On Blocking Probability of Multicast Networks

Yuanyuan Yang, *Member, IEEE*, and Jianchao Wang

***Abstract*— Multicast is a vital operation in both broad-band integrated services digital networks (BISDN) and scalable parallel computers. In this paper we look into the issue of supporting multicast in the widely used three-stage Clos network or $v(m,n,r)$ network. Previous work has shown that a nonblocking $v(m,n,r)$ multicast network requires a much higher network cost than a $v(m,n,r)$ permutation network. However, little has been known on the blocking behavior of the $v(m,n,r)$ multicast network with only a comparable network cost to a permutation network. In this paper we first develop an analytical model for the blocking probability of the $v(m,n,r)$ multicast network and then study the blocking behavior of the network under various routing control strategies through simulations. Our analytical and simulation results show that a $v(m,n,r)$ network with a small number of middle switches m, such as $m = n + c$ or dn, where c and d are small constants, is almost nonblocking for multicast connections, although theoretically it requires $m \geq \Theta(n(\log r/\log\log r))$ to achieve nonblocking for multicast connections. We also demonstrate that routing control strategies are effective for reducing the blocking probability of the multicast network. The best routing control strategy can provide a factor of two to three performance improvement over random routing. The results indicate that a $v(m,n,r)$ network with a comparable cost to a permutation network can provide cost-effective support for multicast communication.**

***Index Terms*—Blocking probability, discrete event simulation, multicast communication, performance analysis, routing algorithm.**

I. Introduction

MULTICAST or one-to-many communication is highly demanded in broad-band integrated services digital networks (BISDN) and scalable parallel computers. Some examples are video conference calls and video-on-demand services in BISDN networks, and barrier synchronization and write update/invalidate in directory-based cache coherence protocols in parallel computers. In general, providing multicast support at hardware/network level is the most efficient way supporting such communication operations [1], [2]. In this paper we look into the issue of supporting multicast in the well-known three-stage Clos network [3], [4]. Clos-type networks have been widely used in various interconnection problems. Some recent applications include the NEC ATOM switch designed for BISDN [5], the IBM GF11 multiprocessor [6], and the ANSI Fiber Channel Standard for interconnection of processors to the input/output (I/O) system. More recently, it was shown [7] that the network in the IBM SP2 [8] is functionally equivalent to the Clos network.

Clos-type networks have been extensively studied for both one-to-one communication and multicast communication in the literature. For this type of network, it has been shown [9]–[13] that a nonblocking multicast network requires a much higher network cost than a permutation network [3], [4]. However, little has been known on the blocking behavior of the multicast network with only a comparable network cost to a permutation network. In this paper we first develop an analytical model for the blocking probability of the multicast network and then study the blocking behavior of the network under various routing control strategies through simulations. Our analytical and simulation results show that a network with a comparable cost to a permutation network is almost nonblocking for multicast connections and can provide cost-effective support for multicast communication. We will also demonstrate that routing control strategies are effective for reducing the blocking probability of the multicast network. The best routing control strategy can provide a factor of two to three performance improvement over random routing.

The rest of the paper is organized as follows. In Section II background knowledge for this work is given. In Section III the analytical model for the blocking probability of the multicast network is presented. Section IV shows the simulation results that demonstrate the blocking behavior of the multicast network. Section V compares the analytical model with the simulation results. Section VI concludes the paper.

II. Preliminary and Previous Work

In general, a three-stage Clos network or a $v(m,n,r)$ network has r $(n \times m)$ switches in the first stage (or input stage), m $(r \times r)$ switches in the middle stage, and r $(m \times n)$ switches in the third stage (or output stage). The network has exactly one link between every two switches in its consecutive stages. Fig. 1 illustrates a general schematic of a $v(m,n,r)$ network. Since two of the $v(m,n,r)$ network parameters n and r are restricted by the number of network I/O ports, the main focus of the study is to determine the minimum value of the network parameter m for a certain type of connecting capability to achieve the minimum network cost.

When the $v(m,n,r)$ network is considered for supporting multicast, it is reasonable to assume that every switch in the network has multicast capability. Since output switches have multicast capability, a *multicast connection* from an input port

Paper approved by A. Pattavina, the Editor for Switching Architecture and Performance of the IEEE Communications Society. Manuscript received March 10, 1997; revised September 5, 1997 and February 13, 1998. The work of Y. Yang was supported by the U.S. Army Research Office under Grant DAAH04-96-1-0234 and by the U.S. National Science Foundation under Grant OSR-9 350 540 and Grant MIP-9 522 532. This paper was presented in part at the 1997 International Conference on Parallel Processing (ICPP'97), Bloomingdale, IL, August 1997.

Y. Yang is with the Department of Computer Science and Electrical Engineering, University of Vermont, Burlington, VT 05405 USA (e-mail: yang@cs.uvm.edu).

J. Wang is with GTE Laboratories, Waltham, MA 02254 USA (e-mail: jwang@gte.com).

Publisher Item Identifier S 0090-6778(98)05162-9.

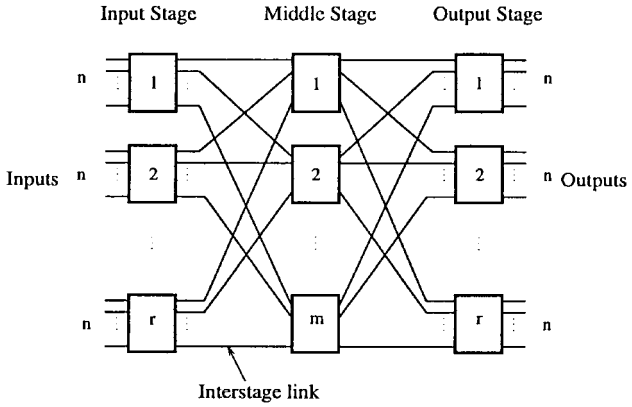

Fig. 1. A general schematic of an $N \times N$ $v(m,n,r)$ network, where $N = nr$.

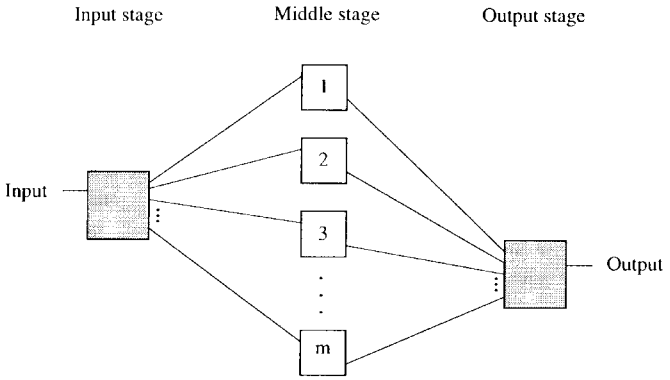

Fig. 2. The paths between a given input and output pair in the $v(m,n,r)$ network.

can be simply expressed in terms of output switches it connects to. The number of output switches in a multicast connection is referred to as the *fanout* of the multicast connection.

Several designs have been proposed for this type of multicast network [9]–[11]. It was shown [9], [10] that a $v(m,n,r)$ network is *nonblocking* for arbitrary multicast connections if the number of middle switches $m \geq cnr$, where c is a constant. By nonblocking, we mean that any arbitrary multicast connection between an idle network input port and a set of idle network output ports can always be realized without any *rearrangement* of the existing connection in the network. A nonblocking multicast network can be considered as a logical crossbar network that supports multicast. As can be seen, the number of middle switches required for a nonblocking multicast network is much larger than that for a nonblocking permutation network which requires only $m \geq 2n-1$ [3]. This is mainly due to the nonuniform nature of multicast connections. Aiming at the nonuniformity of multicast connections, the most recent design [11] employed a routing control strategy which can effectively reduce such nonuniformity and obtained the currently best available sufficient nonblocking condition for multicast connections $m \geq 3(n-1)(\log r/\log\log r)$. Although the new condition significantly improved the previous sufficient condition, it is still considered too large for real applications. On the other hand, a necessary condition $m = \Theta(n(\log r/\log\log r))$ was obtained [12] for this type of multicast network to be nonblocking under three typical routing control strategies, which matches the sufficient condition in [11]. This suggests that there is little room for further improvement on the nonblocking condition for multicast connections. However, note that the previous work has primarily focused on the analysis of worst-case network states, that is, determining the number of middle switches m which can guarantee the network nonblocking for any multicast connections. Little has been known on the behavior of the $v(m,n,r)$ multicast network with only a comparable network cost to a permutation network. There are many important problems concerning the $v(m,n,r)$ multicast networks that remain to be studied. In particular, we are interested in the following questions. How frequent do the worst-case network states occur? If the number of middle switches is reduced, what is the blocking probability for multicast connections? For a given number of middle switches, which routing control strategy performs the best?

