuit switching has the following nice features:

(i) Quasi-circuit switching can be viewed as circuit switching at the time scale of frames. As such, QoS can still be provided at the frame level.

(ii) Quasi-circuit switching allows packets to be multiplexed within a frame. As such, it can achieve statistical multiplexing gain within frames.

(iii) Quasi-circuit switches are much more scalable than packet switches and circuit switches in the context of high speed switching. In particular, the load balanced Birkhoff-von Neumann quasi-circuit switch has the on-line complexity of $O(1)$. Such complexity is much lower than input-buffered packet switches (see e.g., [16], [17], [21], [13], [2], [15]). Moreover, finding routing paths in the Benes network of quasi-circuit switches is much easier than the corresponding Benes network of circuit switches.

(iv) By allowing packets to be dropped in a quasi-circuit switch, statistical QoS can be easily inferred by measuring the average link utilization. There is no need for centralized control as required in circuit switching and complicated packet scheduling as required in packet switching.

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] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi and F. Neri, "On the throughput of input-queued cell-based switches with multicast traffic," *Proc. IEEE INFOCOM'01*, pp. 1664-1672, 2001.

[3] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn, "Effective Envelopes: Statistical Bounds on Multiplexed Traffic in Packet Networks," *Proc. IEEE INFOCOM 2000*, Vol. 3, pp. 1223-1232, Tel Aviv, Israel, 2000.

[4] C.-S. Chang, W.-J. Chen and H.-Y. Huang, "Birkhoff-von Neumann input buffered crossbar switches," *Proc. IEEE INFOCOM 2000*, pp. 1614-1623, Tel Aviv, Israel, 2000.

[5] C.-S. Chang, Y.-M. Chiu and W.-M. Song, "On the performance of multiplexing independent regulated inputs," *ACM Sigmetrics 2001/Performance 2001*, pp. 184-193, 2001

[6] C.-S. Chang, D.-S. Lee, and C.-Y. Yue, "Providing guaranteed rate services in the load balanced Birkhoff-von Neumann switches," *Proc. IEEE INFOCOM*, 2003.

[7] H. J. Chao, C. H. Lam and E. Oki. *Broadband Packet Switching Technologies: A Practical Guide to ATM Switches and IP Routers*. John Wiley & Sons, Inc., 2001.

[8] S. J. Golestani, "Congestion-free communication in high speed packet networks," *IEEE Transactions on Communications*, Vol. 39, pp. 1802-1812, 1991.

[9] S. J. Golestani, "Duration-limited statistical multiplexing of delay sensitive traffic in packet networks," *Proc. IEEE INFOCOM*, pp. 323-332, 1991.

[10] R. Händel, M. N. Huber, S. Schröder. *ATM Networks: Concepts, Protocols, Applications*. Addison-Wesley Publishers Ltd., 1994.

[11] J. Hui, *Switching and Traffic Theory for Integrated Broadband Networks.* Boston: Kluwer Academic Publishers, 1990.

[12] (IETF RFC 2702) D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell and J. McManus, "Requirements for Traffic Engineering Over MPLS," Informational, RFC 2702, September, 1999.

[13] S. Li and N. Ansari, "Input-queued switching with QoS guarantees," *IEEE INFOCOM'99*, pp. 1152-1159, New York, 1999.

[14] S.-Y. R. Li. *Algebraic Switching Theory and Broadband Applications*. Academic Press, 2001.

[15] Y. Li, S. Panwar and H. J. Chao, "On the performance of a dual round-robin switch," *Proc. IEEE INFOCOM'01*, pp. 1688-1697, 2001.

[16] N. McKeown, "Scheduling algorithms for input-queued cell switches," *PhD Thesis. University of California at Berkeley*, 1995.

[17] N. McKeown, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch," *Proc. IEEE INFOCOM'96*, pp. 296-302, 1996.

[18] R. Nelson, *Probability, Stochastic Processes, and Queueing Theory: the Mathematics of Computer Performance Modeling*. Springer-Verlag: New York, 1995.

[19] K. W. Ross. *Multiservice Loss Models for Broadband Telecommunication Networks*. Springer-Verlag: New York, 1995.

[20] M. Schwartz, *Broadband Integrated Networks*. New Jersey: Prentice Hall, 1996.

[21] 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.

