Blocking and Nonblocking Multirate Clos Switching Networks

Soung C. Liew, *Senior Member, IEEE*, Ming-Hung Ng, and Cathy W. Chan, *Member, IEEE*

***Abstract*—This paper investigates in detail the blocking and nonblocking behavior of multirate Clos switching networks at the connection/virtual connection level. The results are applicable to multirate circuit and fast-packet switching systems. Necessary and sufficient nonblocking conditions are derived analytically. Based on the results, an optimal bandwidth partitioning scheme is proposed to reduce switch complexity while maintaining the nonblocking property. The blocking behavior of blocking switches supporting multicast connections is investigated by means of simulation. We propose a novel simulation model that filters out external blocking events without distorting the bandwidth and fanout (for multicasting) distributions of connection requests. In this way, the internal blocking statistics that truly reflect the switch performance can be gathered and studied. Among many simulation results, we have shown that for point-to-multipoint connections, a heuristic routing policy that attempts to build a narrow multicast tree can have relatively low blocking probabilities compared with other routing policies. In addition, when small blocking probability can be tolerated, our results indicate that situations with many large-fanout connection requests do not necessarily require a switch architecture of higher complexity compared to that with only point-to-point requests.**

***Index Terms*—ATM, Clos networks, multirate switching, nonblocking switches, routing.**

I. Introduction

In 1953 Clos [12] published a seminal paper that gives the construction for a class of networks. A symmetric three-stage Clos network $C(n, m, p)$ is shown in Fig. 1. The general idea is to build a larger switch out of smaller switch modules. There are p switch modules in the first stage, each with n input links and m output links. The second stage has m switch modules, each with p input links and p output links. The third is similar to the first stage but the numbers of inputs and outputs are reversed. Each switch is connected to each switch at the next stage by one link.

Clos's work concerns circuit switching in which each link can be used by at most one connection at any given time. Melen and Turner laid out the foundation for the study of multirate networks [4] in which each link can be used by a number of connections as long as the sum of their data rates does not exceed that of the link. Among many network configurations, the Clos network, in particular, has been widely proposed as a way to build scalable fast-packet/asynchronous transfer mode (ATM) switches [3], [7], [6].

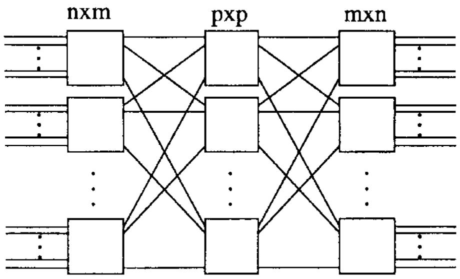


Fig. 1. Clos network $C(n, m, p)$.

In ATM networks the data rate of a connection can be constant or time-varying. For constant-bit-rate connections, the connection bandwidth is simply the data rate. For variable-bit-rate and bursty connections, it is simplest (although probably not efficient) for bandwidth allocation purposes to interpret the connection bandwidths as their peak data rates. Thus, as long as the aggregate peak data rate of connections multiplexed onto a link does not exceed the link capacity, performance of individual connections is guaranteed. An alternative approach is *not* to allocate bandwidth according to the peak rate so that more connections can be multiplexed onto the same link. This approach has been taken in [10].

These two approaches may yield rather drastically different results and conclusions. For instance, in the case of peak-rate allocation, it is desirable to route a new connection request along a densely packed route (i.e., one with almost all of the bandwidth of the route exhausted but with sufficient bandwidth to accommodate the new request) rather than a loosely packed route. This is to avoid bandwidth fragmentation among the alternative routes from inputs to outputs, leaving no single route with sufficient bandwidth to accommodate future high-bandwidth connections. Since peak-rate bandwidth is allocated, acceptable performance is achievable even along the busiest route. When less than peak rate is allocated, it may be more desirable to route a connection along a less busy route to achieve acceptable performance in terms of delay and loss probability.

The work in this paper adopts the former framework. Nonblocking conditions in Clos networks are studied analytically and the blocking probabilities in blocking Clos networks are

Manuscript received August 15, 1997; reassigned to Editor H. S. Kim February 3, 1997; revised December 20, 1997; approved by IEEE/ACM Transactions on Networking Editor H. S. Kim. This work was supported by the RGC Earmarked Grant of the Hong Kong University and Polytechnic Grant Council under Grant CUHK 269/94E and by the Direct Grant of the Chinese University of Hong Kong under Grant 220500870.

S. C. Liew and C. W. Chan are with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: soung@ie.cuhk.edu.hk).

M.-H. Ng was with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, NT, Hong Kong.

Publisher Item Identifier S 1063-6692(98)03713-3.

investigated by means of simulation. There are three major contributions related to this work.

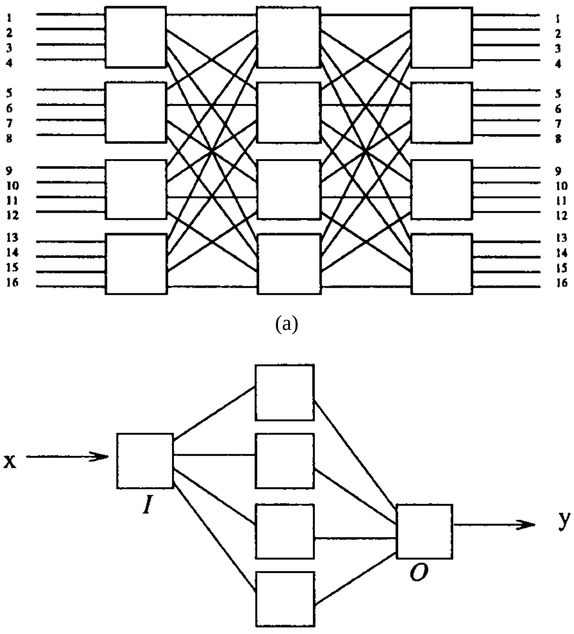
First, although [4] has derived a sufficient nonblocking condition for Clos networks, the condition is not the best achievable result in that it is not a necessary condition. This paper derives conditions that are both sufficient and necessary.

Second, we consider a simple nonblocking routing scheme, called the bandwidth partitioning scheme, that reduces the switch complexity rather effectively. In this approach connections with bandwidth greater than some fixed value α are routed along a subset of routes while those with bandwidth lower than α are routed along another disjoint subset of routes. It is proven that $\alpha = 0.5$ is optimal for reducing the switch complexity. We found that a similar idea has been briefly mentioned in [4], but the optimality of $\alpha = 0.5$ was not proven and the resulting switch is more complex than necessary (due to the use of "nonoptimal" nonblocking conditions in switch sizing).

Third, it is desirable to separate external blocking from internal blocking in switch simulation. The former refers to external links (inputs or outputs) not having enough bandwidth to accommodate a connection request and the phenomenon is independent of the switch design: the problem should be tackled by properly sizing the trunk capacities between switching centers. Therefore, external blocking should be factored out in the study of switch performance. We can simply filter out external blocking events so that requests presented to the switch are those that are not externally blocked. However, this will distort the bandwidth and fanout (for multicast connections) distributions of the requests so that requests used to test internal blocking are skewed toward smaller bandwidths and fanouts, leading to overly optimistic results. We can also simply ignore the external blocking events so that externally blocked requests are still presented to the switch. But this will lead to overly pessimistic internal blocking results. After trying several simulation models and considering their relevance to actual switch performance, we propose in this paper a model that can filter out external blocking without distorting the bandwidth and fanout distributions of requests.

