Nonblocking Broadcast Switching Networks

Yuan yuan Yang, Student Member, IEEE, and Gerald M. Masson, Fellow, IEEE

Abstract—In a broadcast connection through a multistage network, an input port can be connected to more than one output port, with the restriction that at no time can an output port be connected to more than one input port. There are broad ranges of applications for multistage networks which can realize broadcast connection requests in a nonblocking manner in the sense that a request from an idle input port to be connected to some set of idle output ports can always be satisfied without any disturbance (that is, rearrangement) of other broadcast connections already existing in the network. We present results that lead to the currently best-known explicit constructions of nonblocking broadcast switching networks with limited numbers of stages relative to both crosspoint and control algorithm complexity. In three-stage versions of our designs denoted as $v(m,n,r)$ networks, wherein there are $r$ switch modules in each of the first and third stages, $n$ input ports on each switch module in the first stage, and $n$ output ports on each switch module in the third stage, we prove that if the number of switch modules in the middle stage, $m$, satisfies $m > \min(n - 1, x + r^{3/2})$, where $1 \leq x \leq \min(n - 1, r)$, the resulting network is nonblocking for broadcast assignments. This condition on the minimum number of switch modules in the middle stage represents an improvement from $O(nr)$ to $O(n \log r / \log \log r)$ relative to previously known results. We also present a linear algorithm for satisfying new broadcast connection requests in the network. Finally, we compare our nonblocking broadcast networks to other nonblocking and rearrangeable broadcast networks that have been presented in the literature.

Index Terms—Broadcast connection, interconnection network, multistage, nonblocking, switching network.

I. INTRODUCTION

In an increasing number of computing and communication environments, it is necessary to simultaneously transfer text/voice/video/graphics information from a set of transmitting devices to a set receiving devices in various combinations. Similarly, all parallel processing systems depend critically on the transfer of data among processing and memory units. This can be accomplished using an interconnection network called a multistage switching network. When a transmitting device simultaneously sends information to more than one receiving device, the one-to-many connection required between the transmitting device and the receiving devices is called a broadcast connection. A set of broadcast connections is referred to as a broadcast assignment. Multistage switching networks that can satisfy such broadcast requirements are called broadcast networks.

In this paper, we will be concerned with the analysis and constructive design of broadcast networks to provide so-called nonblocking connecting capability in multistage switching networks. In our nonblocking broadcast networks, any broadcast connection request from an idle network input port to a set of idle network output ports can be realized without any disturbance (that is, rearrangement) of other existing broadcast connections. Additionally, a network output port that is connected to a network input port in some broadcast connection, can upon disconnection from that network input port be included in future broadcast connection requests made by network input ports. We also present a linear algorithm for satisfying new broadcast connection requests in the network. Our nonblocking broadcast network designs will be seen to be an improvement in terms of required switching elements and network control complexity over other previously known designs, even including some rearrangeable broadcast network designs.

II. MULTISTAGE SWITCHING NETWORKS

Multistage switching networks are composed of crosspoint switching elements or, more simply, crosspoints that are usually grouped together into building-block subnetworks called switch modules. In an $(N_1 \times N_2)$ multistage switching network with $N_1$ input ports and $N_2$ output ports, the switching modules used as building blocks to implement the network might each have, for example, $n_1$ inputs and $n_2$ outputs, where $n_1 < N_1$ and $n_2 < N_2$. These would be referred to as $(n_1 \times n_2)$ switch modules. An $(n_1 \times n_2)$ switch module is assumed to be composed of $n_1 n_2$ crosspoints. Fig. 1(a) shows an $(3 \times 4)$ switch module.

The connectivity available among the $n_1$ inputs and the $n_2$ outputs of the $(n_1 \times n_2)$ switch modules depends upon implementation details, but a case of general interest which will be considered exclusively in the following will be that in which the switch module has sufficient crosspoint switching elements to provide broadcast capability from the $n_1$ inputs to the $n_2$ outputs in the sense that any input of the switch module can always be connected to any idle output (that is, an output from the switch module that is not currently connected to any input of the switch module). The input to output connections implemented in a switch module characterize the state of the switch module. Fig. 1(b) shows a broadcast state of a $(3 \times 4)$ switch module.

The $(n_1 \times n_2)$ switch modules in a multistage switching network are interconnected by means of links. The links are dedicated, fixed connection paths between the inputs and outputs of different switch modules. The switch modules in a multistage switching network are grouped together into stages.
multistage switching network implies less complexity in the control algorithm for establishing connection paths between input and output ports; this follows because there will in general be many available paths from which to choose for making a requested connection. Similarly, with a small number of switching modules in the middle stage(s), establishing a connecting path would be expected to be more difficult and the network control algorithm correspondingly more complex. Indeed, this intuition is correct: there is an inherent tradeoff between network control algorithm complexity and the number of switching modules used in a design of a multistage switching network.

An awareness of this tradeoff has led to the development of network structures in which providing for new connecting paths between ports can sometimes require the rearrangement of existing connection paths to alleviate a blocking condition so that a requested connection can be made. In rearrangeable networks, an attempt is made to reduce the switch hardware costs of the structure at the expense of the complexity and time required to set up connection paths. Multistage switching networks which handle blocking conditions by rearrangement must be cautiously utilized. This is because of the possible disruption of on-going communication caused by the rearrangements as well as the resulting time delay in setting up the requested path. There are many applications in which the difficulties associated with rearrangements cannot be tolerated.

In general, it is desirable to minimize or, indeed, even eliminate the need for rearrangements to existing connections in order to satisfy a new connection request. Switching networks in which there is no need for rearrangements are usually referred to as having nonblocking connecting capability. When considering nonblocking connecting capability in broadcast networks, there is an aspect of the nature of requests for broadcast connections that must be specified: namely, it must be determined whether broadcast connections originate in the form of idle input ports requesting connections to some set of idle output ports or whether they originate in the form of an idle output port requesting a connection to some (busy or idle) input port. In the following, it should be understood that we will consider broadcast connection requests that emanate from idle input ports. This type of nonblocking connecting capability has been considered previously [10], [19] and has important applications in parallel processing for processor-memory interconnection as well as in a variety of general communication environments.