In this paper we study the blocking behavior of the $v(m,n,r)$ multicast networks with various m values, especially with smaller m than the theoretical nonblocking condition. We report in the following the work performed along two parallel lines: 1) develop an analytical model for the blocking probability of the $v(m,n,r)$ multicast networks and 2) look into the blocking behavior of the networks under various routing control strategies through simulations.

III. The Analysis of Multicast Blocking Probability

In this section we provide an analytical model for the blocking probability of the $v(m,n,r)$ multicast network.

A. Previous Analytical Models for $v(m,n,r)$ Networks

In general, determination of blocking probability in a multistage network (even for permutation networks) is inherently complex and difficult. This is due to the fact that there are many possible paths to consider in a typical large network, and the dependencies among links in the network lead to combinatorial explosion problems. To the best of our knowledge, previous work on blocking probability of $v(m,n,r)$ networks was done only for permutation networks. Several analytical models have been proposed in the literature, for example, [14]–[19]. C. Y. Lee [14], [17] gave the simplest method for analyzing the blocking probability for the $v(m,n,r)$ permutation network, in which the events that individual links are busy are assumed to be independent. To see how this model works, let's consider the paths between a given I/O pair in Fig. 2. Let the probability that a typical input port is busy be a and the probability that a typical output port is busy be also a, and assume that the incoming traffic is uniformly distributed over the m interstage links. Then the probability that an interstage link is busy is given by

$$p = \frac{an}{m} \tag{1}$$

and the probability that an interstage link is idle is given by

$$q = 1 - p. \tag{2}$$

Since one path from the source input to the destination output consists of two interstage links, if any link in the path is busy,

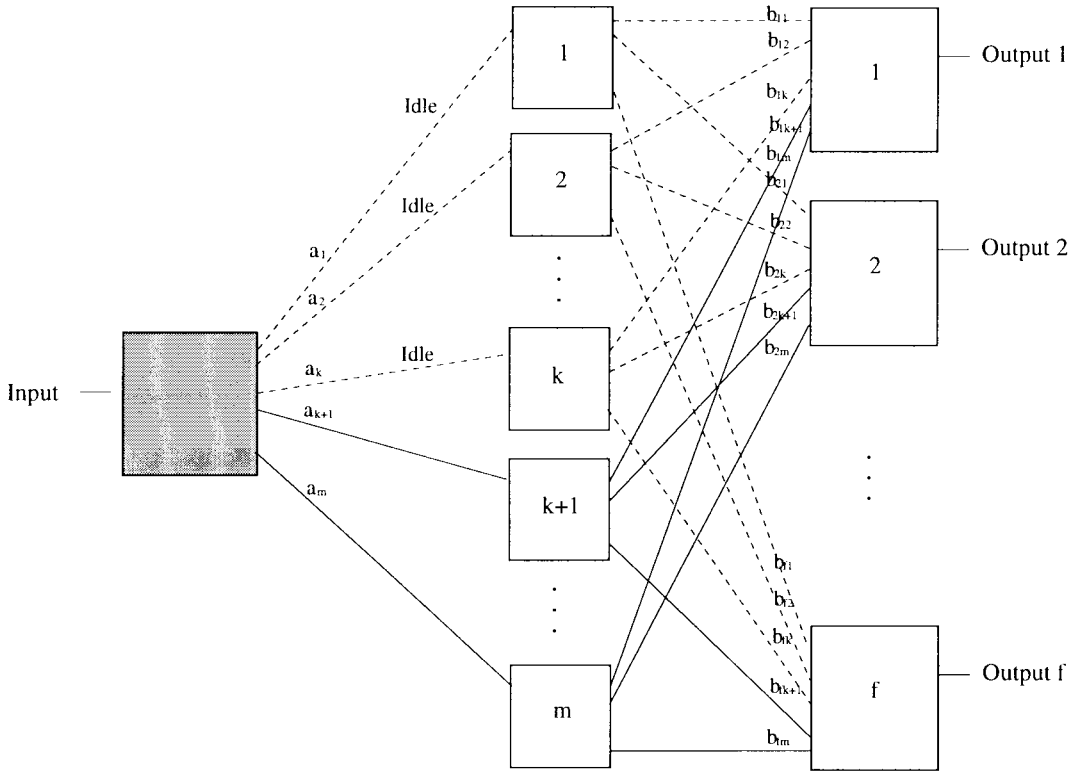

Fig. 3. The subnetwork associated with a multicast connection with fanout f. The dashed lines indicate the idle link subnetwork.

the path cannot be used for realizing the new connection from the source input to the destination output. Thus, the probability that one path cannot be used is $(1-q^2)$. As shown in Fig. 2, there are a total of m distinct paths from the source input to the destination output and any two of such paths are link disjoint (i.e. no shared links). With the link independence assumption, the probability that no idle path is available for making the connection between the given I/O, or the blocking probability, is given by

$$P_B = (1-q^2)^m. \tag{3}$$

B. Generalizing Lee's Approach to Multicast

It is interesting to see if we can generalize Lee's approach to $v(m,n,r)$ multicast networks. Recall that a multicast connection is represented by the output switches it connects to. Given a multicast connection request with fanout f $(1 \le f \le r)$, let $P_B(f)$ be the probability that this connection request cannot be satisfied, that is, the blocking probability for this connection request. Clearly, according to our definition of fanout, this connection will connect to f distinct output switches. Fig. 3 depicts the subnetwork associated with this multicast connection. In the subnetwork, one input switch is linked to m middle switches and each of the m middle switches is linked to f output switches. Denote the interstage links between the input switch and m middle switches as $a_1, a_2, \cdots, a_m$ (also referred to as input–middle interstage links), and the interstage links between the m middle switches and the kth $(1 \le k \le f)$ output switches as $b_{k1}, b_{k2}, \cdots, b_{km}$ (also referred to as middle–output interstage links). All paths realizing a multicast connection in the network can be considered as a multicast tree. There are following possible ways (i.e., possible multicast trees) to realize a multicast connection with fanout f.

Case 1: The connection is routed through an input–middle interstage link to a middle switch and then multicast to f destination output switches through f middle–output interstage links [see Fig. 4(a)]. In this case the probability of success is $q \cdot q^f = q^{f+1}$ and the probability of failure is $(1-q^{f+1})$. With a total of m middle switches, there are m possible ways to realize this connection request.

Case k $(2 \le k \le f)$: The connection is routed through k input–middle interstage links to k middle switches and then multicast from these k middle switches to a total of f destination output switches through f middle–output interstage links. Fig. 4(b) illustrates an example of $k=2$. In this case the probability of success is $q^k \cdot q^f = q^{f+k}$ and the probability of failure is $(1-q^{f+k})$. Note that there are $\binom{m}{k}$ ways to choose the k middle switches. For the given k middle switches, there are $S(f,k) \cdot k!$ ways to partition the f destination output switches to k disjoint sets so that each of the k middle switches is routed to a different set of destination output switches, where $S(f,k)$ is the Stirling number of the second kind [20].