II. Preliminaries

Let us review Clos networks for circuit switching before moving to the multirate situation. Fig. 2 shows a Clos switch with $n = m = p = 4$ and that the switch is not strictly nonblocking. We want to derive the relationship between the parameters that will guarantee nonblocking operation. First of all, for a Clos switch $C(4,4,4)$, there are $m = 4$ alternative paths between an input and an output, as illustrated in Fig. 2(b). Blocking between the input and output occurs when none of the four routes is free from existing connections so that a connection request between the input and output cannot be accommodated. By making m larger, more alternative paths between stage-1 and stage-3 modules are made available and, therefore, we should expect the likelihood of blocking to be smaller. In fact, if m is large enough, blocking can be eliminated altogether. On the other hand, larger m also implies higher switch complexity. The idea, then, is to find the minimum m that can guarantee nonblocking operation.



(a)

(b)

Fig. 2. (a) A Clos $C(4,4,4)$ switch. (b) Four alternative paths to route a request.

Definition (Strictly Nonblocking): A switch is strictly nonblocking [11], [13] if a connection can always be set up between any idle input and output without the need to rearrange the paths taken by existing connections.

Clos Theorem: A Clos network [12] is strictly nonblocking for circuit switching if and only if the number of second-stage switch modules

$$m \geq 2n - 1. \quad (1)$$

Proof: With reference to Fig. 2(b), suppose that an input link x of a first-stage switch module I asks for connection to an output link of a third-stage switch module O. In the worst case, the other $(n-1)$ input links of I are active and they use up $(n-1)$ outgoing links of I, and the other $(n-1)$ output links of O are active and they use up $(n-1)$ incoming links of O. Furthermore, none of the $(n-1)$ outputs of I and the $(n-1)$ inputs of O are attached to a common second-stage switch module. In other words, at most $2(n-1)$ paths cannot be used for the new request. So, to be strictly nonblocking, we must have

$$m \geq 2(n-1) + 1$$

so that at least one of the middle-stage modules is available for setting up the new path. □

Definition (Wide-Sense Nonblocking): A switch is wide-sense nonblocking [11], [13] if a route-selection policy exists for setting connections in such a way that a new connection can always be set up between any idle input and output without the need to rearrange the paths of the existing connections.

Thus, associated with wide-sense nonblocking is an algorithm for setting the internal paths of the switch. The strictly nonblocking property poses a more stringent requirement than the wide-sense nonblocking property since the former means that the switch must be nonblocking regardless of the route-selection policy used. The study and the proof of the wide-sense nonblocking property is generally not easy since not only the arrivals of connection requests must be considered but also the departures (terminations) of existing connections must be considered. For the circuit-switching situation, there is no known routing policy that will lower the required m if only the wide-sense nonblocking property is desired. As will be seen, the multirate-switching case is different: the required m can be substantially reduced by adopting a simple routing policy.

III. Nonblocking Conditions

In Section II we have reviewed the derivation of the strictly nonblocking condition for circuit-switching Clos networks. However, with broadband systems, the basic hypotheses related to circuit switching have to be changed. In multirate systems each connection induces a load on the network which depends on its bandwidth characteristics. This will be modeled by associating a weight $0 < \omega \leq 1$ to each connection. A connection request is denoted by (x, y, ω), where x is the input, y is the output, and ω is the weight. Physically, ω is the ratio of the connection bandwidth to the link bandwidth. For the rest of the paper, the term bandwidth refers to the normalized bandwidth with the link bandwidth equal to one. Many connections may share a common physical link, provided the sum of their weights does not exceed one. Thus, a new connection with weight ω can use a link if and only if the load that the link is already carrying is no more than $(1 - \omega)$.

The definitions for nonblocking properties are the same as those in the circuit-switching case (see preceding section) except that the term "idle input and output" is replaced by "input and output with at least ω remaining capacity." For a connection request (x, y, ω) with x being an input to a first-stage module I and y being an output of a third-stage module O, we say that a second-stage switch module U is accessible [4] from x (y) if the link between I (O) and U has an existing weight of no more than $1 - \omega$. Thus, the connection setup problem is to find a second-stage switch module that is accessible from both x and y.

This section derives the strictly nonblocking conditions for multirate Clos networks. Compared with the results obtained in [4], we are able to improve the bounds on m for the nonblocking property. Results of similar work can also be found in [8] and [2].

A. Unrestricted-Weight Nonblocking Conditions

Suppose the weights of connections are unrestricted and can be anywhere between zero and one. The standard reasoning for determining the nonblocking condition for the Clos network can be extended in a straightforward manner to obtain the following nonblocking condition. This condition also appears in a different form in [8] and the reader is referred to [8] for a proof from another approach.

Nonblocking Condition 1: For a connection request (x, y, ω), the Clos network $C(n, m, p)$ is strictly nonblocking if and only if

$$m \geq 2\left\lceil \frac{(n-1)}{(1-\omega)} \right\rceil + 1 \tag{2}$$

where $\lceil z \rceil$ denotes the minimum integer greater than or equal to z.

Proof: Denote by I the first-stage switch module to which input x is connected and denote by O the third-stage switch module to which output y is connected. Similar to the proof of the Clos theorem, in the worst case, all other $(n-1)$ inputs of I are fully occupied. In addition, the existing weight on input link x is $(1-\omega)$. Then, the sum of the weights on all links out of I is

$$(n-1) + (1-\omega) = n - \omega.$$

A link out of I does not have enough bandwidth for the connection request and, hence, a corresponding second-stage module is inaccessible from x if its existing weight is more than $(1-\omega)$. Consequently, the number of links out of I that carry a weight of more than $(1-\omega)$ is *strictly less than* $\lceil (n-\omega)/(1-\omega) \rceil$. In other words, the maximum number of inaccessible second-stage modules from x is

$$\lceil (n-\omega)/(1-\omega) \rceil - 1.$$

By a similar argument, the maximum number of inaccessible second-stage modules from y is also this value. In the worst case, these $2(\lceil (n-\omega)/(1-\omega) \rceil - 1)$ links connect to different second-stage modules. To be strictly nonblocking, we need at least one more path to set up the connection from x to y. This leads to the following result

$$\begin{aligned} m &\geq 2\left\lceil \frac{n-\omega}{1-\omega} \right\rceil - 1 \\ &= 2\left\lceil \frac{n-1}{1-\omega} + 1 \right\rceil - 1 \\ &= 2\left\lceil \frac{n-1}{1-\omega} \right\rceil + 1. \end{aligned} \tag{3}$$

To see the necessity of the above bound, suppose that $m \leq 2\lceil \frac{n-1}{1-\omega} \rceil$. We can construct a blocking situation as follows. For switch module I, create blocked outgoing links by assigning a weight of $1 - \omega + \epsilon$ to each of them, where ϵ is an arbitrarily small positive number. The number of blocked links that can be created this way is

$$\lim_{\epsilon \to 0} \left\lfloor \frac{n-\omega}{1-\omega+\epsilon} \right\rfloor = \left\lceil \frac{n-\omega}{1-\omega} \right\rceil - 1.$$

Create the same number of blocked links from O. We can therefore make all of the $m = 2\lceil \frac{n-\omega}{1-\omega} \rceil - 2$ second-stage modules inaccessible. □

Now, let use compare our result with that in [4], where the number of second-stage modules sufficient for nonblocking operation is given by

$$m^* > 2\left\lfloor \frac{n-\omega}{1-\omega} \right\rfloor. \tag{4}$$

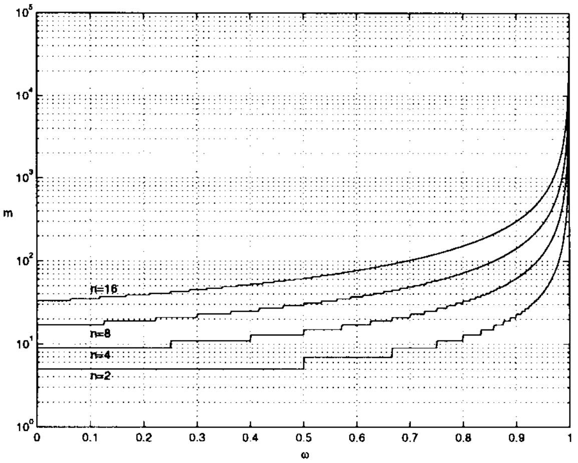


Fig. 3. m required for different values of ω to achieve strictly nonblocking for $n = 2, 4, 8$, and 16.