Nonblocking connecting capability can still be further categorized. In a strictly nonblocking broadcast network, for any legitimate broadcast connection request from an input port to some set of output ports, it is always possible to provide a connection path through the network to satisfy the request without disturbing other existing broadcast connections, and if more than one such path is available, any of them can be selected without being concerned about satisfaction of future potential broadcast connection requests. In wide-sense nonblocking broadcast networks, it is again always possible to provide a connection path through the network to satisfy the request without disturbing other existing broadcast connections, but in this case the path used to satisfy the connection request must be

such that the inputs of the switch modules of one stage are linked only to outputs of switch modules of another stage, and, similarly, the outputs of the switch modules of one stage are linked only to inputs of switch modules of another stage. The inputs of the switch modules of the first stage of a multistage switching network are referred to as the network input ports. This first stage is often referred to as the input stage of the network. The outputs of the last stage of a multistage switching network are referred to as the network output ports. This last stage is often referred to as the output stage of the network. In a three-stage network, the second stage is referred to as the middle stage. The switch modules in the input stage are sometimes referred to as input switch modules or input switches; the switch modules in the output stage are sometimes referred to as output switch modules or output switches; and the switch modules in the middle stage are sometimes referred to as middle switch modules or middle switches. Fig. 2 shows a three-stage switching network with $N_1 = N_2 = 9$ input/output ports and comprised of three $(3 \times 4)$ input switches, four $(3 \times 3)$ middle switches, and three $(4 \times 3)$ output switches. The set of switch module states in a network characterizes the state of the network. The network of Fig. 2 is shown in a state realizing a broadcast assignment.

In general, an $(N_1 \times N_2)$ three-stage network has $r_1 \times r_2$ switch modules in stage 1, $m \times m$ switch modules in stage 2, and $N_2 \times m$ in stage 3. Such a multistage switching network is denoted as a $v(m, n_1, r_1, n_2, r_2)$ network. A general schematic of an $(N_1 \times N_2)$ three-stage network is shown in Fig. 3. For the special symmetrical case where $n_1 = n_2 = n$ and $r_1 = r_2 = r$, the three-stage network is denoted as a $v(m, n, r)$ network. The three-stage network of Fig. 2 is a $v(4, 3, 3)$ network. In general, the set of input ports is denoted as $\{1, 2, \cdots, r_1 n_1\}$ and the set of switch modules in the output stage is denoted as $\emptyset = \{1, 2, \cdots, r_2\}$.

The results on nonblocking multistage broadcast switching networks that follow will initially be described primarily in terms of three-stage networks. It should be understood, however, that our results are applicable to $k$-stage networks for $k > 3$ by recursively using the design criteria developed on the switch modules. Indeed, in a later section we will make explicit comparisons of our network to other networks with both limited and unlimited number of stages. The presentation of the results in terms of three-stage networks is only for notational convenience.

A. Network Control

The network controller of a multistage switching network executes a network control algorithm for establishing connection paths between input and output ports. Intuitively, a large number of switch modules in the middle stage(s) of a
Fig. 2. A $(9 \times 9)$ three-stage switching network denoted as a $v(m, n, r)$ network realizing a broadcast assignment characterized as $I_1 = \{1, 2, 3\}, I_2 = \{1\}, I_3 = \{2\}, I_4 = \{3\}$, and all other $I_j = \phi$. The destination sets for the middle switches for the shown network state are $M_1 = \{1, 2\}, M_2 = \{2, 3\}, M_3 = \{3\}$, and $M_4 = \{1\}$.

Fig. 3. General schematic of an $(N_1 \times N_2)$ three-stage switching network denoted as a $v(m, n_1, r_1, n_2, r_2)$ network.

carefully selected so as to maintain the nonblocking connecting capability for future potential broadcast connection requests. These categorizations are essentially due to Benes [1].

Without in any way attempting to criticize the term wide-sense nonblocking, as it was chosen at a time when algorithm analysis was not well-developed, the term is nevertheless somewhat of a misnomer as it does not convey the notion of nonblocking connecting capability that it is meant to imply. This particular type of nonblocking capability might perhaps be better referred to as control-algorithm-dependent nonblocking capability, since the control algorithm used to satisfy requests figures critically in the nonblocking connection capability provided. In fact, once a network is built, the time complexity of the control algorithm will repeatedly affect every connection request and therefore the overall performance of the network. Accordingly, control-algorithm-dependent nonblocking capability must be evaluated from the perspective of the complexity of the control algorithm as opposed to only a proof of its existence. It is this type of nonblocking capability we address in this paper, and we elect to refer to it as simply nonblocking connection capability. That is, the nonblocking connection capability considered in this paper is control-algorithm-dependent nonblocking with broadcast connection requests emanating from idle input ports. We will show constructive multistage designs for such nonblocking broadcast networks for which a linear network control algorithm is given that permits a broadcast connection request from an idle input port to some set of idle output ports to be satisfied without any rearrangement of connection paths of other existing broadcast connections. Furthermore, if any output port connected to an input port in a broadcast connection is disconnected, this output port can be included in future broadcast connection requests made by input ports.

III. PREVIOUS RESULTS

Certainly, the most well-known nonblocking multistage switching network design is due to Clos [2]. The so-called Clos network provides strictly nonblocking connecting capability for permutation assignments between the input ports and the output ports in each input port can be connected to at most one output port in a one-to-one fashion. For a $v(m, n, r)$ network, Clos showed that if $m \geq 2n - 1$, the network was strictly nonblocking; and for a $v(m, n_1, r_1, n_2, r_2)$ network, if $m \geq n_1 + n_2 - 1$, the network was again strictly nonblocking. The Clos network structure demonstrated that strictly nonblocking multistage switching networks could be designed at a considerable savings in switching costs as compared to the obvious alternative of using a large single-stage switch with a crosspoint switching element between each input port and output port. Cantor [3] later improved upon the Clos design with an alternative multistage structure that could provide strictly nonblocking connecting capability for permutation assignments with asymptotically fewer crosspoints.
It is not surprising that nonblocking networks designed for permutation assignments are not in general nonblocking for broadcast assignments, and, indeed, do not even necessarily satisfy all broadcast assignments [4]. Masson [5, 6] first gave designs for strictly nonblocking and rearrangeable multistage switching networks for broadcast assignments. For the case of a three-stage \( v(m, n, r) \) network, it was shown for broadcast assignments that if \( m \geq n(r + 1) - 1 \), the network was strictly nonblocking, and if \( m \geq nr \), the network was rearrangeable. Hwang [7] later pointed out that if, for some reason, the middle stage switch modules did not have broadcast capability so that all connection path fan-out must take place in the input and output stage switch modules, then Masson’s condition on the number of middle stage switch modules for rearrangeable connection capability was necessary and sufficient.

Hwang and Jajszczyk [8–10] have given a set of design conditions for nonblocking multistage multiconnection switching networks. A multiconnection is a generalization of a broadcast connection in the sense that input sets are connected to output sets. The nonblocking broadcast capability discussed in this paper is analogous to that considered by Hwang and Jajszczyk with the restriction that our input sets are singletons. Similar to that of Masson’s strictly nonblocking and rearrangeable broadcast networks, it was shown that relative to a \( v(m, n, r) \) networks, their nonblocking multiconnection networks require \( m = O(nr) \) in the case where the input sets are singletons.

Masson [11, 12] has also shown a two-stage design of a rearrangeable broadcast switching network which cascades sparse crossbar switching structures that function as concentrators [13] with broadcast switching modules. Later, Kufa and Varcoux [14] and then Richards and Hwang [15–17] used Masson’s two-stage concept as the basis of reconfigured and extended—but nevertheless fundamentally similar—rearrangeable broadcast network designs.

