Better Necessary Conditions for Rearrangeably Nonblocking f-cast d-ary Multi-log Networks under Fanout and Crosstalk Constraints

Hung Q. Ngo, Yang Wang, Anh Le
Department of Computer Science and Engineering
State University of New York at Buffalo
Amherst, New York 14260
Email: {hungngo, yw43, anhle}@cse.buffalo.edu

Xiaohong Jiang
School of Information Sciences
Tohoku University,
Sendai, Japan
Email: jiang@ecei.tohoku.ac.jp

***Abstract*—We derive necessary conditions for the d-ary multi-log switching networks to be f-cast rearrangeably nonblocking when the input stage does not have fanout capability, with and without crosstalk-free constraint. In electronic switching networks, connection routes cannot share a link but they can share a switching node. This is the case when there is no crosstalk-free constraint. In optical switching networks, it is often desirable to enforce the crosstalk-free constraint which forbids connection routes to share switching nodes.**

The most novel contribution of this paper is the analytical technique, which addresses the problem from an algebraic angle. Our necessary conditions are much better than previously known conditions. Moreover, our results are on general d-ary multi-log networks, while known results are on 2-ary networks only.

Keywords: multicast, switches, f-cast, rearrangeably nonblocking, d-ary multi-log switching networks, crosstalk-free, fanout constraints.

I. Introduction

Many current and future Internet applications demand multicast support. To support multicast efficiently, the switching networks which serve as the switching fabric architectures at the core of electronic routers, or as the switching topology for optical cross-connects must be multicast capable.

Current multicast switch designs mostly focus on the broadcast case [1]–[3]. Although broadcast switches are certainly capable of supporting multicast with any fanout requirement, they are not scalable due to their prohibitively high hardware requirement. Almost all the multicast applications are restricted to a group of users, where broadcasting is rarely required. Hence, allocating expensive broadcast capability to each network switch is cost-inefficient for most practical purposes. Moreover, from the viewpoints of resource fairness and network security (e.g., limiting virus and worm propagation), we have other good reasons to impose a restriction on the maximum fanout of each request.

Consequently, there have been some recent research efforts on designing and analyzing the so-called *f-cast* switches, in which the maximum fanout of each request is upperbounded by the parameter f [**?**], [4]–[8]. An f-cast switch usually requires significantly lower hardware cost than its broadcast counterpart. Furthermore, a good design of an f-cast switch covers both the unicast design ($f = 1$) and the broadcast design ($f = N$) as special cases. Consequently, studying general f-cast switches is both mathematically pleasing and practically useful, as the results potentially can offer network designers more flexibility in selecting architectures for future multicast-intensive networks.

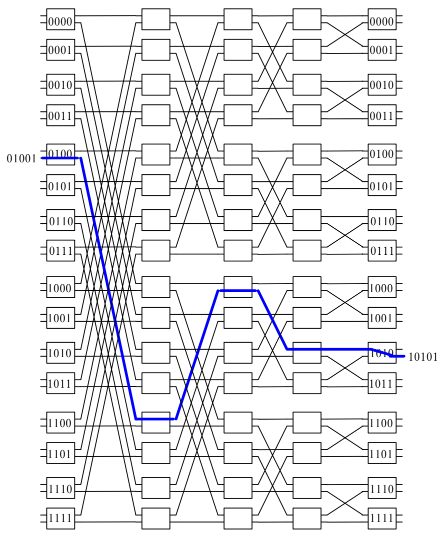

Fig. 1. The inverse Banyan network $\mathrm{BY}^{-1}(5)$

The design of a scalable and hardware-inexpensive switch usually employs the multistage architecture. The most popular multistage architectures are Banyan-type [9] and Clos-type [10] architectures. This paper focuses on analyzing the Banyan-type (e.g., see Figure 1). In particular, we study the general d-ary multi-log switch architecture with multiple vertically stacked inverse Banyan switches, as illustrated in Figure 2. The d-ary multi-log switches have been attractive for both electronic and photonic domains [11]–[17], because they have small depth ($O(\log N)$), absolute signal loss uniformity, and good fault tolerance. Hereafter, we use $\log_d(N, 0, m)$ to denote a d-ary multi-log switch with m vertically stacked

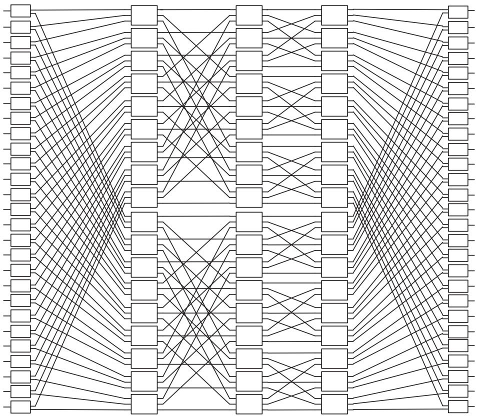

Fig. 2. Illustration of the $\log_3(27, 0, 2)$ network

inverse Banyan planes.