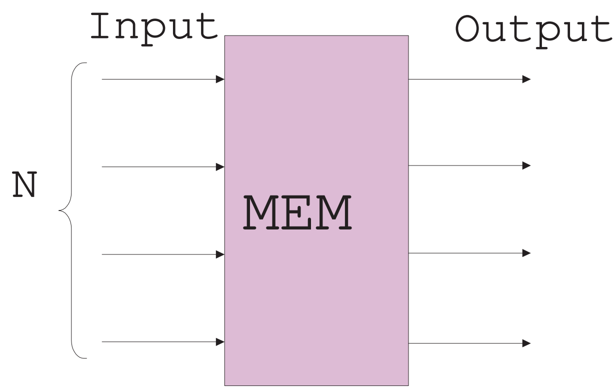

Fig. 1. A shared memory switch

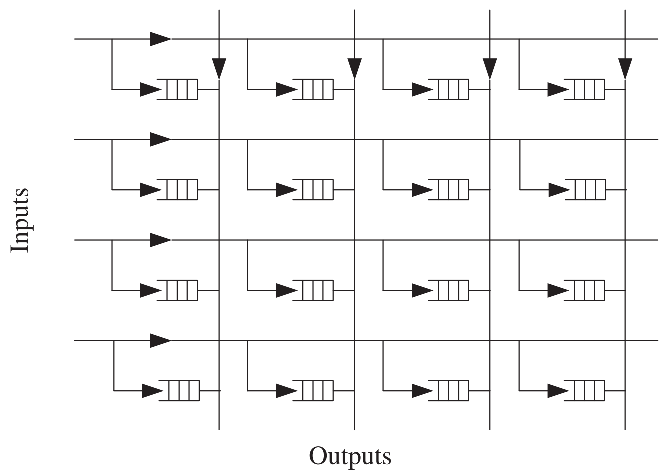

Fig. 2. A crosspoint buffered switch

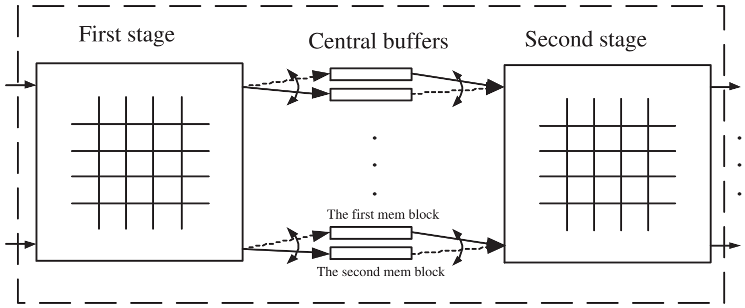

Fig. 3. The load balanced architecture for quasi-circuit switches

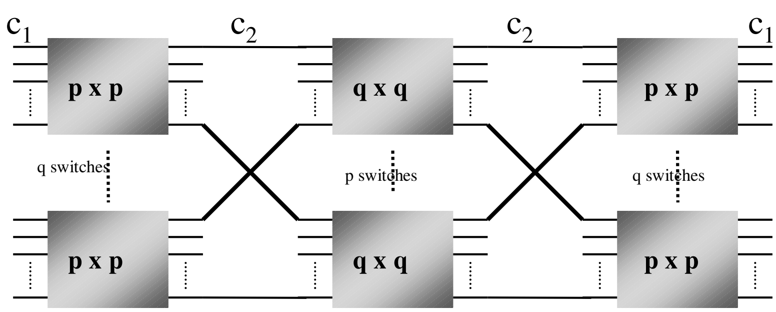

Fig. 4. A three-stage Clos quasi-circuit switch

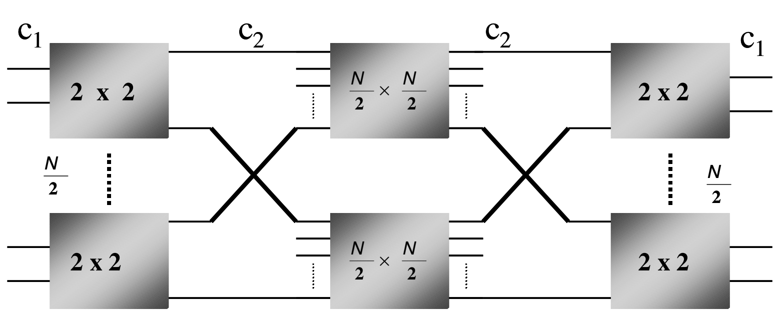

Fig. 5. Recursive construction of the Benes quasi-circuit switch

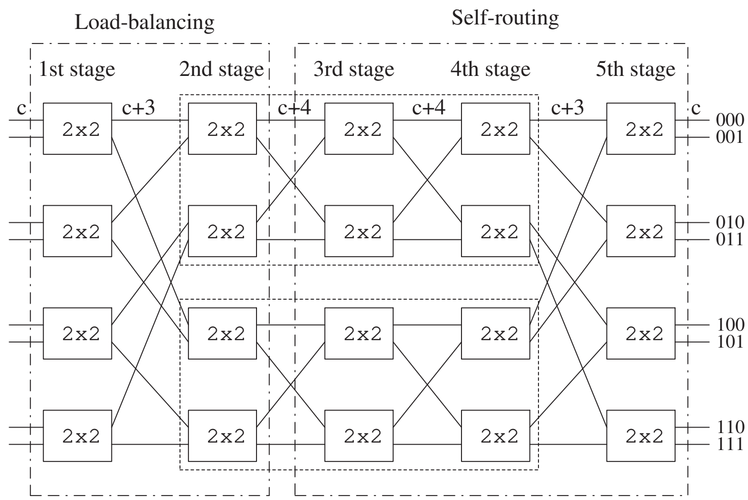

Fig. 6. An 8×8 Benes quasi-circuit switch using the round-robin splitting scheme in (R2)

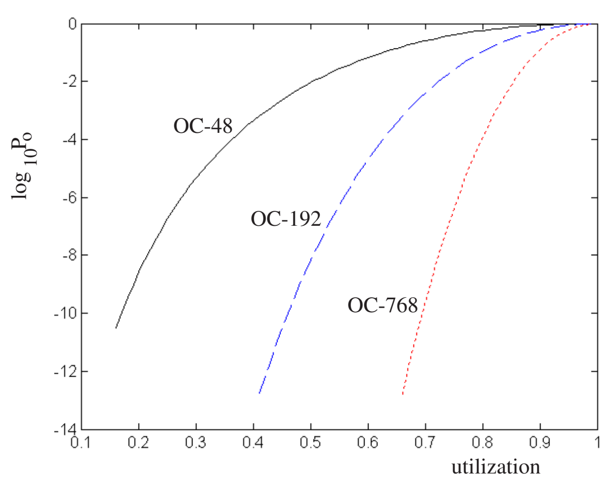

Fig. 7. $\log_{10} P_o$ as a function of the average link utilization