Other techniques of cascading networks of various types to achieve broadcast capability have been considered. Lea [18] has studied cascading a spreading (or fan-out) multistage network with a permutation network for the design of rearrangeable broadcast networks (in a manner similar to that suggested by Ofman [25] and later considered by Thompson [26]). Turner [19] has considered the cascading of Cantor and Clos networks to achieve nonblocking broadcast connection capability.

Dolev, Dwork, Pippenger, and Wigderson [20] have given minimum possible upper bounds on the number of crosspoints required for \( k \)-stage rearrangeable broadcast networks. Subsequently, Feldman, Friedman, and Pippenger [21] showed improved upper bounds for \( k \)-stage nonblocking broadcast networks. But neither explicit constructions nor efficient control algorithms for networks satisfying these bounds are known. However, Dolev, Dwork, Pippenger, and Wigderson [20] did offer a construction for a \((3k - 2)\)-stage rearrangeable broadcast network (where \( k \geq 1 \)) and Feldman, Friedman, and Pippenger [21] gave constructions for two-stage and three-stage nonblocking broadcast networks. Finally, by means of an application of a hypergraph-hypercoloring theorem, Kirpatrick, Klawe, and Pippenger [22] gave constructive designs for multistage rearrangeable broadcast networks. We will later compare our nonblocking broadcast networks to all of these previously known designs and results.

IV. PRELIMINARIES

In this section, we present some basic definitions, notations, and observations that will be useful to us in our analysis of nonblocking broadcast multistage switching networks.

A. Fan-Out

Every switch module in our networks will be assumed to have broadcast capability. In a \( v(m, n_1, r_1, n_2, r_2) \) network, if a network input port is to be connected to more than one output port on the same switch module in the output stage of the network, then it is only necessary for the input port to have one connection path to that switch module. This follows because that path can be broadcast within the switch module in the output stage to as many output ports as necessary. Broadcast assignments can therefore be described in terms of connections between input ports and switch modules in the output stage. An existing connection or a requested connection from an input port to output ports on \( r' \) output stage switches is said to have fan-out \( r' \). It is clear that if all broadcast assignments wherein each input is to be connected to at most one output port on any switch module in the output stage are realizable, then broadcast assignments wherein any input port is to be connected to more than one output port on the same output stage switch module can also be realized. Hence, we are primarily interested in general broadcast connections with fan-out \( r', 1 \leq r' \leq r_2 \). However, we will also at times be interested in restricted broadcast connections with upper bounds on their fan-out, say, \( d \) where \( 1 \leq d \leq r_2 \).

B. Connection Characterizations

To characterize a broadcast assignment, for each input port \( i \in \{1, \ldots, r_1\} \), let \( I_i \subseteq O = \{1, \ldots, r_2\} \) denote the subset of the switch modules in the output stage to which \( i \) is to be connected in the broadcast assignment. For example, for the broadcast assignment shown in Fig. 2, \( I_1 = \{1, 2, 3\}, I_2 = \{1\}, I_3 = \{2\}, I_6 = \emptyset \), and all other \( I_j = \emptyset \).

C. Middle Stage Characterizations

To characterize the state of the \( m \) switch modules in the middle stage of a three-stage switching network, let \( M_j \subseteq O = \{1, \ldots, r_2\}, j = 1, 2, \ldots, m \) denote the subset of the switch modules in the output stage to which the \( M_j \) is providing connection paths from the input ports. In other words, each \( M_j \) denotes the connection paths passing through middle switch \( j \) in terms of their output switch destinations. We will refer to the sets \( M_j \subseteq O = \{1, \ldots, r_2\}, j = 1, 2, \ldots, m \), as the destination sets of the middle switches. For example, for the broadcast assignment shown in Fig. 2, the destination sets of the four middle switches are \( M_1 = \{1, 2\}, M_2 = \{2, 3\}, M_3 = \{3\}, \) and \( M_4 = \{1\} \). It should be clear that in general for any state of a three-stage switching
network
\[ \sum_{i=1}^{n} |I_i| = \sum_{j=1}^{m} |M_j| \leq n_2 r_2. \]

In the example of Fig. 2, \( \sum_{i=1}^{3} |I_i| = 3 + 1 + 1 + 1 = 6 \), and
\( \sum_{j=1}^{3} |M_j| = 2 + 2 + 1 + 1 = 6. \)

V. NONBLOCKING CONNECTING CAPABILITY

In this section, we will present the main results of the paper. We will first show that a \( v(m, n, r) \) network is nonblocking for general broadcast assignments when \( m \geq O(n \log r / \log \log r) \). We will then extend the results to the more general \( v(m, n_1, n_2, r_1, r_2) \) networks and restricted broadcast assignments.

A. Sufficient Conditions on \( m \) for \( v(m, n, r) \) Networks

Assume that a \( v(m, n, r) \) network is currently providing some set of broadcast connections from the input ports to the output ports. Suppose there is a new request for a connection from an idle input port \( i (i \in \{1, 2, \cdots, nr\}) \) to some set of \( r' \) idle output ports (each assumed to be on a distinct output switch). We denote this new connection request as \( I_i \subseteq O_i \), where \( r'=|I_i|, 1 \leq r' \leq r \), is the fan-out of the request. We will refer to the set of middle switches with currently unused links to the input switch associated with input \( i \) as the available middle switches for this connection request.

To satisfy this request \( I_i \), we scan the available middle switches. If we find an empty middle switch (that is, a middle switch, say, \( i_1, i_2 \in \{1, 2, \cdots, m\} \), where \( M_{i_1} = \phi \), we can surely satisfy \( I_i \) through this middle switch. Similarly, if our scan determines that there are two middle switches, say, \( i_1 \) and \( i_2 \) for which
\[ M_{i_1} \cup M_{i_2} = \{1, 2, \cdots, r\} \]
where \( M_i = \{1, 2, \cdots, r\} - M_i \) we can satisfy \( I_i \) through these two middle switches. In general, if there exist \( x \) \((1 \leq x \leq m)\) middle switches. Then, if there exist \( x \) \((1 \leq x \leq m)\) middle switches with
\[ \bigcup_{j=1}^{x} M_{i_j} = \{1, 2, \cdots, r\}, \]
we can satisfy \( I_i \) through these \( x \) middle switches.

The above naturally leads us to inquire as to the smallest number of available middle switches among which we can always find less than or equal to \( x \) middle switches satisfying (1) regardless of the state of the network. Obviously, this number will be a function of \( x \). To answer this question, we will use the following lemmas, which pertain to the states of middle switches in \( v(m, n, r) \) networks when realizing broadcast assignments.