Equivalently, since m^* must be an integer

$$m^* \geq 2\left\lfloor \frac{n-\omega}{1-\omega} \right\rfloor + 1. \tag{5}$$

Let us only consider the lower bounds of both cases. Note that if $(n-\omega)$ is not divisible by $(1-\omega)$

$$m = 2\left\lceil \frac{n-\omega}{1-\omega} \right\rceil - 1 = 2\left\lfloor \frac{n-\omega}{1-\omega} \right\rfloor + 1 = m^*.$$

However, if $(n-\omega)$ is divisible by $(1-\omega)$

$$m = m^* - 2.$$

So, our derivation improves the bound in [4] by two middle-stage nodes when $(n-\omega)$ is divisible by $(1-\omega)$.

It should be pointed out that the bound is with respect to a connection request with bandwidth ω. Obviously, a switch with fixed m may be blocking to some connection requests while nonblocking to others. Specifically, requests with ω beyond a certain value may be blocked given a fixed m. The graph in Fig. 3 plots the required m versus ω. Note that m increases more than exponentially with ω, and $\omega = 1$ requires infinite m to achieve the nonblocking property. This can be easily seen as follows. For any finite m, we can create a blocking situation: make all outgoing links from I blocking to the new request by placing an arbitrarily small weight ϵ on each of them to correspond to an existing connection. In practice this would not occur because the smallest bandwidth of connections should be larger than zero. Nevertheless, exceedingly large m may still be needed to make the switch nonblocking to all connection requests.

We now consider two approaches to reducing m. The first approach increases the speed of the internal links. Suppose the switch is speeded by S times ($S > 1$). A request (x, y, ω) at the input can be viewed as $(x, y, \omega/S)$ with respect to the internal-link capacity. So, the total weight at any external link will be no greater than $1/S$. In other words, the sum of the weights of an external link is limited to $1/S$. The second approach restricts the request bandwidth to the interval $[b, B]$, where either $b > 0$ or $B < 1$ (or both). Drawing on the results of the second approach, we shall show that a wide-sense nonblocking switch can be constructed to accommodate *unrestricted* bandwidth requests using an optimal bandwidth partitioning scheme.

B. Switch With Speedup Factor S

Nonblocking Condition 2: The Clos network $C(n, m, p)$ with speedup factor S is strictly nonblocking for packet switching if and only if

$$m \geq 2\left\lceil \frac{n-\omega}{S-\omega} \right\rceil - 1. \tag{6}$$

Proof: Consider a request (x, y, ω). Let x be one of the inputs of a first-stage switch module I and y be one of the outputs of a third-stage switch module O. With respect to the internal structure of the switch, the connection is $(x, y, \omega/S)$. As in the proof of nonblocking condition 1, in the worst case, the sum of the weights of all links out of I is

$$\frac{n-1}{S} + \frac{1-\omega}{S} = \frac{n-\omega}{S}.$$

Consequently, the number of links out of I that carry a weight of more than $(1 - \omega/S)$ is strictly less than

$$\left\lceil \frac{n-\omega}{S} \right\rceil \Big/ \left(1 - \frac{\omega}{S}\right) = \left\lceil \frac{n-\omega}{S-\omega} \right\rceil$$

and the maximum number of inaccessible second-stage modules from x is

$$\left\lceil \frac{n-\omega}{S-\omega} \right\rceil - 1.$$

By similar argument, the maximum number of inaccessible second-stage modules from y is also this value. To be strictly nonblocking, we need at least one more path to make connection from x to y. This leads to the following result:

$$m \geq 2\left\lceil \frac{n-\omega}{S-\omega} \right\rceil - 1.$$

The "only if" part is similar to that in the proof of nonblocking condition 1. □

C. Restricted-Weight Nonblocking Conditions

Now, let us consider the case where the bandwidth ω required by a connection request is restricted to the interval $[b, B]$. We assume that the internal links and external links of the switch are of the same speed in this subsection. We divide our analyses into several cases as below. A general condition for all cases is provided in [4], but it turns out that the condition in [4] is sufficient but not necessary. The conditions below, however, are both sufficient and necessary (i.e., they are the tightest possible bounds). These conditions are also discussed and proven in [2]. In the following we provide an alternative proof to these conditions.

Case 1 ($b = 0$ and $B < 1$): The worst case occurs when ω is at its maximum B. The proof for nonblocking condition 1 can also be used to establish the following result.

Nonblocking Condition 3: For a connection request (x, y, ω) with $\omega \in [0, B]$, where $B < 1$, the Clos network $C(n, m, p)$ is strictly nonblocking if and only if

$$m \geq 2 \max_{\omega} \left\lceil \frac{n-1}{1-\omega} \right\rceil + 1 = 2 \left\lceil \frac{n-1}{1-B} \right\rceil + 1. \qquad (7)$$

Case 2 ($b > 0$ *and* $B = 1$). This case requires a different approach because the minimum bandwidth of a request cannot arbitrarily approach zero.

Nonblocking Condition 4: For a connection request (x, y, ω) with $\omega \in [b, 1]$, where $b > 0$, the Clos network $C(n, m, p)$ is strictly nonblocking if and only if

$$m \geq 2(n-1) \left\lfloor \frac{1}{b} \right\rfloor + 1. \qquad (8)$$

Proof: In the worst case, a request $(x, y, 1)$ arrives. Furthermore, the current state of the network is such that all of the other $n-1$ input links of I are fully occupied and are connecting to the maximum number of requests, each of which is occupying one outgoing link. Because the maximum number of requests each input link can support is $\lfloor 1/b \rfloor$, the maximum number of outgoing links that are made blocking by an input link is also $\lfloor 1/b \rfloor$. So, the maximum total number of blocked outgoing links is $(n-1)\lfloor 1/b \rfloor$.

A similar argument applies to the output y. In the worst case, none of the blocked links from I and O are attached to a common second-stage switch module. To be strictly nonblocking, we need at least one more path to establish the connection from x to y. This leads to the following result:

$$m \geq 2(n-1) \left\lfloor \frac{1}{b} \right\rfloor + 1.$$

The argument for the necessity of this condition is similar to that in the proof of nonblocking condition 1. □

Note that if $b > 0.5$, each link can only serve one connection. This special case is similar to circuit switching, and so $m \geq 2n - 1$. Let us also compare our derivation to those in [4], where

$$m^* \geq 2 \left\lfloor \frac{n-1}{b} \right\rfloor + 1. \qquad (9)$$

One can notice that $m \leq m^*$ and the equality holds only when one is divisible by b. For example, Fig. 4 shows how our derivation improves the bound when $b = 0.34$ and $B = 1$.

Case 3 ($b > 0$ *and* $B < 1$)*:* When $b > 0$ and $B < 1$, the situation becomes more difficult to analyze. There are two subcases. If $b > 1 - B$, then the bound in nonblocking condition 4 applies. This is because the minimum weight on each link b is sufficient to block out a connection requesting a bandwidth of B. If $b \leq 1 - B$, the situation becomes more complicated. The bound in nonblocking condition 3 is sufficient but not necessary. We cannot find a single inequality that applies to this subcase and it appears that this subcase needs to be further divided into subsubcases. Since we do not depend on this subcase for latter discussion, we will omit its details here.