To support multicast (or generally f-cast) in a $\log_d(N, 0, m)$ switch, a certain degree of fanout capability must be provided in the switch. Similar to that of the fanout definitions for Clos network [5], a $1 \times m$ switch or a basic $d \times d$ switching elements (SE) in a $\log_d(N, 0, m)$ switch is said to have *fanout capability* if any one-to-many mapping between its inputs and outputs can be realized. We will say that the input stage (resp. the middle Banyan stage) of a $\log_d(N, 0, m)$ network has the fanout capability if each of its $1 \times m$ switches (resp. $d \times d$ SEs) has the fanout function. Thus, the fanout capability of a $\log_d(N, 0, m)$ switch can be provided either in its input stage, or in its central Banyan stage or in both. The additional fanout requirement of a stage usually makes its implementation more complex and costly than its unicast counterpart, so the fanout capability should be carefully allocated for the efficient design of a multicast-capable switch.

There are three levels of nonblockingness typically studied in the switching network literature: rearrangeably nonblocking (RNB), wide-sense nonblocking (WSNB), and strictly nonblocking (SNB). The reader is referred to [4] for their precise definitions.

In addition to the fanout constraint, we also consider both the link-blocking and node-blocking (i.e. crosstalk-free) constraints. Under the link-blocking constraint, only one request is allowed to use a link at one time, which is relevant to electronic switches [1]–[8], [13], [14], [16]. The node-blocking constraint allows only one request to use a switching element at one time, reducing the crosstalk effect in all-optical cross-connect designs [11], [12], [15], [17]–[19].

This paper focuses on analyzing the RNB f-cast $\log_d(N, 0, m)$ network, in which only the middle Banyan stage has fanout capability. This is a continuation of the work done in [7], where necessary conditions for a $\log_2(N, 0, m)$ network to be f-cast RNB were derived. In particular, let $N = 2^n$, for the case when only the Banyan stage has fanout capability, it was shown in [7] that

$$m \geq \begin{cases} \sqrt{N} + 1 & \text{for } n \text{ even} \\ \sqrt{2N} + 1 & \text{for } n \text{ odd} \end{cases}$$

is necessary for $\log_2(N, 0, m)$ to be f-cast RNB without crosstalk-free constraint, and that

$$m \geq \begin{cases} \sqrt{N} & \text{for } n \text{ even} \\ \frac{1}{2}\sqrt{2N} + 1 & \text{for } n \text{ odd} \end{cases}$$

is necessary for $\log_2(N, 0, m)$ to be f-cast RNB with crosstalk-free constraint.

In this paper, we improve the above results in two fronts: (1) we derive much better necessary conditions, and (2) our results are applicable to the general d-ary network case. This is done via an algebraic view of the multi-log network. However, due to space limitation, we will only present our results for the $d = 2$ case, leaving the general d-ary case for a future, more complete version of the research.

The rest of the paper is organized as follows. Section II establishes basic notations and presents a simple algebraic view of $\log_d(N, 0, m)$ networks, which are used throughout the paper. Section III presents the main results of this paper: better necessary conditions for the $\log_d(N, 0, m)$ network to be f-cast RNB under link/node-blocking constraints when only the central stage has fanout capability. Section IV concludes the paper with a few remarks.

II. Preliminaries

We first establish notations which will be used throughout the paper. For any positive integers l, d, let

- $[l]$ denote the set $\{1, \ldots, l\}$;
- $\mathbb{Z}_d$ denote the set $\{0, \ldots, d-1\}$ which can be thought of as d-ary "symbols";
- $\mathbb{Z}_d^l$ denote the set of all d-ary strings of length l;
- b^l denote the string with symbol $b \in \mathbb{Z}_d$ repeated l times (e.g., $3^4 = 3333$);
- $|\mathbf{s}|$ denote the length of any d-ary string $\mathbf{s}$ (e.g., $|31| = 2$);
- $\mathbf{s}_{i..j}$ denote the substring $s_i \cdots s_j$ of a string $\mathbf{s} = s_1 \ldots s_l \in \mathbb{Z}_d^l$, when $j > i$ we agree on the convention that $\mathbf{s}_{i..j}$ is the empty string.

Let $N = d^n$. We consider the $\log_d(N, 0, m)$ network, which denotes the stacking of m copies of the d-ary inverse Banyan network $\mathrm{BY}^{-1}(n)$ with N inputs and N outputs. We label the inputs and outputs of $\mathrm{BY}^{-1}(n)$ with d-ary strings of length n. Specifically, each input $\mathbf{x} \in \mathbb{Z}_d^n$ and output $\mathbf{y} \in \mathbb{Z}_d^n$ have the form $\mathbf{x} = x_1 \cdots x_n$, $\mathbf{y} = y_1 \cdots y_n$, where $x_i, y_i \in \mathbb{Z}_d$, $\forall i \in [n]$.

Also, label the $d \times d$ SEs in each of the n stages of $\mathrm{BY}^{-1}(n)$ with d-ary strings of length $n - 1$. It is easy to see that an input $\mathbf{x}$ (resp. output $\mathbf{y}$) is connected to the SE labeled $\mathbf{x}_{1..n-1}$ in the first stage (resp. $\mathbf{y}_{1..n-1}$ in the last stage).