Lemma 1: We can satisfy a new connection request with fan-out \( r \) using some \( x \) \((x \geq 1)\) middle switches, say, \( i_1, i_2, \cdots, i_x \), from among the available middle switches if and only if the current destination sets of these \( x \) middle switches are such that
\[ \bigcup_{j=1}^{x} M_{i_j} = \phi. \]

Proof: If there exist \( x \) available middle switches say, \( i_1, \cdots, i_x \), for which \( \bigcap_{j=1}^{x} M_{i_j} = \phi \) (that is, there are no common output switch(es) to which each of these \( x \) middle switches is providing a connection path), then for every output switch \( i, 1 \leq i \leq r \), we can always find a middle switch, say \( j_1, 1 \leq j \leq x, i \notin M_{i_j} \), through which a connection path to \( i \) is available. Thus, we can satisfy the new connection request through these \( x \) middle switches. Similarly, if we can satisfy a connection request with fan-out \( r \) using \( x \) switches, say, \( i_1, \cdots, i_x \), then \( \bigcap_{j=1}^{x} M_{i_j} = \phi \) before we satisfy this connection request. Otherwise, if \( k \in \bigcap_{j=1}^{x} M_{i_j}, 1 \leq k \leq r \), then a connection path could not be provided to output switch \( k \) through any middle switch in the set of \( x \) available middle switches.

Lemma 2: Assume that a \( v(m, n, r) \) network is in a state in which there exist at most \( n' \) connection paths, \( 1 \leq n' \leq n \), to each of the output switches. Then the intersection of more than \( n' \) \( M_i \)'s is empty.

Proof: It is clear that there are at most \( n' \) \( 1 \)'s, \( n' \) \( 2 \)'s, \( \cdots, n' \) \( r \)'s distributed among the destination sets of the middle switches. Suppose the intersection of some \( n'' > n' \) destination sets were not empty. Then there exists some output switch \( j \) to which \( n'' > n' \) connection paths were being provided through the middle switches (i.e., there existed \( n'' \) \( j \)'s among the destination sets). This contradicts our assumption.

For the following lemma, we will assume that a \( v(m, n, r) \) is in an arbitrary state in which there exist at most \( n' \) connection paths, \( 1 \leq n' \leq n \), to each of the output switches. Fig. 2 is an example of such a state for a \( v(4, 3, 3) \) network where \( n' = 2 \).

Lemma 3: For all \( n', 1 \leq n' \leq n \), and for all \( x, 1 \leq x \leq \min(n', r) \), let \( m' \) be the maximum number of middle switches whose destination sets have the following properties:

1) there are at most \( n' \) \( 1 \)'s \( n' \) \( 2 \)'s, \( \cdots, n' \) \( r \)'s distributed among the destination sets;
2) the intersection of any \( x \) of the destination sets is not empty.

Then
\[ m' \leq n' r^{1/x}. \]

Proof: Clearly, if \( x > n' \), by Lemma 2, the intersection of the destination sets of any \( x \) middle switches is already empty. Therefore, we only need to consider \( x \leq \min(n', r) \).

Without loss of generality, suppose these \( m' \) middle switches are \( 1, 2, \cdots, m' \) with destination sets \( M_1, M_2, \cdots, M_{m'} \). Obviously, these \( m' \) destination sets all are nonempty. Let \( c_1(i) = |M_i|, 1 \leq i \leq m' \). Then, clearly,
\[ \sum_{i=1}^{m'} c_1(i) \leq n' r. \]

Let
\[ c_1 = \min_{i} c_1(i). \]

Then, we obtain that
\[ m' c_1 \leq \sum_{i=1}^{m'} c_1(i) \leq n' r. \]
Note that \( c_1 \neq 0 \). Therefore,

\[
m' \leq \frac{n'r}{c_1}
\]  

(2)

Without loss of generality, suppose that the destination set of middle switch 1 has cardinality \( c_1 \), and \( M_1 = \{1, 2, \ldots, c_1\} \). Intersect each of the destination sets \( M_1, M_2, \ldots, M_{m'} \) with \( M_1 \) and obtain \( m' \) sets \( M_1', M_2', \ldots, M_{m'}' \). (Obviously, \( M_1 = M_1' \).) Note that these resulting \( M_i' \)'s consist of only elements in \( \{1, 2, \ldots, c_1\} \), and distributed among the \( M_i' \) are at most \( n' \) 1's, \( n' \) 2's, \ldots, and \( n' \) \( c_1 \)'s.

Now considering the following sequence of operations:

For the intersected destination sets \( M_1', M_2', \ldots, M_{m'}' \), let \( c_2(i) = |M_i'|, 1 \leq i \leq m' \). We have that

\[
\sum_{i=1}^{m'} c_2(i) \leq n' c_1.
\]

Let

\[
c_2 = \min_i \{c_2(i)\}.
\]

Without loss of generality, suppose that \( M_2' \) has cardinality \( c_2 \), and \( M_2' = \{1, 2, \ldots, c_2\} \). We then get that

\[
m' c_2 \leq \sum_{i=1}^{m'} c_2(i) \leq n' c_1.
\]

Note that \( c_2 \neq 0 \) since by hypothesis \( M_1 \cap M_2 \neq \emptyset \). We can thus conclude that

\[
m' \leq \frac{n' c_1}{c_2}.
\]

(3)

Finally, intersect each of \( M_1', M_2', \ldots, M_{m'}' \) with \( M_2 \) and obtain \( m' \) sets \( M_1'', M_2'', \ldots, M_{m''}'' \), which consist of only elements in \( \{1, 2, \ldots, c_2\} \).

We want to perform the above sequence of operations \( x - 1 \) times. For the \( k \)th performance, \( 1 \leq k < x \), we will start with a set of \( m' \) intersected destination sets \( M_1^{k-1}, M_2^{k-1}, \ldots, M_{m'}^{k-1} \). (For consistency, define \( M_0^{k} = M_i \) for \( i = 1, 2, \ldots, m' \).) We will let \( c_k(i) = |M_i^{k-1}|, 1 \leq i \leq m' \) and have that

\[
\sum_{i=1}^{m'} c_k(i) \leq n' c_{k-1}.
\]

Then we will let

\[
c_k = \min_i \{c_k(i)\}.
\]

We will, without loss of generality, suppose that \( M_2^{k-1} \) has cardinality \( c_k \), and \( M_2^{k-1} = \{1, 2, \ldots, c_k\} \). We then get that

\[
m' c_k \leq \sum_{i=1}^{m'} c_k(i) \leq n' c_{k-1}.
\]

Note that \( c_k \neq 0 \) since by hypothesis the intersection of any \( k \) destination sets, \( 1 < k < x \), is not empty. We then conclude that

\[
m' \leq \frac{n' c_{k-1}}{c_k}.
\]

(4)

Finally, we get a set of sets \( M_1^{k}, M_2^{k}, \ldots, M_{m'}^{k} \), which consist of only elements in \( \{1, 2, \ldots, c_k\} \). After performing the above sequence of operations \( x - 1 \) times, we have

\[
m' \leq \min \left\{ \frac{n' r}{c_1}, \frac{n' c_1}{c_2}, \ldots, \frac{n' c_{x-2}}{c_{x-1}} \right\}.
\]