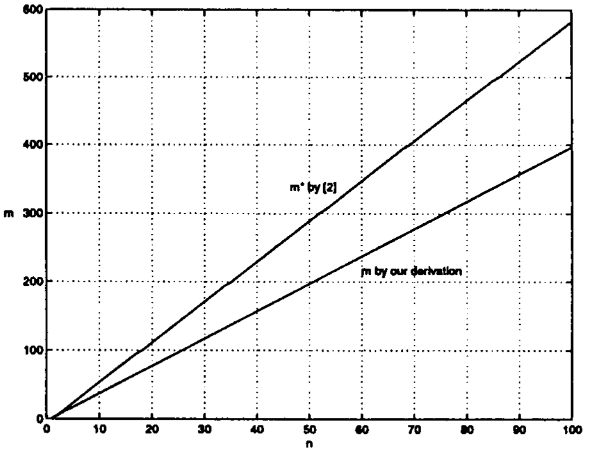


Fig. 4. Comparing two bounds on m for strictly nonblocking property.

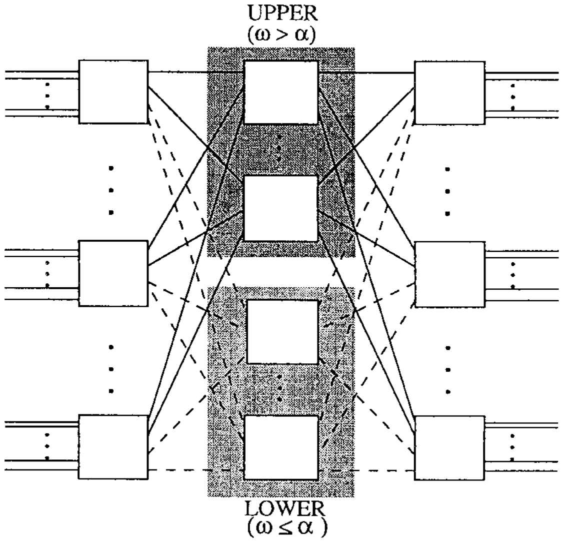


Fig. 5. Construction of a wide-sense nonblocking Clos switch.

D. Bandwidth Partitioning Scheme

As mentioned previously, we cannot construct a strictly nonblocking Clos network for unrestricted packet switching ($b = 0$ and $B = 1$) because m goes to infinity for $\omega = B = 1$. Fortunately, (7) and (8) suggest that we can construct a wide-sense nonblocking network as an unrestricted packet switch by segregating connections based on weights. As shown in Fig. 5, the middle-stage switch modules are divided into two groups, UPPER and LOWER. Let us define a partition bandwidth α. All of the connections with weight $\leq \alpha$ are routed through the LOWER modules and all of the connections with weight $> \alpha$ are routed through the UPPER modules.

We want to show that the best value of the partition bandwidth α is 0.5 and that $m \geq 6n - 4$ is sufficient to achieve the nonblocking property. Since a specific algorithm (albeit a simple one) is used to route connections, the switch is wide-sense nonblocking rather than strictly nonblocking [11], [13].

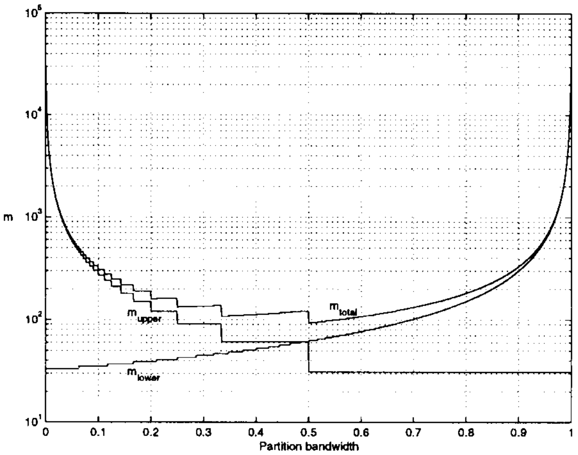


Fig. 6. Minimum value of m occurring at $\alpha = 0.5$ for the case $n = 16$.

For connections routed to the LOWER modules, $b = 0$ and $B = \alpha$. From (7)

$$m_{\text{lower}} \geq 2\left\lceil \frac{n-1}{1-\alpha} \right\rceil + 1. \tag{10}$$

For the UPPER modules, where $B = 1$ and $b > \alpha$, we write $b = \epsilon + \alpha$, where ϵ is an arbitrarily small positive number. From (8)

$$\begin{aligned} m_{\text{upper}} &\geq 2(n-1)\left\lfloor \frac{1}{b} \right\rfloor + 1 \\ &= 2(n-1)\lim_{\epsilon \to 0}\left\lfloor \frac{1}{\alpha+\epsilon} \right\rfloor + 1 \\ &= 2(n-1)\left(\left\lceil \frac{1}{\alpha} \right\rceil - 1\right) + 1. \end{aligned} \tag{11}$$

The required m for the overall Clos network is simply

$$m = m_{\text{lower}} + m_{\text{upper}}. \tag{12}$$

Let us consider $n = 16$ and plot m as a function of α. From Fig. 6, it can be seen that the minimum value of m occurs at $\alpha = 0.5$ for the case $n = 16$. The figure suggests that $\alpha = 0.5$ may be the best value for all n greater than one. The general case is proven below.

Let us break the possible value of α into many subintervals S_i $(i = 1, 2, \cdots)$ such that

$$S_i = \left\{\alpha : \frac{1}{i+1} \leq \alpha < \frac{1}{i}\right\}$$

and let us also define the lower boundary of the subinterval S_i as s_i. That is

$$s_i = \frac{1}{i+1}.$$

Within each subinterval S_i, m_{upper} is fixed at a constant according to (11). Specifically, $\lceil 1/\alpha \rceil = i + 1$ so that m is fixed at $2(n-1)i - 1$.

Now, it can be easily seen that m_{lower} is an increasing function of α. Therefore, within each subinterval S_i, the minimum value of $m = m_{\text{lower}} + m_{\text{upper}}$ is achieved at $\alpha = s_i$, the lower boundary of S_i. Writing the minimum m as a function of the subinterval index i, we have

$$\begin{aligned} m(i) &= m_{\text{lower}}(i) + m_{\text{upper}}(i) \\ &\geq 2\left\lceil \frac{n-1}{1-1/(i+1)} \right\rceil + 1 + 2(n-1)i + 1 \\ &= 2\left[(n-1)i + \left\lceil \frac{n-1}{i} \right\rceil + n\right] \\ &= 2[Q(i) + n] \end{aligned}$$

where $Q(i) = (n-1)i + \lceil \frac{n-1}{i} \rceil$. It is easy to show that $Q(i+1) > Q(i)$ for all $i \geq 1$ if $n > 1$. So we can conclude that $m(i)$ increases with i and $i = 1$ is the best. Then

$$\alpha_{\text{optimum}} = s_1 = 0.5$$

and

$$m \geq 6n - 4. \tag{13}$$

Although we can reduce m from infinity to $6n-4$ by the bandwidth partition scheme, m is still rather large to achieve the nonblocking property. The next section investigates blocking switches using simulation methods.

IV. Blocking Switches

The analytical study of blocking switches is difficult. Many assumptions must be made and there are generally no simple closed-form solutions [9]. The analysis becomes even more difficult when sophisticated routing schemes are considered. For these reasons, we choose to investigate the blocking switches by simulation. As will be discussed, simulation also presents new difficulty in which the results are very sensitive to the simulation model used.