After considering all possible cases, there are a total of $\Sigma_{j=1}^{f} \binom{m}{j} S(f,j)j!$ ways to realize the connection request. Note that the blocking probability for the multicast connection is the probability that all possible multicast trees fail to realize the connection. Under the link independence assumption, if there are not any shared links between any two multicast trees, the blocking probability for the connection is simply the product of each probability that an individual multicast tree fails to realize the connection. Now let's examine whether these multicast trees are link disjoint or not. In Case 1 m

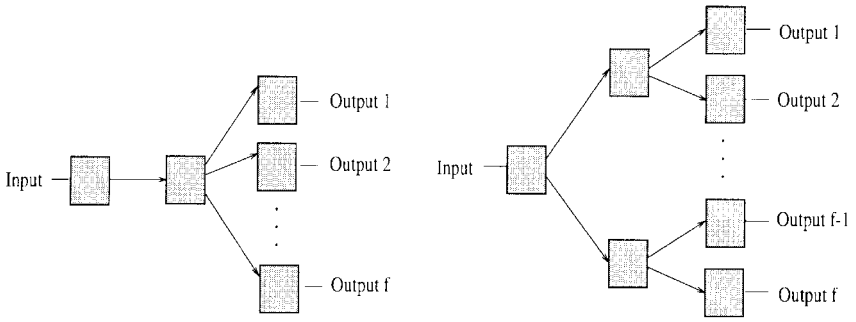

Fig. 4. Different ways to realize a multicast connection. (a) Fanout to one middle switch. (b) Fanout to two middle switches.

multicast trees are indeed link disjoint. Thus, the probability that no multicast tree can be used to realize the connection is $(1-q^{f+1})^m$. However, in Case k $(k \geq 2)$, some multicast trees share common interstage links. In addition, some multicast trees in different cases also share common interstage links. All of these dependencies among the multicast trees make the problem almost intractable. Consequently, we cannot simply extend Lee's approach to multicast networks.

C. Blocking Probability of Multicast Connections

In the following we employ a different approach to derive the blocking probability for the $v(m,n,r)$ multicast networks. We still follow Lee's assumption that the events that individual links are busy are independent.

Consider the subnetwork in Fig. 3. Let $\boldsymbol{\varepsilon}$ be the event that the connection request with fanout f cannot be realized in the subnetwork. Notice that any interstage link is either busy or idle. Denote the event that the link a_i is busy as $\boldsymbol{a_i}$ and the event that the link a_i is idle as $\overline{\boldsymbol{a_i}}$ for $1 \leq i \leq m$. Let σ represent the state of the input–middle interstage links $a_1, a_2, \cdots, a_m$, $P(\varepsilon|\sigma)$ be the conditional blocking probability in this state, and $P(\sigma)$ be the probability of being in state σ. If, in state σ, ka_i's are idle and the rest of a_i's are busy, by the link independence assumption we have $P(\sigma) = q^k p^{m-k}$. Considering all states of input–middle interstage links $a_1, a_2, \cdots, a_m$ and using symmetry of the states, it is apparent that the *blocking probability for a multicast connection with fanout* f is given by

$$\begin{aligned} P_B(f) &= P(\boldsymbol{\varepsilon}) \\ &= \sum_{\sigma} P(\sigma) P(\boldsymbol{\varepsilon}|\sigma) \\ &= \sum_{k=0}^{m} \binom{m}{k} q^k p^{m-k} P(\boldsymbol{\varepsilon}|\overline{\boldsymbol{a_1}}, \cdots, \overline{\boldsymbol{a_k}}, \boldsymbol{a_{k+1}}, \boldsymbol{a_m}). \end{aligned} \tag{4}$$

Under the condition that $a_{k+1}, \cdots, a_m$ are busy and the rest of a_i's are idle, finding the blocking probability of the network is equivalent to finding the blocking probability of a smaller subnetwork which is obtained by removing the busy input–middle interstage links, the middle switches connected to these links, and the middle–output interstage links associated with these middle switches in the original network. Such a subnetwork is shown in Fig. 3 by dashed lines. We have the following lemma concerning the blocking property of this subnetwork.

Lemma 1: Assume that the interstage links $a_1, a_2, \cdots, a_k$ in the subnetwork shown in Fig. 3 all are idle. A multicast connection from an input of the input switch to the f distinct output switches cannot be realized if and only if there exists an output switch whose all k inputs (i.e., middle–output interstage links) are busy.

Proof: If there exists an output switch whose all k inputs are busy, then there is not any idle path to connect the input switch to this output switch. On the other hand, if there exists at least one idle input on each of the f output switches, noticing that all input–middle interstage links $a_1, a_2, \cdots, a_k$ are idle, then there exist idle paths from the input switch to all the f output switches, and these paths form a multicast tree which can be used to realize the connection request. □

Let $\boldsymbol{\varepsilon'}$ be the event that the connection request with fanout f cannot be realized in the idle link subnetwork in Fig. 3. Based on the above discussion, we have

$$P(\varepsilon') = P(\boldsymbol{\varepsilon}|\overline{\boldsymbol{a_1}}, \cdots, \overline{\boldsymbol{a_k}}, \boldsymbol{a_{k+1}}, \cdots, \boldsymbol{a_m}). \tag{5}$$

On the other hand, for a middle–output interstage link b_{ij} which is an input to the ith output switch in Fig. 3, let $\boldsymbol{b_{ij}}$ be the event that b_{ij} is busy, where $1 \leq i \leq f$ and $1 \leq j \leq k$. From Lemma 1, event $\boldsymbol{\varepsilon'}$ can be expressed in terms of events $\boldsymbol{b_{ij}}$'s as follows:

$$\begin{aligned} \boldsymbol{\varepsilon'} = (\boldsymbol{b_{11}} \cap \boldsymbol{b_{12}} \cap \cdots \cap \boldsymbol{b_{1k}}) \cup (\boldsymbol{b_{21}} \cap \boldsymbol{b_{22}} \cap \cdots \boldsymbol{b_{2k}}) \\ \cup \cdots \cup (\boldsymbol{b_{f1}} \cap \boldsymbol{b_{f2}} \cap \cdots \cap \boldsymbol{b_{fk}}). \end{aligned} \tag{6}$$

By the link independence assumption, for any $i \neq i'$ or $j \neq j'$, event $\boldsymbol{b_{ij}}$ and $\boldsymbol{b_{i'j'}}$ are independent, and for any $i \neq i'$, event $(\boldsymbol{b_{i1}} \cap \boldsymbol{b_{i2}} \cap \cdots \cap \boldsymbol{b_{ik}})$ and event $(\boldsymbol{b_{i'1}} \cap \boldsymbol{b_{i'|bf2}} \cap \cdots \cap \boldsymbol{b_{i'k}})$ are independent since there are not any shared links. Therefore, from (6) and by De Morgan's laws, the probability of event $\boldsymbol{\varepsilon'}$ is given by

$$\begin{aligned} P(\varepsilon') &= 1 - \prod_{i=1}^{f} P(\overline{\boldsymbol{b_{i1}} \cap \boldsymbol{b_{i2}} \cap \cdots \cap \boldsymbol{b_{ik}}}) \\ &= 1 - \prod_{i=1}^{f} [1 - P(\boldsymbol{b_{i1}} \cap \boldsymbol{b_{i2}} \cap \cdots \cap \boldsymbol{b_{ik}})] \\ &= 1 - \prod_{i=1}^{f} (1 - p^k) = 1 - (1-p^k)^f. \end{aligned} \tag{7}$$

Combing (4), (5), and (7), we obtain the blocking probability for a multicast connection with fanout f

$$P_B(f) = \sum_{k=0}^{m} \binom{m}{k} q^k p^{m-k}[1-(1-p^k)^f]. \quad (8)$$

In particular, letting $f = 1$, we have

$$\begin{aligned} P_B(1) &= \sum_{k=0}^{m} \binom{m}{k} q^k p^{m-k}[1-(1-p^k)] \\ &= p^m \sum_{k=0}^{m} \binom{m}{k} q^k \\ &= p^m(1+q)^m = (1-q)^m(1+q)^m = (1-q^2)^m. \end{aligned}$$

This is exactly Lee's blocking probability for the $v(m,n,r)$ permutation network [14], [17].