(5)

We now have a set of \( m' \) intersected destination sets \( M_1^{x-1}, M_2^{x-1}, \ldots, M_{m'}^{x-1} \). By hypothesis, each of the \( M_i^{x-1} \)'s is nonempty. Moreover, each \( M_i^{x-1} \) consists of only elements in \( \{1, 2, \ldots, c_{x-1}\} \), and there are at most \( n' \) 1's, \( n' \) 2's, \ldots, and \( n' \) \( c_{x-1} \)'s distributed among these \( m' \) sets. Thus, we have that

\[
m' \leq n' c_{x-1}.
\]

(6)

Therefore, from (5) and (6), we get that

\[
m' \leq \min \left\{ \frac{n' r}{c_1}, \frac{n' r}{c_2}, \ldots, \frac{n' r}{c_{x-1}} \right\}.
\]

(7)

We are interested in the upper bound of the right-hand side of (7). Since the geometric mean of \( \left\{ \frac{n' r}{c_1}, \frac{n' r}{c_2}, \ldots, \frac{n' r}{c_{x-1}} \right\} \) is

\[
\left( \frac{n' r}{c_1} \cdot \frac{n' r}{c_2} \cdot \ldots \cdot \frac{n' r}{c_{x-1}} \right)^{1/x} = n' r^{1/x}
\]

we have that

\[
n' r^{1/x} \geq \min \left\{ \frac{n' r}{c_1}, \frac{n' r}{c_2}, \ldots, \frac{n' r}{c_{x-1}} \right\}.
\]

(8)

Therefore,

\[
m' \leq n' r^{1/x}.
\]

We are now in a position to prove a fundamental result on the number of middle switches sufficient to realize a general broadcast assignment with nonblocking connecting capability.

We begin with the following.

**Theorem 1:** In a \( v(m, n, r) \) network, for a new connection request with fan-out \( r' \), \( 1 \leq r' \leq r \), if there exist more than \((n - 1)r'^{1/x}, 1 \leq x \leq \min(n - 1, r')\), available middle switches for this connection request, then there will also always exist \( x \) middle switches through which this new connection request can be satisfied.

**Proof:** Without loss of generality, suppose that the new connection request is \( I_i = \{1, 2, \ldots, r'\} \). Clearly, to satisfy this new request, we need not be concerned with the connection paths through the middle switches to the output switches \( O - I_i \). There will be at most \( n - 1 \) 1's, \( n - 1 \) 2's, \ldots, and \( n - 1 \) \( r' \)'s distributed among the destination sets of the middle switches. If we replace \( r \) with \( r' \) in Lemmas 1, 2, and 3, and interpret the terms "empty" and "common element" with respect to only the elements in the set \( \{1, 2, \ldots, r'\} \), these lemmas still hold. Thus, from Lemma 3, we know that if the number of available middle switches is greater than \((n - 1)r'^{1/x}, 1 \leq x \leq \min(n - 1, r')\), then there must exist \( x \) destination sets without
a common element in \{1, 2, \ldots, r\}. Hence, by Lemma 1, the new connection request can be satisfied.

**Theorem 2**: A \(v(m, n, r)\) network is nonblocking for general broadcast assignments if

\[
m > \min_{1 \leq x \leq \min\{n, n-1, r\}} \left\{ (n-1) \left( x + \frac{1}{r} \right) \right\}.
\]  

(9)

**Proof**: Suppose that in a \(v(m, n, r)\) network an input port on an input switch makes a connection request with fan-out \(r\). Clearly, the middle switches providing connection paths for the other \(n-1\) input ports on this input switch are not available for satisfying this connection request. Given \(x, 1 \leq x \leq \min\{n, n-1, r\}\), if we have that

\[
m > (n-1)x + (n-1)x^{1/r} = (n-1)(x + x^{1/r})
\]  

(10)

then from Theorem 1 those existing connection paths from the \(n-1\) other input ports can pass through at most \((n-1)x\) middle switches. This means that there are more than \((n-1)x^{1/r}\) middle switches available for the connection request. Again from Theorem 1, we can thus satisfy the connection request. Taking some \(x\) which minimizes \((x + x^{1/r})\), we can obtain the nonblocking condition (9).

For a given \(n\) and \(r\) in a \(v(m, n, r)\) network, we could use Theorem 2 to find \(x\) such that a minimum \(m\) can be determined for nonblocking connection capability for broadcast assignments. But it is also of interest to determine bounds on \(m\) as a function of \(n\) and \(r\). The following theorem addresses this issue.

**Theorem 3**: A \(v(m, n, r)\) network is nonblocking for general broadcast assignments if

1. \(m > (n-1)(\log r + 2)\)
2. \(m \geq O\left(\frac{n \log r}{\log \log r}\right)\)

**Proof**: The first bound follows from letting \(x = \log r\) in (10). We get

\[
x^{1/r} = \frac{1}{\log \log r} = 2.
\]

Therefore, \(m > (n-1)(\log r + 2)\) is sufficient for nonblocking capability.

To get the second, tighter bound for \(m\), for any given constant \(\delta, 0 < \delta < 1\), let \(x = \frac{1}{\delta} \frac{\log r}{\log \log r}\). Then

\[
x^{1/r} = \left( \frac{\log r}{\log \log r} \right)^{\frac{1}{\delta}} \approx (\log r)^{\delta}.
\]

Therefore,

\[
x + x^{1/r} = \frac{1}{\delta} \frac{\log r}{\log \log r} + (\log r)^{\delta} = O\left(\frac{\log r}{\log \log r}\right).
\]

Thus,

\[
m \geq O\left(\frac{n \log r}{\log \log r}\right).
\]

**B. Some Extensions**

The above results for \(v(m, n, r)\) networks can be generalized in two ways: first, the more general asymmetrical \(v(m, n_1, r_1, n_2, r_2)\) networks can be considered; second, restricted fan-out connection assignments can be considered in which each input port can have connection paths to at most \(d\) outputs, \(1 \leq d \leq r_2\), output switches [5]. We will state corollaries to the above theorems that address such generalizations with brief proofs. Detailed proofs to these corollaries would be similar to that of the previous theorems and lemmas.