For the sake of clarity, let us first consider a small example. Consider the unicast request $(\mathbf{x}, \mathbf{y}) = (01001, 10101)$ when $d = 2, n = 5$. The input $\mathbf{x} = 01001$ is connected to the SE labeled 0100 in the first stage, which is connected to two SEs labeled 0100 and 1100 in the second stage, and so on. The unique path from $\mathbf{x}$ to $\mathbf{y}$ in $\text{BY}^{-1}(n)$ can be explicitly written out (see Figure 1):

input $\mathbf{x}$	01001
stage-1 SE	0100
stage-2 SE	**1**100
stage-3 SE	**10**10
stage-4 SE	**101**0
stage-5 SE	**1010**
output $\mathbf{y}$	10101

We can see clearly the pattern: the prefixes of $\mathbf{y}_{1..n-1}$ are "taking over" the prefixes of $\mathbf{x}_{1..n-1}$ on the path from $\mathbf{x}$ to $\mathbf{y}$. In general, the unique path from an arbitrary input $\mathbf{x}$ to an arbitrary output $\mathbf{y}$ is exactly the following:

input $\mathbf{x}$	$x_1x_2\dots x_{n-1}x_n$
stage-1 SE	$x_1x_2\dots x_{n-1}$
stage-2 SE	$y_1x_2\dots x_{n-1}$
stage-3 SE	$y_1y_2\dots x_{n-1}$
$\vdots$	$\vdots$
stage-n SE	$y_1y_2\dots y_{n-1}$
output $\mathbf{y}$	$y_1y_2\dots y_{n-1}y_n$

Now, consider two unicast requests $(\mathbf{a}, \mathbf{b})$ and $(\mathbf{x}, \mathbf{y})$. In the node-blocking case, these two requests cannot be routed through the same copy of $\text{BY}^{-1}(n)$ if and only if the two corresponding paths intersect at some SE in the middle (if they were to be routed through the same copy). More precisely, $(\mathbf{a}, \mathbf{b})$ and $(\mathbf{x}, \mathbf{y})$ are said to *node-block* each other if and only if there is some $j \in [n]$ such that $b_{1..j-1} = y_{1..j-1}$ and $a_{j..n-1} = x_{j..n-1}$. In this case, the two paths intersect at a stage-j SE. It should be noted that two requests' paths may intersect at more than one SE. In a $\log_d(N, 0, m)$ network, two requests which are node-blocking one another have to be routed through different copies of $\text{BY}^{-1}(n)$.

For any two d-ary strings $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_d^l$, let $\text{PRE}(\mathbf{u}, \mathbf{v})$ denote the *longest common prefix*, and $\text{SUF}(\mathbf{u}, \mathbf{v})$ denote the *longest common suffix* of $\mathbf{u}$ and $\mathbf{v}$, respectively. For example, if $\mathbf{u} = 0100110$ and $\mathbf{v} = 0101010$, then $\text{PRE}(\mathbf{u}, \mathbf{v}) = 010$ and $\text{SUF}(\mathbf{u}, \mathbf{v}) = 10$. From the observation made in the previous paragraph, the following proposition is immediate.

Proposition II.1. *Let* $(\mathbf{a}, \mathbf{b})$ *and* $(\mathbf{x}, \mathbf{y})$ *be two unicast requests in a* $\log_d(N, 0, m)$ *network, then the two requests node-block one another if and only if*

$$|\text{SUF}(\mathbf{a}_{1..n-1}, \mathbf{x}_{1..n-1})| + |\text{PRE}(\mathbf{b}_{1..n-1}, \mathbf{y}_{1..n-1})| \geq n - 1. \quad (1)$$

In the link blocking case, two requests *link-block* each other if and only if they share a common link in $\text{BY}^{-1}(n)$ (if they were to be routed through the same copy). More precisely, two requests $(\mathbf{a}, \mathbf{b})$ and $(\mathbf{x}, \mathbf{y})$ are said to *link-block* each other if and only if there is some $j \in [n]$ such that $b_{1..j-1} = y_{1..j-1}$, $a_{j..n-1} = x_{j..n-1}$, $b_{1..j} = y_{1..j}$, and $a_{j+1..n-1} = x_{j+1..n-1}$. The four conditions are equivalent to just to conditions $b_{1..j} = y_{1..j}$ and $a_{j..n-1} = x_{j..n-1}$. We easily obtain the link-blocking analog of Proposition II.1 as follows.

Proposition II.2. *Let* $(\mathbf{a}, \mathbf{b})$ *and* $(\mathbf{x}, \mathbf{y})$ *be two unicast requests in a* $\log_d(N, 0, m)$ *network, then the two requests link-block one another if and only if*

$$|\text{SUF}(\mathbf{a}_{1..n-1}, \mathbf{x}_{1..n-1})| + |\text{PRE}(\mathbf{b}_{1..n-1}, \mathbf{y}_{1..n-1})| \geq n. \quad (2)$$

III. Main Results

To find a necessary condition for $\log_2(N, 0, m)$ to be RNB f-cast, the most natural strategy is to construct a set $\mathcal{R}$ of requests requiring as many copies of the inverse Banyan network as possible. A more precise way to describe this strategy is as follows.