Since the study of multicast connections does not present much additional difficulty as far as simulation is concerned, we have included multicasting in our investigations. Correspondingly, a connection request is characterized by (x, Y, ω) in which Y is a subset of switch outputs. All of the switch modules are assumed to have multicast capability (i.e., the data on an input can be forwarded to any subset of the outputs). For each subset Y, there is a subset of third-stage modules, say S_o, to which the outputs in Y are attached. The problem is to find to a multicast tree with I, the first-stage module to which input x is attached, being the root and S_o being the leaves.

A. Simulation Models

Dynamic simulation is adopted in which connections arrive randomly with a certain rate and depart after a random holding time. The set of outputs Y, the fanout f (numbers of elements in Y), and the requested bandwidth are also randomly generated. For the simulation, it is important to separate *external blocking* from *internal blocking.*

Blocking Definitions: A connection request is blocked externally if either input x or any of the output in Y has less than ω remaining bandwidth. It is internally blocked if it is not externally blocked but an internal route with sufficient remaining bandwidth cannot be found inside the switch architecture.

Note that external blocking is independent of the switch architecture and it can be solved only by properly sizing the trunk capacities between switching centers. While it can be excluded rather easily in the analytical study of the preceding sections, separating it from internal blocking in simulation without affecting the targeted bandwidth and fanout distributions requires more thought. We now discuss several models that we have tested and argue for the use of one of them.

Ignoring External Blocking: Perhaps the simplest approach is to ignore external blocking. An incoming request (x, Y, ω) is used to test for internal blocking whether or not it is blocked externally. It will be accepted if an internal path consisting of two links with sufficient bandwidth can be found. Obviously, this leads to pessimistic results as far as internal blocking probability is concerned (consider, for instance, the corollary that it is then possible for a first-stage module to have a total of more than n units of incoming traffic, which is physically impossible in reality).

From the engineering viewpoint, this conservative approach has the appeal that once the switch is engineered this way, it should perform well in the real setting. For simulation, the offered load, and the fanout and bandwidth distributions of requests can be controlled precisely with this approach. However, our experimentation indicated that the results generated are simply too pessimistic, especially when n is small, that they may not be very useful.

Filtering Out External Blocking Events: We can simply filter out external blocking events so that requests presented to the switch are those that are not externally blocked. An incoming request (x, Y, ω) is more likely to be blocked externally if the bandwidth and fanout $|Y|$ are large. Therefore, the bandwidth and fanout (for multicast connections) distributions of the requests used to test internal blocking are skewed toward smaller bandwidths and fanouts, since requests with larger bandwidths and fanouts are more likely to be blocked externally and, therefore, filtered. This leads to overly optimistic internal blocking statistics. Compounding the problem is the fact that the distortions on the bandwidth and fanout distributions vary with the offered load. Fig. 7 shows qualitatively an observed inconsistent result that higher offered load leads to lower internal blocking probability. This is because at high offered load almost all high-bandwidth requests are already blocked externally and only those low-bandwidth requests are presented to the switch for sampling of internal blocking events.

One possibility is to perform "equalization" on the prefiltered distributions to obtain the desired postfiltered distributions. For instance, if we desire a uniform bandwidth distribution we can intentionally increase the probability density of higher bandwidths beyond that of the uniform distribution. Our experiments, however, showed that precise control of postfiltered distributions is difficult with this approach.

Filtering Out External Blocking Events with Feedback: To maintain the bandwidth and fanout profiles of requests, the external blocking events that have been filtered out can be fed back to the system until it is accepted. Thus, an externally blocked request (x, Y, ω) may wait until enough connections have departed from x and Y before entering the system for internal blocking testing. This approach prefixes x and Y before testing for external blocking. The problem is that the actual offered load to the switch is decreased. In fact, we have experienced difficulty in testing high offered-load situations with this approach.

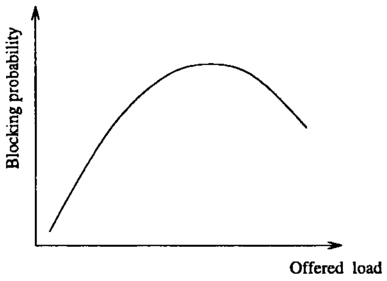


Fig. 7. Inconsistent result due to external blocking distorting the bandwidth distribution requests.

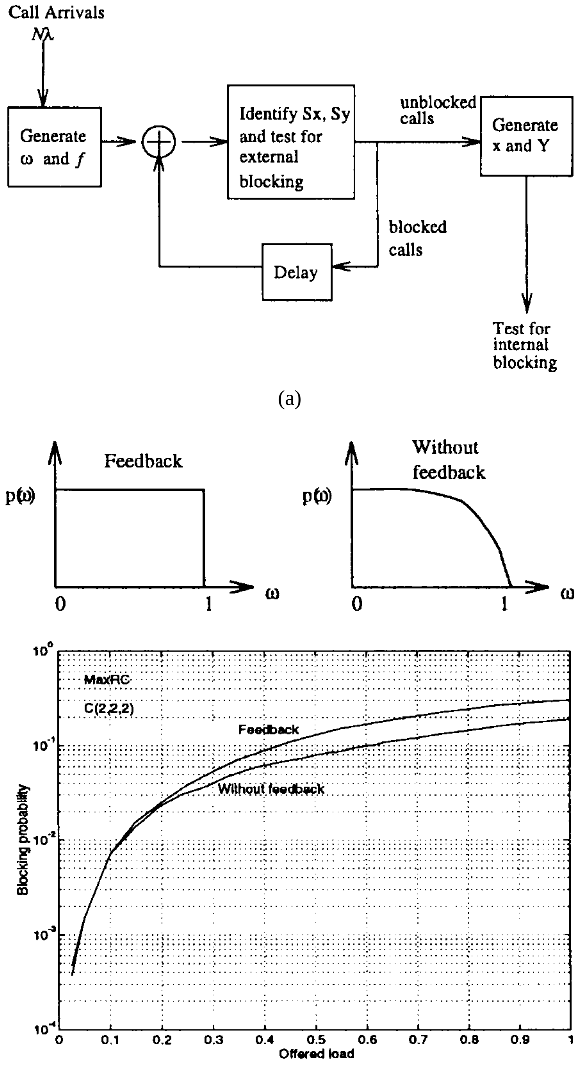
The key to circumventing this problem is to decrease the likelihood of a call being externally blocked in the first place. One way is to generate the input x and outputs Y only after ω and f have been generated so as to make sure x and Y can accommodate bandwidth ω. Note that this strategy does not imply a contrived situation because in practice one would not attempt to set up a call between x and Y if there is not enough bandwidth on them anyway.

A simulation model as depicted in Fig. 8(a) is used. Call requests arrive at the rate of $N\lambda$, where λ is the arrival rate on an input. The calls are not associated with any input x and outputs Y in the beginning. The bandwidth ω and fanout f of a request is generated upon its arrival using a random-number generator according to the targeted distributions. Based on ω, the subsets of inputs S_x and outputs S_y that have remaining bandwidths not smaller than ω are identified. If $S_x = 0$ or $S_y < f$, then the call is blocked externally and filtered out. The externally blocked call is fed back to the system with the same ω and f after an exponentially distributed delay. The previous S_x and S_y, however, are not kept in the feedback request. The system identifies a new S_x and S_y. The process is repeated until $S_x > 0$ and $S_y \geq f$ can be found, in which case an input $x \in S_x$ and output subset $Y \subseteq S_y$ are chosen randomly to make up the request specification (x, Y, ω) for internal blocking testing. An internally blocked event will simply be discarded and will not be fed back. This approach guarantees that the distributions of f and ω assumed by the random-number generator are also the distributions presented to the switch, since each f and ω generated will eventually be used.