**Corollary 1**: A \(v(m, n_1, r_1, n_2, r_2)\) network is nonblocking for general broadcast assignments if

\[
m > \min_{1 \leq x \leq \min\{n_2-1, r_2\}} \left\{ (n-1)x + (n_2-1)x^{1/r_2} \right\}.
\]  

(11)

In particular, this condition can be written as

\[
m > (n_1 - 1) \log r_2 + 2(n_2 - 1)
\]  

(12)

or, more precisely,

\[
m > 2(n_1 - 1) \log \frac{r_2}{\log \log r_2} + (n_2 - 1)(\log r_2)^{1/2}.
\]  

(13)

**Proof**: Note that for any state of a \(v(m, n_1, r_1, n_2, r_2)\) network, there will be at most \(n_2\) 1's, \(n_2\) 2's, \(\ldots\), \(n_2\) 2's distributed among the destination sets \(M_1, M_2, \ldots, M_{n_2}\). In our statements and proofs of Theorem 1 and Lemmas 1, 2, and 3, we can replace \(n\) and \(r\) with \(n_2\) and \(r_2\), respectively. Also, in Theorem 2, we can have \(n_1 - 1\) input ports on an input switch with connection paths to at most \((n_1-1)x\) middle switches while we consider satisfying the connection request from the remaining input port. Thus,

\[
m > (n_1 - 1)x + (n_2 - 1)x^{1/r_2}
\]

is sufficient for nonblocking connection capability for \(v(m, n_1, r_1, n_2, r_2)\) networks realizing broadcast assignments. Letting \(x = \log r_2\) and \(x = \frac{\log r_2}{\log \log r_2}\) in the above inequality results in (12) and (13), respectively.

**Corollary 2**: A \(v(m, n_1, r_1, n_2, r_2)\) network is nonblocking for restricted broadcast assignments, in which each input port can be connected to at most \(d\) \((1 \leq d \leq r_2)\) output switches, if

\[
m > \min_{1 \leq x \leq \min\{n_2-1, d\}} \left\{ (n-1)x + (n_2-1)x^{1/d} \right\}.
\]

In particular, this condition can be written as

\[
m > (n_1 - 1) \log d + 2(n_2 - 1)
\]

or, more precisely,

\[
m > 2(n_1 - 1) \log \frac{d}{\log \log d} + (n_2 - 1)(\log d)^{1/2}.
\]
Proof: Note that now the input port fan-out $r'$ used in Theorem 1 is between 1 and $d$. Therefore, we can replace $r$ with $d$ in the proof of Theorem 2 to yield the desired result.

It is interesting to note that for the restricted fan-out case, when $d = 1$ we have the special case of permutation connection capability. The above result then takes on the form of the sufficient condition for strictly nonblocking connection capability of Clos [2].

Corollary 3: Setting $d = 1$ in Corollary 2 yields $m \geq n_1 + n_2 - 1$, which is the bound on $m$ associated with the classical Clos strictly nonblocking permutation networks [2].

Proof: For this case, $x = 1$ and the bound on $m$ becomes $m \geq (n_1 - 1) \times 1 + (n_2 - 1) \times 1^{1/1} = n_1 + n_2 - 2$, or $m \geq n_1 + n_2 - 1$. This is the Clos result for strictly nonblocking connection capability for permutation networks.

VI. A NETWORK CONTROL ALGORITHM

The theorems in the previous section provide the basis for a network control strategy. Assuming the network satisfies the conditions of Theorem 3, this control strategy can be briefly described as follows: to satisfy each connection request, we use at most $x$ middle switches whose destination set intersections are empty from among the guaranteed more than $(n - 1)^{r'/2}$ available middle switches. Note that based on this control strategy, each of the connection paths from each input port on the input switch associated with the connection request passes through no more than $x$ middle switches, and any middle switch providing a connection path from an input port on a particular input switch is not available to the other input ports of that particular input switch. In this section, we present efficient algorithms to implement this control strategy. We present an $O(N)$ algorithm for satisfying a general broadcast connection request and an $O(nN)$ algorithm for realizing an entire broadcast assignment in our nonblocking $v(m,n,r)$ networks. Extensions of these algorithms to more general $v(m,n,r_1,r_2,r_3)$ networks and restricted fan-out broadcast assignments are not difficult. Finally, these network control algorithms can be easily implemented in either software or hardware.

A. An $O(N)$ Algorithm for Satisfying a Broadcast Connection

Given a $v(m,n,r)$ network satisfying the condition on $m$ in (10), we have some $t \in \{1,2,\ldots,\min\{n-1,r\}\}$. Given a connection request $I_t$, $t \in \{1,2,\ldots,nr = N\}$, with $|I_t| = r' \leq r$, we take a set of any $m' = (n - 1)^{r'/2} + 1$ available middle switches; without loss of generality, let the destination sets of these middle switches be $M_1,M_2,\ldots,M_{m'}$. The following algorithm will generate a set of middle switches through which the connection request can be satisfied.

Algorithm:

Step 1: $mid\_switch \leftarrow \emptyset$

for $j = 1$ to $m'$ do

find $S_j(1 \leq k \leq m')$ such that
$|S_k| = \min\{|S_1|,|S_2|,\ldots,|S_{m'}|\}$;

$min\_set \leftarrow S_k$;

$mid\_switch \leftarrow mid\_switch \cup \{k\}$;

if $min\_set \neq \emptyset$ then

for $j = 1$ to $m'$ do

$S_j \leftarrow S_j \cap min\_set$

until $min\_set = \emptyset$;

Step 2: connect $I_t$ through the middle switches in $mid\_switch$

and update the destination sets of these middle switches.

End

It would be appropriate to provide some explanations of the steps in the above algorithm before we discuss its complexity. In the algorithm, $mid\_switch$ designates a set of middle switches which is initially empty, but will contain at the end of the algorithm at most $x$ middle switches the intersection of whose destination sets is empty. The sets $S_1,S_2,\ldots,S_{m'}$ are temporary holding sets which store the results of the intersection operations, and $min\_set$ indicates the minimum cardinality set of the $S_1,S_2,\ldots,S_{m'}$ at each iteration.

Step 1 takes the destination sets of the $m'$ available middle switches and removes all elements other than those in $I_t$ since we clearly need only be concerned with connection paths to these output switches in our attempt to satisfy the connection request. Step 2 repeatedly finds the minimum cardinality set among $S_1,\ldots,S_{m'}$, and intersects each $S_j(1 \leq j \leq m')$ with this minimum cardinality set until $min\_set$ becomes empty. At this point, $mid\_switch$ contains a set of middle switches the intersection of whose destination sets is empty. These middle switches are used in Step 3 to satisfy the connection request. The destination sets are also updated in Step 3.