Moreover, by expanding $(1-p^k)^f$ in (8), we can write $P_B(f)$ in a different way

$$\begin{aligned} P_B(f) &= \sum_{k=0}^{m} \binom{m}{k} q^k p^{m-k} \sum_{i=1}^{f} \binom{f}{i} (-1)^{i-1} p^{ik} \\ &= \sum_{i=1}^{f} \binom{f}{i} (-1)^{i-1} \sum_{k=0}^{m} \binom{m}{k} (qp^i)^k p^{m-k} \\ &= \sum_{i=1}^{f} \binom{f}{i} (-1)^{i-1} (qp^i + p)^m \\ &= p^m \sum_{i=1}^{f} \binom{f}{i} (-1)^{i-1} [1+(1-p)p^{i-1}]^m. \quad (9) \end{aligned}$$

It is easy to verify that the blocking probability $P_B(f)$ in (8) is an increasing sequence of fanout f. In other words, it is more difficult to realize a multicast connection with a larger fanout, which is consistent with our intuition. Fig. 5 gives some numerical examples of $P_B(f)$ in (5). From Fig. 5 we can see that for a fixed m, the blocking probability increases as the fanout gets larger, and for a fixed fanout, the blocking probability decreases sharply as m gets larger. We will have more discussions and comparisons on the property of the blocking probability in a later section.

In general, we may be more interested in the typical behavior of the blocking probability and ask about its "average" value over all fanouts. Suppose the probability distribution for different fanouts in a multicast connection is

$$\left\{ w_f | 0 \le w_f \le 1, 1 \le f \le r, \sum_{i=1}^{r} w_i = 1 \right\}.$$

Then the "average" value of the blocking probability can be written as

$$P_B = \sum_{f=1}^{r} P_B(f) \cdot w_f. \quad (10)$$

Now, suppose the fanout is uniformly distributed over 1 to r. Then (10) becomes

$$P_B = \frac{1}{r} \sum_{f=1}^{r} P_B(f) = \frac{1}{r} \sum_{f=1}^{r} \sum_{k=0}^{m} \binom{m}{k} q^k p^{m-k}[1-(1-p^k)^f]. \quad (11)$$

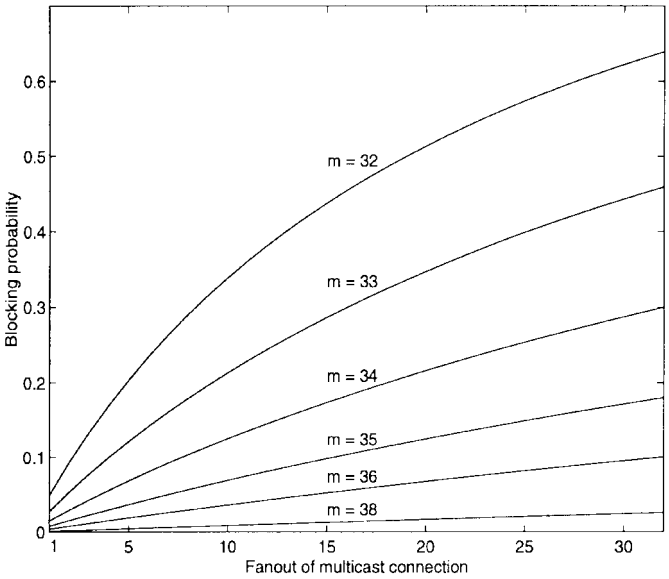

Fig. 5. Blocking probabilities for $v(m,32,32)$ network with fanouts between 1 and 32. $N = 1024$, $n = r = 32$, and $a = 0.7$.

In the rest of the paper we simply refer to P_B as the *blocking probability of the $v(m,n,r)$ multicast network.*

D. Asymptotic Bound on the Blocking Probability

Since there is (apparently) no closed form for the blocking probability P_B in (11), it is appropriate that we derive a closed form for the asymptotic bound on it.

Since we are interested in the networks with small m values, the following two cases are considered.

Case 1: $m = n + c$ for some constant integer $c > 1$.

Case 2: $m = dn$ for some constant $d > 1$.

In our analysis, we need the following inequality:

$$1-(1-x)^l < lx \quad (12)$$

where $0 < x < 1$, and l is an integer ≥ 1.

By applying (12) to (11), we can obtain an upper bound on P_B

$$\begin{aligned} P_B &< \frac{1}{r} \sum_{f=1}^{r} \sum_{k=0}^{m} \binom{m}{k} q^k p^{m-k} \cdot f \cdot p^k \\ &= \frac{1}{r}(1-q^2)^m \sum_{f=1}^{r} f = \frac{r+1}{2}[1-(1-p)^2]^m. \quad (13) \end{aligned}$$

Consider Case 1 first. Suppose $m = n + c$ for some constant integer $c > 1$. As discussed in Section III-A, we have $p = (an)/m$, where a is a constant and $0 \le a < 1$. Then

$$[1-(1-p)^2]^m = \left[1-\left(1-\frac{an}{m}\right)^2\right]^m < [1-(1-a)^2]^m$$

which implies

$$P_B = O(r \cdot \delta^m) \quad (14)$$

where $\delta = 1-(1-a)^2$. Clearly, δ is a constant such that $0 < \delta < 1$.

TABLE I
COMPARISON BETWEEN THE $P_B(f)$ IN (8) AND THE $P_B(f)$ IN (17) FOR $n = 32$, $m = 64$, $r = 32$, AND $a = 0.7$

Fanout f	$P_B(f)$ in (8)	$P_B(f)$ in (17)
1	5.46×10^{-16}	2.77×10^{-17}
2	1.09×10^{-15}	5.55×10^{-17}
5	2.74×10^{-15}	1.46×10^{-16}
8	4.38×10^{-15}	2.33×10^{-16}
12	6.57×10^{-15}	3.51×10^{-16}
16	8.76×10^{-15}	4.66×10^{-16}
20	1.10×10^{-14}	5.87×10^{-16}
24	1.31×10^{-14}	6.99×10^{-16}
28	1.53×10^{-14}	8.12×10^{-16}
32	1.75×10^{-14}	9.33×10^{-16}

Now consider Case 2. Suppose $m = dn$ for some constant $d > 1$. Since $p < (n/m) = (1/d)$, (13) becomes

$$P_B < \frac{r+1}{2}\left[1 - \left(1 - \frac{1}{d}\right)^2\right]^m = O(r \cdot \delta'^m) \qquad (15)$$

where $\delta' = 1 - (1 - (1/d))^2$. Similarly, δ' is a constant such that $0 < \delta' < 1$.

Notice that, in both cases, if $r = O(n)$, we obtain $P_B = O(e^{-\epsilon n})$, where ϵ is a constant >0. We can see that the blocking probability tends to zero very quickly as n increases. In other words, for a sufficiently large n, the network is almost nonblocking for multicast connections. This means that, in practice, even when the network parameter m is as small as dn or $n + c$, the network performance is still fairly good. Such m values are much smaller than the theoretical bound $\Theta(n(\log r / \log\log r))$ given in [11] and [12].

E. More Accurate Blocking Probability

In the following we derive a more accurate formula for the blocking probability. Note that we consider only one-to-many or one-to-one connections, and there are n outputs on an output switch. If an output switch is chosen as one of destinations in a multicast connection, this output switch must have at least one idle output and have at most $n-1$ busy inputs. Recall Lemma 1 and Fig. 3. In the case of $k > n - 1$ there must exist some idle input on each of f output switches. Thus, the conditional blocking probability in this case becomes zero. Therefore, the conditional blocking probability can be modified to

$$P(\varepsilon|\overline{a_1}, \cdots, \overline{a_k} a_{k+1}, \cdots, a_m) = \begin{cases} 1 - (1 - p^k)^f, & \text{if } 1 \le k \le n-1 \\ 0, & \text{if } k \ge n. \end{cases} \qquad (16)$$

and the blocking probability for a multicast connection with fanout f is now given by

$$P_B(f) = \sum_{k=m-n}^{m} \binom{m}{k} q^k p^{m-k}[1 - (1 - p^k)^f] = \sum_{i=0}^{n} \binom{m}{m-n+i} q^{m-n+i} p^{n-i} \cdot [1 - (1 - p^{m-n+i})^f]. \qquad (17)$$

The $P_B(f)$ in (17) is slightly smaller than that in (8). Table I shows the difference between them for $m = 2n$.