For any (valid) request set $\mathcal{R}$, construct a graph $G(\mathcal{R}) = (\mathcal{R}, E)$. The vertices of $G(\mathcal{R})$ are requests in $\mathcal{R}$. Two requests R_1, R_2 are connected in $G(\mathcal{R})$ (i.e. $R_1R_2 \in E$) if and only if R_1 and R_2 node-block each other. In order to satisfy $\mathcal{R}$, it is necessary that $m \geq \chi(G(\mathcal{R}))$ (the chromatic number of $G(\mathcal{R})$). To see this, think of each copy of the inverse Banyan network as a color. We thus want to construct a request set $\mathcal{R}$ for which $\chi(G)$ is as large as possible. Formally, we can summarize the above description in the following proposition.

Proposition III.1. *The necessary and sufficient condition for* $\log_2(N, 0, m)$ *to be* f*-cast rearrangeable is*

$$m \geq \max_{\mathcal{R}} \chi(G(\mathcal{R})),$$

where the max *function is over all valid request sets* $\mathcal{R}$*.*

To illustrate our idea, we consider three simple examples.

Example III.2. This example reproduces Theorem 1 and Corollary 1 in [7], illustrating the power of our algebraic approach. In the node-blocking case, let $k = \lfloor n/2 \rfloor$; and define $\mathcal{R}$ as follows.

$$\mathcal{R} = \left\{ (\mathbf{x}, \mathbf{y}) \mid \mathbf{x} = \mathbf{s}0^k u, \mathbf{y} = 0^k \mathbf{s} u, \mathbf{s} \in \mathbb{Z}_2^{n-1-k}, u \in \mathbb{Z}_2 \right\} \quad (3)$$

For any two requests $(\mathbf{a}, \mathbf{b})$ and $(\mathbf{x}, \mathbf{y})$ in $\mathcal{R}$, we have

$$|\text{SUF}(\mathbf{a}_{1..n-1}, \mathbf{x}_{1..n-1})| + |\text{PRE}(\mathbf{b}_{1..n-1}, \mathbf{y}_{1..n-1})| \geq |0^k| + |0^k| = 2k \geq n - 1.$$

Hence, by Proposition II.1 any two requests in $\mathcal{R}$ node-block each other. The graph $G(\mathcal{R})$ is a complete graph of size $|\mathcal{R}| = 2^{n-k} = 2^{\lceil n/2 \rceil}$, which is precisely Theorem 1 in [7].

In the link-blocking case, let $k = \lceil n/2 \rceil$, and define $\mathcal{R}$ exactly like in (3). We have

$$|\text{SUF}(\mathbf{a}_{1..n-1}, \mathbf{x}_{1..n-1})| + |\text{PRE}(\mathbf{b}_{1..n-1}, \mathbf{y}_{1..n-1})| \geq 2k \geq n.$$

Hence, by Proposition II.2 any two requests in $\mathcal{R}$ link-block each other. The graph $G(\mathcal{R})$ is a complete graph of size $|\mathcal{R}| = 2^{n-k} = 2^{\lfloor n/2 \rfloor}$, which is precisely Corollary 1 in [7].

Theorem III.3 (Necessary condition, basic version, node-blocking case)**.** *Let* $k = \lfloor n/2 \rfloor$, $N = 2^n$. *Let*

j be any integer such that $0 \le j \le (k-1)/3$. If $f \ge 1 + 4 + \cdots + 4^j = (4^{j+1}-1)/3$, then for the $\log_2(N, 0, m)$ network to be f-cast RNB in the node-blocking case it is necessary that $m \ge 2^{n-k+j}$.

Proof: We will create 2^{n-k+j} f-cast requests, each of which has fanout equal to $(4^{j+1}-1)/3 \le f$. The inputs of these requests are the $\mathbf{x}$'s such that

$$x_{n-1-k+j} = x_{n-k+j} = \cdots = x_{n-1} = 0.$$

In other words, we will create requests from inputs $\mathbf{x}$ for which the last $k-j$ bits of $\mathbf{x}_{1..n-1}$ are equal to 0. This means, the number of such inputs $\mathbf{x}$ is 2^{n-k+j}. The idea is to construct the requests such that every two of them node-block each other.

Each such input $\mathbf{x}$ can be expressed in the following form

$$\mathbf{x} = \mathbf{suv}0^{k-j}b,$$

where $\mathbf{s} \in \mathbb{Z}_2^{n-1-k-2j}, \mathbf{u} \in \mathbb{Z}_2^{2j}, \mathbf{v} \in \mathbb{Z}_2^{j}$, and $b \in \mathbb{Z}_2$. More precisely, $\mathbf{x}$ can also be written as

$$\mathbf{x} = \mathbf{s}\mathbf{u}_{1..2j}\mathbf{v}_{1..j}0^{k-j}b.$$

This input $\mathbf{x}$ requests a set $Y(\mathbf{x})$ of outputs of size

$$1 + 4 + \cdots + 4^j = (4^{j+1}-1)/3.$$

The output set $Y(\mathbf{x})$ is defined as follows. For convenience, define $v_0 = 0$. For each i from 0 to j, and for each string $\mathbf{t} \in \mathbb{Z}_2^{2i}$,