To show that at most $x$ middle switches are selected at the end of the algorithm and to analyze the complexity of the algorithm, we can assume that $m' = (n - 1)^r$ for some $w > r'/2 > 1$. Since before Step 2 is executed in the first iteration, there are a total of at most $(n - 1)^r$ elements distributed among $S_1,S_2,\ldots,S_{m'}$, it follows that after Step 2 is executed in the first iteration, $min\_set = S_k$ and $|S_k| \leq r'/w$, where $|S_k| = \min\{|S_1|,|S_2|,\ldots,|S_{m'}|\}$. Similarly, after Step 2 of the second iteration, $min\_set \leq r'/w^2$. This reduction of the cardinality of $min\_set$ will continue from one iteration to the next, each time the value of $min\_set$ being reduced to at most $1/w$ times its value at the end of the previous iteration. There can be at most $1 + \log_{w} r'$ iterations. Note that since $\log_{w} r' = \log_{w} r' < \log_{w} r' = x$, we get $1 + \log_{w} r' \leq x$, that is, at most $x$ middle switches are selected. We also note that all sets here are subsets of $\{1,2,\ldots,r\}$, then they can be represented with a "bit-vector" data structure. Thus, the time of Step 1 is proportional to $m'r'$ and the time of each iteration of Step 2 is proportional to $m'[\log_{w} r']$. This means that the
total time of the algorithm is proportional to
\[
m'(r' + \frac{r}{w} + \frac{r'}{w^2} + \cdots + \frac{r'}{w^N}) \leq 2m' r' \prod_{i=1}^{N} \frac{1}{w^i}.
\]

Therefore, the complexity of the our algorithm is \(O(m'r')\).

Taking \(x = \log r\), we get that the complexity is \(O(nr')\) or \(O(N)\) for arranging one input port broadcast connection request. Finally, it should be mentioned that although this complexity analysis is based on the software implementation of the algorithm, this control algorithm can also be realized with a relatively simple hardware implementation to satisfy those applications demanding very short network setup times.

**B. An \(O(nN)\) Algorithm for Realizing an Entire Broadcast Assignment**

It is interesting to note that with our algorithm if we are given a network through which no connections have been made, and if we proceed to provide connection paths in response to requests from input ports until each output port has been connected to some input port, then the total time will be \(O(nN)\). To see this, note that we are handling at most \(nr'\) input port requests \(I_i, 1 \leq i \leq nr\). The time to satisfy each input port request is at most \(cn[I_i]\) for some constant \(c\). Thus, the total time to satisfy all the input port requests is
\[
\sum_{i=1}^{nr'} cn[I_i] = cn \sum_{i=1}^{nr'} |I_i| \leq cn \times nr = cn^2 r.
\]

This shows that the complexity of our algorithm for satisfying an entire broadcast assignment is \(O(nN)\).

**VII. ASYMPTOTIC CROSSPOINT GROWTH OF MULTI-STAGE NETWORK CONSTRUCTIONS**

A traditional means of evaluating and comparing switching network designs is to determine the asymptotic growth of crosspoint switching elements as the number of input ports and output ports increase. Although the availability and use of very large scale integration (VLSI) circuits for the implementation of switching networks might suggest that other measures be employed, it is nevertheless appropriate that we analyze the asymptotic crosspoint growth of our networks and compare it to that of others.

In general, for any \(k \geq 1\), we can construct a symmetric \(2k\)-stage \((N \times N)\) broadcast network by replacing each switch module at the middle stage in a \(v(m, n, r)\) network with a \((2k-1)\)-stage \((r \times r)\) broadcast network. Let \(G_j(N)\) denote the minimum number of crosspoints of our \((N \times N)\) broadcast network with \(j\) stages. We will show that the following is an upper bound on the minimum number of crosspoints for this network:
\[
G_{2k+1}(N) \leq O \left( N^{1 + \frac{1}{2k+1}} (\log N / \log \log N)^{\frac{k+1}{2k+1} - \frac{1}{2}} \right)
\]

(14)

Note that in (14) \(k = 1\) corresponds to the case of a three-stage network. Clearly,
\[
G_3(N) = 2rnm + mr^2 \leq cnr(\log r / \log \log r)(2n + r)
\]

where \(c\) is some constant.

Taking \(n = r = N^{1/2}\), we have
\[
G_3(N) \leq \frac{3c}{2} N^{3/2} \left( \log N / \log \left( \frac{1}{2} \log N \right) \right).
\]

Thus, there exists some constant \(c_3\),
\[
G_3(N) \leq c_3 N^{3/2} (\log N / \log \log N).
\]

Now we assume that \(G_{2k-1}(N)\) satisfies (14) and calculate \(G_{2k+1}(N)\). We can consider a \((2k+1)\)-stage network as a three-stage network \(v(m, n, r)\) where each middle switch has been replaced by a \((2k-1)\)-stage network. Thus,
\[
G_{2k+1} \leq m(2nr + G_{2k-1}(r))
\]
\[
\leq cn(\log \log r / \log \log \log r) \left( \frac{2N}{r} + G_{2k-1}(r) \right)
\]
\[
\leq cn(\log r / \log \log r) \left( \frac{2N}{r} + G_{2k-1}(r) \right)
\]
\[
\cdot \left( \frac{2N}{r} + c_{2k-1} \frac{1}{r} (\log r / \log \log r)^{\frac{k+1}{k} - \frac{1}{k+1}} \right).
\]

Let
\[
r = \frac{N^{\frac{1}{2k+1}}}{(\log N / \log \log N)^{\frac{k+1}{2k+1} - \frac{1}{2k+1}}}
\]

Note that
\[
\frac{\log r}{\log \log r} = \frac{k}{k+1} \log N / \log N^\alpha_N
\]

where \(\alpha_N\) is shown in the equation at the bottom of this page and \(\lim_{N \to \infty} \alpha_N = 1\). We have that
\[
G_{2k+1} \leq c_{2k+1} N^{1 + \frac{1}{2k+1}} (\log N / \log \log N)^{\frac{k+1}{2k+1} - \frac{1}{2k+1}}
\]

where, \(c_{2k+1}\) is some constant. Therefore, (14) holds for all integer \(k \geq 1\).

The above multistage network construction and asymptotic analyses are based on a symmetrical version of the network which is very easy to implement in practice. If we allow

\[
\alpha_N = 1 - \left( \frac{k+1}{2} - \frac{1}{k} \right) (\log \log N / \log N^\alpha_N)
\]
asymmetrical switch modules in the middle stage, we can get an improved bound of the number of crosspoints is \(O(N^{1 + \frac{1}{k+1}}(\log N)^{\frac{3}{2}})\) for \((2k + 1)\)-stage network by using a technique employed by Kirkpatrick, Klawe, and Pippenger [22].

VIII. COMPARISONS

Hwang and Jajsyczyk's [8]–[10] nonblocking multiconnection broadcast networks require that \(m = O(nr)\) even for the case where the input set is a singleton (that is, the broadcast case). Thus, since our networks only require that \(m = O(n \log r / \log \log r)\), extensive comparison of our nonblocking broadcast networks to that Hwang and Jajsyczk's network is not very illuminating as our designs clearly require significantly fewer middle switches and therefore have an overwhelming advantage in crosspoint growth.