F. Some Extensions to the Asymmetric Clos Networks

The above analysis of blocking probability can be easily extended to the asymmetric Clos-type networks or $v(m, n_1, r_1, n_2, r_2)$ networks, where n_1 is the number of inputs on each input switch, r_1 is the number of input switches, n_2 is the number of outputs on each output switch, and r_2 is the number of output switches. In this case the probability that an input–middle interstage link is busy might differ from the probability that a middle–output interstage link is busy. In fact, in Fig. 3 let $P(\overline{a_i}) = p_a = (an_1/m), q_a = 1 - p_a$ for $1 \le i \le m$, and $P(\overline{b_{i,j}}) = p_b = (an_2/m, q_b = 1 - p_b$ for $1 \le i \le f$ and $1 \le j \le m$, where p_a is not necessarily equal to p_b. Then the blocking probability for a multicast connection with fanout f in (8) or (9) can be rewritten as

$$P_B(f) = \sum_{k=0}^{m} \binom{m}{k} q_a^k p_a^{m-k}[1 - (1 - p_b^k)^f] \qquad (18)$$

or

$$P_B(f) = \sum_{i=1}^{f} \binom{f}{i} (-1)^{i-1} (q_a p_b^i + p_a)^m. \qquad (19)$$

In particular, letting $f = 1$ in (18), Lee's blocking probability for permutation networks becomes

$$\begin{aligned} P_B(1) &= \sum_{k=0}^{m} \binom{m}{k} q_a^k p_a^{m-k}[1 - (1 - p_b^k)] \\ &= \sum_{k=0}^{m} \binom{m}{k} (q_a p_b)^k p_a^{m-k} \\ &= (q_a p_b + p_a)^m = [q_a(1 - q_b) + (1 - q_a)]^m \\ &= (1 - q_a q_b)^m. \end{aligned}$$

Also, the $P_B(f)$ equivalent to (17) can be written as

$$P_B(f) = \sum_{i=0}^{n} \binom{m}{m-n+i} q_a^{m-n+i} p_a^{n-i}[1 - (1 - p_b^{m-n+i})^f]. \qquad (20)$$

IV. EXPERIMENTAL STUDY OF THE BLOCKING BEHAVIOR OF MULTICAST NETWORKS

In the last section we have developed an analytical model under the link independence assumption for the blocking probability of the $v(m, n, r)$ multicast network. Our model indicates that the blocking probability is very small even for small m, such as $m = n + c$ or dn. In this section we look into this issue through the simulation of real networks. As discussed in [11], a routing control strategy plays an important role in reducing the nonuniformity of multicast connections and, in turn, reducing the blocking probability of the $v(m, n, r)$ multicast network. Therefore, it is more appropriate to study the blocking behavior of the network under a good routing control strategy. In our simulation we employ seven different routing control strategies and compare the blocking probabilities under all of these control strategies.

A. Generic Routing Algorithm

We start from describing a generic routing algorithm in which different routing control strategies can be embedded.

First of all, we need the following definitions on the states of the $v(m,n,r)$ network.

To characterize the connection state between the input stage and middle stage in a $v(m,n,r)$ network, for any input port $i \in \{1,2\cdots,nr\}$, we refer to the set of middle switches with currently unused links to the input switch associated with input port i as the *available middle switches*.

To characterize the state of m switches in the middle stage of a $v(m,n,r)$ network, let $M_j, j \in \{1,2,\cdots,m\}$ denote the subset of output switches to which middle switch j is currently providing connection paths from the input ports. M_j is referred to as the *destination set* of middle switch j. Clearly, we have $M_j \subseteq \{1,2,\cdots,r\}$.

Given a $v(m,n,r)$ multicast network with destination sets $M_1, M_2, \cdots, M_m$ and a new connection request from input port i, I_i (I_i is defined as the output switches to be connected from input port i in the multicast connection), the main function of a routing algorithm is to choose a set of middle switches which can satisfy the connection request. It was shown [12] that a connection request I_i can be satisfied by using some x $(x \geq 1)$ middle switches, say, $j_1, j_2, \cdots, j_x$, from among the available middle switches of a $v(m,n,r)$ network if and only if

$$I_i \cap \left(\bigcap_{k=1}^{x} M_{j_k} \right) = \phi. \tag{21}$$

Note that both I_i's and M_j's are subsets of set $\{1,2,\cdots,r\}$. Setting $I_i = \{1,2,\cdots,r\}$ in (21), we obtain

$$\bigcap_{k=1}^{x} M_{j_k} = \phi. \tag{22}$$

This means that any x middle switches satisfying condition (22) can be used to satisfy an arbitrary connection request.

Now, we provide the generic algorithm for routing in a $v(m,n,r)$ multicast network.

Algorithm:

Step 1: If there are no available middle switches for the current connection request, then exit without making the connection; otherwise go to Step 2.

Step 2: Choose a nonfull middle switch (i.e., a middle switch with at least one idle output link) among the available middle switches for the connection request according to some control strategy. If no such middle switch exists, then exit without making the connection.

Step 3: Realize as large as possible portion of the connection request in the middle switch chosen in Step 2.

Step 4: Update the connection request by discarding the portion that is satisfied by the middle switch chosen in Step 2.

Step 5: If the connection request is nonempty, go to Step 1.

End.

It is easy to see that, at the normal termination of the above algorithm, the middle switches chosen by the algorithm satisfies (21), while abnormal termination in Steps 1 or 2 represents a blocking case.

B. Routing Control Strategies

In Step 2 of the above generic routing algorithm there are many ways to choose middle switches among available middle switches for satisfying a connection request. Due to the nonuniform nature of multicast connections, if no control strategy is employed, we can expect that the number of middle switches required for nonblocking becomes large. Hence, we must employ some type of "intelligent" control strategy to reduce such nonuniformity of multicast connections. In the following we describe seven control strategies for choosing middle switches from the available middle switches in a $v(m,n,r)$ multicast network.

- *Smallest Absolute Cardinality Strategy*: Choose a middle switch whose destination set has the smallest cardinality.
- *Largest Absolute Cardinality Strategy*: Choose a middle switch whose destination set has the largest cardinality.
- *Average Absolute Cardinality Strategy*: Choose a middle switch such that the cardinality of its destination set is equal to the average cardinality of all available middle switches.
- *Smallest Relative Cardinality Strategy*: Choose a middle switch whose destination set has the smallest cardinality with respect to the connection request (that is, first intersect the connection request with the destination sets and then choose the smallest cardinality).
- *Largest Relative Cardinality Strategy*: Choose a middle switch whose destination set has the largest cardinality with respect to the connection request.
- *Average Relative Cardinality Strategy*: Choose a middle switch such that the cardinality of its destination set with respect to the connection request is equal to the average cardinality of all available middle switches with respect to the connection request.
- *Random Strategy*: Choose a middle switch at random.

Note that for a given connection request in a given network state, the routability of the connection request does not depend on the control strategy used. However, different strategies may lead to different network states after satisfying this connection request and thus have a long-term effect on the blocking behavior of the network. In the next two subsections we demonstrate how these control strategies affect the blocking probability of the network through simulations.

C. Simulation Model

We have developed a discrete event simulator which simulates the $v(m,n,r)$ multicast network to study the blocking behavior of the network under different routing control strategies.

1) Model Assumption: The discrete event simulator used to evaluate the performance of $v(m,n,r)$ multicast network is based on the following assumptions.

- Three types of traffic distributions are considered: uniform traffic, uniform/constant traffic, and Poisson traffic. In the uniform traffic model both the interarrival time of connection requests and the connecting time of each multicast connection follow the uniform distribution. In the uniform/constant traffic model the interarrival time of connection requests follows the uniform distribution and the connecting time of each multicast connection is a constant. In the Poisson traffic model the arrival process of connection requests is a Poisson process (that is, the interarrival time of connection requests follows the exponential distribution) and the connecting time of each multicast connection follows the exponential distribution.
- The network is considered as a multiple-server queueing system with the number of servers varies from n to N, depending on the network state.
- In the steady state the arrival rate of the connection requests is approximately equal to the departure rate (service rate) of the connections.
- A new multicast connection request is randomly generated among all idle network input ports and idle network output ports. In particular, the fanout of a new connection request is the smaller of a number randomly chosen from $\{1, 2, \cdots, r\}$ and the number of output switches with idle output ports in the current network state. Clearly, only legal connection requests are generated in the simulation.
- During the network operation, a certain workload is maintained. The workload is measured by the network utilization, which is defined as

$$\text{Network utilization} = \frac{\text{The total number of busy output ports}}{N}.$$

- The blocking probability in the simulation is computed by

$$P_B = \frac{\text{The total number of connection requests blocked}}{\text{The total number of connection requests generated}}.$$

2) The Simulator: The network simulator can accept any network size, workload, routing control strategy, and connection request distribution. The simulator has three main components: *network initializer*, *connection/disconnection handler*, and *data collector*.