- if $v_i = 0$, add the following output to $Y(\mathbf{x})$
$$0^{k-1-j-2i}1\mathbf{v}_{i+1..j}\mathbf{t}_{1..2i}\mathbf{u}_{1..2i}\mathbf{u}_{2i+1..2j}\mathbf{v}_{1..i}\mathbf{s}b,$$
- otherwise, if $v_i = 1$, add the following output to $Y(\mathbf{x})$
$$0^{k-1-j-2i}1\mathbf{v}_{i+1..j}\mathbf{u}_{1..2i}\mathbf{t}_{1..2i}\mathbf{u}_{2i+1..2j}\mathbf{v}_{1..i}\mathbf{s}b,$$

Thus, for each $0 \le i \le j$, there are 2^{2i} outputs, one for each $\mathbf{t} \in \mathbb{Z}_2^{2i}$, that are added to $Y(\mathbf{x})$, for a total of $1 + \cdots + 4^j$ outputs in $Y(\mathbf{x})$. It is straightforward to verify that all the sets $Y(\mathbf{x})$ are disjoint. Thus, the set f-cast requests is valid.

To this end, consider any two f-cast requests $(\mathbf{x}, Y(\mathbf{x}))$ and $(\overline{\mathbf{x}}, Y(\overline{\mathbf{x}}))$, where

$$\begin{aligned} \mathbf{x} &= \mathbf{s}\ \mathbf{u}\ \mathbf{v}\ 0^{k-j}b \\ \overline{\mathbf{x}} &= \overline{\mathbf{s}}\ \overline{\mathbf{u}}\ \overline{\mathbf{v}}\ 0^{k-j}\overline{b}. \end{aligned}$$

To complete the proof, we will show that there are outputs $\mathbf{y} \in Y(\mathbf{x})$ and $\overline{\mathbf{y}} \in Y(\overline{\mathbf{x}})$ such that

$$|\text{SUF}(\mathbf{x}_{1..n-1}, \overline{\mathbf{x}}_{1..n-1})| + |\text{PRE}(\mathbf{y}_{1..n-1}, \overline{\mathbf{y}}_{1..n-1})| \ge n-1.$$

Suppose $\text{SUF}(\mathbf{v}, \overline{\mathbf{v}}) = j - i$, for some $0 \le i \le j$. We have

$$|\text{SUF}(\mathbf{x}_{1..n-1}, \overline{\mathbf{x}}_{1..n-1})| = |\mathbf{v}_{i+1..j}0^{k-j}| = k - i.$$

To this end, consider two cases as follows.

Case 1: $i = 0$. In this case $\mathbf{v} = \overline{\mathbf{v}}$. By definition, the following output belongs to $Y(\mathbf{x})$:

$$\mathbf{y} := 0^{k-1-j}1\mathbf{v}_{1..j}\mathbf{u}_{1..2j}\mathbf{s}b$$

And, the following output belongs to $Y(\overline{x})$:

$$\overline{\mathbf{y}} := 0^{k-1-j}1\overline{\mathbf{v}}_{1..j}\overline{\mathbf{u}}_{1..2j}\overline{\mathbf{s}}\overline{b}$$

Furthermore,

$$|\text{PRE}(\mathbf{y}_{1..n-1}, \overline{\mathbf{y}}_{1..n-1})| = |0^{k-1-j}1\mathbf{v}_{1..j}| = k.$$

Hence,

$$\begin{aligned} |\text{SUF}(\mathbf{x}_{1..n-1}, \overline{\mathbf{x}}_{1..n-1})| + |\text{PRE}(\mathbf{y}_{1..n-1}, \overline{\mathbf{y}}_{1..n-1})| \ge \\ (k-0) + k = 2k \ge n-1. \end{aligned}$$

Case 2: $i \ge 1$. In this case we have $v_i \ne \overline{v}_i$, $\mathbf{v}_{i+1..j} = \overline{\mathbf{v}}_{i+1..j}$. Without loss of generality, assume $v_i = 0$ and $\overline{v}_i = 1$. Since $v_i = 0$ the following output belongs to $Y(\mathbf{x})$:

$$\mathbf{y} := 0^{k-1-j-2i}1\mathbf{v}_{i+1..j}\overline{\mathbf{u}}_{1..2i}\mathbf{u}_{1..2i}\mathbf{u}_{2i+1..2j}\mathbf{v}_{1..i}\mathbf{s}b$$

And, because $\overline{v}_i = 1$ the following output belongs to $Y(\overline{x})$:

$$\overline{\mathbf{y}} := 0^{k-1-j-2i}1\overline{\mathbf{v}}_{i+1..j}\overline{\mathbf{u}}_{1..2i}\mathbf{u}_{1..2i}\overline{\mathbf{u}}_{2i+1..2j}\overline{\mathbf{v}}_{1..i}\overline{\mathbf{s}}\overline{b}$$