Fig. 8(b) compares the internally blocking probabilities of the feedback system in Fig. 8(a) and a system without feedback for a Clos switch $C(2, 2, 2)$. It can be seen that the system without feedback has a lower loss probability because the bandwidth distribution has been skewed toward lower values, thanks to external blocking.

B. Routing Strategies

Once external blocking has been excluded, the next question is whether there is an internal route that can support the connection. When there are several alternative routes, which



(a)

(b)

Fig. 8. (a) A Simulation model for filtering out external blocking events with feedback to maintain desired bandwidth and fanout distributions of connection requests. (b) Comparison between feedback and without feedback.

one should be chosen? There are many possible routing strategies, of which we only focus on three in this paper.

Maximum Residual Capacity (MaxRC) Routing: This strategy always selects the route with the maximum remaining bandwidth among the available routes for an incoming request (x, y, ω). The maximum remaining bandwidth of a route is defined to be $\min(C_1, C_2)$ where C_1 and C_2 are the remaining bandwidths of the first- and second-stage internal links, respectively. If remaining bandwidths on all routes are smaller than ω, internal blocking occurs and the request is rejected. This routing algorithm can be used for both point-to-point and point-to-multipoint connections. For a point-to-multipoint connection (x, Y, ω), one route is established for each module in S_o based on the point-to-point algorithm and the resulting multicast tree is used to support the multicast connection. Because the switch modules are assumed to have the multicast capability, only one data stream from the same connection needs to be sent along any link in the tree. In other words, each link in the tree uses only ω to support the connection.

Ordered Routing: Another strategy is to order the second-stage modules. Attempts will be made to route a connection via the lower ordered second-stage modules before the higher ordered ones. The first available route is chosen. The idea is to prevent bandwidth fragmentation so that part of the network can remain relatively less busy to accommodate future high-bandwidth connections. For extension to multicasting, we try to route as many connections via the first-ordered module, and if there are any remaining connections, we try to route through the second-ordered middle-stage modules, and so on, until there is a path to all outputs in the connection.

Narrow-Tree Routing: Given the output node set S_o, the number of links used in the multicast tree between the second-stage modules and the third-stage modules S_o is fixed at $|S_o|$. The number of links between the first-stage module I and the second-stage modules used, on the other hand, is dependent on the routing strategy. A narrow multicast tree is one in which there are very few second-stage modules, and it has the advantage of reducing blocking of future connections at the links between the first- and second-stage modules.

Finding the narrowest possible multicast tree is a hard algorithmic problem: it can be posed as a Steiner-tree problem with link cost equal to one for all links that have sufficient bandwidth [10], [1]. Therefore, a heuristic that attempts to find a narrow tree is considered here.

Some of the links from I to the middle-stage modules may be blocked. Let us focus only on the middle-stage modules that are accessible from I and suppose that there are $m' \leq m$ such modules. Let $S_o = \{O_1, O_2, \cdots, O_k\}$. For each O_j, we define a 0–1 accessible vector $U_j = \{u_{1j}, u_{2j}, \cdots, u_{m'j}\}$ such that $u_{ij} = 1$ if the link from the ith middle-stage module to O_j has sufficient bandwidth to support the connection and $u_{ij} = 0$ otherwise. We form an $m' \times k$ matrix $U = [U_1 U_2 \cdots U_k]$. Thus, the rows correspond to the accessible middle-stage modules and the columns correspond to the third-stage modules in the tree.

If the sum along any column is zero, then the connection is blocked because the third-stage module corresponding to the column is not accessible. Otherwise, we sort the rows according to the number of one entries in an ascending order. Starting from row one (the one with the least number of one entries) until row m' (the one with the most number of one entries), we perform the following to attempt to eliminate as many middle-stage modules from the tree as possible. Remove a row from matrix U. If as a result any column of U sums to zero, then the middle-stage module associated with the row cannot be eliminated from the multicast tree, and therefore the row will be put back into U. Otherwise, the row can be eliminated from the multicast tree. After the procedure is performed for all rows, the remaining rows define the middle-stage modules in the resulting multicast tree. In the solution there could be multiple middle-stage modules with links having enough bandwidths to a common third-stage module O_j. In this case one of the links is chosen at random to be included in the tree.

The motivation for first sorting the rows before the above procedure of eliminating middle-stage modules is that to build a narrow tree; we must retain the middle-stage modules with many links to S_o. Therefore, their elimination should be considered last. For further improvement, the middle-stage modules are also ordered so that modules with the same number of one entries in their rows in U are further sorted according to their orders. In this way, narrow-tree routing is similar to ordered routing for point-to-point connections.

C. Simulation Results

We now present the simulation results. Since these are the results related to specific switch parameters and traffic characteristics, only the qualitative natures of the results are important. One should exercise restraints in extrapolating the implications of the results. However, the simulation model proposed in this paper should be useful for further detailed study.

Assumptions: Several assumptions are made in the simulation programs. The interarrival time of connections on each input is exponentially distributed with mean $1/\lambda$. The holding time of connections is also assumed to be exponentially distributed with mean $1/\mu$. The load on each output is given by

$$\rho = \frac{\lambda}{\mu} \cdot \bar{\omega} \cdot \bar{f} \tag{14}$$

where $\bar{\omega}$ is the mean bandwidth and $\bar{f}$ is the mean fanout of connections. Note that because of the filtering of external blocking events and the feedback process, the interarrival times of connections presented to the internal structure of the switch are actually differently distributed. For the rest of this paper, the offered load is defined to be the offered load at an output.

The bandwidths of connections are assumed to be uniformly distributed between the lower bound b and upper bound B, and the fanout is assumed to be uniformly distributed between 1 and some upper bound F.

Unless otherwise noted, the simulation data are related to a Clos network $C(16, 16, 16)$. In addition, collection of statistics does not begin until the system is perceived to have reached some steady state.

Bandwidth Distribution: Let us now examine how the bandwidth distribution affects the blocking behavior. Fig. 9 shows the blocking probability as a function of the offered load with different bandwidth distributions. The connection requests are point-to-point and the MaxRC routing algorithm is used. The curves A, B, C, and D are obtained by uniform bandwidth distributions in the intervals $[0, 0.5]$, $[0.3, 0.8]$, $[0, 1.0]$, and $[0.3, 0.7]$, respectively.

Distributions of A and B have the same standard deviation, but B has a higher mean. We see from curves A and B that higher mean bandwidth implies higher blocking probability. This result is not surprising and merely confirms our intuition that higher bandwidth connections are more easily blocked.

Distributions of C and D have the same mean with C having a higher standard deviation. The blocking probabilities are comparable at all loads.

Routing Algorithms: The performance under the three routing strategies are shown in Fig. 10. For point-to-point connections, two routing strategies are compared in Fig. 10(a). The request–bandwidth distribution is uniformly distributed between zero and one. From this graph, we see that ordered routing offers better performance than MaxRC routing, and the difference is quite substantial at low blocking probability. Ordered routing prevents the small bandwidth connections from spreading out all over the fabric, reducing the likelihood of them blocking the large-bandwidth connections.