The *network initializer* module initializes the network to a prespecified network utilization ratio. Starting from an empty network, connection requests are randomly generated and realized according to the routing control strategy. If a connection is blocked, the initializer discards this request and increments the blocking counter. The initialization process terminates when the utilization ratio is reached.

The *connection/disconnection handler* module performs basic network operations. It generates connection requests and connecting times according to the traffic model. An event queue is maintained to hold all future connection/disconnection events sorted by arrival/departure time. The event at the head of the queue is handled first. For a connection event, the handler tries to find middle switches from the available middle switches in the network according to the control strategy. If such middle switches are found, the handler realizes the connection and updates the network state, and then inserts a disconnection event for this connection into the event queue according to its departure time. If not found, the handler discards this connection request and increments the blocking counter. For a disconnection event, it releases the switches and links that this connection occupies, and updates the network state.

The *data collector* module records all information regarding the network blocking behavior. It collects the number of connection requests, number of disconnections, total number of connections realized, total number of connections blocked, number of connections, and number of blockings when the network first reaches the utilization ratio.

D. The Simulation Results

Extensive simulations were carried out on the $v(m, n, r)$ multicast networks for different m values under seven routing control strategies. We present and discuss the simulation results for the following two configurations of the $v(m, n, r)$ multicast networks:

Configuration 1: $N = 1024$, $n = r = 32$, and $32 \leq m \leq 48$.

Configuration 2: $N = 4096, n = r = 64$, and $64 \leq m \leq 84$.

For each network size, control strategy, and traffic model, the network is simulated for five runs with different initial network states and the final results are averaged over these five runs. In each run, 5000 connection requests are handled for Configuration 1 and 10 000 connection requests are handled for Configuration 2. In both cases 95% confidence interval is achieved.

In Fig. 6 we plotted the blocking probability corresponding to $32 \leq m \leq 48$ for Configuration 1 and the blocking probability corresponding to $64 \leq m \leq 84$ for Configuration 2. The results were obtained under seven routing control strategies for uniform traffic, uniform/constant traffic, and Poisson traffic with initial network utilization $= 90\%$.

Although network sizes and traffic models differ, the simulation results demonstrate a similar trend. We observe that for all seven control strategies, when $m = n$, the blocking probability is relatively high, and as the number of middle switches increases, the blocking probability decreases quickly. In particular, in Fig. 6 when $m \geq 48 = n + 16 = 1.50n$ for network size 1024 and when $m \geq 84 = n + 20 \approx 1.31n$ for network size 4096, the blocking probabilities approach to zero.

We also see that for any of the three traffic models, the *smallest relative* strategy leads to the lowest blocking probability, the *largest relative* strategy has the highest blocking probability, and other strategies lie in between. This observation indicates that the *smallest relative* strategy, which was employed in the routing algorithm in [11] to achieve the currently best available sufficient nonblocking condition, is also the best among the seven strategies for a $v(m, n, r)$ multicast network with a much smaller m than the nonblocking condition. It is not surprising that the *largest relative*

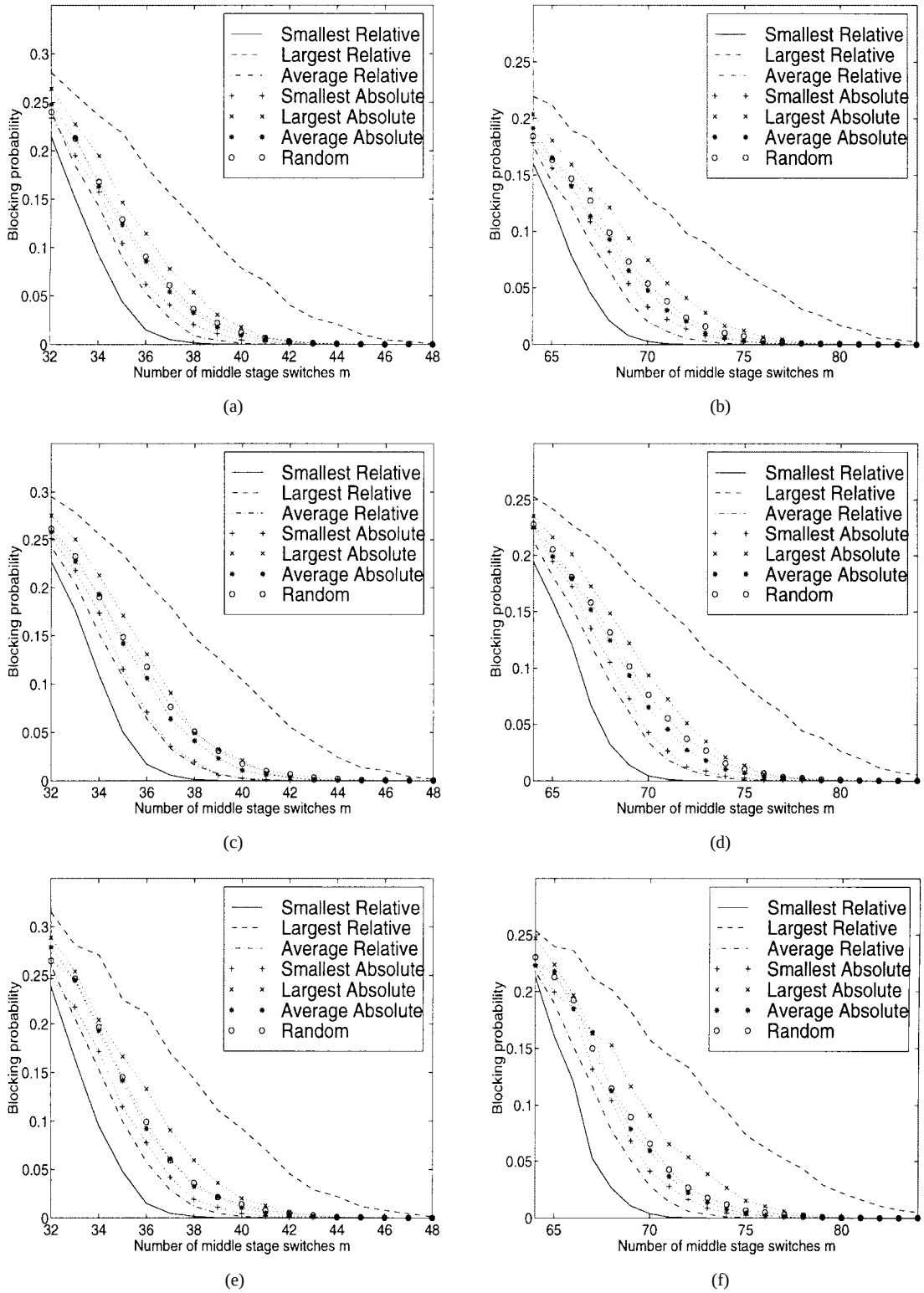

Fig. 6. The blocking probability of the $v(m, 32, 32)$ and $v(m, 64, 64)$ multicast networks under seven routing control strategies. (a) $N = 1024$ under uniform traffic. (b) $N = 4096$ under uniform traffic. (c) $N = 1024$ under uniform/constant traffic. (d) $N = 4096$ under uniform/constant traffic. (e) $N = 1024$ under Poisson traffic. (f) $N = 4096$ under Poisson traffic.

strategy performs the worst. This is because the *largest relative* strategy first tries the middle switch which can realize the smallest portion of the current connection request. The simulation results show that the *average relative* strategy ranks second in achieving lower blocking probability. This is reasonable because this strategy uses some knowledge of the connection request and the middle switch states but does not use it as aggressively as the *smallest relative* strategy. Other strategies (including the *random* strategy) use either no knowledge or less accurate knowledge of the connec-