Furthermore,

$$\begin{aligned} |\text{PRE}(\mathbf{y}_{1..n-1}, \overline{\mathbf{y}}_{1..n-1})| = |0^{k-1-j-2i}1\mathbf{v}_{i+1..j}\overline{\mathbf{u}}_{1..2i}\mathbf{u}_{1..2i}| = \\ (k-j-2i) + (j-i) + 4i = k + j. \end{aligned}$$

Consequently, as desired we obtain

$$\begin{aligned} |\text{SUF}(\mathbf{x}_{1..n-1}, \overline{\mathbf{x}}_{1..n-1})| + |\text{PRE}(\mathbf{y}_{1..n-1}, \overline{\mathbf{y}}_{1..n-1})| \ge \\ (k-i) + (k+i) = 2k \ge n-1. \end{aligned}$$

■

The proof of the following theorem is exactly the same as that of Theorem III.3. The only difference is that, in this case $2k = 2\lceil n/2 \rceil \ge n$.

Theorem III.4 (Necessary condition, basic version, link-blocking case)**.** *Let $k = \lceil n/2 \rceil$, $N = 2^n$. Let j be any integer such that $0 \le j \le (k-1)/3$. If $f \ge 1 + 4 + \cdots + 4^j = (4^{j+1}-1)/3$, then for the $\log_2(N, 0, m)$ network to be f-cast RNB in the node-blocking case it is necessary that $m \ge 2^{n-k+j}$.*

Theorems III.3 and III.4 can be stated more precisely in terms of f (instead of j) as follows.

Corollary III.5. *Let $k_n = \lfloor n/2 \rfloor$, $k_l = \lceil n/2 \rceil$,*

$$j_n = \lfloor \min\{(k_n-1)/3, \log_4(3f+1) - 1\} \rfloor,$$

and

$$j_l = \lfloor \min\{(k_l-1)/3, \log_4(3f+1) - 1\} \rfloor.$$

Then,

(i) *in the node-blocking case, the $\log_2(N, 0, m)$ network is f-cast rearrangeable only if $m \ge 2^{n-k_n+j_n}$; and, it is broadcast rearrangeable only if*
$$m \ge 2^{n-k_n+\lfloor (k_n-1)/3 \rfloor} = 2^{2n/3+O(1)}.$$

(ii) *in the link-blocking case, the* $\log_2(N,0,m)$ *network is* f*-cast rearrangeable only if* $m \geq 2^{n-k_l+j_l}$*; and, it is broadcast rearrangeable only if*

$$m \geq 2^{n-k_l+\lfloor(k_l-1)/3\rfloor} = 2^{2n/3+O(1)}.$$

It is intuitively obvious that the necessary lowerbound for m gets higher as f is larger. This intuition is reflected in Theorems III.3 and III.4 because each time f is increased by a power of 4, the lowerbound is doubled. However, when $1+4+\cdots+4^{j-1} < f < 1+4+\cdots+4^j$, the lowerbounds in the Theorems remain the same as the case when $f = 1+4+\cdots+4^{j-1}$. It would thus be desirable to have better lowerbounds for values of f in between "jumps" of powers of 4. The following theorem partially fulfills this desire.

Theorem III.6 (Necessary condition, finer version, node-blocking case)**.** *Let* $k = \lfloor n/2 \rfloor$, $N = 2^n$. *Suppose*

$$1+4+\cdots+4^{j-1} < f < 1+4+\cdots+4^j$$

for some integer j *such that* $0 \leq j \leq (k-1)/3$. *Let*

$$\overline{f} = 1+4+\cdots+4^j - f.$$

Then, for the $\log_2(N,0,m)$ *network to be* f*-cast RNB in the node-blocking case it is necessary that*

$$m \geq 2^{n-k+j} - \overline{f} \cdot 2^{n-k-j}.$$

Proof: Note that $\overline{f} \leq 4^j$. Let $\overline{U} \subseteq \mathbb{Z}_2^{2j}$ be a fixed set of strings of length $2j$ where $|\overline{U}| = \overline{f}$. Similar to the proof of Theorem III.3, we will create requests from inputs $\mathbf{x}$ of the form

$$\mathbf{x} = \mathbf{s}\mathbf{u}_{1..2j}\mathbf{v}_{1..j}0^{k-j}b,$$

where $\mathbf{s} \in \mathbb{Z}_2^{n-1-k-2j}, \mathbf{u} \in \mathbb{Z}_2^{2j} - \overline{U}, \mathbf{v} \in \mathbb{Z}_2^j$, and $b \in \mathbb{Z}_2$.

This input $\mathbf{x}$ requests a set $Y(\mathbf{x})$ of outputs of size

$$1+4+\cdots+4^j = (4^{j+1}-1)/3.$$

The output set $Y(\mathbf{x})$ is defined as follows. For convenience, define $v_0 = 0$.

For each i from 0 to $j-1$, add outputs to $Y(\mathbf{x})$ in exactly the same manner as that in the proof of Theorem III.3.