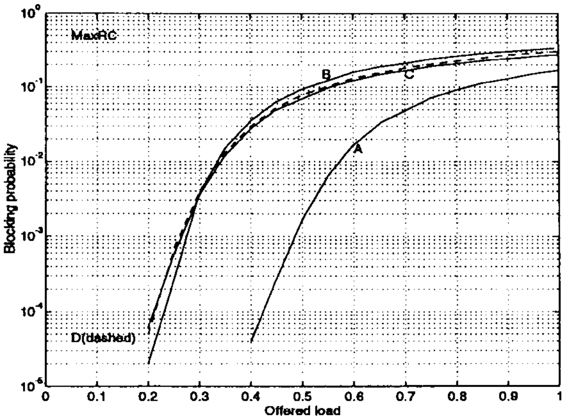
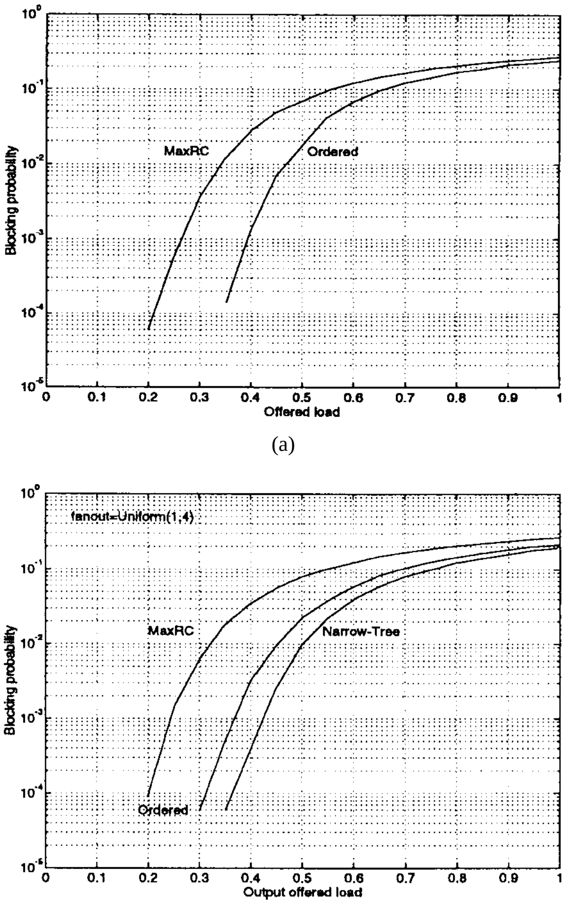


Fig. 9. Effects of bandwidth distribution.

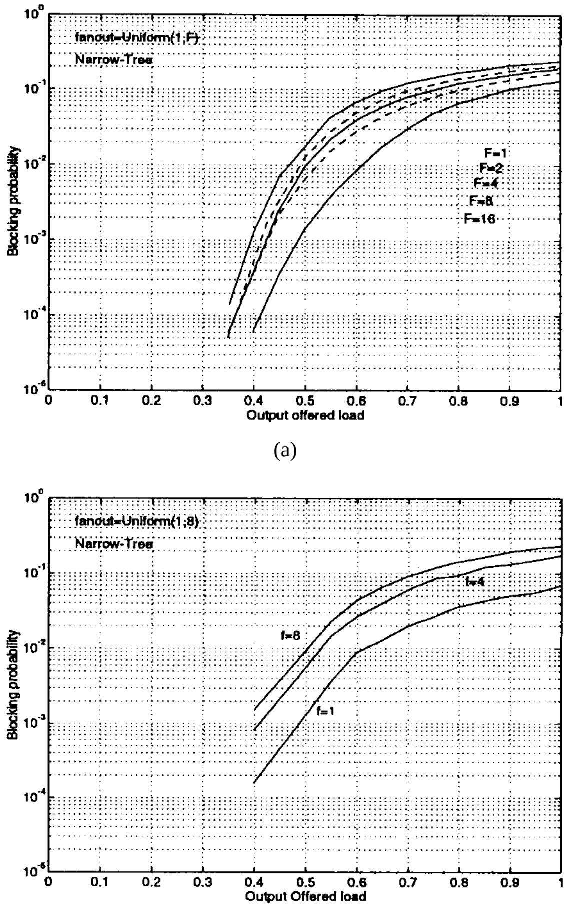


(a)

(b)

Fig. 10. Comparison among routing algorithms. (a) Point-to-point connections. (b) Point-to-multipoint connections.

Fig. 10(b) reports the results for point-to-multipoint connections. The fanout is uniformly distributed between one and four. The graph shows that ordered routing is better



(a)

(b)

Fig. 11. (a) Effects of fanout distribution. (b) The blocking probabilities of connection with different fanouts when $F = 8$.

than MaxRC routing and that narrow-tree routing is better than ordered routing. Both narrow-tree and ordered-routing policies reduce bandwidth fragmentation. In addition, narrow-tree routing reduces blocking at the first-stage links.

Fanout Distribution: Fig. 11(a) shows that the fanout distribution also affects the blocking behavior. The request bandwidth is uniformly distributed between zero and one, and the fanout numbers are integers uniformly distributed between 1 and F. Narrow-tree routing has been adopted. Several F values have been tested. From the graph, for the same output offered load, it can be seen that as F increases, blocking probability decreases.

There are two opposing factors affecting the blocking probability. As F increases, there are more large-fanout calls, which are more easily blocked compared with small-fanout calls. On the other hand, as F increases, the internal link usage of the switch decreases for a fixed output-link offered load, thanks to the multicast capability of the switch modules of the three stages. That is, the ratio of internal load to output load decreases. This makes internal blocking less likely to occur. The results in Fig. 11 simply indicate that the latter is a more dominant factor.

The results are interesting for the following reason. It is generally known that the complexity of a switch that supports point-to-multipoint connections has to be much larger than the one that supports only point-to-point connections if the nonblocking property is desired [5]. On the other hand, our results indicate that if small blocking probability can be tolerated, higher switch complexity is not necessary.

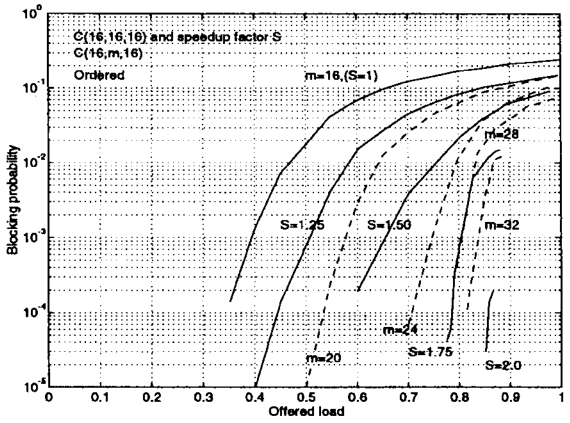


Fig. 12. Internal expansion versus speedup.

For $F = 8$, Fig. 11(b) plots the loss probability of connections of different fanouts. As expected, larger fanout requests are more easily blocked than lower fanout connections. It can be seen that the blocking probability of the $f = 8$ connections is larger than that of the $f = 1$ connections by slightly less than one order of magnitude. Also, the blocking probability of the $f = 8$ connections, the worst case when $F = 8$, is rather comparable to the blocking probability when $F = 1$ [Fig. 11(a)].

Expansion Versus Speedup: The blocking probability can be reduced further by reducing the internal-link loading of the switch. This load can be reduced by two approaches: we can either increase number of intermediate switch modules m or speed up the operation of the switch with respect to the external links. Speeding up the switch by S times reduces the effective offered load by S times.

Fig. 12 shows how these two approaches affect the blocking behavior. In the graph, the dashed curves are obtained by the speedup method while the solid curves are obtained by the expansion method. Point-to-point connections with bandwidth uniformly distributed between zero and one are considered, and ordered routing has been used. For speedup, a $C(16, 16, 16)$ Clos network was chosen. For expansion, n and p of the Clos network were both fixed at 16 while m varies. The figure shows that the blocking probability decreases as m/n or S increases, as expected.