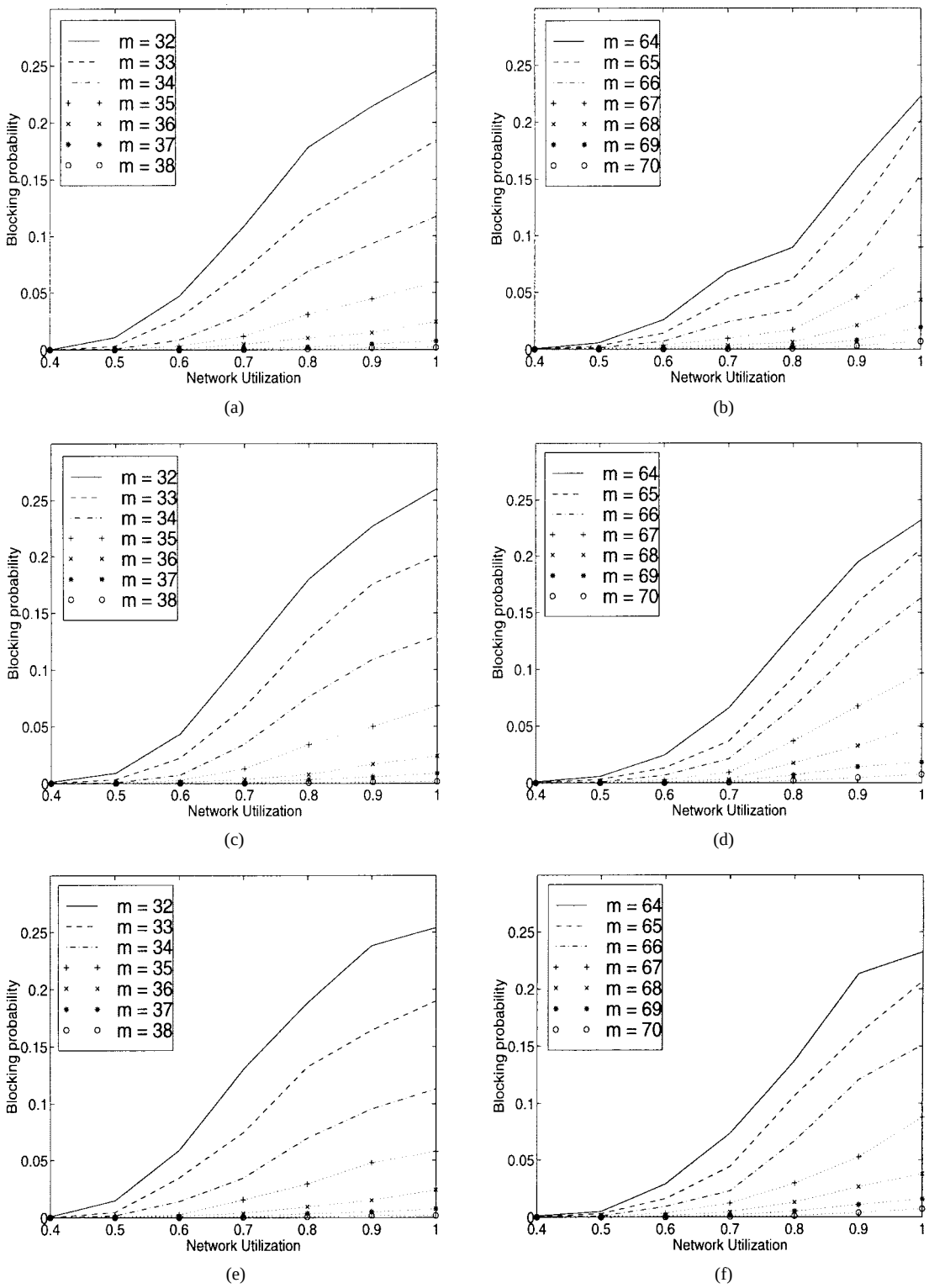

Fig. 7. The blocking probability of the $v(m, 32, 32)$ and $v(m, 64, 64)$ multicast networks under different network utilization for the smallest relative strategy. (a) $N = 1024$, uniform traffic, smallest relative. (b) $N = 4096$, uniform traffic, smallest relative. (c) $N = 1024$, uniform/constant traffic, smallest relative. (d) $N = 4096$, uniform/constant traffic, smallest relative. (e) $N = 1024$, Poisson traffic, smallest relative. (f) $N = 4096$, Poisson traffic, smallest relative.

tion request and the middle switch states, and demonstrate a moderate performance. This is also consistent with our intuition.

Moreover, we observe that in both configurations and under three types of traffic distributions, the "best" strategy (*smallest relative* strategy) can approximately provide a factor of two to three performance improvement over the "average" strategy (*random* strategy).

We have also carried out simulations for different network utilization ranging from 40% to 100%. The blocking probabilities for both Configurations 1 and 2 with different m values are shown in Fig. 7. The *smallest relative* strategy is used

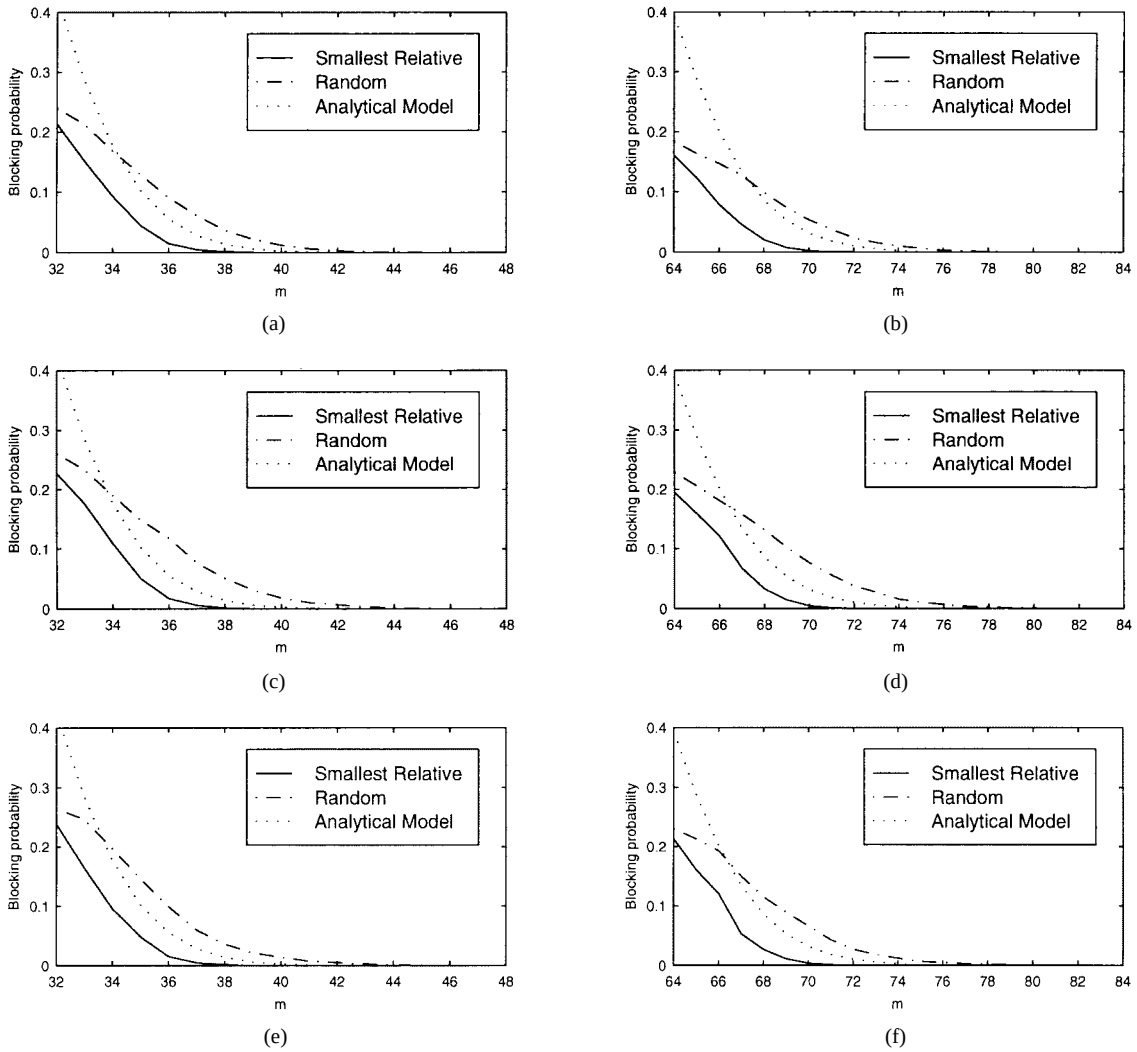

Fig. 8. The comparison between the analytical model and the simulation results for the $v(m, 32, 32)$ and $v(m, 64, 64)$ multicast networks. (a) $N = 1024$ under uniform traffic. (b) $N = 4096$ under uniform traffic. (c) $N = 1024$ under uniform/constant traffic. (d) $N = 4096$ under uniform/constant traffic. (e) $N = 1024$ under Poisson traffic. (f) $N = 4096$ under Poisson traffic.