Even though our networks are nonblocking, our results warrant comparisons not only to other nonblocking broadcast switching networks, but with some rearrangeable networks as well. Dolev, Dwork, Pippenger, and Wigderson [20] have given a minimum possible upper bound of \(O\left(\frac{N}{(\log N)^{1+1/k}}\right)\) on the number of crosspoints needed for \(k\)-stage rearrangeable broadcast networks (which they refer to as generalized connectors). Subsequently, Feldman, Friedman, and Pippenger [21] gave an upper bound \(O\left(N^{1+1/k}(\log N)^{1-1/k}\right)\) on the number of crosspoints in \(k\) stage nonblocking broadcast networks. Since the latter is a superior bound, it therefore also represents the best upper bound on crosspoints for rearrangeable networks. Clearly, our results are not in agreement with this latter upper bound. But there are no known explicit constructions satisfying these bounds nor are there efficient control algorithms, rendering these constructions to be primarily of theoretic interest. Dolev, Dwork, Pippenger, and Wigderson [20] also gave an explicit construction for \((3k - 2)\)-stage rearrangeable broadcast network \((k \geq 1)\) with \(\gamma_{3k-2}(N) = O\left(N^{1+1/k}\right)\) crosspoints, where \(g_k\) is the number of crosspoints in an \(i\)-stage network. In the previous section we showed that the number of crosspoints of our \((2k + 1)\)-stage network is \(G_{2k+1}(N) = O\left(N^{1 + \frac{1}{k+1}}(\log N/\log \log N)^{\frac{1}{3k-2} + \frac{1}{k+1}}\right)\). In other words, for any limited \(i\)-stage network, \(g_i = O\left(N^{1 + \frac{1}{k+1}}\right)\) and \(G_i = O\left(N^{1 + \frac{1}{k+1}}(\log N/\log \log N)^{\frac{1}{3k-2} + \frac{1}{k+1}}\right)\). Obviously, for any \(i > 1\), we have an improved result.

Richards and Hwang [15], [16] have proposed a multi-stage rearrangeable broadcast network for video teleconferencing applications [24]. The proven two-stage version of the Richards/Hwang network has \(O(N^{7/4})\) crosspoints and the proven three-stage version has \(O(N^{23/14})\) crosspoints. Thus, our nonblocking three-stage network demonstrates superior crosspoint growth for these cases. Richards and Hwang conjectured that the two-stage versions of their rearrangeable networks can be constructed with \(O(N^{3/3})\) crosspoints and that three-stage versions of their rearrangeable networks (which result after one recursive decomposition of their two-stage design) can be constructed with \(O(N^{3/2})\) crosspoints. Richards and Hwang have not proven this conjecture. Finally, the Richards/Hwang two-stage rearrangeable structure is most efficient relative to crosspoint utilization when the number of input ports equals the square of a prime. However, after repeated decompositions, this condition will no longer be satisfied and inefficiency can occur. Our nonblocking networks have no such restrictions and can be decomposed efficiently.

Feldman, Friedman, and Pippenger [21] have also given explicit constructions for nonblocking broadcast networks with \(O\left(N^{3/2}\right)\) crosspoints for two-stage networks and \(O\left(N^{11/7}\right)\) crosspoints for three-stage networks, where the nonblocking connecting capability is relative to idle output ports requesting a connection to input ports. Clearly, these networks have more crosspoints than our networks.

By utilizing a hypergraph-hypercoloring theorem, Kirkpatrick, Klawe, and Pippenger [22] gave a constructive result for multistage rearrangeable broadcast networks where, from the perspective of a three-stage network, \(m \geq (n_1 - 1) \log 2r_2 + 2n_2\). Recall that by Corollary 1, our result is \(m \geq 2(n_1 - 1) \log 2r_2 / \log \log r_2 + (\log 2r_2)^{1/2}/(n_2 - 1)\). Therefore, in any case we would have slightly fewer number of middle switches as well as crosspoints. However, our asymptotic results are the same as theirs.1 In contrast to our linear time control algorithm, Kirkpatrick, Klawe, and Pippenger [22] did not specify a control algorithm, and their proof implies only an exponential time control algorithm. We have also addressed the restricted fan-out case which renders our results somewhat more practical for applications involving limited broadcast.

IX. CONCLUSIONS

We have presented new results for nonblocking multistage broadcast networks wherein a request from an idle input port to be connected to some set of idle output ports can be satisfied without any disturbance of other broadcast connections already existing in the network. We have furthermore given a linear network control algorithm for realizing such a broadcast connection request. These results represent the best known explicit constructions with limited numbers of stages relative to both crosspoint and control algorithm complexity. Thus, our networks are highly useful for practical applications involving the movement of and collaboration with voice/video/text/graphics information that require broadcast capability. These networks are also useful for the interconnection of processor and memory units in parallel processing systems.

ACKNOWLEDGMENT

The authors would like to acknowledge the thoughtful, generous comments of Prof. N. Pippenger of the Department of Computer Science, University of British Columbia, regarding the results reported in this paper.

1On the basis of a probabilistic proof technique, Pippenger and Spencer [23] have reported achieving an improved \(O\left(N^{3/2}\right)\) bound on the number of crosspoints for three-stage rearrangeable networks. This result, however, remains unpublished.
REFERENCES


Yuanyuan Yang (S'90) received the B.S. and M.S. degrees in computer science and engineering from Tsinghua University, Beijing, China, in 1982 and 1984, respectively.

Since September 1987, she has been a Graduate Assistant in the Department of Computer Science, The Johns Hopkins University, Baltimore, MD, where she received the M.S.E. degree in computer science in 1989 and is currently working toward her Ph.D. degree. Her research interests include computer communication networks, parallel processing, algorithm design, computer architecture, and fault-tolerant computing.

Gerald M. Masson (S'67–M'70–SM'86–F'89) was born in Chicago, IL. He received the B.S.E.E. degree from the Illinois Institute of Technology, Chicago, in 1966, and the M.S. and Ph.D. degrees from the Department of Electrical Engineering, Northwestern University, Evanston, IL, in 1968 and 1971, respectively.

In 1966, he worked as a Digital System Designer for the Teletype Corporation, Skokie, IL, for approximately one year. In 1971, he joined the Department of Electrical Engineering, University of Pittsburgh, Pittsburgh, PA. In 1975, he joined the Department of Electrical Engineering, The Johns Hopkins University, Baltimore, MD. He spent the 1984–1985 academic year as a Visiting Professor and half of the 1985–1986 academic year as a Professor in the Department of Electrical and Computer Engineering, Carnegie-Mellon University, Pittsburgh, PA. He returned to Johns Hopkins in 1986 where he is currently Professor and Chairman of the Department of Computer Science. He is coauthor of the textbook Distributed Processor Communication Architecture (Lexington, MA: Heath, 1979). His current research interests include fault-tolerant computing and computer interconnection structures.

Dr. Masson is a member of Sigma Xi and Eta Kappa Nu. He has served as Technical Program Chairman of a number of conference/workshops, including the 1979 International Symposium on Fault-Tolerant Computing. He has participated in the development and presentation of several tutorials for the IEEE. He is on the Editorial Boards of the IEEE TRANSACTIONS ON COMPUTERS and of JETTA.