When $i = j$, there is a slight difference. For each string $\mathbf{t} \in \mathbb{Z}_2^{2j} - \overline{U}$,

- if $v_i = 0$, add the following output to $Y(\mathbf{x})$

$$0^{k-1-j-2i}1\mathbf{v}_{i+1..j}\mathbf{t}_{1..2i}\mathbf{u}_{1..2i}\mathbf{u}_{2i+1..2j}\mathbf{v}_{1..i}\mathbf{s}b,$$

- otherwise, if $v_i = 1$, add the following output to $Y(\mathbf{x})$

$$0^{k-1-j-2i}1\mathbf{v}_{i+1..j}\mathbf{u}_{1..2i}\mathbf{t}_{1..2i}\mathbf{u}_{2i+1..2j}\mathbf{v}_{1..i}\mathbf{s}b,$$

Thus, for each $0 \leq i \leq j-1$, there are 2^{2i} outputs added to $Y(\mathbf{x})$; and, when $i = j$ there are $2^{2j} - \overline{U}$ outputs added to $Y(\mathbf{x})$. Thus,

$$|Y(\mathbf{x})| = 1+4+\cdots+4^j - \overline{f} = f.$$

In other words, the requests created are all f-cast requests. It is straightforward to verify that all the sets $Y(\mathbf{x})$ are disjoint.

The rest of the proof is completely the same as that of Theorem III.3. The number of requests created is $2^{n-k+j} - \overline{f} \cdot 2^{n-k-j}$, which are pairwise node-blocking as desired. ■

The same proof strategy works for the link-blocking case. The only difference is that we set $k = \lceil n/2 \rceil$ in the link-blocking case.

Theorem III.7 (Necessary condition, finer version, link-blocking case)**.** *Let* $k = \lceil n/2 \rceil$, $N = 2^n$. *Suppose*

$$1+4+\cdots+4^{j-1} < f < 1+4+\cdots+4^j$$

for some integer j *such that* $0 \leq j \leq (k-1)/3$. *Let*

$$\overline{f} = 1+4+\cdots+4^j - f.$$

Then, for the $\log_2(N,0,m)$ *network to be* f*-cast RNB in the link-blocking case it is necessary that*

$$m \geq 2^{n-k+j} - \overline{f} \cdot 2^{n-k-j}.$$

Theorems III.6 and III.7 are only better than their respective counterparts III.3 and III.4 when $\overline{f} \leq 2^j$, or equivalently when

$$1+4+\cdots+4^{j-1}+2^j < f < 1+4+\cdots+4^j.$$

The natural question is: can we give a lowerbound better than that for $f = 1+4+\cdots+4^{j-1}$ when the fanout f is in between $f = 1+4+\cdots+4^{j-1}$ and $f = 1+4+\cdots+4^{j-1}+2^j$. The answer to this question seems to be positive, but we have not found a sufficiently "neat" answer to be given here. Let us show a better bound when $j = 1$ as an example.

Example III.8. In this example, we derive a necessary condition for the case when $f = 3 = 1+2^1$ ($j = 1$ in the language of previous theorems). For simplicity, assume $n = 2k+1$ and $k \geq 2$. We will show that $m \geq 2^{k+1} + 2^{k-1}$ is necessary for the network to be f-cast rearrangeable in the node-blocking case. Note that Theorem III.6 only gives a lower bound of

$$m \geq 2^{n-k+1} - 2^1 \cdot 2^{n-k-1} = 2^{n-k} = 2^{k+1}.$$

Hence, our lowerbound here of $2^{k+1} + 2^{k-1}$ is better.

We will create requests from the inputs $\mathbf{x}$ of the following forms: $\mathbf{s}000^kc, \mathbf{s}010^kc, \mathbf{s}100^kc, \mathbf{s}0010^{k-1}c, \mathbf{s}0110^{k-1}c$, where $c \in \mathbb{Z}_2$, and $\mathbf{s} \in \mathbb{Z}_2^{k-2}$. The total number of such inputs is our desired lowerbound of $5 \cdot 2^{k-1} = 2^{k+1} + 2^{k-1}$.

For each such input $\mathbf{x}$, we create a multicast request $(\mathbf{x}, Y(\mathbf{x}))$ according to the following rules:

- For $\mathbf{x} = \mathbf{s}000^kc$, set

$$Y(\mathbf{x}) = \{0^k00\mathbf{s}c, 0^{k-2}0110\mathbf{s}c, 0^{k-2}1100\mathbf{s}c\}$$

- For $\mathbf{x} = \mathbf{s}010^kc$, set

$$Y(\mathbf{x}) = \{0^k01\mathbf{s}c, 0^{k-2}0100\mathbf{s}c, 0^{k-2}1110\mathbf{s}c\}$$

- For $\mathbf{x} = \mathbf{s}100^kc$, set

$$Y(\mathbf{x}) = \{0^k10\mathbf{s}c, 0^{k-2}1000\mathbf{s}c, 0^{k-2}1010\mathbf{s}c\}$$

- For $\mathbf{x} = \text{s}0010^{k-1}c$, set
$$Y(\mathbf{x}) = \{0^{k-2}0101\text{s}c, 0^{k-2}1001\text{s}c, 0^{k-2}1101\text{s}c\}$$
- For $\mathbf{x} = \text{s}0110^{k-1}c$, set
$$Y(\mathbf{x}) = \{0^{k-2}0111\text{s}c, 0^{k-2}1011\text{s}c, 0^{k-2}1111\text{s}c\}$$

It is straightforward to check that these requests are valid and that they pair-wise node-block one another.