here, and all three traffic models are examined. We can see that as network utilization increases, the blocking probabilities increase monotonically in all cases.

V. Comparison Between Analytical Model and Simulation Results

In this section we compare the analytical model with the simulation results. For simulation results, it is reasonable that we choose two typical routing control strategies: *smallest relative* and *random*.

Fig. 8 depicts the comparisons between the analytical blocking probability P_B in (11) and the simulation results under the *smallest relative* and *random* strategies for Configurations 1 and 2. From Fig. 8, we observe that in Configuration 1 the analytical blocking probability approaches zero when $m \geq 42 \approx 1.31n$ and in Configuration 2 it approaches zero when $m \geq 76 \approx 1.19n$. We can see that when m gets larger, the analytical model matches better with the simulation results under both strategies. Notice that although the analytical model and the simulation results were obtained under quite different assumptions, they reveal the same trend in the blocking behavior of the $v(m, n, r)$ multicast network: when m gets slightly larger than n, the network becomes almost nonblocking.

VI. Conclusions

We have studied the blocking behavior of the $v(m, n, r)$ multicast networks with small m values along two parallel lines: 1) we developed an analytical model for the blocking probability of the $v(m, n, r)$ multicast network and 2) we studied the blocking behavior of the network under various routing control strategies through simulations. Our observations can be summarized as follows.

- A network with a small m, such as $m = n + c$ or dn, where c and d are small constants, is almost nonblocking for multicast connections, although theoretically it requires $m \geq \Theta(n(\log r / \log\log r))$ to achieve nonblocking for multicast connections.
- Routing control strategies are effective for reducing the blocking probability of the multicast network. The best

routing control strategy can provide a factor of two to three performance improvement over random routing.

The results are encouraging and indicate that a $v(m,n,r)$ network with a comparable cost to a permutation network can provide cost-effective support for multicast communication. Our analytical and simulation results also provide a basis for further study on this type of multicast network.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees of this article for their insightful and constructive comments.

REFERENCES

[1] L. M. Ni, "Should scalable parallel computers support efficient hardware multicast?" in *Proc. 1995 ICPP Workshop Challenges for Parallel Processing*, St. Charles, IL, 1995, pp. 2–7.
[2] ——, "Issues in designing truly scalable interconnection networks," in *Proc. 1996 ICPP Workshop Challenges for Parallel Processing*, Bloomingdale, IL, 1996, pp. 74–83.
[3] C. Clos, "A study of nonblocking switching networks," *Bell Syst. Tech. J.*, vol. 32, pp. 406–424, 1953.
[4] V. E. Benes, "Heuristic remarks and mathematical problems regarding the theory of switching systems," *Bell Syst. Tech. J.*, vol. 41, pp. 1201–1247, 1962.
[5] A. Itoh *et al.*, "Practical implementation and packaging technologies for a large-scale ATM switching system," *J. Select. Areas Commun.*, vol. 9, pp. 1280–1288, 1991.
[6] J. Beetem, M. Denneau, and D. Weingarten, "The GF11 supercomputer," in *Proc. 12th Annu. Int. Symp. Computer Architecture*, 1985, pp. 108–115.
[7] M. T. Bruggencate and S. Chalasani, "Equivalence between SP2 high-performance switches and three-stage Clos networks," in *Proc. 25th Int. Conf. Parallel Processing*, Bloomingdale, IL, 1996, pp. I-1–I-8.
[8] G. B. Stunkel, D. G. Shea, B. Abali, *et al.*, "The SP2 high-performance switch," *IBM Syst. J.*, vol. 34, no. 2, pp. 185–204, 1995.
[9] G. M. Masson and B. W. Jordan, "Generalized multi-stage connection networks," *Networks*, vol. 2, pp. 191–209, 1972.
[10] F. K. Hwang and A. Jajszczyk, "On nonblocking multiconnection networks," *IEEE Trans. Commun.*, vol. COM-34, pp. 1038–1041, 1986.
[11] Y. Yang and G. M. Masson, "Nonblocking broadcast switching networks," *IEEE Trans. Comput.*, vol. 40, pp. 1005–1015, 1991.
[12] ——, "The necessary conditions for Clos-type nonblocking multicast networks," in *Proc. 10th IEEE Int. Parallel Processing Symp.*, Honolulu, HI, 1996, pp. 789–795.
[13] Y. Yang, "A class of interconnection networks for multicasting," *IEEE Trans. Comput.*, to be published.
[14] C. Y. Lee, "Analysis of switching networks," *Bell Syst. Tech. J.*, vol. 34, no. 6, pp. 1287–1315, Nov. 1955.
[15] C. Jacobaeus, "A study on congestion in link systems," *Ericsson Tech.*, vol. 51, no. 3, 1950.
[16] P. M. Lin, B. J. Leon, and C. R. Stewart, "Analysis of circuit-switched networks employing originating office control with spill forward," *IEEE Trans. Commun.*, vol. COM-26, pp. 754–765, 1978.
[17] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*. Reading, MA: Addison-Wesley, 1987.
[18] Y. Mun, Y. Tang, and V. Devarajan, "Analysis of call packing and rearrangement in a multistage switch," *IEEE Trans. Commun.*, vol. 42, pp. 252–254, Feb./Mar./Apr. 1994.
[19] Y. Yang, "An analytical model on network blocking probability," *IEEE Commun. Lett.*, vol. 1, pp. 143–145, Sept. 1997.
[20] K. P. Bogart, *Introductory Combinatorics*, 2nd ed. Orlando, FL: Harcourt Brace Jovanovich, 1990.

Yuanyuan Yang (S'91–M'91) received the B.S. and M.S. degrees in computer engineering from Tsinghua University, Beijing, China, in 1982 and 1984, respectively, and the M.S.E. and Ph.D. degrees in computer science from The Johns Hopkins University, Baltimore, MD, in 1989 and 1992, respectively.

She is currently an Associate Professor with the Department of Computer Science, University of Vermont, Burlington. Her research interests include parallel and distributed computing and systems, high-speed networks, optical networks, high-performance computer architecture, computer algorithms, and fault-tolerant computing. She has published extensively in archival journals and conference proceedings. She also holds a U.S. patent in the area of multicast communication. She has served on the program committees of several international conferences.

Dr. Yang is a member of the Association for Computing Machinery (ACM), the IEEE Computer Society, and the IEEE Communications Society.

Jianchao Wang received the B.S. degree in computer engineering from Tsinghua University, Beijing, China, in 1982, and the M.S. and Ph.D. degrees in computer science from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1985 and 1988, respectively.

He is currently a Principal Member of the Technical Staff of GTE Laboratories, Inc., Waltham, MA. Before he joined GTE, he was with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, The Johns Hopkins University, Baltimore, MD, and Legent Corporation, Marlboro, MA. His research interests include databases, programming languages, computer communication networks, computer algorithms, and fault-tolerant computing.

Dr. Wang is a member of IEEE Computer Society and the Association for Computing Machinery (ACM).