Note that if $m/s = S$ and they are both slightly greater than one (see the case of 1.25 and 1.5 in Fig. 12), expansion offers better performance. This is because when small bandwidth connections are spread across many internal links, most of them will not have enough capacity for subsequent large bandwidth requests. However, increasing m increases the number of alternative routes and makes this kind of blocking less likely to happen.

When S is sufficiently large, say two, speedup is always better than expansion if $m/n = S$, since the internal link would have been sped up enough that the above effect does not come into play anymore. In fact, when $S = 2$, each internal

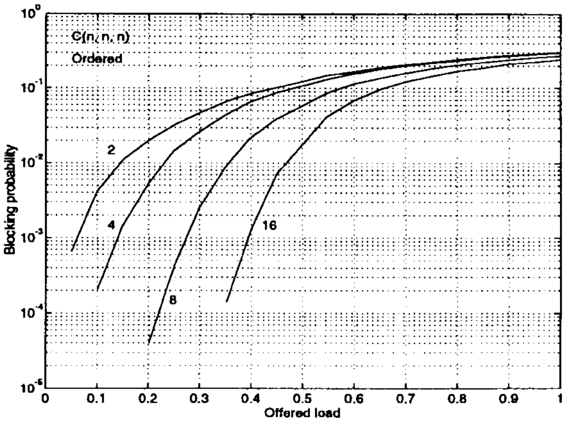


Fig. 13. Effects of switch-module size.

speedup link is at least as good as two nonspeedup links. Furthermore, bandwidth fragmentation is even less likely to occur.

When $m/n = S > 2$, our simulation failed to generate any blocking event, indicating that an expansion or speedup factor of two is probably enough to build a close-to-nonblocking switch. This compares well with the results in the previous section that a speedup factor of about 3 [see (6)] and an expansion factor of about 6 [see (3)] are needed to build a nonblocking switch.

Finally, our experiments indicated a potential problem with the simulation model in Fig. 8. When the offered load λ/μ is close to one and when internal blocking is rare, the input and output links are highly loaded. Therefore, the external blocking probability is large. A call may be blocked many times before being accepted for internal blocking testing. This may lead to an unstable situation where the number of calls waiting in the feedback loop in Fig. 8(a) grows indefinitely and where the arrival rate of calls for internal blocking testing does not match the external arrival rate $N\lambda$. This happens in the case of $S = 1.75$ and 2.0 in Fig. 12: the measured offered load is lower than the external load, and that is why no data for load equal to one was generated. But as mentioned, this occurs only under the combined effect of low internally blocking probability and high offered load; in other words, when the switch is likely to be good enough anyway. Therefore, our model is still useful for switch design in general in that poor designs can be screened out easily.

Size of Switching Modules: We now look at the effects of switch-module size on blocking probability. A set of Clos networks $C(n, n, n)$ where n ranges from 2 to 16 has been considered for point-to-point connections with uniform bandwidth distribution between zero and one. Ordered routing has been used. Fig. 13 shows that the blocking probability decreases as switching-module size increases. This can be explained by the higher degree of sharing of the internal links among connections. That is, each internal link can be accessed by connections from a larger number of external links when n is larger.

V. Summary and Conclusions

This paper has investigated in detail the blocking and nonblocking behavior of multirate Clos switching networks at the connection/virtual connection level. Necessary and sufficient nonblocking conditions which improve on previously known results are derived analytically. Based on the conditions, an optimal bandwidth partitioning scheme can be devised to reduce switch complexity substantially while maintaining the nonblocking property. In this approach, connections with bandwidths greater than 0.5 are routed along a subset of routes while those with bandwidths lower than 0.5 are routed along another disjoint subset of routes. The optimality of the partition bandwidth, 0.5, has been proven. The corresponding switch has an internal bandwidth expansion factor of six.

The blocking behavior of blocking switches supporting multicast connections has been investigated by means of simulation. The advantages and disadvantages of several simulation models, and their relevance to actual switch performance, have been discussed. Although not fully presented in this paper, our experimentation indicates that different models can lead to drastically different simulation results. A novel simulation model has been proposed to factor out the effects of external blocking events without distorting the bandwidth and fanout distributions of requests. In this way, the internal blocking statistics that truly reflect the switch performance can be gathered and studied.

The effects of routing policies, fanout distribution, bandwidth distribution, internal speedup and expansion factor, and switch-module size have been investigated. Among many simulation results, we have shown that for point-to-multipoint connections, a heuristic routing policy that attempts to build a narrow multicast tree can have relatively low blocking probabilities compared with other routing policies. In addition, when small blocking probability can be tolerated, our results indicate that situations with many large-fanout connection requests do not necessarily require a switch architecture of higher complexity compared to that with only point-to-point requests. This contrasts drastically with the nonblocking case, where it is much more costly to build a nonblocking switch when multicasting capability is desired.

References

[1] C.-H. Chow, “On multicast path finding algorithm,” in *Proc. IEEE INFOCOM’91*, Bal Harbour, FL, Apr. 1991, pp. 1274–1283.

[2] S.-P. Chung and K. W. Ross, “On nonblocking multirate interconnection networks,” *SIAM J. Comput.*, vol. 20, no. 4, pp. 726–736, Aug. 1991.

[3] K. Hajikano, K. Murakami, E. Iwabuchi, O. Isono, and T. Kobayashi, “Asynchronous transfer mode switching architecture for broadband ISDN,” in *Proc. IEEE ICC’88*, June 1988, pp. 911–915.

[4] R. Melen and J. S. Turner, “Nonblocking multirate networks,” *SIAM J. Comput.*, vol. 18, no. 2, pp. 301–313, Apr. 1989.

[5] ——, “Nonblocking multirate distribution networks,” *IEEE Trans. Commun.*, vol. 41, pp. 362–369, Feb. 1993.

[6] Y. Sakurai, N. Ido, S. Gohara, and N. Endo, “Large scale ATM multistage switching network with shared buffer memory switches,” in *Proc. ISS’90*, vol. 4, Stockholm, Sweden, May 1990, pp. 121–126.

[7] H. Suzuki, H. Nagano, T. Suzuki, T. Takeuchi, and S. Iwasaki, “Output-buffer switch architecture for asynchronous transfer mode,” in *Proc. IEEE ICC’89*, Boston, MA, June 1989, pp. 99–103.

[8] I. Svinnset, “Nonblocking ATM switching networks,” *IEEE Trans. Commun.*, vol. 42, pp. 1352–1358, Feb./Mar./Apr. 1994.

[9] E. Valdimarsson, “Blocking in multirate interconnection networks,” *IEEE Trans. Commun.*, vol. 42, pp. 2028–2035, Feb./Mar./Apr. 1994.

[10] S. C. Liew, “Multicast routing in 3-stage Clos ATM switching networks,” *IEEE Trans. Commun.*, vol. 42, pp. 1380–1390, Feb./Mar./Apr. 1994.

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

[12] C. Clos, "A study of nonblocking switching networks," *Bell Syst. Tech. J.*, vol. 32, pp. 406–424, 1953.

[13] J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*. Norwell, MA: Kluwer, 1990.

Soung C. Liew (S'84–M'87–SM'92), for photograph and biography, see p. 55 of the February 1998 issue of this TRANSACTIONS.

Ming-Hung Ng received the B.Eng. degree in information engineering from Chinese University of Hong Kong, Shatin, Hong Kong, in 1994. He worked on this paper as part of his undergraduate final year project.

Cathy W. Chan (M'97) received her B.Eng. and Ph.D. degrees in information engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 1994 and 1997, respectively.

She is currently a Postdoctoral Fellow at the Chinese University of Hong Kong. Her research interests include routing and multicasting strategies in broadband switching networks and traffic control in ATM networks.