IV. Discussions

There are obviously several gaps in our results. The most difficult task is to find sufficient conditions that will match the necessary conditions. In this paper, we have used a simple bound for the chromatic number of $G(\mathcal{R})$: the clique-number of the graph. Any attempt at a better bound must exploit the structure of the graph much further than what derived here. For sufficiency, any upperbound for graph coloring gives a sufficient condition. In [20] we used Brook's theorem (greedy coloring) to show sufficiency conditions for the strictly nonblocking case, which are certainly also sufficient for the rearrangeably nonblocking case. We expect the sufficient conditions for rearrangeability to be much better than those of strictly nonblockingness, however. This question is wide open for further research.

References

[1] W. Kabacinski and G. Danilewicz, "Wide-sense and strict-sense nonblocking operation of multicast multi-log_2n switching networks," *IEEE Transactions on Communications*, vol. 50, pp. 1025–1036, Jun 2002.

[2] F.K.Hwang and B.-C. Lin, "Wide-sense nonblocking multicast $\log_2(n,m,p)$ networks," *IEEE Transactions on Communications*, vol. 51, pp. 1730–1735, Oct 2003.

[3] Y. Yang and G. M. Masson, "Nonblocking broadcast switching networks," *IEEE Trans. Comput.*, vol. 40, no. 9, pp. 1005–1015, 1981.

[4] F. K. Hwang, *The mathematical theory of nonblocking switching networks*. River Edge, NJ: World Scientific Publishing Co. Inc., 2004.

[5] H.-B. Chen and F. K. Hwang, "On multicast rearrangeable 3-stage clos networks without first-stage fan-out," *SIAM Journal on Discrete Mathematics*, vol. 20, no. 2, pp. 287–290, 2006.

[6] A. Pattavina and G. Tesei, "Non-blocking conditions of multicast three-stage interconnection networks," *IEEE Transactions on Communications*, vol. 46, pp. 163–170, Dec. 2005.

[7] X. Jiang, A. Pattavina, and S.Horiguchi, "Rearrangeable f-cast multi-$\log 2n$ networks," *IEEE Transactions on Communications*, 2007. to appear.

[8] F.K.Hwang, Y. Wang, and J. Tan, "Strictly nonblocking f-cast $\log_d(n,m,p)$ networks," *IEEE Transactions on Communications*, vol. 55, pp. 981–986, May 2007.

[9] R. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proceedings of the 1st Annual Symposium on Computer Architecture (ISCA'73)*, pp. 21–28, Dec 1973.

[10] C. Clos, "A study of non-blocking switching networks," *Bell System Tech. J.*, vol. 32, pp. 406–424, Mar 1953.

[11] C.-T. Lea, "Muti-$\log_2 n$ networks and their applications in high speed electronic and photonic switching systems," *IEEE Transactions on Communications*, vol. 38, no. 10, pp. 1740–1749, 1990.

[12] M. M. Vaez and C.-T. Lea, "Strictly nonblocking directional-coupler-based switching networks under crosstalk constraint," *IEEE Transactions on Communications*, vol. 48, pp. 316–323, Feb 2000.

[13] F. Hwang, "Choosing the best $\log_2(n,m,p)$ strictly nonblocking networks," *IEEE Transactions on Communications*, vol. 46, pp. 454–455, Dec 1998.

[14] D.-J.Shyy and C.-T. Lea, "$\log_2(n,m,p)$ strictly nonblocking networks," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1502–1510, 1991.

[15] G. Maier and A. Pattavina, "Design of photonic rearrangeable networks with zero first-order switching-element-crosstalk," *IEEE Transactions on Communications*, vol. 49, no. 7, pp. 1268–1279, 2001.

[16] C.-T. Lea and D.-J.Shyy, "Tradeoff of horizontal decomposition versus vertical stacking in rearrangeable nonblocking networks," *IEEE Transactions on Communications*, vol. 39, pp. 899–904, 1991.

[17] X.Jiang, H. Shen, M. M.R.Khandker, and S.Horiguchi, "Blocking behaviors of crosstalk-free optical banyan networks on vertical stacking," *IEEE/ACM Trans. Networking*, vol. 11, no. 6, pp. 982–993, 2003.

[18] V. R. C. et al, "Crosstalk in a lossy directional coupler switch," *J. Lightwave Technol.*, vol. 13, no. 7, pp. 1530–1535, 1995.

[19] D.Li, "Elimination of crosstalk in directional coupler switches," *Optical Quantum Electron.,*, vol. 25, no. 4, pp. 255–260, 1993.

[20] Y. Wang, H. Q. Ngo, and X. Jiang, "Strictly nonblocking f-cast d-ary multi-log networks under fanout and crosstalk constraints," *IEEE Transactions on Communications*, 2007. Submitted.